UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

Habilitation à Diriger les Recherches

Spécialité : Informatique

Présentée par

Rébecca Deneckère
Maître de conférences – Section CNU 27

Ingénierie des Processus Intentionnels

Habilitation soutenue publiquement le 16 Décembre 2022 devant le jury composé de :

- Oscar Pastor, Professeur, Universidad Politécnica de Valencia (Espagne), rapporteur
- Christine Verdier, Professeur, Université Grenoble Alpes, rapporteur
- Régine Laleau, Professeur, Université Paris-Est Créteil, rapporteur
- Sjaak Brinkkemper, Professeur, Universiteit Utrecht (Pays-Bas)
- Carine Souveyet, Professeur, Université Paris 1 Panthéon-Sorbonne
- Bénédicte Le Grand, Professeur, Université Paris 1 Panthéon-Sorbonne

Centre
de Recherche
en Informatique

Remerciements

Tout d'abord je remercie les rapporteurs de ce mémoire, merci du fond du cœur d'avoir accepté cette charge de travail supplémentaire alors que l'on a tous déjà beaucoup de travail ! Gracias Oscar, por todos estos años compartidos! ; Christine, pour t'intéresser à mes travaux de recherche ; et Régine, que je connais depuis le tout début de mes enseignements et qui participe régulièrement aux soutenances de thèse de mes doctorants. Merci à tous les trois pour m'avoir suivi depuis maintenant plus de vingt ans.

Merci également à tous les membres du jury : Sjaak, qui m'a accompagné tout au long de ma carrière, avec qui j'ai commencé par échanger autour de l'ingénierie des méthodes et avec qui je rentre maintenant dans la « Vie Intelligente » (En ik hoop dat we lang zullen samenwerken!) ; Bénédicte, collègue, amie, co-bureau, complice, avec qui j'ai les meilleures « réunions de direction » du monde ; et bien sûr Carine, qui m'a suivi de la thèse à l'HDR, toujours là pour moi, pour me conseiller et m'aider.

Je tiens aussi à remercier Colette, sans qui je ne serais certainement pas là, la seule personne que je connais capable de réfléchir à plusieurs niveaux en même temps sans perdre le fil, impressionnante.

Merci aussi à tous mes autres collègues du CRI (Daniel, Irina, Camille, Manuele, Selmin, Nicolas), qui font que venir au labo est toujours un plaisir, grâce à qui partager un repas à la va-vite entre deux cours est aussi une occasion de rire, se détendre et communiquer.

Et que serais-je sans mes étudiantes ? Ce mémoire est aussi le vôtre après tout ! Elena (K), ma première doctorante, devenue mon amie et ma co-chercheuse privilégiée, avec qui la recherche est un plaisir ; Ghazaleh, toujours efficace et intelligente ; Elena (E), brillantissime doctorante et amie également ; et Ramona, la « petite dernière ». Vous avez toutes été complètement différentes et notre manière de travailler à de facto aussi été complètement différente, mais tout aussi efficace. Bravo à vous et à tout ce que l'on s'est apporté mutuellement au cours des années.

Et pour finir, merci à ma famille et mes amis. Toujours là fidèles au poste, sans faillir. Mes parents qui m'ont inculqué ma curiosité et mon indépendance qui m'ont suivi tout au long de ces années ; mes sœurs ; et mes enfants, mes grands rayons de soleil. Et mention spéciale à mon chéri qui a dû me supporter pendant l'écriture de ce mémoire et où je n'ai pas forcément été la plus facile à vivre…

Tables des matières

I. Introduction 7

I.1. Les processus intentionnels 7

I.1.1. Le modèle de la Carte 10

I.1.2. Fonctionnement d'un processus intentionnel 15

I.2. Mes travaux de recherche 16

I.2.1. Mes principaux encadrements 17

I.2.2. L'ingénierie des processus intentionnels 18

I.2.3. Application de l'ingénierie des processus intentionnels à l'ingénierie des méthodes situationnelles 19

I.2.4. Application de l'ingénierie des processus intentionnels à la fouille d'intentions 21

I.2.5. Structure du document 23

II. Définir un modèle intentionnel 25

II.1. Définir un modèle intentionnel par réingénierie 26

II.1.1. Construction manuelle d'un processus métier : l'approche MMMR 26

II.1.2. Notion de famille de processus 28

II.2. Définir un modèle intentionnel par combinaison de composants 32

II.2.1. Composant de méthode 32

II.2.2. Famille de méthode 33

II.2.3. Construction d'une famille de méthodes 36

II.2.4. Famille de méthode appliquée à la prise de décision : l'approche MADISE 40

II.2.5. Famille de méthode appliquée au lancement de projets agiles : la famille LPA 48

II.2.6. Famille de méthode appliquée à l'analyse statistique : la famille SAM 52

II.3. Définir un modèle intentionnel par découverte dans les traces utilisateurs . 57

II.3.1. Définir un modèle de processus intentionnel en se basant sur les traces utilisateurs : l'approche FlexPAISSeer 59

II.3.2. Définir un modèle de processus intentionnel en se basant sur les traces utilisateurs : l'approche Map Miner Method (MMM) 66

II.3.3. Définir un modèle intentionnel par adaptation en fonction du comportement des utilisateurs : l'approche MMM 74

III. Utiliser un modèle intentionnel 81

III.1. Utiliser un modèle intentionnel par guidage en fonction du contexte 82

III.1.1. Utiliser un modèle intentionnel par guidage en fonction du contexte : l'approche MIG (Map Indicator-based Guidance) 82

III.1.2. Utiliser un modèle intentionnel par guidage en fonction du contexte : l'approche de la Théorie des Graphes 89

III.2. Utiliser un modèle intentionnel par guidage en fonction du comportement des utilisateurs 98

III.2.1. Utiliser un modèle intentionnel par guidage en fonction du comportement des utilisateurs : l'approche Map-TBS 98

III.2.2. Utiliser un modèle intentionnel par guidage en fonction du comportement des utilisateurs : l'approche FlexPAISSeer 104

IV. Perspectives et Conclusion 109

V. Annexes 113

VI. Bibliographie 133

VII. Curriculum Vitae 145

I. Introduction

Résumer vingt ans de carrière en recherche n'est pas forcément un exercice facile, surtout que le propre d'un chercheur est d'être particulièrement curieux. Je me suis intéressée à différentes thématiques de recherche pendant cette période mais j'ai choisi ici de limiter mon mémoire sur les travaux réalisés dans le cadre de l'ingénierie des processus intentionnels.

L'ingénierie des processus, et, par extension, l'ingénierie des processus intentionnels (Rolland 1998), est un domaine phare en ingénierie des systèmes d'information (SI). L'étude des traitements, des règles de gestions, des intentions des utilisateurs, des différentes stratégies qui leurs sont offertes, offre de nombreuses possibilités et techniques différentes. La fouille de processus (Van der Aalst 2011), quant à elle, a ouvert un tout nouvel horizon à ce domaine en intégrant des techniques beaucoup plus poussées (et surtout automatiques) pour définir et modifier les modèles de processus. De nouvelles approches apparaissent également pour pouvoir faire de la recommandation à la volée et guider les utilisateurs pendant leur exécution de leur processus. La Figure 1 illustre l'arrivée des différents domaines dans lesquels j'ai travaillé, ainsi que les différentes grandes approches que nous avons proposées dans le domaine de l'ingénierie des processus intentionnels.

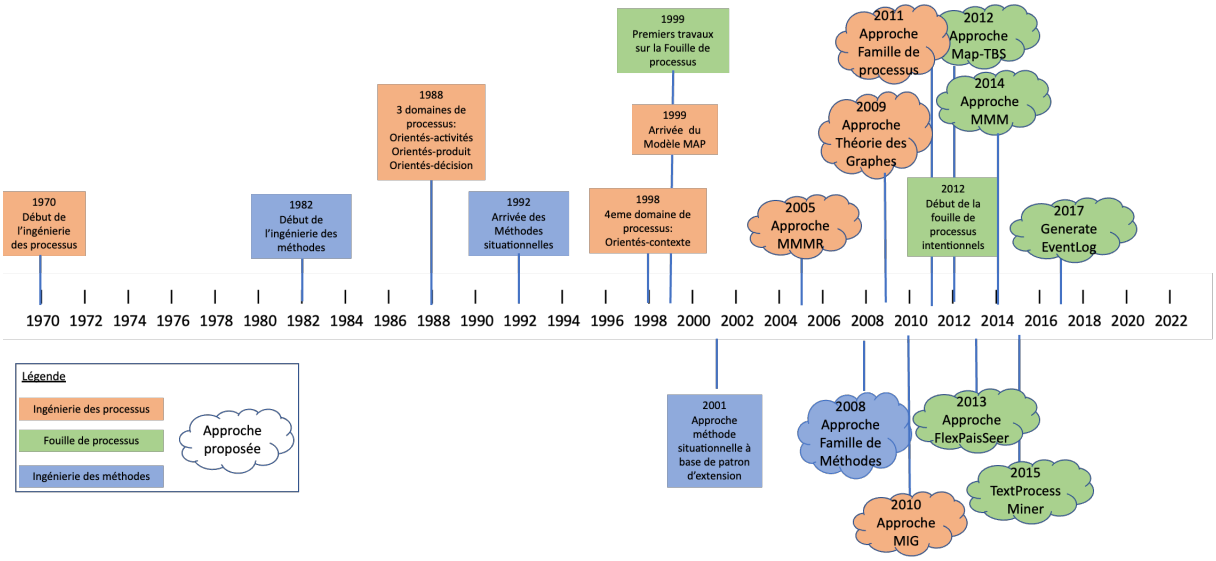


Figure 1. Frise chronologique avec l'arrivée des différents domaines et les approches que nous avons proposés dans ceux-ci

I.1. Les processus intentionnels

Un processus intentionnel (aussi appelé *téléologique*), est un processus qui prend en compte le comportement téléologique d'une exécution d'un processus (comportement attaché à la notion de but). Il décrit les intentions (buts, objectifs) associées à un résultat que le concepteur veut obtenir. (Taylor 1964) propose une classification de comportement comme étant soit axé sur les buts (comportement téléologique) soit axé sur la réponse aux stimuli. (Malcom 1967) définit un comportement téléologique comme suit : lorsqu'un comportement spécifique est nécessaire pour satisfaire un objectif, alors ce comportement se produit. En fait,

il est nécessaire d'étudier la possibilité que les règles changent au cours du processus de développement. (Cayla 2008) souligne le lien entre cette possibilité de changement et les principes téléologiques de Veblen (Veblen 1898) qui intègrent les intentions de l'acteur dans les processus. Ce travail définit un processus téléologique comme « un équilibre dynamique dont les règles sont déterminées de façon endogène, comme une relation entre le comportement du système et de ses intentions ».

Un tel processus peut exprimer les changements dans une trajectoire de processus. Par exemple, même si l'ingénieur a décidé d'un chemin général pour l'exécution de son processus, il a l'occasion de modifier ce chemin (au cours de l'exécution), en prenant en compte les informations et les connaissances qu'il a recueillies de la situation en cours.

(Van de Ven 1995) définit les processus téléologiques comme décrivant le comportement des acteurs faisant des actions pour atteindre un but ou un objectif (un état final envisagé). (Ralph, 2008) souligne que cette théorie est compatible avec le développement de logiciels de la façon suivante : les agents téléologiques sont l'équipe de développement logiciel et le chef de projet ; l'état final représente le produit logiciel, l'équipe de développement effectue des actions (codage, tests...) qui correspondent aux étapes téléologiques et le chef de projet surveille les performances. La notion de but est également essentielle pour toute organisation. Le système d'information est créé car il y a plusieurs objectifs à atteindre et ses fonctions et propriétés sont définies par les objectifs de l'organisation que le système vise à remplir (Loucopoulos 1995). Par exemple, les objectifs des processus sont envisagés dans des approches comme KAOS (Dardenne 1993) et I* (Yu 1997). Ces approches définissent la notion d'option alternative existant dans les processus d'ingénierie des exigences. Les intentions alternatives sont évaluées pour améliorer ces processus (Van Lamsweerde 2009). Plus spécifiquement, dans I*, il existe un langage des exigences orienté objectif appelé GRL (Goal-oriented Requirement Language). Le GRL utilise les exigences non-fonctionnelles (NFR – Non-Functional Requirements). Ce langage permet d'identifier les objectifs et les solutions pour atteindre ces objectifs. Les solutions sont représentées comme des alternatives qui sont mesurées par rapport à leur contribution aux objectifs (Amyot 2002).

Les processus téléologiques intègrent, par le biais des décisions à prendre pour réaliser les intentions, une certaine variabilité. Au fil du temps, la *Variabilité* en génie logiciel est devenu de plus en plus importante. Dans les premiers temps, le SI permettait de réaliser les objectifs d'une seule organisation et d'un ensemble assez simple de clients, alors que, de nos jours, il doit être conçu dans une perspective plus large, pour répondre aux objectifs de plusieurs organisations, être adaptable à des situations et des clients différents (Rolland 2007). La variabilité est la capacité à être sujet à la variation. La notion de variabilité du logiciel est donc définie comme la capacité d'un système logiciel à être modifié, adapté ou configuré dans un certain contexte (Van Gurp 2000). Alors que la communauté du logiciel considère la variabilité comme un problème de conception et se concentre sur les questions d'implémentation (Van Gurp 2001) (Bosh 2001) (Bachmann 2001), nous estimons, comme (Halmans 2003), que l'expression de la variabilité au niveau des buts est essentielle pour satisfaire la nature multi-objectifs de ces nouveaux SI, intégrant la variabilité et capables de s'auto-adapter à la situation en cours.

(Liaskos 2007) établit que la modélisation de la variabilité est utile tant pour l'acquisition de variabilité – pour découvrir des points de variation – que pour un contexte et une situation donnés. Le grand nombre de caractéristiques observées dans l'analyse de la variabilité – pour évaluer l'applicabilité de chaque variant identifié dans les systèmes informatiques modernes, ainsi que la difficulté à comprendre comment les détails techniques des décisions affectent les intentions des parties prenantes, conduit à la nécessité d'explorer et d'analyser la variabilité à un niveau supérieur d'abstraction. Plusieurs typologies de variabilité ont été proposées dans cette optique. Par exemple, (Salifu 2007) utilise les descriptions de problèmes pour représenter et analyser la variabilité dans les logiciels sensibles au contexte ; ce travail identifie le lien entre les besoins et le contexte comme une étape fondamentale pour concevoir des systèmes sensibles au contexte. Ils proposent quatre dimensions à considérer pendant l'établissement de la variabilité contextuelle et les préoccupations liées aux exigences de qualité, les phénomènes physiques, l'application d'un processus de prise de décision aux besoins de qualité et l'identification des catégories connues de problèmes. (Bashmann 2001) distingue plusieurs sources de variation telles que les fonctions, les données, les contrôles de flux, la technologie, les objectifs qualité et l'environnement. Dans (America 2003), la classification décrit les vues clients, application, fonctionnelle, conceptuelle et développement. Dans (Kuloor 2003) l'identification des dimensions de variabilité des besoins est considérée comme un problème de séparation des besoins spécifiques du problème étudié alors que, dans (Halmans 2003), la variabilité essentielle (la variabilité d'un point de vue utilisateur) se distingue de la variabilité technique (problèmes d'implémentation).

La variabilité d'un processus se doit de prendre en compte le contexte du projet sur lequel il s'applique. Pour cela, on parle de la *sensibilité au contexte*.

Les processus intentionnels et la sensibilité au contexte

(Dey 2001) définit la notion de *contexte* par toute information pouvant être utilisée pour caractériser la situation d'une entité (personne, objet physique ou informatique). Et, plus généralement, tout élément pouvant influencer le comportement d'une application. (Rey *2*002) propose plusieurs axes de définitions du contexte : il n'y a pas de contexte sans contexte (i.e. le contexte doit se définir en fonction d'une finalité), le contexte est un espace d'information qui sert l'interprétation (i.e. la capture du contexte n'est pas l'objectif mais les données capturées doivent servir un but), le contexte est un espace d'information partagé par plusieurs acteurs (i.e. utilisateur et système), le contexte est un espace d'information infini et évolutif (i.e. il se construit au cours du temps).

Les modèles de contexte sont pluridisciplinaires (Bradley 2005). La recherche linguistique s'intéresse à l'analyse du contexte d'utilisation des signes (mots) dans un langage. (Bunt, 1997) définit cinq types de contexte pour les aspects communication : la linguistique (lié au matériel linguistique), la sémantique (lié à la description du domaine), le physique (lié à la description de l'environnement dans lequel l'action ou l'interaction se produit), le social (lié à la situation interactive qui se produit entre acteurs) et le cognitif (lié aux intentions des participants, leur évolutions dépendant de la perception, la production, l'évaluation et l'exécution).

Le contexte est également formalisé selon des modèles mathématiques. Par exemple, (Coutaz 2002) propose un modèle cumulatif où le contexte (*Ctx*) est une agrégation

temporaire de situations. Une situation est un descripteur d'état pour un utilisateur (*U*) exécutant une tâche (*T*) à un certain moment (*t*). Le modèle est décrit par la formule suivante :

$$Ctx(U,T,t) = \bigcup_{n=1}^{m} (Situation(U,T,t_n))$$

Le terme '*Context-awareness'* (sensibilité au contexte) vient des travaux sur l'informatique pervasive, ou *informatique ubiquitaire*. Ces systèmes utilisent les modifications de l'environnement dans les systèmes informatiques. Même si c'est un terme d'origine informatique, il a également été appliqué à la théorie des entreprises dans le domaine de la gestion des processus métiers (Rosemann 2006). C'est dans (Schilit 1994) que l'on trouve la première apparition de ce terme qui ne concerne alors que des variables comme la localisation, l'identité des personnes proches et des objets ainsi que les changements dans ces objets. Plusieurs définitions existent, comme (Dey 2001) pour qui un système est sensible au contexte s'il utilise le contexte pour offrir des informations ou des services pertinents pour l'utilisateur alors que (Salber 1998) définit la sensibilité au contexte comme étant la meilleure capacité d'un système à agir en temps réel avec des données provenant du contexte. Plus en rapport avec notre étude, les modèles de contexte sont également proposés pour la réringénierie des processus métiers (Bessai 2008), l'informatique (bradley 2005), la sélection de services (Kirsh Pinheiro 2008) et la prise de décision en situation militaire (Rosen 2008) (Drury 2008). Dans ces derniers cas, le modèle de contexte est perçu comme une manière d'analyser une *situation* donnée afin de guider une exécution de tâches. Ainsi, les modèles de contexte sont principalement utilisés pour résoudre le problème de manque de flexibilité et d'adaptabilité au sein des processus.

Nous avons mentionné que les processus téléologiques, ou intentionnels, contiennent des étapes de prise de décision. Ces décisions peuvent porter sur les modèles à utiliser, actions à effectuer, scénarios possibles etc. Ce sont notamment les décisions prises au cours de l'exécution des processus qui permettent de les adapter au contexte et, par conséquent, d'assurer leur variabilité. Chaque fois qu'un ingénieur doit choisir entre plusieurs possibilités pour continuer son processus, il est confronté à un problème de prise de décision. Un problème de prise de décision est défini par la présence d'au moins deux alternatives. Le concept d'alternative désigne l'objet de la décision. Les alternatives doivent être identifiées clairement. Ces alternatives sont comparées entre elles selon un ou plusieurs critères. Les critères représentent des informations qui permettent d'évaluer et de comparer les alternatives.

I.1.1. Le modèle de la Carte

La littérature suggère plusieurs formalismes de modèles intentionnels, tels que KAOS (Dardenne 1993), Map (Rolland 1999), i* (Yu 2011), Tropos (Bresciani 2004) et GRL (Amyot 2009) entre autres.

Le modèle i* a été créé par Eric Yu (1997, 2009) et ses principes reposent sur deux concepts : les acteurs intentionnels et les dépendances intentionnelles entre les acteurs. Deux types d'objectifs sont présentés dans i* : des objectifs concrets qui peuvent être atteints et des objectifs plus souples qui peuvent être satisfaits, ce qui induit un manque de précision dans la perception de la satisfaction (Yu 1997) (Maria 2009). L'élément central de i* est

l'acteur qui dépend d'autres acteurs pour atteindre un but ou un objectif plus facile. Derrière la formulation d'un plan de satisfaction des objectifs, l'acteur dépend des autres acteurs pour fournir les ressources nécessaires et pour effectuer certaines tâches. Par conséquent, un acteur est qualifié de stratégique, essayant de satisfaire ses intentions tout en faisant face à ses opportunités et vulnérabilités (Maria 2009). Les acteurs sont une catégorie générale comprenant les agents (les entités concrètes et actives), les rôles (l'ensemble des responsabilités des agents) et les positions (les rôles joués par les agents). Selon Yu (1997, 2009), les acteurs sont intentionnels donc ils ont des croyances, des aptitudes et des engagements. Généralement, le modèle i* se concentre sur l'environnement social en intégrant l'autonomie de l'acteur (le comportement ne peut pas être entièrement contrôlable ou connaissable), l'intentionnalité (le comportement n'est pas aléatoire mais plutôt dirigé par des intentions), la socialité (l'acteur dépend des relations sociales pour atteindre ses objectifs), la rationalité (les raisons du comportement peuvent être comprises), la réflectivité stratégique (l'acteur réfléchit sur les relations qui peuvent le positionner stratégiquement par rapport à ses objectifs). La principale difficulté avec i* est qu'il n'y a pas de mécanismes systématiques de raffinement des buts (tous les buts sont définis au même niveau), et qu'il n'y a pas de couple objectif-stratégie pour aider à clarifier les multiples façons dont un but peut être atteint. Les modèles i* trouvent leurs limites lorsque la situation devient plus complexe. Tropos est une méthodologie de génie logiciel orientée agent qui inclut i*; GRL est un langage de modélisation intentionnel basé sur un sous-ensemble de i*. Ainsi, ils ont tous les deux les mêmes limites que i*.

Alors que i* se concentre sur la capture de l'intentionnalité derrière le comportement de l'agent en modélisant les humains comme des entités cherchant à satisfaire leurs objectifs, KAOS se concentre directement sur les objectifs. KAOS est caractérisé comme un cadre multi-paramètres qui expose différents degrés d'expression : semi-formelle pour la modélisation des objectifs, qualitative pour la sélection des alternatives et formelle pour le raisonnement. Il est composé d'une représentation externe utilisant des concepts, des attributs et des relations, et d'une couche interne capturant la temporalité (Van Lamsweerde 2004). Dans le cadre du KAOS, les objectifs peuvent être abordés au moyen des conditions requises. Les conditions sont en outre classées dans les exigences se rapportant aux produits logiciels, et les hypothèses se référant au comportement des agents externes. Les objets utilisés dans la réalisation du but représentent la perspective structurelle du cadre et ils peuvent être actifs ou passifs. Le modèle d'objectifs est représenté sous forme de graphique ET/OU où un objectif est réduit récursivement à des sous-objectifs. Les objectifs fonctionnels (services) et non fonctionnels (qualités des services) sont pris en compte (Van Lamsweerde 2004) (Maria 2009). Le processus d'élaboration du modèle d'objectifs comprend plusieurs étapes : l'identification des objectifs, la formalisation des objectifs, la détection et la gestion des conflits liés aux objectifs, le raffinement des objectifs, l'attribution des responsabilités aux agents et la détection et la gestion des opérations. Les objectifs sont classés en fonction des agents comme suit : les objectifs de satisfaction sont ce que les agents veulent atteindre, les objectifs d'information ont pour but d'informer les agents et les objectifs d'exactitude saisissent les aspects non fonctionnels (Van Lamsweerde 2004). KAOS a une décomposition rigide des tâches – l'affinement ne peut pas être séparé de la décomposition OU et de la décomposition ET, ce qui introduit une complexité artificielle dans la hiérarchie des buts; par conséquent, la modélisation de processus intentionnels complexes est difficile. Une autre difficulté dans le modèle KAOS est de trier les objectifs selon leur niveau d'abstraction et de relier les objectifs

lorsqu'ils appartiennent au même niveau d'abstraction. De plus, KAOS est moins impliqué dans l'aspect intentionnel des acteurs des systèmes d'information puisqu'il ne soutient pas les objectifs stratégiques/organisationnels.

La plupart des modèles n'intègrent pas réellement le concept d'intention, se concentrant sur la manière dont le processus est exécuté et externalisant dans l'intention ce que le processus est destiné à accomplir (Dietz, 2004). En revanche, les modèles de processus intentionnels se concentrent sur ce que le processus essaie d'atteindre, offrant ainsi la raison d'être du processus, à savoir pourquoi le processus est exécuté. Le modèle MAP permet de spécifier les processus d'une manière flexible en mettant l'accent sur les intentions de processus et sur les différentes manières d'atteindre chacune de ces intentions, il est donc d'une nature téléologique. Une carte (une instance du méta-modèle) est présentée comme un diagramme dont les nœuds sont les intentions et les arcs les stratégies. Les arcs orientés montrent quelles sont les intentions qui peuvent précéder ou suivre les autres. Un arc entre dans un nœud si sa stratégie associée peut être utilisée pour réaliser l'intention cible (le nœud). Comme il peut y avoir plusieurs arcs entrant dans un nœud, cela représente tous les moyens pour réaliser une intention. Le choix entre les différentes alternatives se fait en fonction de la situation. Cela permet une grande flexibilité lors de la réalisation des processus et la prise en compte pertinente de la sensibilité au contexte. Une carte offre une vue synthétique de la variabilité d'un processus qui est relativement simple à comprendre. Les variations sont de deux natures : soit elles sont le produit du mouvement graduel entre les différents niveaux de granularité de la carte, soit elles sont représentées par les chemins/stratégies alternatives à un moment donné (Rolland 2007a).

Comme le montre l'exemple donné en Figure 2, une carte est présentée comme un diagramme dont les nœuds sont les intentions et les arcs les stratégies. Cette carte représente le processus de l'approche CREWS-L'écritoire et est décrite en détails dans (Ralyté 1999) et (Rolland 1998b).

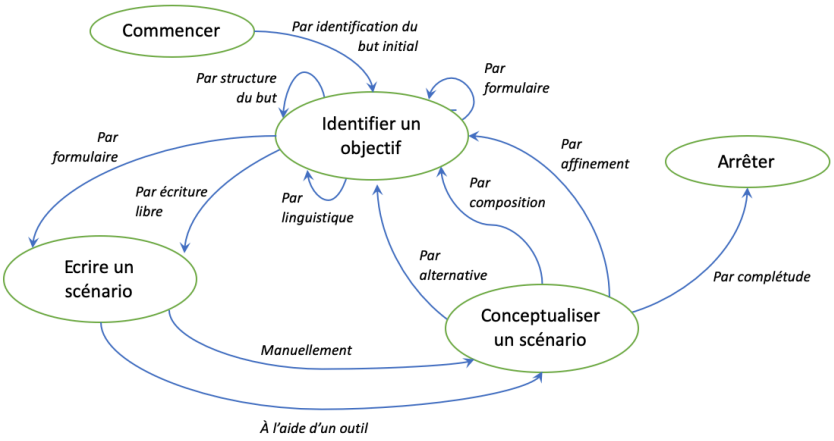


Figure 2. Un exemple de carte - celle de l'approche CREWS-L'écritoire (Ralyté 1999)

Les arcs orientés montrent quelles sont les intentions qui peuvent précéder ou suivre les autres. Un arc entre dans un nœud si sa stratégie associée peut être utilisée pour réaliser l'intention cible (le nœud). Comme il peut y avoir plusieurs arcs entrant dans un nœud, cela représente tous les moyens pour réaliser une intention.

Le but de cette modélisation est de décrire ce que l'entreprise veut réaliser ou éviter. Raisonner sur les objectifs facilite la compréhension des problèmes et la communication sur les aspects essentiels (quoi et pourquoi, au lieu de qui, quand, où et comment) (Nurcan, 2004).

Le méta-modèle de processus MAP est défini à la Figure 3. Le concept principal de ce méta-modèle est celui de Section. Une carte est composée d'un ensemble de sections. Une section est associée avec deux intentions, l'intention source et l'intention cible, reliées entre elles par une stratégie. Elle intègre les connaissances correspondant à une étape spécifique du processus pour réaliser une intention (l'intention cible) à partir d'une situation spécifique (l'intention source) en suivant une technique particulière (la stratégie). L'intention est un objectif qui peut être atteint par l'exécution d'un processus. Chaque carte comprend deux intentions prédéfinies : « Démarrer/Commencer » et « Arrêter/Terminer » (début et fin du processus). Une manière spécifique de réaliser une intention est capturée dans une section de la carte, sachant que toutes les sections ayant les mêmes sources et cibles permettent de représenter les différentes stratégies qui peuvent être utilisées pour réaliser cette intention cible. De la même façon, il peut y avoir plusieurs sections avec la même intention source, mais des intentions cibles différentes. Cela permet de montrer toutes les intentions qui peuvent être atteintes après la réalisation de l'intention source.

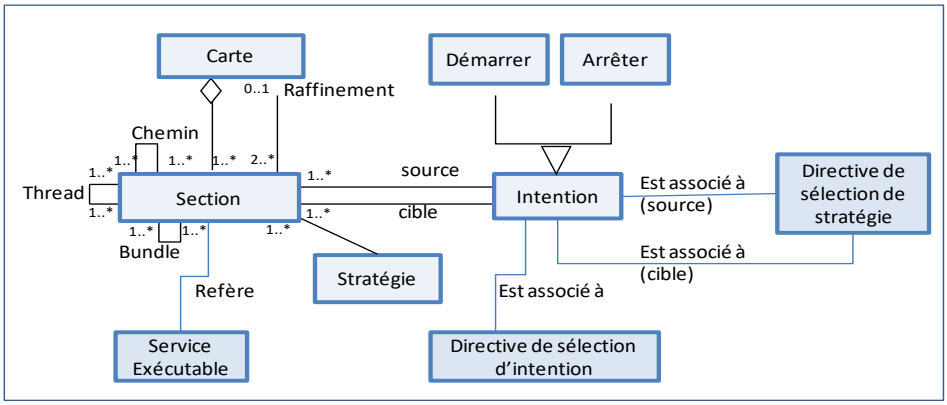


Figure 3. Le méta-modèle de processus MAP

Les sections sont à un niveau d'abstraction qui permet de capturer la variabilité du processus puisque chaque section correspond à l'application d'un service exécutable. Comme dans (Rolland 2007b) (Rolland 2007a), notre point de vue est qu'une section représente une caractéristique visible du processus et une abstraction d'un flux cohérent d'activités exprimées de manière intentionnelle. Chaque section de la carte identifie ce que nous pouvons appeler une variante.

Les variantes (sections) d'une carte sont liées les unes aux autres par trois types de relations (thread, bundle et chemin) qui génèrent des multi-threads et des multi-chemins dans une carte (Prakash 2006).

– Un *thread* montre la possibilité d'atteindre la même intention cible, à partir d'une intention source unique, de plusieurs façons. Chacune de ces techniques est exprimée dans la carte comme une section spécifique. L'ensemble des sections participant à une telle relation est appelée un multi-thread. Cette relation exprime directement la variabilité

associée à un processus multi-objectifs puisque chaque but est intégré dans les différentes stratégies permettant d'atteindre l'intention cible commune (Rolland 2007a).

- Un *chemin* établit une relation de précédence entre les sections. Pour qu'une section puisse succéder à une autre, son intention source doit être l'intention cible de la section précédente. Une carte contient un nombre fini de chemins allant de l'intention Start à l'intention Stop. Chacun d'eux est un modèle de processus. En ce sens la carte peut être considérée comme un multi-modèle. En naviguant dans la carte l'ingénieur de développement sélectionne le chemin et construit dynamiquement le modèle de processus qu'il applique. Aucun chemin n'est recommandé *a priori* ; au contraire c'est en fonction des situations qu'il rencontre que l'ingénieur choisira un chemin plutôt qu'un autre. Avec les relations de thread et de chemins, un but peut être atteint par plusieurs combinaisons de sections. Un tel ensemble est appelé un multi-chemins et peut contenir plusieurs threads. La relation multi-chemins est une autre façon de représenter la perspective pluri-objectifs du modèle de processus (Rolland 2007a).
- Un *bundle* indique la possibilité, pour plusieurs sections, ayant les mêmes intentions sources et cibles, d'être mutuellement exclusives.

Une relation d'affinement montre qu'une section de la carte peut être affinée par une autre carte. C'est un mécanisme d'abstraction par lequel un assemblage complexe de sections au niveau i +1 est considéré comme une section unique au niveau i. Cela introduit le concept de niveaux dans la représentation du processus puisque cela implique que chaque carte peut être représentée comme une hiérarchie de cartes. Cette hiérarchie permet d'obtenir une carte de haut niveau du processus ainsi que des cartes affinant ses sections particulièrement stratégiques, et ceci de manière récursive.

Il existe des directives associées. La première est la directive de sélection d'intention (DSI). Elle définit l'ensemble des intentions qui peuvent être obtenues à l'étape suivante (lorsque la section précédente aura été réalisée) et propose des arguments de sélection. La seconde est appelée directive de sélection de stratégie (DSS). Elle complète la première car elle détermine toutes les stratégies pouvant être utilisées pour atteindre le but visé (identifiée avec la directive précédente) ainsi que les arguments permettant à l'ingénieur de faire son choix.

Chaque section est reliée à un service exécutable. Il fournit un moyen opérationnel de réaliser l'intention. Il implique la transformation du produit en cours d'élaboration par l'exécution d'un service (qui peut être une directive, un workflow, un algorithme, etc.). On l'appelle aussi la directive de réalisation de l'intention (DRI).

Chaque section est décrite avec un descripteur qui permet de savoir dans quels cas (dans quel contexte) elle peut être exécutée. Les descripteurs de chaque section permettent de donner les informations nécessaires à l'utilisateur pour savoir si la section est utilisable dans son contexte particulier. Ils peuvent contenir : la situation dans laquelle la section peut agir, la situation qui sera obtenue après utilisation, l'expertise de l'utilisateur, le type de processus impliqué, divers éléments de contexte, etc.

Différentes extensions ont été proposées sur le modèle de la carte, comme par exemple l'intégration de la variabilité des rôles (Deneckère 2011a), les contraintes d'alignements (Thevenet 2009) et bien d'autres encore.

I.1.2. Fonctionnement d'un processus intentionnel

Le fonctionnement d'un processus intentionnel peut être lui aussi représenté à l'aide d'une carte.

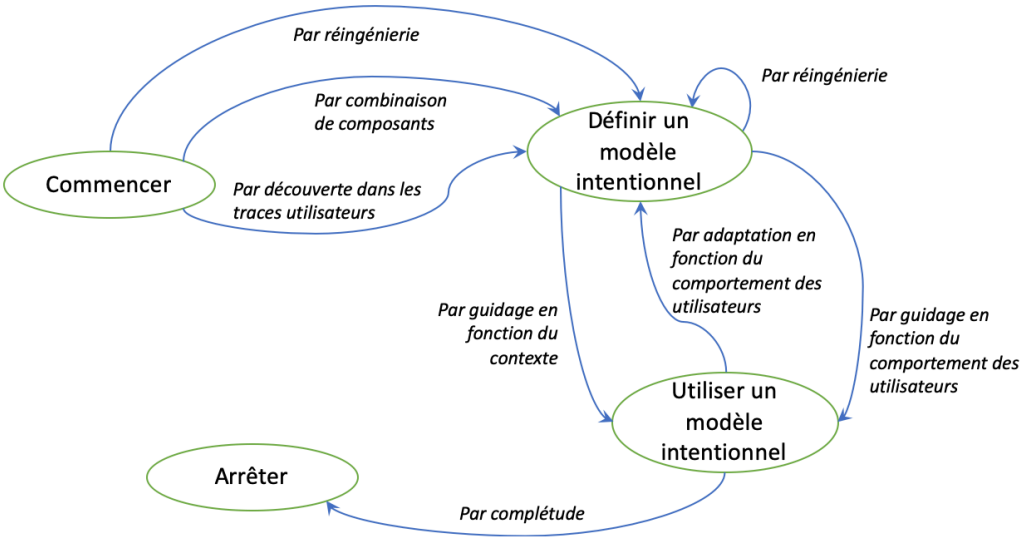


Figure 4. Processus de l'ingénieur de processus intentionnels

Il y a deux étapes principales correspondant au fonctionnement d'un processus intentionnel.

Il faut tout d'abord le définir avant de pouvoir l'utiliser. Il existe différentes manières de définir un modèle intentionnel : soit *par réingénierie* (en étudiant les documents existants, en interviewant les experts du métier, les processus potentiels déjà existants, etc.), soit *par combinaison de composants* (en identifiant des composants de processus que l'on peut ensuite agencer ensemble), soit *par découverte dans les traces utilisateurs* (en déduisant le processus directement à partir de ce qui est réellement exécuté). Une fois le processus identifié, il est aussi possible de l'améliorer en se basant sur les traces réelles des utilisateurs *par adaptation en fonction du comportement des utilisateurs*.

Une fois le processus construit, on peut l'utiliser et les processus intentionnels sont fournis avec des outils de guidage pour aider les utilisateurs à les exécuter : soit *par guidage en fonction du contexte*, soit *par guidage en fonction du comportement des utilisateurs*.

I.2. Mes travaux de recherche

Mes travaux de recherche s'organisent donc essentiellement autour de l'ingénierie des processus intentionnels. Ce domaine s'entrecoupe avec d'autres domaines et certaines notions développées sont utilisées ailleurs. J'ai notamment appliqué certains concepts en ingénierie des méthodes situationnelles et en fouille d'intentions.

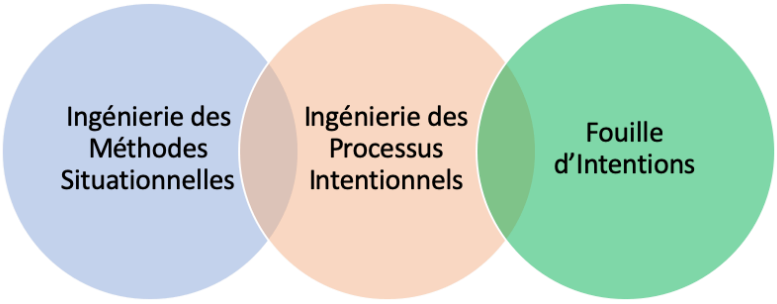


Figure 5. Domaines de recherche

Le tableau suivant montre l'évolution de mes publications en fonction du domaine concerné. La figure illustre également les principaux encadrements que j'ai pu faire dans ces domaines (j'ai omis tous les encadrements de masters qui ont eu moins d'impact, même s'ils ont donné lieu à des publications présentes dans le tableau).

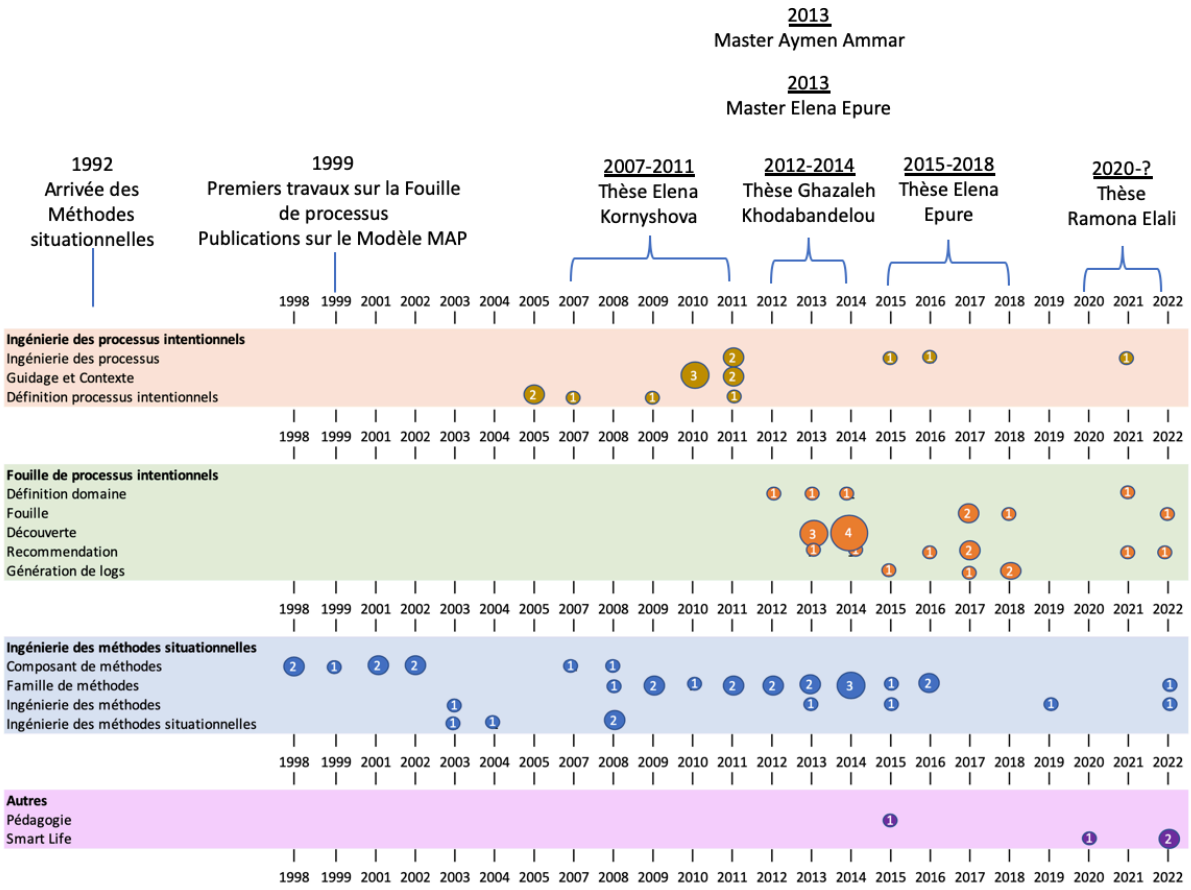


Figure 6. Évolution de mes publications en fonction du domaine de recherche

On peut constater que l'ingénierie des méthodes situationnelles est un domaine que j'affectionne particulièrement, y ayant commencé ma recherche lors de ma thèse, mais dans lequel je continue à publier régulièrement. La fouille de processus et, par extension, la fouille de processus intentionnels est un domaine auquel je me suis intéressée à partir de 2012 et dans lequel je suis toujours active aujourd'hui. J'ai bien entendu également des publications dans le domaine pur de l'ingénierie des processus intentionnels depuis 2005. J'ai également, de manière anecdotique, quelques publications dans d'autres domaines, notamment le domaine de la Smart Life (Deneckère 2020) (Alsayed 2022) (Kornyshova 2022a) (Kornyshova 2022c) dans lequel je m'investis depuis quelques années.

I.2.1. Mes principaux encadrements

Aymen Ammar : Aymen a effectué un stage de recherche en 2013 en ingénierie des méthodes sur la définition d'une famille de méthodes statistiques à destination des chercheurs en sciences humaines.

Elena Kornyshova : thèse soutenue en ingénierie des méthodes et prise de décision (co-encadrement avec Pr. Camille Salinesi). J'ai commencé à co-encadrer Elena pendant sa thèse, sous le conseil de son encadrant principal, qui avait vu le potentiel d'une collaboration entre nous dans le domaine de l'ingénierie des méthodes. Elena a brillamment soutenue sa thèse en 2011 et est maintenant Maitre de Conférence au CNAM (laboratoire Cédric). Nous continuons à collaborer ensemble sur de nombreux sujets (ingénierie des méthodes situationnelles, fouille de processus, Smart Life). Nous encadrons ensemble Ramona Elali, ma dernière doctorante en cours.

Ghazaleh Khodabandelou : thèse soutenue en fouille de processus intentionnels (co-encadrement avec Pr. Camille Salinesi). C'est avec Ghazaleh que nous avons défini le nouveau domaine de la fouille d'intentions et que nous avons commencé à publier dans ce domaine. Ghazaleh est maintenant Maitre de Conférences à l'université de Paris 12 (laboratoire LISSI). Elle co-encadre actuellement deux doctorants en intelligence artificielle.

Elena Epure : Elena a tout d'abord effectué un stage de recherche dans notre laboratoire sur le domaine de la recommandation en fouille d'intentions. Suite à son master et à une année dans le monde professionnel, elle a effectué une thèse en fouille d'intentions (co-encadrement avec Dr. Charlotte Hug et Pr. Camille Salinesi). Elena a soutenu une excellente thèse en 2018 et a obtenu deux prix de thèse (prix de thèse de l'université Paris 1 et prix de thèse de INFORSID). Elle travaille maintenant dans le service de recherche et développement de Deezer dans le domaine du traitement du langage naturel et encadre elle-même des doctorants (une thèse soutenue en 2021, une thèse en cours).

Ramona Elali : Ramona a commencé sa thèse en fouille de processus intentionnels en 2020, sous la direction du Pr. Camille Salinesi, Dr Elena Kornyshova et moi-même. Elle a trois publications dans le domaine.

I.2.2. L'ingénierie des processus intentionnels

L'ingénierie des processus est considérée comme essentielle à la fois par la communauté de l'ingénierie des logiciels et par la communauté de l'ingénierie des systèmes. Par exemple, (Dowson 1993) (Armenise 1993) (Jarke 1994) démontrent que l'amélioration d'un processus de développement conduira à une meilleure productivité et à une meilleure qualité des systèmes.

Les processus de développement traditionnels ont été conçus comme des suites d'étapes, chaque étape dominée par une activité particulière organisée en différentes séquences (par exemple, « l'analyse », « la conception », « le développement », « les tests » (Royce, 1970) (Boehm 1988). Plus tard, ces séquences d'activités ont été remplacées par des valeurs et des pratiques (Beck 2005) (Abrahamsson 2002). (Ralph 2008) met en évidence le fait que, dans la pratique, le développeur du logiciel est confronté à une myriade de choix méthodologiques, puisqu'il peut choisir d'exécuter une activité entièrement ou partiellement, combiner des aspects de deux ou plusieurs activités différentes, ou en ignorer une autre. Les décisions prises au cours de l'exécution conduisent l'ingénieur à adopter un processus qui est unique et complètement adapté à la situation en cours.

Toutefois, les points de décision ne sont souvent abordés que de manière partielle. La plupart des processus connus n'ont pas les moyens de prendre en compte la prise de décisions inhérente à tout processus de développement de systèmes d'information pour le développement. Selon Dowson (Dowson 1988), les modèles de processus peuvent être classés en trois groupes de modèles appelés orientés-activités, orientés-produit et orientés-décision. (Rolland, 1998) affine cette classification avec un quatrième groupe appelé orientés-contexte. Les processus orientés-décision et orientés-contexte prennent en compte la prise de décision en intégrant les concepts d'objectifs et d'arguments. Les modèles orientés-décision sont non seulement capables d'expliquer comment le processus est effectué, mais aussi pourquoi il est effectué (Rolland 1998) (Ralph 2008). Les modèles de processus orientés-contexte couplent la notion de contexte de décision à la décision elle-même (Grosz 1997) (Rolland 2000). Cela rend la notion de contexte (le couplage d'une situation et d'une décision) un concept central des processus de modélisation. Ces deux types de modèles de processus peuvent être définis comme des processus téléologiques.

Apport au domaine : Mes apports en ingénierie des processus intentionnels se situent à plusieurs niveaux (Figure 7) puisque j'ai travaillé à la fois sur plusieurs modèles de processus intentionnels métier (ingénierie des processus), mais également sur la notion de famille de processus (définition processus intentionnels). J'ai défini, avec Elena Kornyshova, un nouveau type de guidage basé sur l'utilisation de composants avec des descripteurs détaillés, en y ajoutant des indicateurs, pour pouvoir naviguer dans les processus intentionnels en sélectionnant les composants les plus pertinents selon le contexte (guidage et contexte).

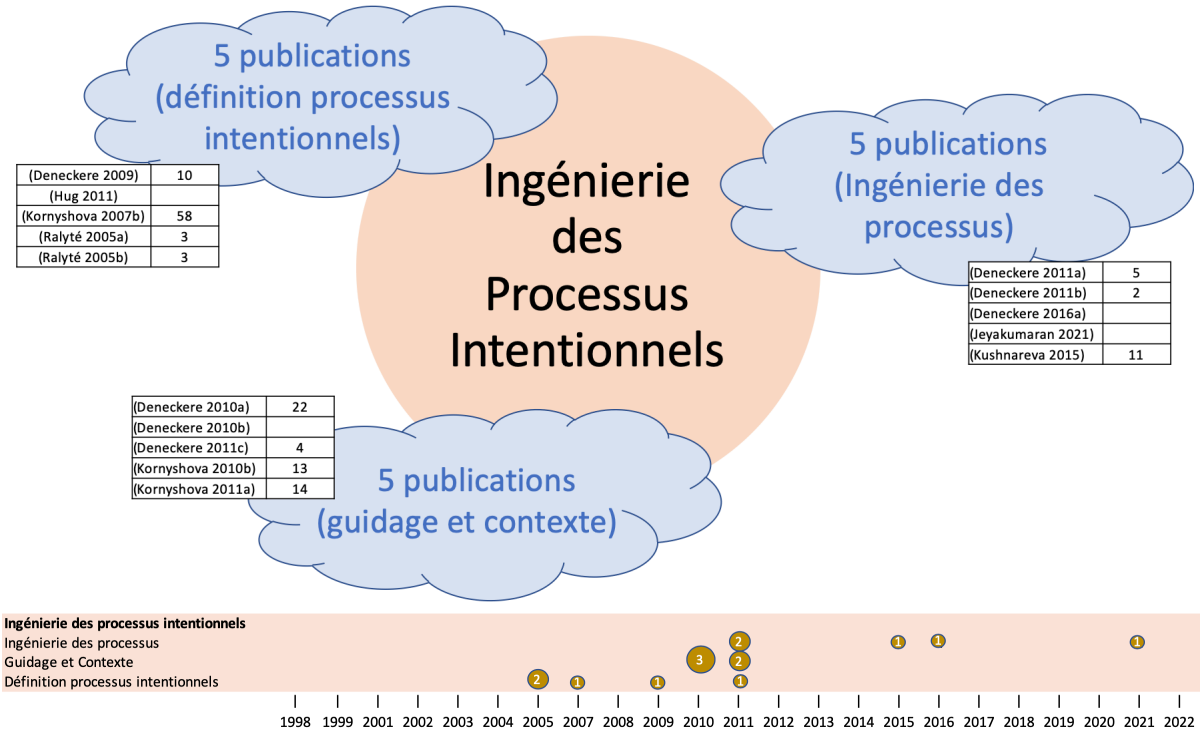


Figure 7. Apports au domaine de l'ingénierie des processus intentionnels (avec le nombre de publications et le nombre de citations par publication)

I.2.3. Application de l'ingénierie des processus intentionnels à l'ingénierie des méthodes situationnelles

Une méthode de développement de systèmes d'information est un ensemble d'idées, d'approches, de techniques et d'outils dont un analyste se sert pour transcrire les besoins d'une organisation en un SI approprié. Cependant, il est maintenant clair qu'il n'existe pas de méthode universelle pouvant s'appliquer à toutes les situations. Le domaine de l'ingénierie des méthodes situationnelles (IMS), depuis les années 90 (Kumar 1992), s'est attaqué à cette problématique pour offrir des techniques permettant la création ou l'adaptation de méthodes selon la situation du projet en cours (la situation d'ingénierie de chaque projet est différente et exige un support méthodologique différent). À cette fin, l'IMS promeut la construction de méthodes spécifiques aux contextes en réutilisant des parties de méthodes existantes conçues comme des composants autonomes et stockés dans des bases de méthodes. En effet, dans toutes les approches proposées, les méthodes sont décomposées en morceaux modulaires pouvant ensuite s'assembler ou s'intégrer les uns avec les autres. Cette notion fondamentale permet d'optimiser la création de méthodes en réutilisant différent morceaux appelés composants. (Ralyté 2001a) décrit le processus d'IMS en deux étapes : (1) la réingénierie des méthodes existantes en composants réutilisables qui seront stockés dans une base de méthodes, (2) la construction de méthodes situationnelles par composition de composants adéquats, ce qui inclut la spécification de la situation du projet et de ses besoins, la recherche des composants qui correspondent le mieux aux spécificités du projet et la construction d'une nouvelle méthode avec les composants sélectionnés (Mirbel 2006). La notion de composant de méthode a été introduite par (Harmsen 1997) par analogie avec la notion de composants logiciels (la construction modulaire d'une méthode à base de

composants de méthodes stockés dans une base de composants est similaire à la notion de construction de logiciels orientée-composants logiciels).

Cependant, malgré la diversité des différentes approches et des techniques que l'IMS propose (par assemblage (Brinkkemper 1998) (Firesmith 2002) (Ralyté 2001b), par extension (Deneckère 2001a), basée sur la configuration (Karlsson, 2004), basée sur l'adaptation de processus (Rossi 2004) ou encore l'orientée-service (Guzelian 2007)), leur usage n'est pas très répandu dans les entreprises car leur mise en œuvre semble difficile. Les entreprises tardent à adopter les approches et techniques proposées bien que conscientes de l'importance du rôle que les méthodes jouent dans leurs activités d'ingénierie. Nous pouvons imaginer les causes probables de cette réticence à l'utilisation des approches IMS. Tout d'abord, les approches ne sont pas interopérables car toutes les approches sont extrêmement liées à la structure même de leurs composants, ce qui fait que les techniques développées dans une approche ne sont pas toujours utilisables dans les autres. Ensuite, les techniques pour retrouver un composant sont assez complexes et peuvent nécessiter d'effectuer des recherches dans plusieurs bases de méthodes différentes. De plus, comme la plupart des approches ne proposent pas de processus organisationnel pour la gestion des composants, cela complique encore le processus d'exécution de la construction de la méthode. Qui dit méthode situationnelle dit situation, or la prise en compte de la caractérisation d'un projet n'est pas toujours très formalisée, de même que la caractérisation d'un composant et la comparaison possible entre les deux. Mais le frein majeur que rencontre la popularisation des méthodes en entreprise est que celles-ci soutiennent que les processus IMS consomment trop de temps et sont trop chers en termes de connaissances et de ressources pour les mettre en œuvre dans des projets d'entreprise. Ils exigent des approches et des outils simples et agiles, faciles à mettre en œuvre et ne requérant pas de vastes connaissances en ingénierie des méthodes.

Apport au domaine : J'ai eu plusieurs publications dans le domaine, à la fois en ingénierie des méthodes et en ingénierie des méthodes situationnelles (Figure 8). J'ai notamment défini un nouveau type de composant appelé patron d'extension permettant d'étendre une méthode existante en fonction des besoins (composants de méthodes) (Deneckère 2001a). J'ai également défini, avec Elena Kornyshova et Colette Rolland, une nouvelle approche d'ingénierie de méthodes situationnelles appelée Famille de Méthode, offrant à la fois la flexibilité du choix des composants de méthodes mais également un processus organisationnel permettant cette sélection de composants en fonction du contexte, processus exprimé sous forme intentionnelle (famille de méthodes). Cette approche a été utilisée avec succès dans plusieurs domaines.

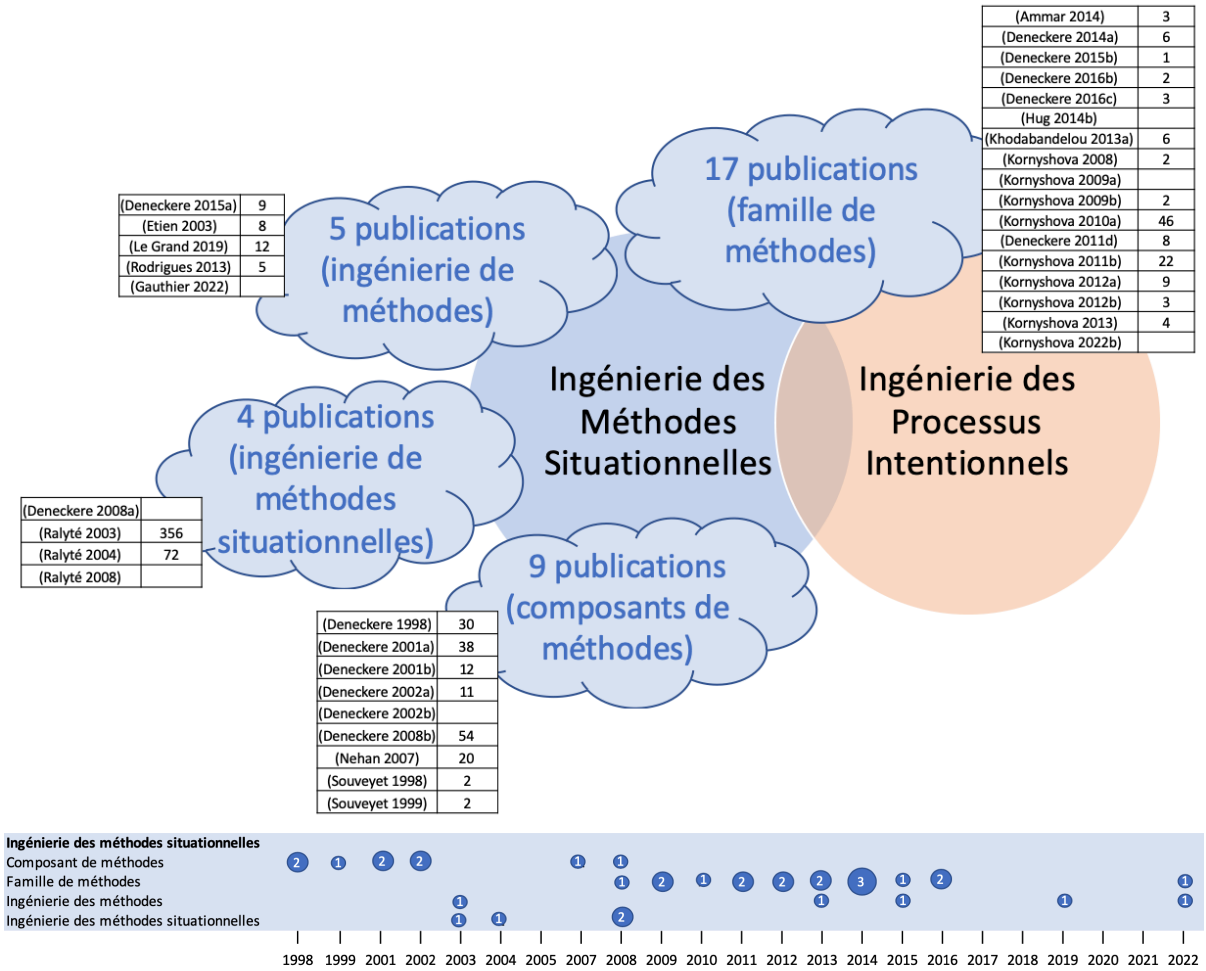


Figure 8. Apports au domaine de l'ingénierie des méthodes situationnelles (avec le nombre de publications et le nombre de citations par publication)

I.2.4. Application de l'ingénierie des processus intentionnels à la fouille d'intentions

Alors que de nombreux modèles de processus ont été proposés pour guider l'ingénierie des SI, La compréhension de leur mise en œuvre effective reste difficile. L'analyse des traces produites par les parties prenantes de projets d'ingénierie de SI permet, entre autres, de détecter l'écart entre le modèle de processus prescrit et ce qui est réellement accompli par les acteurs. Plusieurs approches de fouille de processus ont été proposées pour découvrir des modèles de processus à partir de logs d'événements ou traces (Greco 2006) (van der Aalst 2004a) (van der Aalst 2004b). Bien que ces méthodes permettent de récupérer des séquences d'activités et d'en extraire des informations, elles ignorent les intentions qui les génèrent. Les intentions sont importantes car elles capturent ce que les intervenants ont l'intention d'effectuer (Rolland 1999). Pour atteindre une intention, les acteurs exécutent un ensemble d'activités, ils peuvent également changer d'intentions lors de l'exécution d'un processus et ont parfois des comportements aléatoires. En conséquence, nous pensons que les séquences d'activités représentent un ensemble de comportements exécutés les uns après les autres, alors que le processus fondamental derrière ces activités est souvent intentionnel. La découverte de modèle de processus intentionnels à partir de logs d'événements fait partie

d'un nouveau domaine de recherche que nous appelons fouille d'intentions. L'objectif principal de la fouille d'intentions est d'extraire, à partir de logs d'événements, les intentions des acteurs liées aux activités générant ces évènements. Connaître les intentions sous-jacentes des activités des acteurs permet d'améliorer le guidage. En effet, si nous savons pourquoi un acteur effectue certaines activités, nous pouvons avoir une meilleure vision de ce qu'il peut faire par la suite et ainsi lui offrir de meilleures recommandations.

La fouille de processus définit quatre objectifs qui sont aussi applicables à la fouille d'intentions (Khodabandelou 2013d) :

- Découverte : Identifier les intentions et les stratégies des acteurs à partir des logs pour définir le modèle de processus intentionnel.
- Conformité : Vérifier la conformité d'un modèle de processus intentionnel prescrit par rapport à ce qui est réellement effectué par les utilisateurs.
- Amélioration : Utiliser les mesures de conformité pour définir si le modèle prescrit peut être amélioré en fonction du modèle réellement exécuté identifié à partir des logs.
- Recommandation : Utiliser la base de traces et le processus intentionnel pour guider l'utilisateur pendant l'exécution de ses activités, en se basant sur les intentions que l'on peut déduire de son comportement.

Apport au domaine (Figure 9) : Après avoir commencé à étudier la fouille de processus (fouille), j'ai défini, avec Camille Salinesi et Charlotte Hug, un nouveau domaine appelé la fouille de processus intentionnels, plus communément appelé fouille d'intentions. Nous avons défini, avec Elena Kornyshova, un cadre de référence pour la fouille d'intentions (Deneckère 2021), permettant de caractériser les différentes approches selon un ensemble de critères (définition domaine).

Nous avons proposé plusieurs techniques de découverte de processus intentionnels (découverte), ainsi que plusieurs approches de recommandation à la volée (recommandation). Avec Ghazaleh Khodabandelou, nous avons utilisé les chaines de markov cachées pour identifier des séquences d'activités pouvant représenter des stratégies, qui, une fois regroupées, permettent de définir une ébauche de processus intentionnel (découverte) . Avec Elena Epure, nous avons travaillé sur la découverte d'intentions à partir de textes en langage naturel (fouille). Avec Ramona Elali, nous travaillons sur la combinaison de sources hétérogènes pour la définition de processus intentionnels (fouille).

Nous avons réfléchi à la difficulté de pouvoir obtenir des logs d'activités et avons proposé plusieurs techniques pour pallier à ce problème (génération de logs). Nous nous sommes également intéressés à la fouille d'intentions en elles-mêmes (fouille).

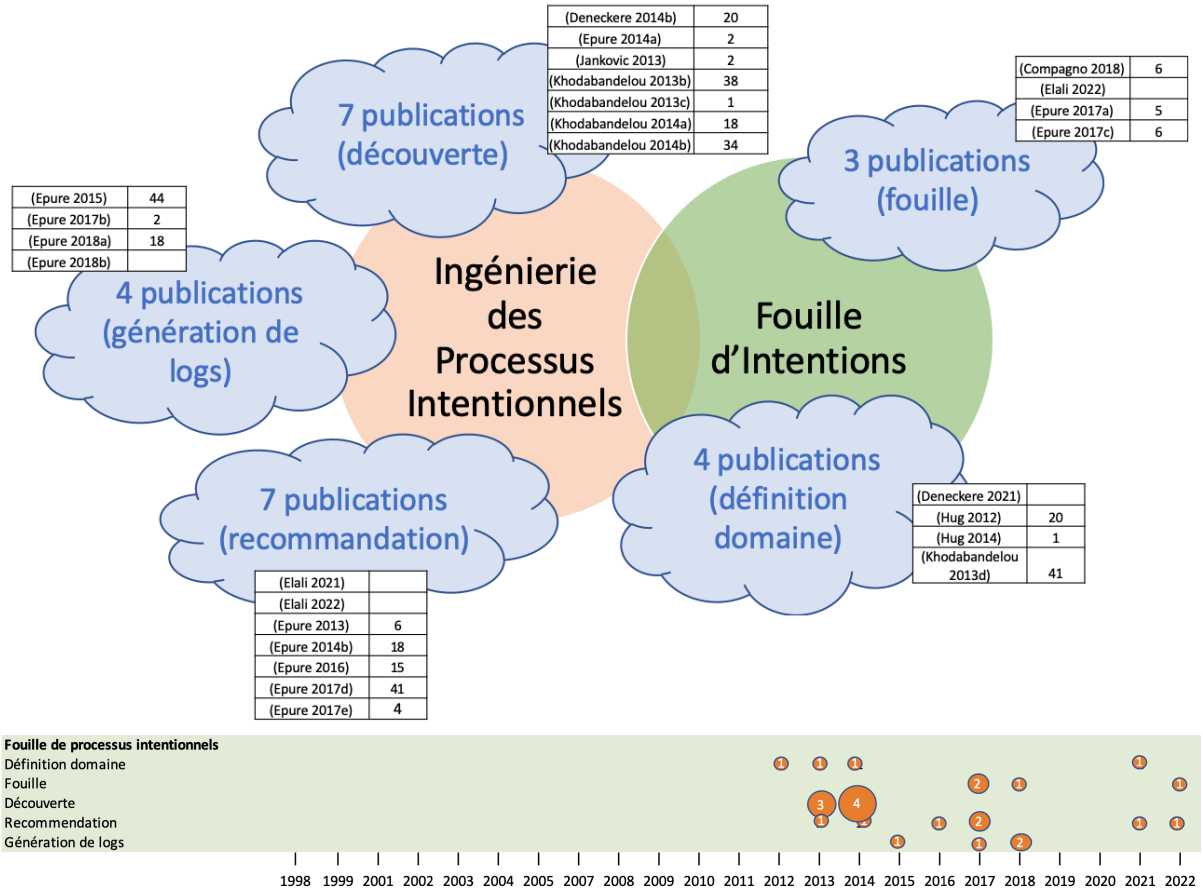


Figure 9. Apports au domaine de la fouille de processus intentionnels (avec le nombre de publications et le nombre de citations par publication)

I.2.5. Structure du document

La carte présentée en Figure 4 représente les différentes intentions et stratégies correspondant à la façon dont un ingénieur de processus intentionnels peut naviguer. La Figure 10 illustre les différentes approches que j'ai pu proposer au cours de ma carrière en relation avec elles et qui sont expliquées dans le reste du document.

La section II expliquera les différentes techniques apportées pour *définir un processus intentionnel*. La section III, quant à elle, illustrera les techniques pour *utiliser un processus intentionnel*. Nous conclurons avec la section IV qui interrogera sur certains challenges qui restent à résoudre dans ce domaine.

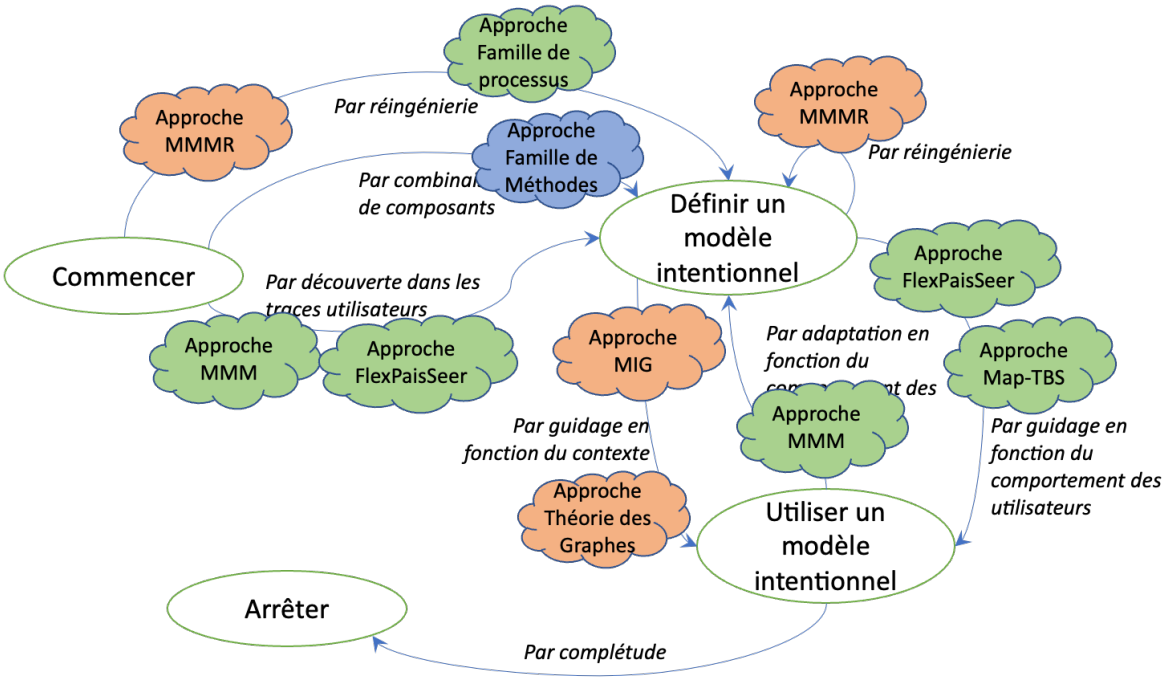


Figure 10. Approches pour l'ingénierie des processus intentionnels proposées dans ce document

II. Définir un modèle intentionnel

Cette intention peut être réalisée de 4 manières différentes en fonction de la situation en cours :

- Par réingénierie (ingénierie des processus)
- Par combinaison de composants (ingénierie des méthodes)
- Par découverte dans les traces utilisateurs (fouille de processus)
- Par adaptation en fonction du comportement des utilisateurs (fouille de processus)

La figure suivante montre les approches que nous avons proposé dans cet objectif : l'approche MMMR (section II.1.1), l'approche des familles de processus (section II.1.2), l'approche des familles de méthodes (section II.2.2), l'approche FlexPAISSeer (section II.3.1) et l'approche MapMinerMethod (section II.3.2 et II.3.3).

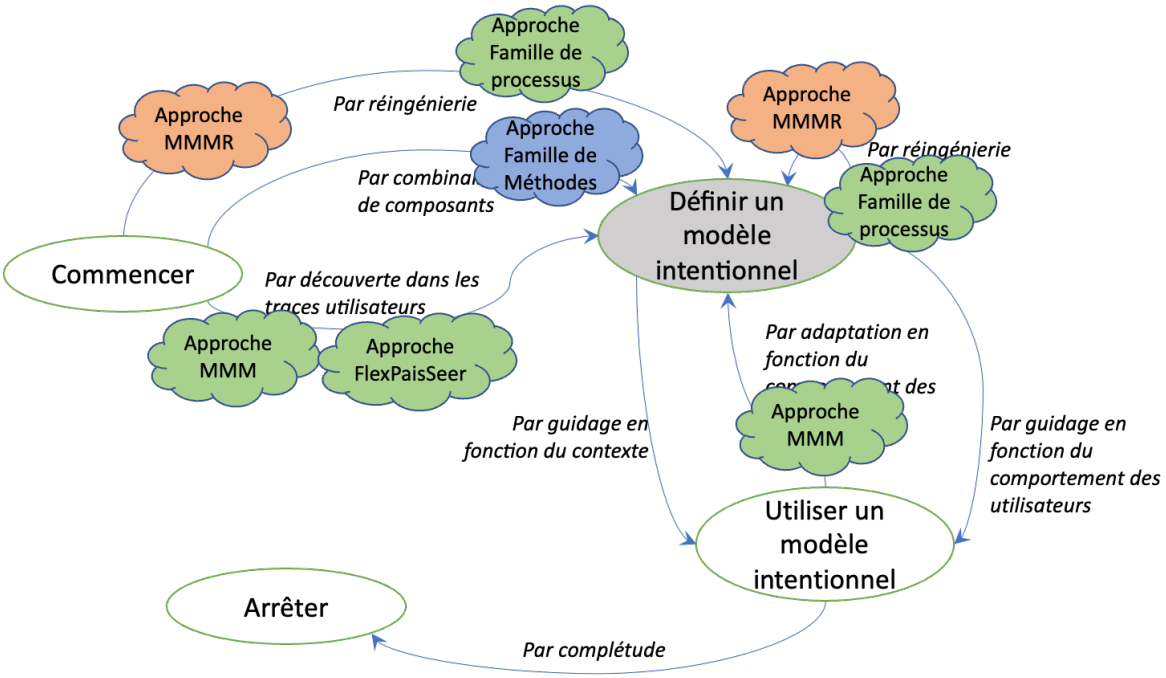


Figure 11. Approches proposées dans le cadre de la définition de processus intentionnels

II.1. Définir un modèle intentionnel par réingénierie

Au cours des dernières décennies, la documentation des processus métiers est devenue une composante essentielle de la capitalisation des connaissances dans les organisations (Koehler 2006) (Maurer 2004). Les utilisateurs ont confirmé l'importance de la modélisation des processus et cette activité s'est développée pour devenir indispensable dans la gestion des processus. Aujourd'hui, cependant, la gestion, la maintenance et l'utilisation efficace de la documentation des processus est devenu un problème important. Beaucoup d'entreprises tentent d'organiser et de maintenir de grandes collections de modèles de processus, mais d'autres décident simplement de renoncer à documenter leurs processus, estimant que l'effort n'en vaut pas la peine. Cela montre que les techniques classiques de modélisation de processus ne fournissent que très peu d'aide aux utilisateurs pour effectuer ces tâches.

Le principe de la construction d'un processus intentionnel par réingénierie (Figure 12) est donc d'utiliser la connaissance sous-jacente des processus que possèdent les entreprises pour la formaliser sous la forme d'un processus intentionnel. Cette connaissance peut se trouver sous la forme d'anciens processus formalisés, en échangeant avec les experts du domaine, en étudiants les documents jalonnant le processus métier concerné, etc. Cette stratégie est typique du domaine de l'ingénierie des processus.

Figure 12. Entrées-Sorties de la construction d'un modèle intentionnel par réingénierie

II.1.1. Construction manuelle d'un processus métier : l'approche MMMR

Nous représentons le modèle de processus comme une carte avec ses directives associées. Comme mentionné auparavant, la structure de la carte offre un haut degré de modularité et fournit les moyens d'évaluer un processus, de le décomposer en composants, de l'améliorer en ajoutant de nouvelles stratégies pour atteindre ses objectifs, etc. Ce modèle a été proposé et utilisé en ingénierie des méthodes dans le cadre de l'approche MMMR (Map-driven Modular Method Re-engineering approach) (Ralyté 2005a) (Ralyté 2005b).

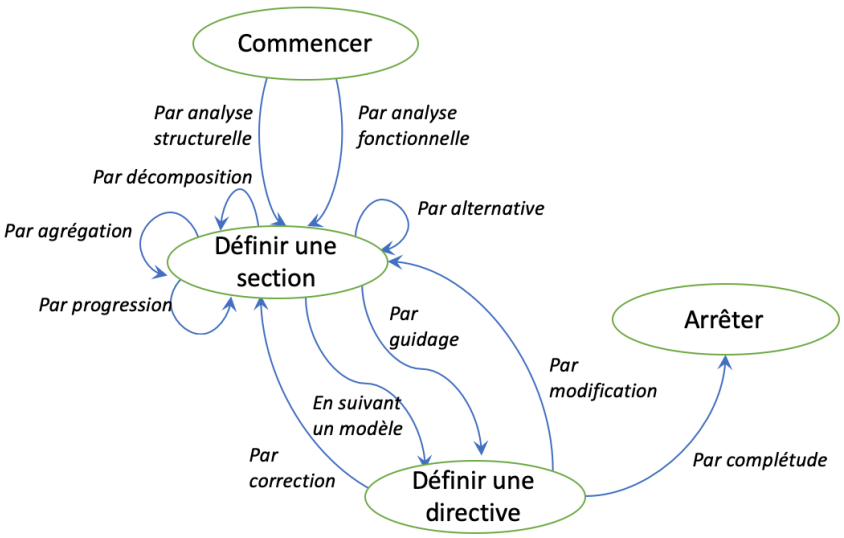


Figure 13. Processus de la construction d'un modèle intentionnel par réingénierie

Selon cette carte, la refonte ou la création d'un modèle de processus nécessite d'abord de le définir en termes de sections et de directives. Pour cette raison, la carte vise à atteindre deux objectifs principaux : *Définir une section* et *Définir une directive* et propose un ensemble de stratégies pour les satisfaire. Par exemple, il y a deux stratégies *Par analyse structurelle* et *Par analyse fonctionnelle* pour atteindre l'intention *Définir une section*. La stratégie d'analyse structurelle est recommandée lorsqu'un processus déjà construit ne fournit pas à l'ingénieur un modèle de processus formellement défini, mais une description simple du produit à construire. Cette stratégie utilise un glossaire des intentions de processus génériques pour appuyer la découverte des intentions. D'autre part, la stratégie d'analyse fonctionnelle devrait être utilisée si le modèle de processus défini est exprimé sous forme d'étapes et d'actions recommandées. Cette stratégie aide à identifier les sections à partir de ces actions et étapes.

Lorsqu'une section est définie, l'ingénieur peut soit définir les directives associées à cette section (pour progresser vers l'intention *Définir une directive*) ou définir de nouvelles sections (pour répéter l'intention *Définir une section*).

La définition des directives de sections consiste à décrire la directive de réalisation d'intention (le service exécutable) associé à chaque section, la directive de sélection d'intentions associée à un ensemble de sections ayant la même intention source et des intentions cibles différentes et la directive de sélection de stratégie associée à chaque ensemble de sections parallèles. La définition de ces directives est appuyée par deux stratégies : la stratégie *en suivant un modèle* et la stratégie *par guidage*. La première fournit un modèle pour chaque type de directive et fournit des conseils aux experts tandis que la seconde aide les novices en fournissant des recommandations plus détaillées.

La définition de nouvelles sections basées sur les sections existantes peut être réalisée de quatre manières différentes : *Par décomposition* d'une section existante en plusieurs, *Par agrégation* d'un ensemble de sections en une nouvelle, *Par alternative*, c.-à-d. avoir une stratégie alternative ou une source alternative ou une intention cible, et *par progression* qui

aide à définir une nouvelle section permettant de progresser dans la carte à partir de celle existante.

Les modifications des sections (décomposition, agrégation) impliquent la révision des directives associées si elles sont déjà définies. La stratégie *par modification* guide l'ingénieur de méthode pour réaliser ces transformations. De même, le processus de définition des directives peut impliquer la transformation de sections existantes. Par exemple, la décomposition d'une directive de réalisation d'intention pourrait mener à la décomposition de la section correspondante. De telles transformations peuvent être accomplies en suivant la stratégie de *correction*.

Le processus de création ou de modification du processus se termine par la stratégie de *validation par complétude*. Cette stratégie permet de vérifier si toutes les directives associées aux sections de la carte ont été définies.

Un exemple complet de l'utilisation de cette carte peut être trouvé dans (Ralyté 2005a).

II.1.2. Notion de famille de processus

L'utilisation des familles de processus permet d'avoir une approche de développement rapide et peu couteuse. Cette notion, tout d'abord proposée dans le domaine de l'ingénierie des logiciels, est orientée sur la production de variantes de processus (a) adaptées aux besoins de l'utilisateur, (b) cohérentes avec leur domaine d'utilisation et (c) consistantes avec la politique de l'entreprise concernée.

Nous avons proposé la notion de famille de processus (Deneckère 2011b) dans l'objectif d'avoir la possibilité de fusionner plusieurs lignes de processus, ayant la même intention globale, mais se différenciant par leur mode opératoire. Cette différence peut être due, par exemple, à l'existence de certaines règlementations différentes selon la politique interne de l'entreprise, le pays/état dans laquelle elle se trouve, etc. Une entreprise souhaitant avoir un développement international doit donc adapter ses processus pour obtenir des processus génériques, uniques mais potentiellement adaptables aux contraintes de chaque pays/état que l'entreprise cible dans sa stratégie d'internationalisation.

La variabilité croissante en génie logiciel a conduit à la notion de *ligne de produits*. L'ingénierie des lignes de produits correspond au développement d'applications logicielles utilisant des plateformes et de la personnalisation de masse, ce qui signifie que les points communs et les différence des applications de la ligne de produit doivent être modélisés de manière générique (Pohl 2005). Un produit d'une famille est défini en sélectionnant ou désélectionnant des caractéristiques (*features*) selon les préférences de l'utilisateur. De manière similaire, une famille de processus est configurable pour être personnalisée en ligne de processus, eux-mêmes adaptables à un projet spécifique (Rolland 2007b)

La modélisation des processus métier permet aux entreprises de décrire leurs processus, d'analyser leur structure et d'améliorer leurs performances. Cependant, l'accumulation de grandes collections de modèles de processus dans les entreprises représente un grand défi pour les utilisateurs de BPM (Business Process Management) aujourd'hui. Non seulement il

est difficile de naviguer dans ces collections et d'utiliser efficacement leur contenu, mais il est aussi très difficile de contrôler leur cohérence et leur intégrité. Les décisions prises au niveau de l'entreprise, l'évolution du processus engendré, peut provoquer de multiples erreurs à moins que toutes les collections ne soient contrôlées pour maintenir leur cohérence.

Souvent, les modèles de processus accumulés dans les collections représentent des versions, des modifications ou des personnalisations locales des mêmes processus. Pour modéliser ces ensembles de modèles de processus relativement similaires, la modélisation de la variabilité peut présenter certains avantages. Les modifications ou les versions d'un même processus (appelé variantes de processus) peuvent être agrégées dans des lignes de processus (par analogie avec les lignes de produits). De multiples lignes de processus peuvent être à leur tour regroupées en famille de processus. De nombreux travaux récents ont abordé cette problématique de l'utilisation des collections de processus de modélisation suivant diverses techniques.

Nous définissons un processus comme la trace d'une démarche exécutée pour un projet spécifique. Une ligne de processus représente une démarche flexible appliquée dans une entreprise, adaptable (configurable) aux différents projets de celle-ci. Une famille de processus est alors définie comme une intégration de plusieurs lignes de processus, générique et configurable selon différents critères. Ces notions sont mises en relations dans le méta-modèle suivant (Figure 14).

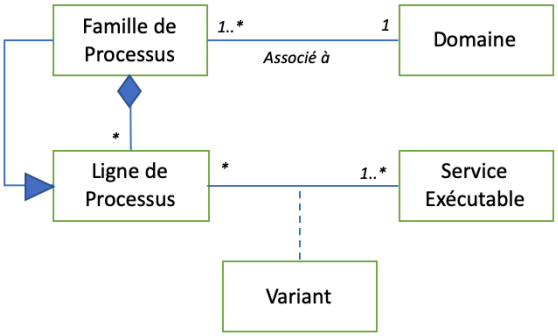


Figure 14. Meta-modèle des familles de processus

Une *famille* de processus est composée de *lignes* de processus, elles-mêmes associées à des *services exécutables* (représentant les différentes activités de la démarche, ce sont donc des composants de processus). Ces services représentent les alternatives possibles dans la ligne de processus - les *variants* - pouvant être, pour une ligne de processus, optionnels ou obligatoires). Nous pouvons également considérer une famille de processus comme une ligne de processus ayant un haut niveau de configuration. La distinction entre famille de processus et ligne de processus est parfois difficile à distinguer : une famille de processus est définie pour un *domaine* particulier alors que la ligne de processus est une configuration de la famille selon un certain nombre de caractéristiques.

Illustration

Nous illustrons notre propos avec un cas d'étude concernant les prêts bancaires immobiliers. Nous avons étudié deux processus existants ayant cours aux Etats-Unis dans deux institutions financières différentes. Nous allons expliquer succinctement ces deux processus pour pouvoir en illustrer les similarités et les différences. Les informations sur lesquelles nous nous sommes basés proviennent de différentes sources (e.g. http://www.homebuyinginstitute.com/, http://www.mortgage-resource-center.com/, http://homebuyereducation.bankofamerica.com/, http://www.homeloancenter.com/ etc.). Elles représentent une compilation de conseils, de recommandations et de descriptions de ce processus, provenant de différentes firmes, de conseillers financiers et de banques en ligne sur le web.

Les principales étapes du processus sont la pré-qualification, la demande de prêt, l'expertise des documents et accord préalable, l'estimation du bien immobilier, l'accord définitif et la fermeture du dossier. Les étapes de pré-qualification et d'accord préalable ne sont pas obligatoires mais conseillées pour une meilleure étude du dossier. L'objectif de l'étape de pré-qualification est de déterminer le montant potentiel que le demandeur peut obtenir. Le but de la demande de prêt est de construire le dossier du demandeur (situation financière en cours, situation professionnelle, historique des crédits). L'expertise des documents suit la demande de prêt et peut inclure une étape d'accord préalable. Le fait de fournir cet accord préalable signifie que le dossier du demandeur a été expertisé de manière très complète et que le montant du prêt a été défini. Cet accord préalable n'est pas obligatoire pour la suite du processus mais il peut s'avérer un atout majeur dans les négociations engagées par le demandeur dans sa recherche de bien immobilier. Une fois que le bien a été sélectionné, la banque demande à effectuer une estimation de celui-ci. Le résultat de l'estimation affine le montant du prêt. La banque prend ensuite la décision finale en se basant sur l'expertise des documents et les résultats de l'estimation. Si le prêt est accordé, une lettre d'engagement est signée par le demandeur et une date de fermeture est définie. La fermeture du dossier représente l'étape finale. Pendant cette étape, la banque achètera le bien immobilier et le conservera jusqu'au dernier paiement du demandeur.

Ce processus a des variantes qui peuvent parfois être très différentes selon la taille de l'entreprise (les petites et moyennes d'un côté et les grandes et très grandes de l'autre). Les petites et moyennes entreprises ont peu de ressources et peu de personnel. Elles n'ont, par exemple, pas les moyens d'investir dans du personnel de support téléphonique et les démarches que font les demandeurs de prêts sont essentiellement faites en rendez-vous, ce qui n'est pas le cas des grandes entreprises qui, elles, peuvent investir dans de nouvelles technologies. Ayant peu de ressources, les petites entreprises demandent au demandeur de payer des frais de dossiers avant de pouvoir l'étudier. Alors que les grandes entreprises ont dans leur personnel des experts internes, les petites entreprises doivent également faire appel à des expert externes sous-traitants à qui le demandeur devra payer l'estimation du bien.

Dans le but de rendre le processus d'attribution de prêts plus générique et plus portable d'une institution à l'autre, nous avons proposé un processus intégrant les particularités des deux précédents processus (Figure 15).

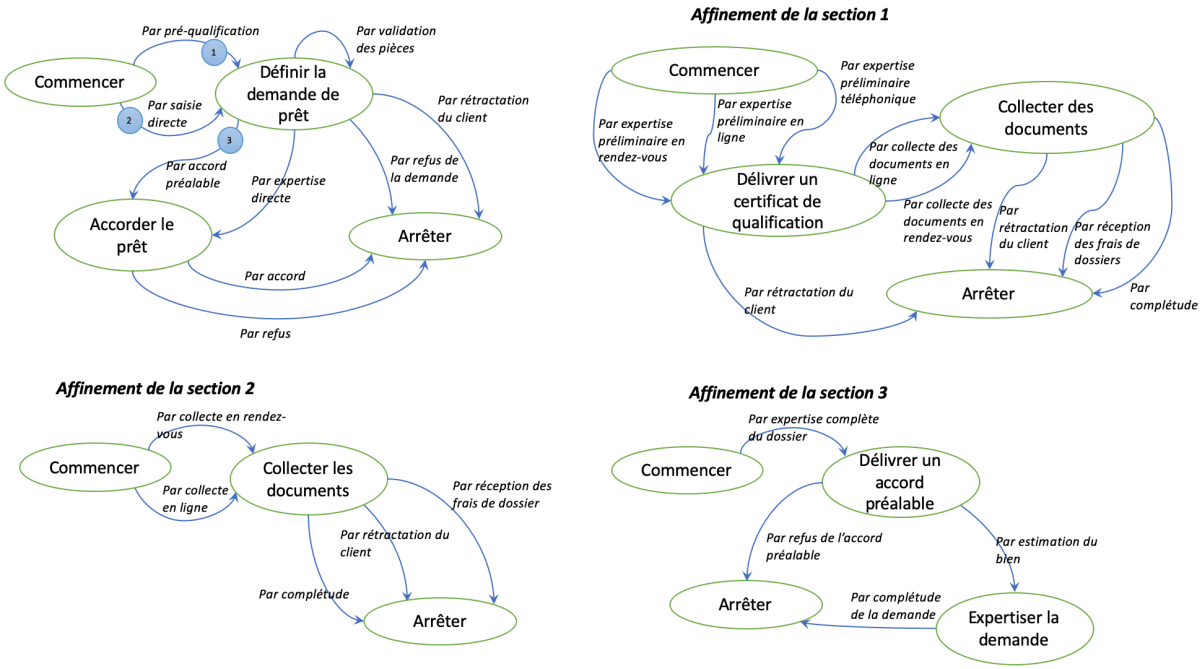


Figure 15. Famille de processus d'accord de prêt aux États-Unis

Nous pouvons imaginer quelques exemples d'avantages engendrés par cette formalisation. Par exemple, si l'institution financière procure un service de support centralisé, alors même les petites agences pourront délivrer le certificat de pré-qualification en ligne, même si cela ne faisait pas partie de leur processus d'origine. Ou encore, si l'institution financière souhaite avoir une politique commune dans toutes ses agences, qu'elles soient grandes ou petites, alors certaines étapes devront être les mêmes pour toutes les agences concernées. En conséquence, la ligne de processus proposée aux agences répondra immédiatement à la modification de la politique de l'institution financière.

II.2. Définir un modèle intentionnel par combinaison de composants

Cette stratégie est utilisée lorsque l'on définit des composants de processus que l'on réorganise ensuite dans un processus intentionnel pour pouvoir les utiliser en fonction du contexte en cours. Comme l'indique la Figure 16, l'ingénieur de processus se base sur l'ensemble des connaissances qui lui sont accessibles (processus existants, entretiens avec les experts métiers, connaissances intrinsèques au domaine, méthodes d'applications, etc.) pour construire une base de composants de processus réutilisables qui seront ensuite assemblés les uns avec les autres pour définir le processus intentionnel.

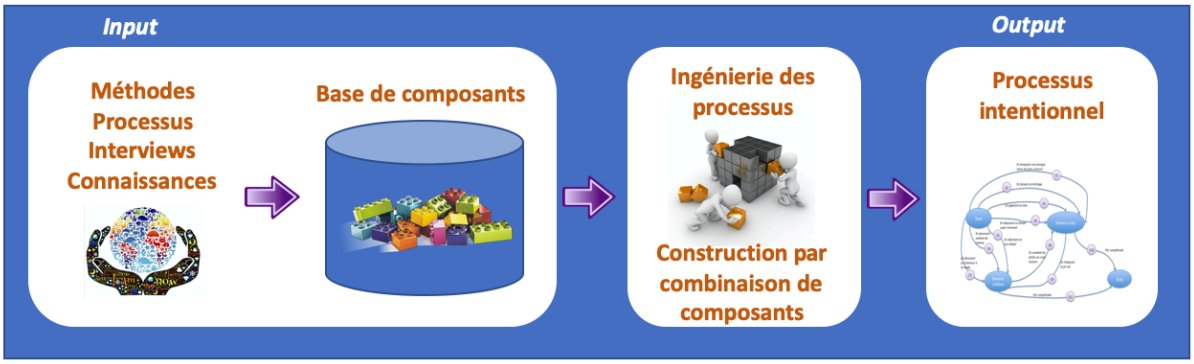


Figure 16. Entrées-Sorties de la construction d'un modèle intentionnel par combinaison de composants

J'ai utilisé cette technique pendant mes travaux en ingénierie des méthodes situationnelles, lors de l'élaboration d'un processus de construction de méthodes à la volée en fonction du contexte. Dans ce cas précis, les composants sont des composants de méthodes et le processus intentionnel construit corresponds à une méthode ou à une famille de méthode (Figure 17).

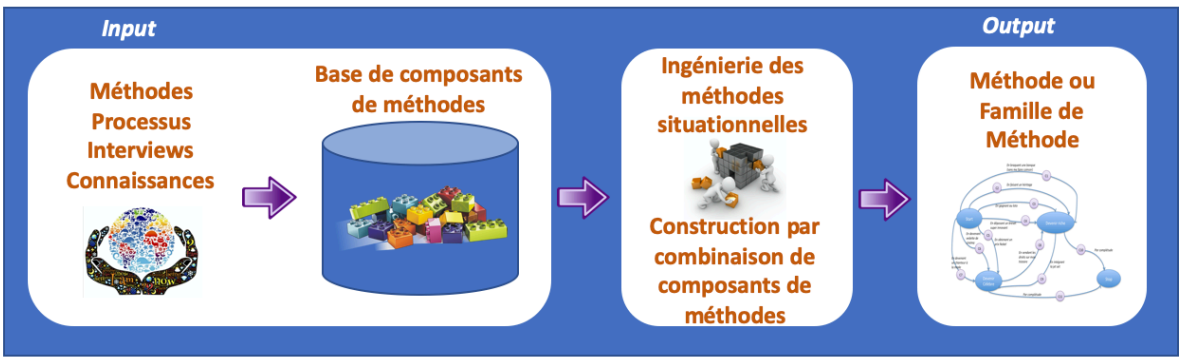


Figure 17. Entrées-Sorties de la construction d'un modèle intentionnel par combinaison de composants de méthodes

II.2.1. Composant de méthode

La notion de composant de méthode a été introduite par (Harmsen, 2007) par analogie avec la notion de composants logiciels (la construction modulaire d'une méthode à base de composants de méthodes stockés dans une base de composants est similaire à la notion de construction de logiciels orientée-composants logiciels). Un *composant de méthode* est une

partie d'une méthode suffisamment autonome pour pouvoir être appliquée de manière indépendante mais qui peut également être combinée avec d'autres composants. Il existe beaucoup de types de composants dans le domaine de l'IMS : fragments (Brinkkemper 1996), chunks (Ralyté 2003), components (Wistrand 2004), OPF (Open Process Framework) fragments (Henderson-Sellers 2002), and method services (Guzélian 2007) (Deneckère 2008b). Une comparaison des différents types de composants a été proposée dans (Nehan 2007).

Lors de ma thèse, j'ai proposé la notion de patron d'extension permettant de construire des composants de méthode à la volée pour étendre une méthode existante et l'adapter en fonction du contexte (Deneckère 2001a). Le concept de composant de méthode que j'ai utilisé ensuite dans mes travaux a la même sémantique que dans les approches de construction de méthodes situationnelles classiques, il s'agit d'un bloc de construction réutilisable, sous-ensemble d'une méthode, qui peut être composé d'un ensemble d'autres composants. Cependant, les différentes approches existantes se focalisent essentiellement sur la description du type de composant utilisé, alors que je considère que le composant n'est qu'une boite noire représentant une partie de méthode, indépendamment de sa structure. Cette distinction permet donc de combiner des composants venant de méthodes différentes, ce qui n'est pas possible avec les approches situationnelles classiques (voir (Henderson-Sellers 2008) pour plus de détails). Une proposition de modélisation modulaire a été présentée dans (Deneckère 2008a).

II.2.2. Famille de méthode

Le concept de famille de méthodes (Asadi 2011) (Kornyshova 2011a) (Deneckère 2011d) (Kornyshova 2012b) (Kornyshova 2013a) (Deneckère 2014a) (Deneckère 2015b) apparait comme une manière de fournir des processus de construction de méthodes aux entreprises de développement de systèmes d'information, chacune consacrée à un certain domaine d'application particulier et/ou un but d'ingénierie spécifique, pouvant être facilement configuré dans des lignes de méthodes spécifiques à un projet selon les caractéristiques et les exigences de ce projet. L'inspiration initiale de ce concept vient de l'ingénierie des lignes de produits logiciel (LdP) et en particulier du concept de Famille de Produit (Clements 2001) (Weiss 1999). L'ingénierie des LdP est un paradigme permettant de développer des applications (systèmes logiciels et produits logiciels) avec de la personnalisation massive. Il est basé sur l'identification et la spécification des similitudes et différences entre applications d'une ligne de produits (Pohl 2005).

La spécification d'un produit logiciel prend toutes les exigences communes de la ligne de produits et quelques exigences spécifiques du logiciel à développer. De même dans l'IMS, nous considérons que la notion de famille de méthodes simplifierait la tâche d'ingénierie de méthode pendant l'exécution du projet en la réduisant à une configuration de la famille de méthodes, c'est-à-dire la sélection de composants de méthode communs et spécifiques de la famille de méthodes. Bien sûr, ceci présume que la famille de méthodes existe avant le démarrage du projet. À cette fin, nous proposons une approche d'ingénierie basée sur les familles de méthodes et inspirée des LdP (la variabilité) et de l'IMS (la réutilisation et la composition de composants de méthode). L'IMS est en général basée sur la décomposition de méthodes existantes en composants de méthode et leur réutilisation pour la construction de

nouvelles méthodes (Ralyté 2001a). Notre proposition tente de répondre à la problématique des entreprises en envisageant la construction de méthodes situationnelles sous un autre angle de vue, celui de la configuration. D'après les principes de l'IMS, notre approche d'ingénierie par famille de méthodes réutilise des méthodes et/ou des composants de méthode existants, dans le but de fournir des familles de méthodes satisfaisant des organisations et/ou des projets de développement divers. La tâche de l'ingénieur de méthode sera réduite à la sélection d'une ligne de méthode simple de la famille de méthodes (processus appelé configuration de ligne de méthode), ce qui limitera également le risque d'erreurs et réduira le temps nécessaire pour produire des méthodes spécifiques à un projet.

Nous définissons une famille de méthodes comme une composition de composants alternatifs ou complémentaires, venant de méthodes diverses et variées mais applicables au même domaine d'application et ayant des objectifs similaires. Une famille de méthodes peut être considérée comme une multi-méthodes (multi-processus et multi-produits) dédiée à un domaine ou un but d'ingénierie particulier et cherchant à satisfaire un ensemble de projets de ce domaine. Une famille est multi-processus puisqu'elle offre un ensemble de démarches possibles pour atteindre le même objectif, et elle est multi-produits puisque les composants peuvent être combinés de différentes manières. L'adaptation d'une famille de méthodes à un projet (configuration) se fait en sélectionnant une ligne de méthode à l'intérieur de la famille. Une ligne de méthode est composée d'un sous-ensemble des composants de la famille, sélectionnés pour un contexte donné (un projet particulier). Cette démarche permet de simplifier la construction de la méthode en réduisant la complexité de la recherche de composants.

Les approches habituelles de l'ingénierie des méthodes sont fondées sur l'hypothèse qu'une méthode doit être composée « à la volée ». Le concept de familles de méthodes aborde cette question : les composants de méthodes du même domaine et ayant le même usage sont organisés dans une famille de méthodes avant leur utilisation dans des projets concrets. Cela permet de réduire la complexité de la gestion de la grande variété de méthodes et de composants de méthodes. Nous utilisons l'idée des variations et des points de variation du domaine des Ldp et nous l'appliquons au domaine de l'ingénierie des méthodes.

Le méta-modèle (Figure 18) comprend trois notions spécifiques : la notion de composants de méthode et leur organisation, la notion de variabilité et la notion de contextualisation.

L'organisation des composants au sein de la famille est très importante car elle permet de spécifier les points communs et la variabilité de méthodes similaires pour leur assurer une meilleure utilisation.

Un composant est définit par son *interface*, identifiant l'objectif principal de la famille, son domaine d'application et sa portée fonctionnelle. Elle peut inclure des composants venant de diverses méthodes (les méthodes initiales) qui correspondent à cette interface. Une analyse informelle des différents éléments permet d'identifier si une méthode initiale correspond à l'interface d'une famille de méthode (si son objectif est un sous-objectif de celui de la famille, si elle s'applique dans le même domaine d'application ou une sous-partie du même domaine et si l'étape du cycle de vie pris en compte fait partie de la portée fonctionnelle de la famille).

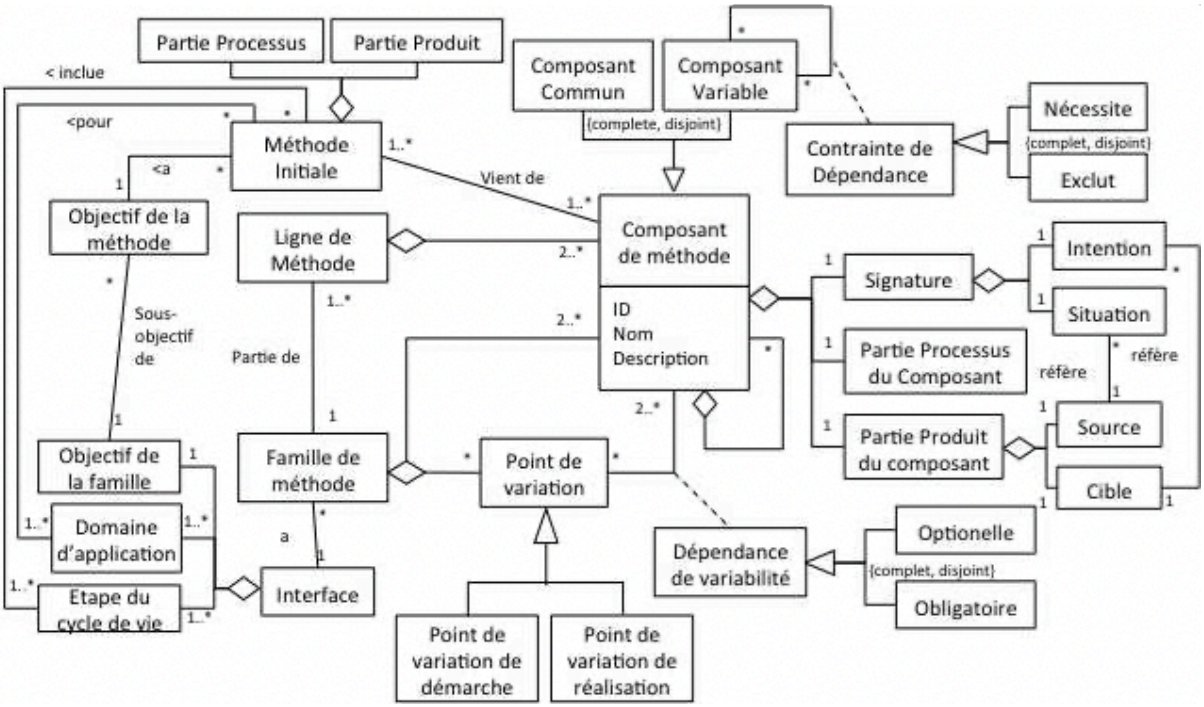


Figure 18. Meta-modèle de la famille de méthodes

Un composant de méthode correspond à plusieurs éléments du domaine des lignes de produit, comme les *features* et les *artefacts*. Une *feature* est un aspect, une qualité ou une caractéristique visible pour les utilisateurs finaux (Kang 1990). Les *features* d'un système permettent de décrire les éléments communs ou variables d'un aspect du système dans un domaine. Les *artéfacts* sont des éléments du développement réutilisables créés dans le sous-processus de l'ingénierie de domaine (Pohl 2005). Ces deux notions sont très liées au produit. En opposition, le composant de méthode contient à la fois du produit et du processus.

L'organisation des composants dans la famille consiste à déterminer ce qui est similaire et ce qui est différent dans les méthodes composant la famille. A l'intérieur d'une famille de méthodes, les composants peuvent donc être considérés comme variables ou communs. Les composants communs correspondent à des composants qui seront présents dans toutes les situations du domaine. Les composants variables, au contraire, sont des composants qui ne seront présents que dans certaines situations. Ces situations engendrant une certaine variabilité sont appelées des *points de variation*. Ceux-ci peuvent être de deux types : de démarche ou de réutilisation. Les points de variation de démarche concerne la variabilité attachée à une situation donnée et permet d'identifier les composants qu'il est possible d'utiliser dans ce contexte précis – les points de variation de réutilisation sont plus axés sur la situation cible et l'identification des composants pouvant atteindre cet objectif. La relation entre le point de variation et le composant définit la *dépendance de variabilité*, pouvant être obligatoire ou optionnelle. Cette représentation de la variabilité est inspirée de (Pohl 2005) où l'on trouve la définition de la variabilité des lignes de produits logiciels.

Un composant de méthode est considéré comme un composant *commun* s'il apparait dans toutes les méthodes initiales composant la famille. Il sera alors considéré comme obligatoire et devra être présent dans toute ligne de méthode générée à partir de la famille. Un composant de méthode *variable* est optionnel – il peut exister d'autres composants

permettant d'atteindre le même but mais avec des techniques différentes, ou la famille permet d'utiliser d'autres composants pour réaliser l'objectif principal. Les points de variation sont utilisés pour spécifier le contexte d'utilisation des composants lorsqu'une alternative est possible. Les composants variables associés à un point de variation peuvent appartenir à plusieurs méthodes différentes ou encore à la même méthode initiale, incluant cette notion d'alternative, de variabilité. La distinction entre composants variables et communs permet d'évaluer le degré de variabilité d'une famille et fournit les outils nécessaires à la configuration des lignes de méthodes puisque celles-ci sont composées des composants communs et d'un certain nombre de composants variables, choisis pour un projet donné selon le contexte.

II.2.3. Construction d'une famille de méthodes

Suivant les recommandations et principes de l'ingénierie des méthodes situationnelles, notre approche pour la construction de familles de méthodes est basée sur la réutilisation des composants de méthodes extraits des méthodes existantes et sur leur analyse de variabilité. En particulier, cette approche vise à soutenir la construction de la famille de méthodes à partir d'un ensemble de méthodes appartenant au même domaine d'application et ayant des objectifs similaires. Le processus de construction est représenté dans la figure suivante (Figure 19).

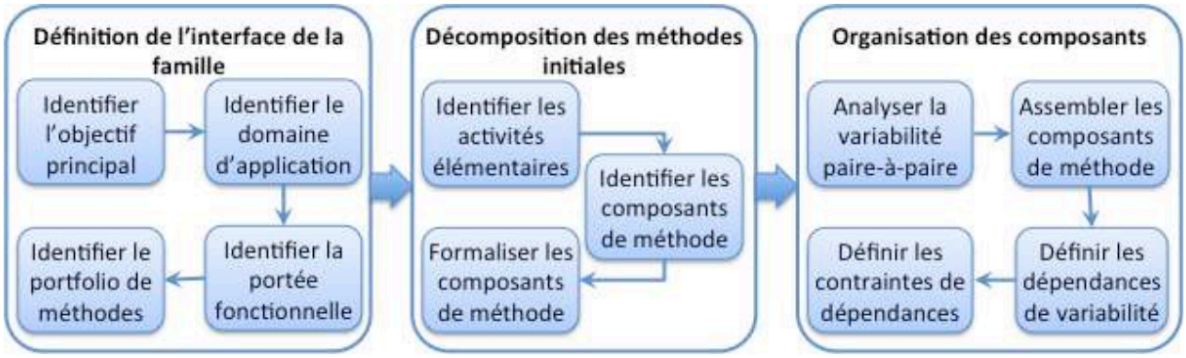


Figure 19. Processus de construction d'une famille de méthodes

Suivant les principes de l'ingénierie des méthodes situationnelles, la construction d'une famille de méthodes est basée sur la réutilisation des composants extraits de méthodes existantes. Ces méthodes doivent avoir le même domaine d'application et des objectifs similaires. Cependant, alors que les approches habituelles de construction de méthode se concentrent sur le processus de construction lui-même et sur ses nombreuses difficultés (comme effectuer une intégration lorsqu'il y a un chevauchement de concepts par exemple), les familles de méthodes offrent une manière de simplifier le travail des utilisateurs avec un détachement du processus de construction. L'ingénieur de méthode construit la famille de méthodes en décomposant en premier lieu les méthodes (avec les approches IMS classiques) et en les regroupant en une famille que l'utilisateur configurera pour obtenir une méthode adaptée à ses besoins. Le processus de construction d'une famille de méthode a été détaillé dans (Kornyshova 2013a).

A. Étape 1 : Définition de l'interface de la famille de méthode.

Différentes méthodes peuvent être fédérées dans une famille uniquement si elles satisfont à la propriété d'« unité téléologique » (i.e. qui ont le même objectif, la même intention de réalisation). Cette propriété caractérise la similarité de différentes méthodes selon leur usage exprimé par leurs intentions (Kornyshova 2011). Cela signifie que seules les méthodes ayant la même utilisation et créées dans le même but pourront être intégrées dans une même famille. Notre phase de cadrage intègre les idées de (Brinkkemper 1998), (Kornyshova 2011) et (Bucher 2007) et comprend quatre étapes : (1) définir l'objectif de la famille de méthodes (par exemple un soutien à la prise de décision), (2) préciser son domaine d'application (par exemple l'ingénierie des systèmes d'information), (3) identifier le portefeuille de méthodes (l'éventail de méthodes correspondant au but de la famille de méthodes et au domaine d'application), et (4) la définition de sa portée fonctionnelle (par exemple des exigences à la conception).

- *Identifier l'objectif principal.* Avant de se lancer dans la construction d'une famille de méthodes, il est nécessaire de comprendre quel sera son objectif principal. Une famille de méthodes peut être dédiée à un certain type de projets (e.g. le développement de systèmes ERP) ou à un certain type d'entreprise (e.g. les fournisseurs d'applications Web), ou encore pour des audiences très larges (e.g. le développement de services dans le secteur public). L'unification de méthodes similaires ou leur standardisation sont autant d'exemples d'objectifs de famille de méthodes.
- *Identifier le domaine d'application.* Le domaine d'application d'une famille de méthodes peut être plus ou moins étendu. Quelques exemples de domaines pourraient être: l'ingénierie du logiciel, la conception de services mobiles, l'ingénierie des besoins, la conception de systèmes d'information, etc.
- *Identifier la portée fonctionnelle.* Cette étape consiste à identifier la partie du cycle de vie prise en compte. Celle-ci peut couvrir l'ensemble du processus de développement ou juste une partie de processus (e.g. la phase d'analyse des besoins), ou encore une activité très spécifique (e.g. l'identification des besoins préliminaires).
- *Identifier le portfolio de méthodes.* La dernière étape consiste à identifier le spectre de méthodes correspondant aux précédents éléments identifiés, i.e. l'objectif principal, son domaine d'application et sa portée fonctionnelle. Seules les méthodes correspondant à cette même unité téléologique pourront être intégrées (Kornyshova 2013).

B. Étape 2 : Décomposition des méthodes en composants de méthode.

Les composants de la méthode sont extraits des méthodes existantes et composent le portefeuille de méthodes défini dans la phase précédente. La décomposition des méthodes est axée sur les processus et comprend trois étapes : (1) mettre en évidence les activités élémentaires de la méthode, (2) identifier les composants de la méthode et (3) les formaliser en suivant une approche d'ingénierie de méthode.

- *Identifier les activités élémentaires.* La décomposition d'une méthode consiste à décomposer son processus en un ensemble d'activités élémentaires. Une activité décrit un ensemble d'actions élémentaires comme : définir une classe objet, calculer une somme pondérée, etc. La description des activités peut être réalisée à l'aide de différents modèles de processus (par exemple avec des diagrammes d'activités UML).

- *Identifier les composants de méthodes.* Les activités élémentaires définies dans l'étape précédente sont regroupées sous la forme de composants de méthodes. Cette étape suit deux grands principes : l'indépendance et la complétude. L'indépendance signifie que chaque composant obtenu doit être autonome – il peut être appliqué séparément, dans différentes situations. La complétude signifie que si une méthode a été décomposée, la réunion de tous les composants issus de celle-ci doit pouvoir aboutir à la méthode initiale – aucune activité ne doit avoir été omise.
- *Formaliser les composants de méthodes.* La formalisation des composants dépend de l'approche de construction de méthode situationnelle choisie (orientée chunks, fragments, services ou autre) étant donné que notre proposition ne se limite pas à un type de composant précis. Nous utilisons ici des composants de type chunk (Ralyté 2001b) qui incluent plusieurs éléments : l'identification du composant (ID, nom, description, composants – si le composant est un agrégat – et signature comprenant la situation et l'intention), la partie produit (avant et après exécution du composant) et la partie processus.

C. *Étape 3 : Organisation des composants dans une famille*

La dernière phase consiste à organiser tous les composants de méthode définis précédemment au sein d'une famille de méthode. Elle est fondée sur (i) l'approche de construction de méthodes situationnelles fondée sur l'assemblage (Ralyté 2001b) (Ralyté 2004) (Ralyté 2008) et (ii) l'analyse de la variabilité avec des mesures de similarité. Cette phase comprend quatre étapes : (1) analyse de la variabilité par paires (définition de la variabilité des composants de la méthode en fonction de mesures de similarité), (2) assemblage des composants de la méthode en utilisant des techniques d'assemblage (Ralyté 2001b), (3) définition des dépendances de variabilité (facultatif ou obligatoire) et (4) la définition des contraintes de dépendance entre les composants de la méthode (exclusifs ou qui s'imposent les uns aux autres).

La similarité sémantique entre deux concepts de produit est basée sur la ressemblance de leurs noms et est appelée Affinité de Noms (AN). Elle utilise la relation de synonyme (SYN) entre les noms des éléments du produit pour l'évaluation de leur ressemblance. SYN définit une relation symétrique entre deux termes T_i et T_j ($T_i \neq T_j$) considérés comme synonymes (e.g. <Objectif SYN But>). La similarité structurelle (SSC) entre deux concepts (c_1,c_2) d'un modèle de produit est basée sur la similitude de leurs propriétés et liens avec les autres concepts (1).

$$SSC(C_1, C_2) = \left(\frac{2*(Nb\ de\ propriétés\ communes)}{\sum_{i=1}^{2} Nb\ de\ propriétés\ dans\ C_i} + \frac{2*(Nb\ de\ liens\ communs)}{\sum_{i}^{2} Nb\ de\ liens\ de\ C_i} \right) \div 2 \qquad (1)$$

Nous définissons des métriques de similitudes entre les concepts du modèle de processus. Leur définition dépend du formalisme de modélisation de processus utilisé (des exemples de métriques de similarité sont donnés dans (Ralyté 2001b). Toutes ces métriques aident à évaluer le degré de variabilité des composants de méthodes (commun, variable ou différent), à identifier les points de variation et à spécifier les dépendances de variabilité entre les composants de méthode. De plus, nous utilisons des techniques d'assemblage pour connecter et ordonner les composants à l'intérieur de la famille. Cette phase est formalisée en quatre étapes.

- *Analyser la variabilité paire-à-paire.* Nous utilisons l'analyse paire-à-paire pour définir la variabilité des composants de méthode (composants variables, composants communs et points de variation), comme illustré dans la Figure 20.

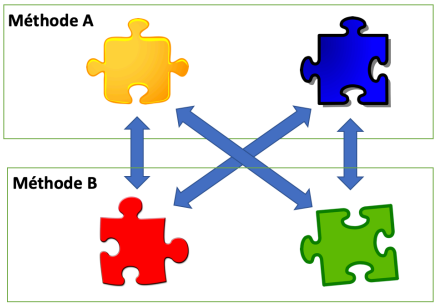


Figure 20. Comparaison des composants paire à paire

En utilisant le type de composant choisi et les mesures de similarité introduites précédemment, nous comparons les éléments suivants des composants de méthode : partie de produit source (PPS), partie de produit cible (PPC) et partie de processus (PPR), d'un point de vue sémantique et structurel. Par exemple, pour les deux composants de méthode cm1 et cm2, la formulation PPS (cm1) ≈ PPS (cm2) signifie que le produit source cm1 est similaire au produit source de cm2. Chaque flèche présente sur la Figure 20 correspond à une comparaison paire-à-paire de ces trois éléments (i.e. PPS, PPC et PPR). Si le degré de similarité totale (structurelle et sémantique) entre deux composants de méthode est élevé, ils sont considérés comme composants communs. S'ils sont similaires seulement du point de vue sémantique, ils sont considérés comme composants variables. La Figure 21 montre les cinq cas principaux utilisés pour analyser la variabilité.

Cas de comparaison	Interprétation
• PPS (cm_1) ≈ PPS (cm_2) • PPC (cm_1) ≈ PPC (cm_2) • PPR (cm_1) ≈ PPR (cm_2)	Les composants cm_1 et cm_2 sont communs.
• PPS (cm_1) ≈ PPS (cm_2) • PPC (cm_1) ≈ PPC (cm_2) • PPR (cm_1) ≠ PPR (cm_2)	Les composants cm_1 et cm_2 sont variables. Le produit source est considéré comme un point de variation avec le même résultat mais différentes démarches pour l'obtenir.
• PPS (cm_1) ≈ PPS (cm_2) • PPC (cm_1) ≠ PPC (cm_2) • PPR (cm_1) ≠ PPR (cm_2)	Les composants cm_1 et cm_2 sont variables. Le produit source est considéré comme un point de variation avec différents résultats.
• PPS (cm_1) ≠ PPS (cm_2) • PPC (cm_1) ≈ PPC (cm_2) • PPR (cm_1) ≠ PPR (cm_2)	Les composants cm_1 et cm_2 sont variables. Le produit cible est considéré comme un point de variation avec différents produits source différents.
• PPS (cm_1) ≈ PPC (cm_2) • PPR (cm_1) ≠ PPR (cm_2)	Les composants cm_1 et cm_2 sont consécutifs.
"≈" signifie "similaire"; ≠ signifie "non similaire"	

Figure 21. Comparaison paire-à-paire des composants de méthodes

- *Assembler les composants de méthode.* Cette étape consiste à finaliser l'organisation des composants de méthode en les intégrant dans une multi-méthode cohérente et complète. Nous utilisons les techniques d'assemblage définies dans (Ralyté 2001b) qui proposent deux démarches, l'association et l'intégration, pour assembler des composants de méthodes en fonction de leur similarité. L'association de composants est utilisée pour connecter des composants variables ou consécutifs selon leur produits source ou cible (par

exemple, dans le cas de composants consécutifs, le produit cible du composant prédécesseur est connecté au produit source du composant successeur). Les composants variables sont connectés à un point de variation. L'intégration de composants est nécessaire pour assembler des composants communs en fusionnant leurs éléments similaires (ex. concepts, propriétés, activités, etc.). Les deux techniques (intégration et association) utilisent des opérateurs d'unification pour unifier les terminologies employées dans les composants et permettre leur assemblage.

- *Définir les dépendances de variabilité.* Cette étape consiste, pour chaque composant variable d'un point de variation, à spécifier la dépendance de variabilité, pouvant être soit optionnelle, soit obligatoire.
- *Définir les contraintes de dépendances.* Cette étape permet de compléter le processus d'assemblage en spécifiant quels composants de méthode sont exclusifs ou complémentaires les uns les autres.

Lorsque toutes ces étapes ont été effectuées, la famille de méthodes est finalisée. A ce stade, la variabilité est prise en compte par l'identification des points de variations et des composants communs et variables. Les composants sont identifiés et organisés comme des variations potentielles à chaque point de variation. Est également prise en compte la variabilité des stratégies pour atteindre un même objectif.

Exemples de familles de méthodes

J'ai défini plusieurs familles de méthodes pendant et depuis l'élaboration de cette approche.

- MADISE : Prise de décision (Kornyshova 2011c)
- Définition des besoins (Deneckère 2011d)
- SAM : Statistical Analysis Method Family pour l'analyse de données dans les sciences humaines et sociales (Ammar 2013) (Hug 2014b)
- LPA : Lancement de Projets Agiles (Deneckère 2014a)
- SEA : Architecture d'entreprise (Kornyshova 2022b)

Cette approche a été utilisée également dans d'autres domaines par d'autres chercheurs, comme par exemple pour la co-création de services (Ralyté, 2013).

Nous allons détailler dans les sous-sections suivantes trois des exemples les plus aboutis de familles de méthodes : Les familles MADISE, LPA et SAM.

II.2.4. Famille de méthode appliquée à la prise de décision : l'approche MADISE

Cette approche a été définie lors de la thèse d'Elena Kornyshova (Kornyshova 2011) (Kornsyhova 2008) (Kornyshova 2009a) (Kornyshova 2009b) et a été la première famille de méthode construite avec le processus expliqué précédemment.

La conception, l'élaboration, la mise en œuvre et les autres processus de l'ingénierie des SI peuvent être considérés comme un ensemble d'étapes de deux types : premièrement, la prise de la décision et, deuxièmement, l'exécution de cette décision. On peut dire que chaque

processus comporte des étapes où des décisions doivent être prises. En outre, considérer certaines exigences comme la qualité, le coût, le temps, etc., implique que de nombreuses décisions doivent être prises pour optimiser le résultat (Ruhe 2003).

En ce qui concerne les prises de décision dans l'ingénierie des systèmes d'information, trois constatations peuvent être faites. (1) Les processus de l'ingénierie des SI sont de nature téléologiques (Loucopoulos 1995). Cela signifie qu'ils contiennent des étapes où des décisions doivent être prises. (2) Il existe de nombreuses méthodes de prise de décision, qui sont de nature différente et plus ou moins efficace en fonction de la situation dans laquelle elles sont appliquées. (3) Il y a de nombreux travaux dans la recherche en ISI portant sur la prise de décision (visant à résoudre les problèmes de prise de décision). La caractéristique principale de ces travaux est qu'ils résolvent un seul problème de prise de décision à chaque fois.

L'absence de prise de décision en ingénierie SI peut être considérée à trois niveaux : (i) au niveau du modèle (les décisions ne sont pas formalisées en termes d'alternatives et de critères, leurs conséquences ne sont pas analysées, les décisions ne sont pas transparentes), (ii) au niveau de la méthode (les décisions intuitives et ponctuelles éclipsent les décisions fondées sur des méthodes), (iii) au niveau des outils (même s'il existe des prises de décision, il n'y en a pas avec un processus complet). L'approche MADISE (MAke Decisions in Information Systems Engineering) a été développée pour gérer ces deux types de difficultés et guider les ingénieurs en systèmes d'information au travers des activités de la prise de décision. Cette approche contient trois éléments : une ontologie de la prise de décision (Kornyshova 2010a) (Kornyshova 2012a), une famille de méthode (Kornyshova 2011b) et une base de directives méthodologiques associés.

Fondamentaux de la prise de décision

Une décision est un acte d'effort intellectuel initié pour satisfaire un but et permettre un jugement sur des actions potentielles afin de prescrire une action finale. Cette prescription est basée sur l'étude de plusieurs aspects caractérisant les actions alternatives. Les décisions sont prises par des acteurs, qui peuvent être aidés par des méthodes et des outils de prise de décision. Les acteurs prennent des décisions afin de satisfaire un objectif. Le concept de décision est également étroitement lié à ce processus (Roy 1996).

Bernard Roy définit les trois concepts de base qui jouent un rôle fondamental dans l'analyse et la structuration des décisions en lien étroit avec le processus décisionnel lui-même (Roy 2005) : le problème décisionnel, les alternatives (actions potentielles) et les critères.

- Le problème décisionnel peut être caractérisé par le résultat attendu de la prise de décision. Lorsque le résultat consiste en un sous-ensemble d'alternatives potentielles (le plus souvent une seule alternative) alors c'est un problème de choix. Lorsque le résultat représente les possibilités d'affectation à certaines groupes prédéfinis, il s'agit alors d'un problème de classification. Lorsque le résultat consiste en un ensemble de solutions ordonnées, il s'agit d'un problème de classement. Lorsqu'une décision n'est mise en évidence que par une description des solutions potentielles et de leur impact, il s'agit d'un problème de description. Ce dernier est rarement utilisé dans les processus de prise de décision.

- Le concept d'alternative désigne l'objet de la décision. Toute décision comporte au moins deux possibles solutions qui doivent être bien identifiées.
- Un critère peut être tout type d'information qui permet l'évaluation des alternatives et leur comparaison. Il existe de nombreux types de critères : caractéristiques intrinsèques des artéfacts ou des processus, opinions des utilisateurs, conséquences potentielles des solutions, etc. Lorsqu'il s'agit de critères, l'ingénieur doit déterminer des "règles de préférence", c.-à-d. la valeur souhaitée du critère (par exemple le maximum ou le minimum pour les critères numériques) selon son besoin spécifique.

La première priorité dans un processus de prise de décision est d'identifier les acteurs. Trois acteurs principaux participent aux activités de prise de décision : les utilisateurs (ceux qui définissent le problème de la prise de décision, donnent leurs préférences sur les alternatives, les critères (Kornyshova 2011b) et qui valident la décision finale), les ingénieurs du SI (ils évaluent les alternatives et font la proposition de la décision finale pour les utilisateurs) et les responsables de la prise de décision (ceux qui définissent qui participera à chaque étape). L'approche MADISE a été définie pour accomplir les fonctions des responsables de la prise de décision.

Herbert Simon (Prix Nobel d'économie de 1978) a été le premier à formaliser le processus décisionnel. Il a suggéré un modèle comprenant trois phases principales : l'intelligence, la conception et le choix (modèle I.D.C.) (Simon 1960). L'intelligence consiste à enquêter sur les conditions qui exigent des décisions. La conception consiste à inventer, à élaborer et à analyser des solutions alternatives. Le choix exige de choisir une solution parmi d'autres. Ce premier processus a été modifié et étendu de diverses manières mais celui qui est le plus courant de nos jours est défini comme suit (Baker 2001): (a) Définir le problème (nécessité de définir les priorités),(b) Identifier les paramètres du problème (par exemple, alternatives et critères), (c) Établir une matrice d'évaluation (estimer les alternatives selon tous les critères), (d) Choisir la méthode de prise de décision, (e) Faire une évaluation basée sur tous les critères permettant une décision finale.

Famille de méthode MADISE

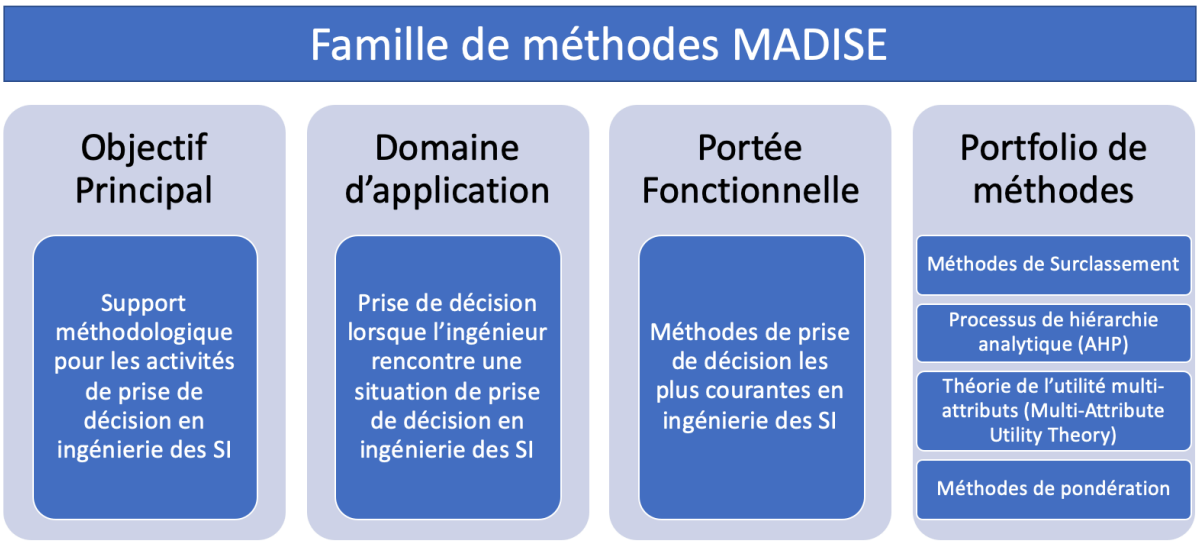


Figure 22. Interface de la famille MADISE

En plus des méthodes sélectionnées, nous les avons complétés avec un ensemble de plusieurs activités supplémentaires (comme la découverte de solutions alternatives, la définition des critères, etc.) afin de couvrir l'ensemble du processus.

La famille de méthodes MADISE a été construite sur la base de la méthodologie expliquée a la section précédente : (i) identification des composants de la méthode de prise de décision et (ii) leur assemblage ultérieur dans la famille de méthodes à l'aide du formalisme MAP. Cette méthodologie permet d'avoir un processus générique qui couvre les activités essentielles de la prise de décision sans répéter les mêmes étapes, tout on les organisant en une multi-méthodes unique.

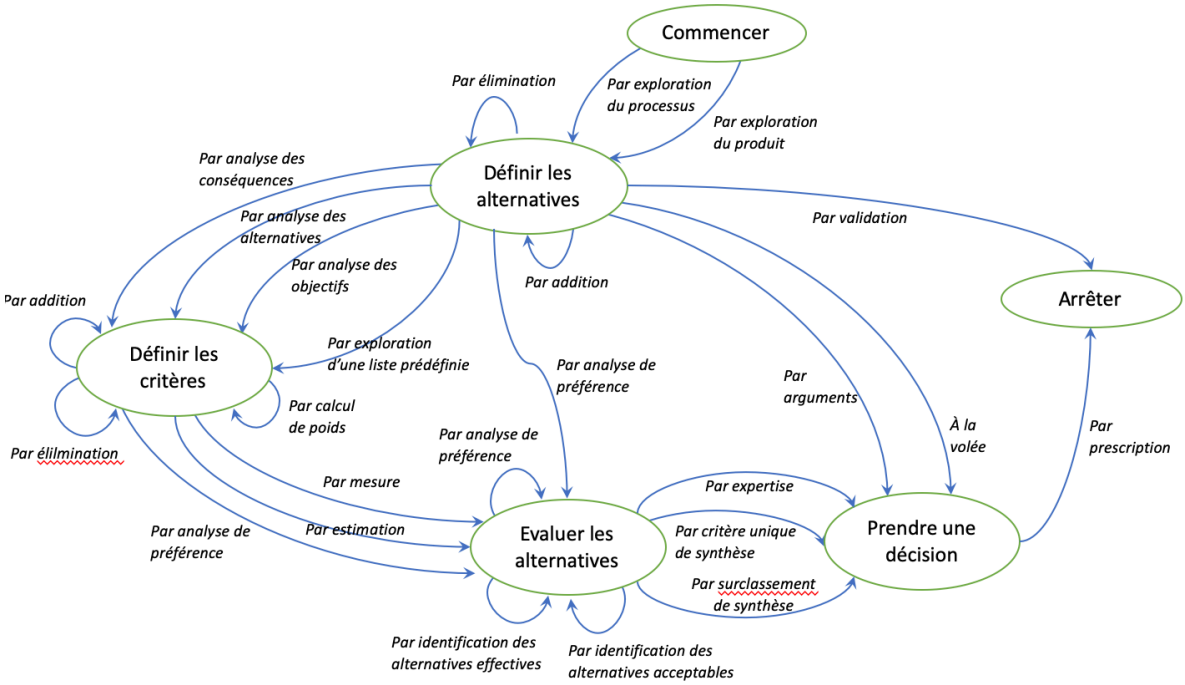


Figure 23. Famille de méthodes MADISE

La carte MADISE (Figure 23) contient quatre intentions principales : *Définir les alternatives*, *Définir les critères, Évaluer les alternatives* et *Prendre une décision*. La carte contient un ensemble de directives qui permettent de choisir entre les intentions et les stratégies de manière exhaustive. L'ingénieur fait ces choix et choisit un ensemble de sections de la carte qui sont exécutées consécutivement. De cette manière, il construit dynamiquement une méthode de prise de décision adaptée à la situation donnée. Par exemple, une trajectoire possible est de Définir des alternatives, Définir des critères, Évaluer des alternatives, Prendre une décision par application de méthodes (par exemple avec la méthode multicritères AHP (Saaty 1980). Un autre est de définir les alternatives et ensuite de prendre une décision directement (par des stratégies ad hoc ou en utilisant des arguments) comme dans (Karlsson 1997) (Wiegers 1999). Par conséquent, la carte MADISE fournit à l'ingénieur du SI un ensemble complet de directives pour la sélection de son chemin et l'exécution des sections.

L'ingénieur commence MADISE en devant définir les alternatives. Comme il est communément admis, les méthodes d'ingénierie des SI contiennent deux parties : une partie processus et une partie produit (Rolland 1998). La partie processus des méthodes saisit leurs aspects comportementaux et procéduraux (étapes, tâches, activités et calendriers) (Agerfalk 2003). La partie produit d'une méthode tient compte de ses aspects structurels et statiques (comme par exemple les documents sur les exigences, modèles et autres livrables) (Agerfalk 2003). L'ingénieur peut tout autant explorer la variabilité des processus ou des produits afin de trouver de possibles alternatives. Il commence donc le processus en choisissant l'une des deux sections permettant de *définir la liste des alternatives par exploration du processus* (cette section est utilisée dans les approches fondées sur les intentions afin de sélectionner une stratégie qui est une action ou une partie du processus (Rolland 1999) ou de *définir la liste des alternatives par exploration du produit* (un exemple habituel est la priorisation des exigences, dans lesquelles les exigences sont les éléments de produit considérés (Karlsson 1997) (Wiegers 1999) (Ngo-The 2005) (Ruhe 2003)).

Les solutions alternatives peuvent être complémentaires ou exclusives. Une fois la liste initiale des alternatives définie, elle peut être affinée par les deux sections suivantes : La stratégie *Par élimination* est appliquée lorsque l'une ou plusieurs alternatives sont évaluées par l'ingénieur comme non réalistes ou non réalisables (Roy 1996) (Chakhar 2006). Dans ce cas, elles sont retirées et ne sont plus étudiées. Un exemple de cette section est l'examen des exigences avant l'application de AHP dans l'approche coût-valeur de la priorisation des exigences (Karlsson 1997). La stratégie *par addition* est disponible lorsque certaines alternatives peuvent être ajoutées en recherchant des alternatives complémentaires. Une telle stratégie est utilisée dans (Ralyté 2003) où la stratégie *par affinement* propose de rechercher un autre composant de méthode dans l'approche de création de méthode par assemblage. À partir de la définition des alternatives, le processus peut être interrompu lorsque le nombre de solutions alternatives identifiées est trop limité (une seule solution possible). L'ingénieur peut valider ou non la solution initialement obtenue.

L'intention de définir les critères n'est pas obligatoire. L'ingénieur le fait s'il veut arbitrer entre les alternatives en se basant sur de multiples facteurs. A ce stade, un ensemble de critères (au moins un critère) pour l'évaluation des alternatives est défini. À partir de l'intention Définir les alternatives, l'intention Définir les critères peut être atteinte en suivant quatre stratégies. La stratégie *d'analyse de la description des alternatives* peut être utilisée dans le cas où les alternatives ont des caractéristiques intrinsèques, qui peuvent être considérées comme des critères. L'ingénieur les analyse afin d'identifier ceux qui sont importants pour faire son arbitrage. La stratégie *d'analyse par conséquences* traite des conséquences alternatives, qui prend en compte les effets produits par les alternatives. Les propriétés futures des alternatives et leurs effets sur le problème de décision sont analysées afin d'identifier les critères. La stratégie *d'analyse par objectif* s'applique lorsqu'un acteur participant au processus a des objectifs en ce qui concerne le problème de décision puisque chaque alternative peut être mesurée en fonction de sa capacité à contribuer à ces objectifs. Dans ce cas, les objectifs deviennent des critères. Les objectifs en tant que critères sont utilisés dans (Kornyshova 2007a) pour la priorisation des processus opérationnels et dans (Maiden 2002) pour la priorisation des besoins. Dans la stratégie *d'exploration d'une liste prédéfinie*, les ingénieurs examinent la liste des caractéristiques de projets qui sont courantes en ingénierie des SI et sélectionnent celles qui conviennent.

Une fois que le premier ensemble de critères est sélectionné, il peut être affiné avec d'autres stratégies. La stratégie *par élimination* est utilisée lors de l'élaboration de l'ensemble de critères, car le décideur doit se conformer à certaines règles pour être cohérent. Par exemple, afin d'éliminer certains critères, l'ingénieur SI les considère dans leur ensemble en utilisant la méthode SMART (Crowe 1997). Les critères doivent être précis, mesurables, réalisables, pertinents et opportuns. L'application de la méthode SMART aux processus opérationnels est présentée dans (Crowe 1997). La stratégie « *Par addition* » est appliquée lorsque les critères peuvent se compléter. Dans ce cas, un critère est ajouté suite à l'analyse des critères existants. La stratégie *de pondération* traite de l'attribution de pondérations aux critères de décision. Les poids sont attribués lorsque l'ingénieur souhaite définir l'importance relative des critères. L'ingénieur peut choisir l'une des techniques suivantes : (i) par simple attribution (Baudry 2002) (Bouyssou 2001), (ii) par indentification du premier critère d'amélioration (Mustajoki 2005), (iii) par technique de compromis (Mustajoki 2005), (iv) par analyse d'importance (Keeney 1999) ou par comparaison par paires (Saaty 1980). Le dernier est couramment utilisé dans la priorisation des exigences en AHP (Karlsson 1997) (Maiden 2002).

L'évaluation des alternatives vise à construire la matrice d'évaluation (ou matrice de décision) (Roy 1996) (Chakhar 2006). Il existe plusieurs façons d'évaluer les alternatives.

À partir de la définition des alternatives, celles-ci peuvent être évaluées à l'aide de la stratégie *d'analyse par préférences* (section S12). Un ingénieur détermine les préférences entre deux alternatives a et b. (Grosz 1997) définit quatre relations élémentaires : (i) Indifférence : $a\mathbf{I}b - a$ et b sont équivalents ; (ii) Préférence stricte : $a\mathbf{P}b - a$ est strictement préféré à b ; (iii) Préférence faible : $a\mathbf{Q}b - a$ est faiblement préféré à b ; (iv) Incomparabilité : $a\mathbf{R}b - a$ et b ne sont pas comparables. Par exemple, une relation de préférence stricte est présente dans la méthode AHP appliquée à la priorisation des exigences (Karlsson 1997) (Maiden 2002).

Il est également possible d'évaluer les alternatives en utilisant les critères définis au préalable : en mesurant, en estimant et en analysant les préférences. La stratégie d'évaluation *par mesure* permet d'utiliser une définition métrique pour produire une valeur (Martin 2003). Une méthode de mesure est une séquence logique d'opérations permettant l'estimation de l'alternative. Les méthodes de mesure sont objectives, car l'évaluation est fondée sur des règles numériques (Martin 2003). Les alternatives peuvent également être évaluées *par estimation* au moyen d'heuristiques. Un acteur évalue les alternatives en fonction de critères subjectifs, par exemple en fonction de son opinion. Cette estimation est donc subjective puisqu'elle fait appel au jugement humain (Martin 2003). Le type de méthode d'évaluation (objective ou subjective) qui dépend de la nature des opérations utilisées pour évaluer une alternative se trouve dans (ISO 2002). La dernière manière d'évaluer les alternatives est *par l'analyse des préférences*. Un acteur détermine les préférences entre deux alternatives a et b selon un critère spécifique. Par exemple, l'ingénieur peut comparer deux exigences en fonction du critère de coût.

Une fois la matrice d'évaluation construite, les évaluations des alternatives peuvent être améliorées par les trois stratégies suivantes. La stratégie *d'identification des alternatives efficaces* permet de supprimer certaines alternatives afin de ne conserver que celles qui sont

« efficaces ». Une alternative est supprimée si ses évaluations selon tous les critères sont pires ou du moins identiques à celles d'une autre alternative (Chakhar 2006) (Vincke 1989). La stratégie *d'identification des alternatives acceptables* permet de qualifier certaines alternatives comme non acceptables et de les supprimer. Un seuil d'acceptation est établi pour un critère. Une telle technique est utilisée dans la méthode WinWin pour la priorisation des exigences (Ruhe 2003). En utilisant la stratégie *par analyse des préférences*, l'ingénieur peut améliorer l'analyse des préférences en définissant des paramètres complémentaires tels que le seuil de préférence, le seuil d'indifférence ou le seuil de veto (Roy 1996) (Chakkar 2006). Ces paramètres sont utilisés dans certaines méthodes de prise de décision, par exemple pour la priorisation des processus opérationnels dans (Kornyshova 2007a).

À ce stade du processus une décision doit être prise. Cela peut être fait *en utilisant des arguments*. (dans plusieurs approches les décisions sont fondées sur un ensemble d'arguments (Grosz 1997) (Rolland 2000) (Rolland 1999) qui indiquent l'alternative à sélectionner en fonction des paramètres donnés), *à la volée* (une décision peut être prise par un décideur sans recourir à une méthode spécifique. Ce genre de décision est propre à chaque situation et dépend fortement des compétences et de l'expérience du décideur), ou *par expertise* (l'expérience et l'expertise du décideur peut être suffisante pour prendre une décision, en particulier si la même situation a déjà été rencontrée).

Les critères peuvent aussi être manipulés pour permettre de prendre la décision. Il est possible de transformer des valeurs partielles (valeurs alternatives selon différents critères) en une valeur agrégée. Cela peut se faire par un critère unique de synthèse ou par surclassement de synthèse.

- Par *critère unique de synthèse*. Cette approche consiste à construire un seul critère à partir d'un ensemble de critères en utilisant une fonction d'agrégation (Roy 1996). La fonction d'agrégation peut être une fonction d'addition ou de multiplication. La plus connue est la méthode basée sur la somme pondérée. Les méthodes de pondération comprennent SMART (Simple Multiattribute Technical Rating) (Keeney 1999), SWING (Mustajoki 2005), Trade-off weighting (Keeney 1999) et MAUT (Keeney 1993). Selon MAUT, une fonction utilitaire est établie pour chaque critère. Ensuite, les fonctions d'utilité partielles sont agrégées en une fonction d'utilité multi-attributs représentant soit une addition, soit une multiplication des fonctions partielles. Toutes les alternatives sont évaluées en utilisant cette fonction. L'alternative qui maximise l'utilité est sélectionnée. Ces méthodes sont disponibles pour les problèmes de choix, de classement ou de décision de classification.
- *Par surclassement de synthèse*. Les méthodes de surclassement (Grosz 1997) s'inspirent de la théorie du Choix Social. La méthode de surclassement la plus connue est ELECTRE (Elimination And Choice Corresponding to Reality). Le surclassement indique le degré de domination d'une alternative sur une autre. Les méthodes de surclassement reposent sur l'identification étape par étape des préférences des décideurs. Les décideurs formulent leurs préférences, puis une analyse détaillée permet de prendre des décisions pour l'un des problèmes (choix, classement ou classification). Les méthodes de surclassement sont utilisées pour la sélection de composants de méthode (Kornyshova 2007b) et pour la priorisation de processus opérationnels (Kornyshova 2007a).

Une fois qu'une décision est prise, l'ingénieur génère une prescription de décision (*Par validation*). Dans ce cas, le processus de prise de décision s'arrête suite à la validation du résultat. Certaines mesures (par exemple, un indice de cohérence (Karlsson 1997)) peuvent être utilisées pour vérifier si le résultat est valide. Sinon, le processus peut se poursuivre en revenant aux étapes précédentes.

Utilisation de la famille de méthode MADISE

Deux types d'utilisateurs peuvent utiliser cette approche. Le premier concerne les ingénieurs de méthode qui spécifient les composants de la méthode de prise de décision et les organisent dans la famille de méthodes. Les ingénieurs possédant des connaissances en prise de décision les représentent dans les composants. Ils utilisent l'ontologie afin de représenter leurs connaissances en matière de prise de décision et de décomposer les méthodes en composants. Les composants sont stockés dans une bibliothèque méthodologique que nous appelons le référentiel MADISE.

Le deuxième groupe représente les ingénieurs des SI qui ont besoin d'aide en matière de prise de décision. Ils précisent leurs exigences et choisissent un ou plusieurs composants de la méthode MADISE afin d'améliorer leurs méthodologies ou de créer une nouvelle méthode. Un ingénieur SI peut avoir l'un des deux objectifs suivants : (i) construire une méthode de prise de décision personnalisée ou (ii) étendre une méthode d'ingénierie de SI avec des composants de prise de décision. Dans les deux cas, le processus commence par préciser les exigences en matière de prise de décision, à savoir la situation et l'intention. L'ontologie (Kornyshova 2010a) est utilisée à cet effet. Ensuite, l'ingénieur sélectionne un ou plusieurs composants adaptés à la situation donnée. Si les composants sont nombreux, ils sont assemblés en un composite. De cette façon, l'ingénieur SI obtient une méthode de prise de décision personnalisée adaptée au problème donné (le premier objectif est atteint). Ensuite, s'il veut étendre une méthode d'ingénierie SI (le second objectif), il intègre les composants sélectionnés dans la méthode en utilisant les approches situationnelles disponibles.

Cette approche a été validée dans (Kornyshova 2012b) avec une comparaison de la famille MADISE avec cinq techniques de priorisation des exigences : l'approche Cost-Value (Karlsson 1997), la matrice de priorisation (PrMatrix) (Wiegers 1999), l'outil de priorisation des besoins (RPT) (Moisiadis 2002), WinWin (Ruhe 2003) et REDEPEND (Maiden 2002). L'objectif principal de cette comparaison était de vérifier si l'approche MADISE était complète et flexible. Ces cinq techniques ont été choisies car elles sont bien connues, représentatives et différentes les unes des autres. En ce qui concerne l'exhaustivité, la comparaison démontre que l'approche MADISE est presque complète. Elle n'inclut pas les activités d'exécution, c'est-à-dire qu'elle reste méthodologique. L'outil de support associé (le référentiel MADISE) permet de sélectionner les composants adaptés à un cas donné de la méthode de prise de décision et ne met pas en œuvre ces composants. Sauf cet aspect, l'approche MADISE est la plus complète car elle comprend des directives pour identifier la liste des alternatives (exigences fixées dans le cas donné) et pour identifier la liste des critères (lorsque les approches de priorisation des exigences ont une liste limitée de critères possibles). En ce qui concerne le critère de flexibilité, l'approche MADISE est la plus souple, ce qui implique qu'elle pourrait être facilement adaptée à différentes situations. En outre, une autre application de l'approche MADISE a été mise en évidence dans les exemples étudiés, car elle peut contribuer à améliorer les modèles de prise

de décision existants. Par exemple, l'approche coût-valeur peut être améliorée par la section <Évaluer les alternatives, Par élimination, Évaluer les alternatives> afin d'éliminer les solutions non intéressantes et, de cette façon, simplifier l'application de la méthode AHP.

Un autre exemple d'application de MADISE peut être trouvé dans (Khodabandelou 2013a) où l'approche a été appliquée pour sélectionner un outil dans un projet spécifique.

II.2.5. Famille de méthode appliquée au lancement de projets agiles : la famille LPA

Cette famille, basée sur les composants agiles définis dans (Iacoveli 2012), a été définie (Deneckère 2014a) puis testée dans un cadre réel (Deneckère 2016b) (Deneckère 2015b).

Fondements des méthodes agiles

Le concept de méthode agile est apparu dans les années 90 puis s'est concrétisé au début du millénaire avec l'arrivée du manifeste agile en 2001 (Beck 2001). Comme il est indiqué dans (Serena 2007), alors que la publication de ce manifeste n'a pas entériné la ruée vers les méthodes agiles, celle-ci ayant commencé bien plus tôt, le manifeste signale tout de même l'acceptation de la philosophie agile dans le monde professionnel. Plusieurs méthodes agiles ont été définies et largement communiquées depuis lors, telles que Lean Software Kanban (Anderson, 2003), Extreme Programming (Beck, 2000), Scrum (Schwaber 2002), Crystal (Cockburn, 2002) et DSDM (Stapleton, 1995) entre autres. Une étude de 2013 (VersionOne, 2013) signale que 57% des répondants travaillent dans des entreprises ayant plus de cinq équipes pratiquant l'agilité et 38% avec plus de 10 équipes. Ces chiffres indiquent que le mouvement agile est un succès d'adoption et qu'il est largement utilisé dans le monde professionnel.

Les méthodes de développement agiles sont en général définies comme des ensembles de bonnes pratiques et de bons comportements à adopter. Depuis le manifeste agile, énormément de livres et de documents ont été édités afin d'expliquer ces bonnes pratiques – par exemple (Shore, 2007) ou (Cohn, 2005) – mais tous ces documents manquent d'une explication claire du processus à appliquer, sur le fait qu'il faut être agile pour faire de l'agilité. Mais le fait qu'il n'y ait pas de processus formel ne signifie pas que les développements agiles ne sont pas structurés, uniquement qu'il y a une différence entre un processus complètement rigide et un processus formalisé. Dans (Meyer, 2014), Bertrand Meyer dit que « Le livre classique sur l'agilité est une succession d'observations générales alternant avec des anecdotes personnelles sur des sauvetages ou des échecs de projets. Ces anecdotes sont en général amusantes et parfois intéressantes, mais un cas d'étude n'est qu'un cas d'étude et nous ne savons jamais jusqu'où il est possible de généraliser ». Dans cette jungle de tâches, principes, conseils et recommandations, les nouveaux utilisateurs de méthodes agiles sont parfois perdus avec toutes les alternatives possibles qui leur sont offertes et avec une compréhension parfois très minimaliste des arguments en faveur ou contre celles-ci. Par exemple, (Meyer, 2014) mentionne que « toute équipe agile construit son propre cocktail de pratiques agiles, rejetant celles qui ne correspondent pas à ses besoins – jusqu'à présent cependant, chaque organisation, pour chacun de ses projets, doit répéter ce processus de tri ». Ce comportement est commun à d'autres champs d'expertise et l'utilisation d'autres

types de méthodes (méthodes de développement, méthodes de déploiement, méthodes de conception, etc.) et le domaine de l'IMS peut peut-être aider à résoudre ce problème.

Les méthodes agiles ont déjà été étudiées plusieurs fois dans le domaine de l'IMS. (Qumer 2007) a montré qu'une approche de méthodes situationnelles couplée à un cadre d'outils agiles (Agile Software Solution Framework - ASSF) peut être utilisée pour créer une méthode de développement hybride utilisable. Ceci peut être réalisé en combinant les pratiques agiles et formelles dans une situation donnée dans de grandes organisations. Dans (Karlsson 2008), une approche de configuration de méthodes a été proposée pour configurer eXtreme Programming. L'une des conclusions était que XP ne procure pas vraiment d'alternatives dans son processus et qu'il était donc assez ardu d'obtenir des possibilités intéressantes. Cette étude n'utilisait cependant qu'une seule méthode agile, ce qui la rend difficile à généraliser. (Abad 2010) (Abad 2012) propose une approche de construction de méthodes situationnelles par assemblage spécifique aux méthodes agiles. Une base de composants de méthodes contient les fragments nécessaires, conformes au méta-modèle SPEM 2.0., et utilisables avec des outils de méta-modélisation permettant d'implémenter ce méta-modèle, incluant l'EPFC (Eclipse Process Framework Composer).

Les méthodes agiles sont en général définies comme des ensembles de bonnes pratiques et beaucoup de documents tentent de les expliquer. Ceux-ci manquent cependant d'explications claires sur le processus à appliquer, sur l'observation que l'on se doit d'être agile si l'on veut faire de l'agile. Nous avons élaboré un ensemble de composants de méthodes concernés par la représentation des méthodes agiles à un haut niveau ainsi que des vues détaillées de composants s'intéressant à la phase de lancement des projets dans les méthodes agiles : Scrum (Schwaber 2002), XP (Beck, 2000), DSDM (Stapleton, 1995) et Crystal Clear (Cockburn, 2002). Ces composants sont intégrés dans une famille de méthodes appelée Lancement de projets agiles (LPA). Cette famille de méthodes agiles propose un ensemble de composants organisés dans un processus clair et facile à comprendre spécialisé dans la phase de lancement des projets agiles. Ces composants de méthodes ont été décrits dans (Iacovelli, 2012) et développés plus avant dans (Deneckère 2014a).

La famille LPA

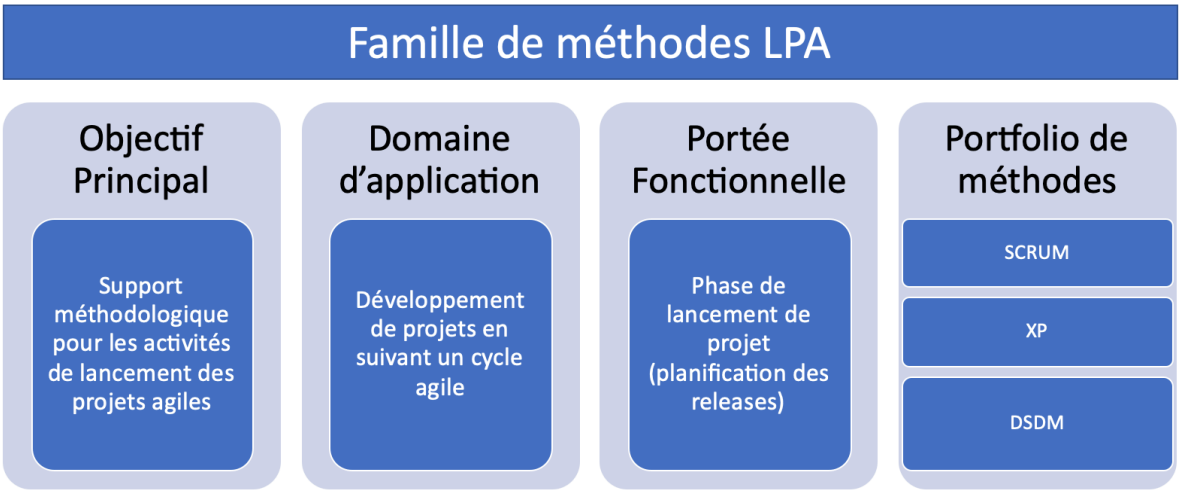


Figure 24. Interface de la famille LPA

Les trois méthodes agiles étudiées comportent clairement deux groupes principaux d'activités, celles liées à la définition des besoins du client vis-à-vis du système à réaliser et celles traitant de la planification globale de haut niveau du projet. La méthode DSDM (Stapleton, 1995) préconise en plus de ces deux groupes d'activités d'effectuer une étude de marché et une étude technique avant toute chose. Cette étape supplémentaire permet de pouvoir abandonner le projet si l'étude de faisabilité montre un trop gros risque d'échec potentiel.

Chacune des trois méthodes agiles propose une méthodologie différente pour définir les besoins (de haut niveau) du client. SCRUM (Schwaber 2002) propose d'identifier les besoins du client, d'estimer leur taille par la technique dite du poker de planification, puis de les prioriser par rapport aux risques et aux incertitudes qu'ils permettent de réduire. XP (Wells, 2009) quand à lui, propose de réaliser une liste des besoins par une itération de définition de besoins et d'estimation. Chaque besoin est identifié de façon similaire à SCRUM mais il est ensuite estimé en temps de développement idéal. Si son temps estimé est trop important, le besoin est alors décomposé en plusieurs besoins. Inversement, s'il est trop petit, il est regroupé à d'autres besoins pour en former un plus important. La méthode XP propose également un style moins formel pour la rédaction des besoins que la méthode SCRUM. La méthode DSDM propose d'inclure les futurs utilisateurs finaux du système directement dans le processus pour la définition et la priorisation d'une liste de besoins de haut niveau répondant à la problématique du projet.

Une fois la liste des besoins établie, la méthode DSDM propose encore une fois de s'interroger sur la viabilité ou la faisabilité du projet avec l'ensemble des besoins tels qu'ils sont définis. Si le projet n'est pas estimé viable, le projet est clôturé. Dans le cas contraire, le processus passe à la partie de planification du projet en utilisant diverses techniques.

L'utilisateur peut ensuite, s'il le désire, arrêter là le processus et terminer ici l'étape de lancement du projet pour ensuite lancer les cycles d'itération de développement. En DSDM cependant, il est possible de commencer un affinement des besoins de haut niveau du projet après avoir établi une première planification. Il peut alors affiner un besoin fonctionnel en divers sous-besoins fonctionnels et réaliser une ébauche de liste des besoins non fonctionnels. Il peut aussi revenir sur la planification du projet pour affecter des acteurs aux livrables et caractériser par un type (modèle fonctionnel, prototypage ou développement) les itérations dans les livrables. L'utilisateur peut exécuter ceci jusqu'à ce que les niveaux de granularité des besoins et du planning soient satisfaisants. Le processus de lancement d'un projet agile s'arrête lorsque le chef de projet estime la planification complète.

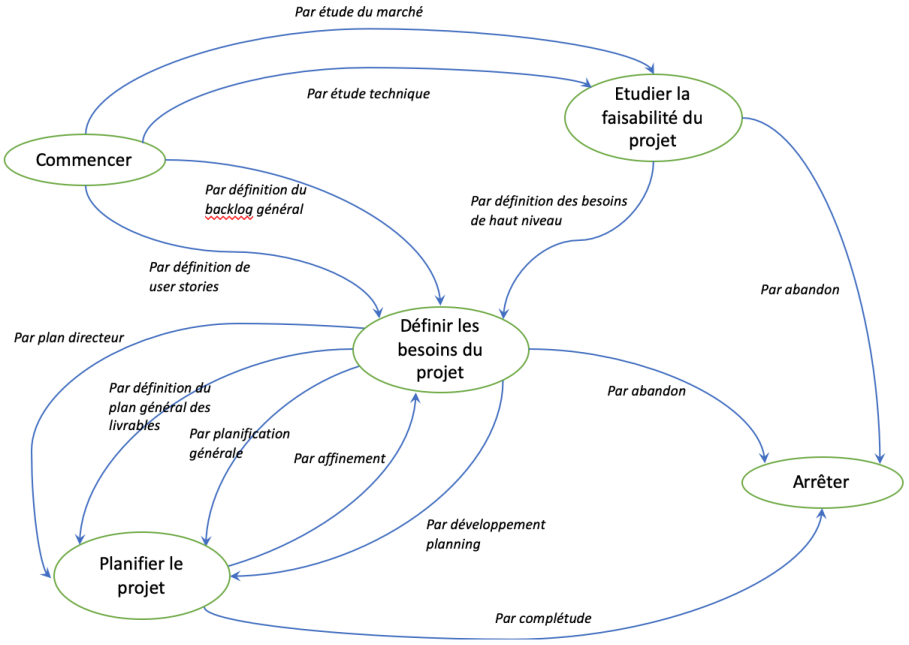


Figure 25. Famille de méthode LPA

Validation de la famille LPA

Une manière d'utiliser les techniques d'IMS de manière plus douce est d'introduire les composants de manière progressive, un à la fois, et d'attendre que les utilisateurs soient à l'aise avec les premiers changements avant d'aller vers une autre modification. Nous appelons cette introduction une « intégration progressive » (Deneckère 2016b). Nous avons validé cette approche sur deux expérimentations observées dans deux entreprises souhaitant améliorer leur processus de gestion de projets informatiques. Les deux cas d'étude ont eu lieu dans des entreprises très différentes par leurs tailles, secteurs, méthodes de travail, mais ayant un point en commun qui était la résistance des membres de l'équipe à la mise en place des méthodes agiles. Pour cette raison, dans les deux cas, il s'agissait d'introduire certains éléments des méthodes agiles au lieu de faire une intégration des méthodes entières en une seule fois. Le premier étude de cas a été décrit dans (Deneckère 2016b), le deuxième dans (Deneckère 2016b).

Ces expériences ont montré qu'une introduction progressive de nouveaux composants de méthodes dans une méthode de développement est possible et peut être plus facilement acceptée par les équipes de développement, mais peut avoir des difficultés liées à la conduite du changement.

Dans le premier cas, l'intégration a été faite de deux manières. Premièrement, les tâches de planification et d'estimation ont été intégrées d'une façon très lente et progressive. Les changements améliorèrent les réunions hebdomadaires puisqu'elles sont passées de 3 heures à 2h, puis 1h30 (pour une équipe de 5-8 personnes). Cette amélioration est essentiellement due au fait qu'il n'y a plus de discussions sur des sujets où tout le monde est déjà d'accord. Il ne s'agissait finalement que de formaliser dans l'outil ce qui existait déjà dans le document Google initial. Deuxièmement, il a été intégré les problématiques de Backlog et de graphique

d'avancement. Comme les premiers changements avaient été acceptés sans trop de difficulté, il était un peu plus facile d'introduire de nouvelles techniques dans le processus de développement. Ces changements améliorèrent l'estimation des tâches à moyen terme, induisirent une meilleure relation clientèle et un partage plus efficace et plus rapide des informations. Dans cette expérience, l'intégration a amené à une amélioration de l'efficacité du travail de l'équipe. L'entreprise peut maintenant annoncer qu'elle utilise la méthode SCRUM dans le développement de ses projets. L'équipe est heureuse de travailler maintenant avec un processus plus formalisé qui permette une meilleure gestion du travail d'équipe, de la durée des réunions, de l'estimation des tâches et de la relation clientèle. Les clients ont un accès à une partie de l'outil utilisé et sont beaucoup plus impliqués dans le processus de développement.

Dans le deuxième cas, les résultats de l'intégration progressive sont plus complexes. Premièrement, la mise en place de certains composants agiles a pu avoir lieu malgré la réticence de certaines personnes aux méthodes agiles. Le progrès a été atteint en ce qui concerne la visibilité du travail et de l'affectation des tâches urgentes. Cependant, pour cette étude, l'intégration progressive peut être considérée comme un échec à cause de l'esprit d'équipe peu présent parmi les participants. L'intégration progressive s'est cependant poursuivi dans cette entreprise afin d'introduire un outil virtuel pour supporter les réunions du Sprint et de mener au mieux la conduite du changement pour identifier les remplaçants sans que le personnel ne se sente menacé.

II.2.6. Famille de méthode appliquée à l'analyse statistique : la famille SAM

Cette famille a été développée dans le cadre d'un mémoire de recherche du master SID (Ammar, 2013) (Hug, 2014b).

Contexte et fondamentaux des méthodes statistiques

Les sciences humaines sont des sciences ayant pour objectif d'étudier les êtres humains : leurs actions, leurs relations ainsi que leurs traces (cultures humaines, modes de vie, comportements sociaux, sociétés). Étudier les modes de vie des êtres humains et leurs comportements sociaux signifie rechercher et analyser des données qui peuvent être décrites sous différents formats ou écrites en langues étrangères, que ce soit sous forme de textes ou de dessins. Ces données peuvent être de grande taille et stockées sous diverses formes. Le développement des outils informatiques a permis la préservation et l'exploitation de grandes masses de données et de nombreuses méthodes statistiques ont été conçues depuis le début du 20ème siècle pour les analyser.

Les méthodes d'analyse statistique sont largement utilisées en sciences humaines pour trouver des corrélations entre les événements, pour faire des prédictions ou des hypothèses sur des faits ou des artéfacts (Canning 2014). Toutefois, l'utilisation de méthodes d'analyse statistique exige des connaissances antérieures en statistique, le choix d'une méthode dépend du type de données, des critères du projet et des objectifs du chercheur. Le choix d'une méthode a alors un grand impact sur les résultats obtenus, donc sur l'interprétation. L'utilisation d'une méthode inadaptée peut conduire à des interprétations erronées qui doivent être évitées à tout prix pour la recherche. Les outils statistiques comme R (R 2014) ou

SAS (SAS 2014) permettent de manipuler des données et d'exécuter des méthodes statistiques, mais les chercheurs doivent être guidés pendant ce processus et pendant l'interprétation des résultats obtenus, surtout s'ils sont débutants. De plus, les chercheurs en sciences humaines ont leurs propres façons de traiter les données : la collecte, le codage, l'analyse et l'interprétation sont souvent propres à chaque chercheur. Ces méthodes ne sont pas partagées - chaque chercheur réinvente la roue lorsqu'il analyse des données sans expérience préalable. Néanmoins, la modélisation et le partage de ces méthodes pourraient être utiles au milieu de la recherche et aux étudiants en sciences humaines, mais il est difficile de les formaliser de manière compréhensible.

L'objectif de la famille SAM était de fournir des conseils à tout acteur des sciences humaines, des étudiants aux chercheurs, tout en effectuant l'analyse des données. Les orientations fournies doivent s'adapter à la situation : chaque projet a ses propres contraintes et spécificités. Il est nécessaire de fournir une méthode souple pour mieux guider et soutenir les acteurs.

La famille de méthodes SAM

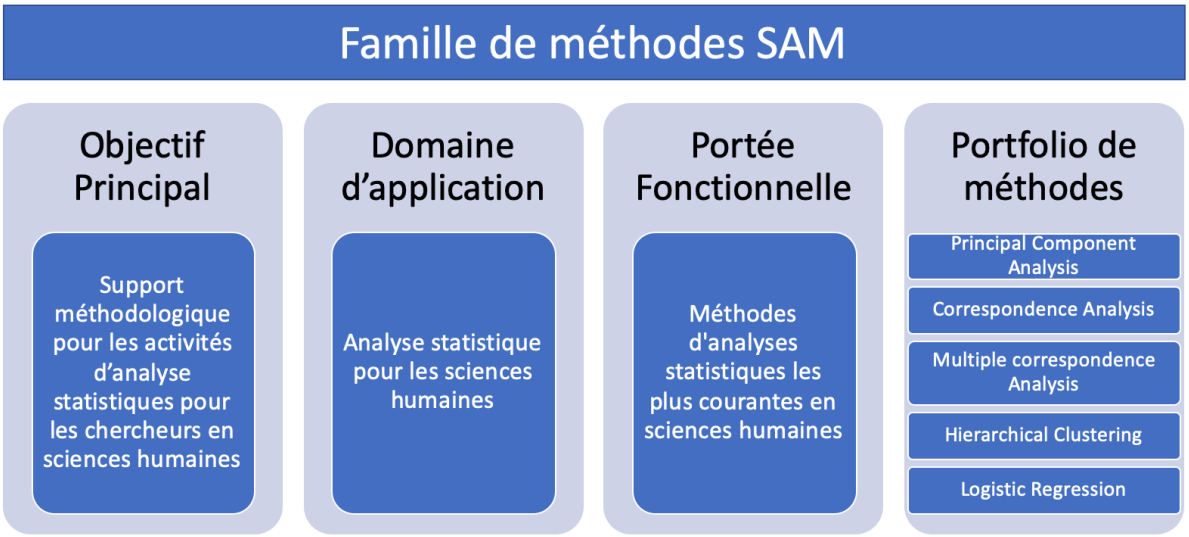


Figure 26. Interface de la famille SAM

Nous avons choisi d'étudier les cinq méthodes d'analyse statistiques qui étaient les plus couramment utilisées par les chercheurs en sciences humaines que nous avons interrogé.

- L'analyse des composantes principales (PCA) (Pearson 1901) (Hotelling 1933) est une technique d'analyse factorielle. L'analyse prend comme entrée un tableau avec n lignes et p colonnes et vise à réduire la taille du tableau en déterminant de nouvelles variables réduites contenant plus d'informations. Cette méthode fournit un diagramme de dispersion (avec ajustement des individus par calcul d'inertie pour mesurer la dispersion du nuage à l'aide d'une matrice de corrélation).
- L'analyse par correspondance (AC) (Benzecri 1982) est semblable au PCA, mais bien que le PCA s'applique aux variables numériques continues, l'AC s'applique à deux variables catégorielles. La méthode est utilisée pour étudier le lien entre deux variables qualitatives. La normalisation des données est effectuée à l'aide d'un test du khi carré (en particulier

par une analyse de corrélation). Un tableau de contingence est établi. Les profils moyens (centroïdes) permettent de déterminer si et comment une classe d'individus diffère de la population générale.

- L'analyse par correspondance multiple (MCA) (Benzécri, 1973) est semblable à une AC appliquée à plus de deux variables. La table de normalisation des données est réalisée de manière à obtenir des nombres binaires. Les lignes représentent les individus, tandis que les colonnes représentent les variables. Ce tableau est ensuite transformé en un tableau disjonctif complet. Les lignes représentent toujours des individus, mais les colonnes représentent les termes (chaque terme est lié à une variable). Une fois que le nombre de variables augmente, les données sont représentées comme une table d'hyper-contingence (appelée table Burt). La similarité entre les individus est déterminée par le nombre de termes communs. Ainsi, deux termes sont semblables s'ils sont présents ou absents chez de nombreuses personnes. Le calcul du profil est également utilisé pour obtenir le total des lignes et des colonnes, comme dans AC. Chaque profil formant un nuage de points est ensuite projeté dans un espace différent. La projection sur un plan unique permet de mettre en évidence une série de directions orthogonales, d'étudier les projections de deux nuages, ce qui permet de choisir le nombre d'axes de projections et d'étudier les valeurs représentant l'inertie de chaque axe.
- L'objectif du regroupement hiérarchique (Sokal 1963) est de classer les individus partageant des similitudes à partir d'un ensemble de variables. La sélection des variables représentant les individus se fait à l'aide de l'écart-type. La distance entre les individus est alors calculée (par exemple avec le calcul de la distance euclidienne ou de la distance Chebyshev), puis la distance entre les groupes (on peut calculer la distance minimale, la distance maximale ou la distance définie par la méthode de Ward (Ward 1963)) pour déterminer une dissimilarité et un indice d'agrégation. Le résultat d'un regroupement hiérarchique est une hiérarchie de classes telle qu'une classe n'est pas vide, chaque individu appartient à une classe et chaque classe est l'union des classes qui sont incluses. Pour interpréter le dendrogramme, une partition est sélectionnée dans la hiérarchie de classe. Le clivage est réalisé à un niveau où l'inertie entre les classes augmente soudainement et significativement. Les données des sous-ensembles restants sont considérées comme pertinentes.
- La régression logistique (Berkson 1944) consiste d'abord à étudier la distribution des différentes variables en calculant les probabilités conditionnelles (Rakotomalala 2011). Deux approches sont proposées pour estimer cette probabilité : basée sur les fréquences ou basée sur le calcul du rapport de vraisemblance. Il est également possible d'utiliser des modèles tels que logit (Berkson 1944) ou probit (Bliss 1934) pour "marquer" chaque individu. On calcule ensuite les corrélations entre les variables indépendantes et la relation entre chaque variable explicative et la variable dépendante (test de Wald). Le modèle de régression logistique est finalement construit en appliquant le test du khi carré pour déterminer si les données contredisent les relations établies. L'ajustement du modèle est nécessaire car il permet de juger de la qualité du modèle final en fonction des données. Cet ajustement peut être fait avec des tests d'analyse résiduelle comme la déviance résiduelle ou Pearson. La présentation et l'interprétation des modèles peuvent être faites par le test Odds ratio et l'analyse des résidus pour vérifier la qualité de la régression.

Ainsi, pour réaliser un projet de recherche on dispose de quatre étapes qui sont : la collecte des données, la définition d'une hypothèse, l'analyse de ces données et à la fin l'interprétation du résultat de l'analyse.

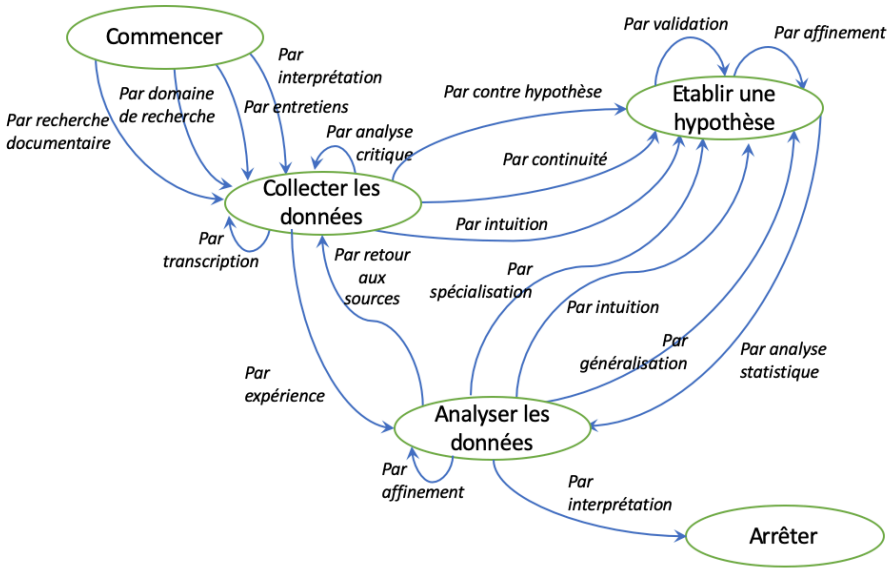


Figure 27. Famille de méthodes SAM

La carte contient quatre stratégies de *collecte de données* : *par des interviews*, *par des recherches documentaires* (sur des archives, des revues publiées, et plus généralement des supports textes ou images)*, par des recherches sur le terrain* (dans le cas des archéologues par exemple) ou *par l'interprétation d'œuvres existantes*. Le chercheur peut mettre en œuvre ces stratégies tant que l'intention de recueillir des données n'est pas atteinte. Le chercheur peut également recueillir des données par analyse critique de sa propre collecte ou par transcription lorsque les données sont entrées dans un fichier informatique sous forme de tableur, de base de données, de fichier xml, etc. Si le chercheur a de l'expérience, il peut analyser directement les données.

Le chercheur a maintenant vérifié les données. Il peut alors *établir une hypothèse* qui correspond à la problématique qu'il veut explorer en fonction des données recueillies. Cela peut se faire à l'aide de 3 stratégies différentes : p*ar continuité des travaux précédents, par contre-hypothèse* (lorsqu'une hypothèse a déjà été établie) ou *par intuition*.

Avant de commencer l'analyse statistique des données, le chercheur doit vérifier si l'hypothèse qu'il/elle veut établir et étudier est originale et qu'aucun chercheur n'y a déjà répondu de la même manière (par validation). Si l'hypothèse est faible, pas assez originale, pas assez spécifique ou au contraire trop spécifique, elle peut être modifiée en appliquant une stratégie *par affinement.*

L'étape suivante consiste alors à *analyser les données* par analyse statistique. Cette section particulière sera détaillée car elle est complexe et nécessite plus d'orientation. Lorsque les données sont analysées, le chercheur peut décider de modifier l'hypothèse si les résultats ne sont pas satisfaisants : par intuition s'il estime que l'hypothèse doit être modifiée, par spécialisation si elle est trop générale, ou par généralisation si elle est trop spécifique.

En analysant les données, le chercheur peut également se rendre compte que les sources ne sont pas complètes ou devraient être modifiées ; la stratégie *par retour aux sources* est alors suivie. Les résultats de l'analyse peuvent être affinés pour améliorer la visualisation des résultats *par affinement*.

Enfin, les données analysées peuvent être interprétées. C'est la dernière étape de la méthode (*par interprétation*).

Sous-carte « Analyser les données par analyse statistique »

L'analyse des données par analyse statistique peut être complexe, car il existe de nombreuses méthodes statistiques et les chercheurs ont besoin de conseils pour les exécuter. Nous affinons la section <Établir une hypothèse, Analyser les données, par analyse statistique> dans une autre carte. Cette carte comprend également les intentions de départ et d'arrêt. Elle a été construite à partir des comptes rendus et des entrevues menées avec les chercheurs en sciences humaines. La Figure 28 présente le modèle de processus obtenu en affinant la section <Établir l'hypothèse, Analyser les données, Par analyse statistique>.

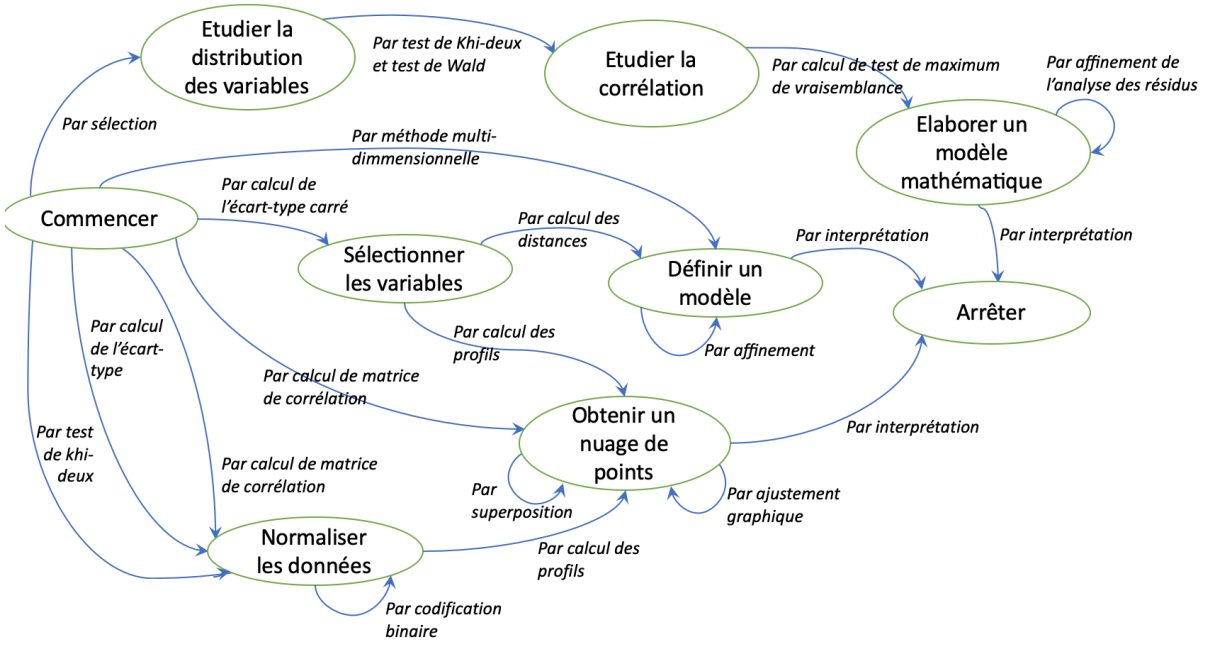


Figure 28. Affinement de la section < Établir une hypothèse, Analyser les données, Par analyse statistique >.

Certains chemins de cette carte correspondent à une méthode particulière. Par exemple, le chemin comprenant les intentions Étudier la distribution des variables, Étudier la corrélation et Élaborer un modèle mathématique ont été extraits de la méthode de la régression logistique. Le chemin comprenant Sélectionner les variables et Définir un modèle vient de la méthode de clustering hiérarchique. Certains chemins sont aussi communs à différentes méthodes d'analyse statistique comme les intentions Normaliser les données et Obtenir un nuage de points qui sont définis à la fois dans PCA, AC et MCA.

II.3. Définir un modèle intentionnel par découverte dans les traces utilisateurs

L'un des objectifs les plus courants de la fouille de processus est la découverte des processus à partir des logs. L'occurrence des activités métiers sont conservées et intégrées dans une base de traces à partir de laquelle on peut ensuite appliquer des techniques de fouille pour découvrir le processus métier réellement exécuté par les utilisateurs (Figure 29). Il est également possible de faire de la découverte de processus intentionnels à partir de ces mêmes traces. Cependant, les logs d'activités ne sont pas les seules sources possibles pour identifier des processus et des documents relatant les activités peuvent également être utilisés à cette fin.

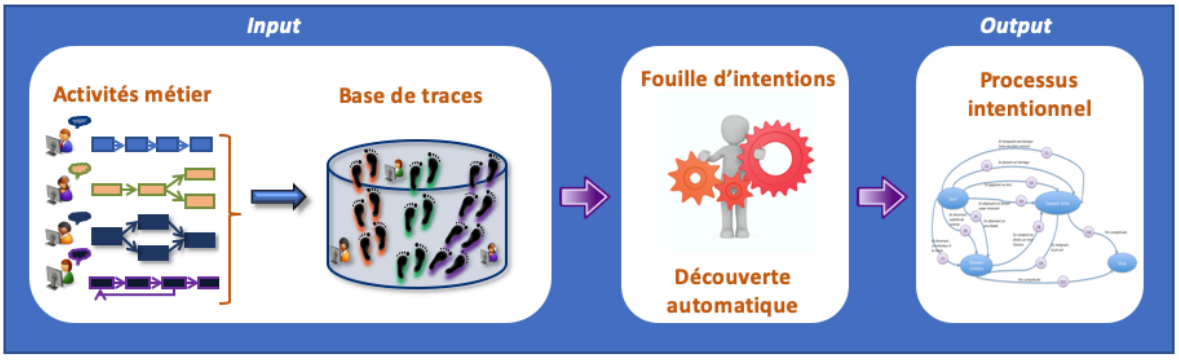


Figure 29. Entrées-Sorties de la construction d'un modèle intentionnel par découverte dans les traces utilisateurs

Les modèles de processus jouent un rôle important dans les organisations. Ils sont utilisés pour diverses raisons lors de la conception, de la refonte, de l'amélioration des processus et de l'introduction de nouveaux systèmes d'information (Van der Aalst 2011). Ils sont utilisés pour voir le processus sous différents angles, pour structurer les discussions des parties prenantes, pour documenter, pour trouver des erreurs dans les systèmes ou les procédures, pour comprendre les facteurs influençant les temps de réponse, pour configurer un système, pour mettre en œuvre différents scénarios et ainsi fournir un retour au concepteur, etc. (Van der Aalst 2011).

La modélisation de processus suit souvent une méthodologie traditionnelle descendante, à partir des exigences, jusqu'à la configuration et la refonte (Van der Werf 2011). L'architecture de ces systèmes doit également évoluer avec le système mais l'architecture et la réalisation ne sont pas souvent d'accord. Autrement dit, les modèles de processus ne sont pas alignés sur la réalité, c'est-à-dire sur ce qui se passe réellement dans les processus ; par conséquent, ils ne peuvent pas fournir de renseignements utiles aux intervenants. Il y a donc un écart entre les systèmes qui soutiennent les processus et ceux qui sont mis en œuvre dans le monde réel et les systèmes deviennent difficiles à entretenir. En effet, dans la plupart des organisations, les changements dans les systèmes ne sont souvent pas documentés de manière systématique et continue (Van der Aalst 2011).

Aujourd'hui, la croissance impressionnante des logs d'événements permet de les analyser. Ces journaux d'événements enregistrés par les systèmes d'information contiennent des informations précieuses sur les processus effectivement mis en œuvre. Ils peuvent être

utilisés pour fournir un aperçu des processus réels ; les écarts peuvent être analysés et la qualité des modèles peut être améliorée. Cependant, la plupart des journaux d'événements enregistrés ne sont pas bien structurés, ce qui cause des problèmes tout en exploitant d'énormes quantités de données. L'un des principaux défis des organisations est d'extraire l'information et la valeur des logs d'événements enregistrés dans les systèmes d'information (Van der Werf 2011) (Van der Aalst 2011).

Compte tenu de la croissance des logs d'événements dans les organisations, des avantages de la modification des processus et de la qualité limitée des modèles de processus fabriqués manuellement, il semble avantageux de relier les logs d'événements produits aux processus. Les processus réels peuvent être découverts pour comprendre ce que les utilisateurs font réellement et les modèles de processus existants peuvent être évalués et améliorés. L'idée de la découverte automatique de processus est apparue initialement dans le domaine de l'ingénierie des logiciels (Cook 1998) et a été appliquée pour la première fois sur des logs de workflow (Agrawal 1998). Cette discipline de recherche se situe entre l'apprentissage automatique et la fouille de données, d'une part, et la modélisation de processus et l'analyse de modèles, d'autre part (Van der Aalst 2011). L'idée de base est d'extraire les connaissances des logs d'événements enregistrés par les systèmes d'information qui contiennent des informations précieuses sur toutes les interactions des utilisateurs avec les systèmes d'information. Les approches de fouille de processus visent à modéliser les comportements des utilisateurs en termes de séquences de tâches et de branchement de manière automatique (Rozinat 2007) (Van der Aalst 2004) (Van der Aalst 2011). Les processus extraits à partir des logs peuvent être utiles pour comprendre comment les gens travaillent vraiment (individuellement ou collectivement), analyser comment les processus réels diffèrent des modèles de processus prescrits, et ainsi améliorer la qualité des processus édictés et augmenter la productivité. Cependant, les approches de fouille de processus sont rigides parce qu'elles ne tiennent compte du processus que du point de vue des activités et de leurs relations et elles rencontrent donc un certain nombre de problèmes tels que le bruit, les tâches cachées, les doublons, les boucles, etc. (Van der Aalst 2004).

La fouille des intentions humaines est un challenge difficile dans un large éventail de domaines. Ils définissent l'intention de différentes façons, les utilisent pour différents objectifs, appliquent différentes techniques pour découvrir les intentions, etc. Cela donne lieu à une nouvelle discipline de recherche, dite Fouille d'intentions.

Au sens large, la fouille d'intention désigne toutes les approches qui visent à découvrir les intentions à partir des interactions des utilisateurs avec les systèmes d'information. Il s'agit de découvrir les intentions des utilisateurs à partir des activités des utilisateurs (logs, traces, etc.). Pourtant, dans le contexte des systèmes d'information, le concept de but ou d'intention est essentiel pour toute organisation puisque les systèmes d'information sont créés pour répondre aux besoins de l'organisation. La notion d'intention peut aussi être considérée comme un objectif qu'un utilisateur veut atteindre sans le dire (Rolland 2007a) et ce concept est largement utilisée en systèmes d'information : ingénierie des services [Rolland, 2010], ingénierie des méthodes (Rolland, 1993) (Salinesi 2003) (Mirbel 2006) (Najar 2011), pour l'adaptation au contexte (Ralyté 2003) (Mirbel 2006), processus opérationnels (Outmazgin 2013), définition des exigences (Yu 1987) (Van Lamsweerde 2001), recherche d'information (Hashemi 2008) (Baeza-Yates 2006) (Chen 2002).

Nous avons étudié plusieurs approches qui permettent de découvrir des processus intentionnels par des techniques de fouilles.

II.3.1. Définir un modèle de processus intentionnel en se basant sur les traces utilisateurs : l'approche FlexPAISSeer

Cette approche a été développée dans le mémoire de master d'Elena Epure (Epure 2013) (Epure 2014b).

Une catégorie de systèmes d'information, très adoptée par les organisations, est le *Process Aware Information System* (PAIS) défini comme un « système logiciel qui gère et exécute des processus opérationnels impliquant des personnes, des applications et/ou des sources d'information sur la base de modèles de processus » (Van der Aalst 2011). Les systèmes d'information tels que les systèmes de workflow, de gestion des processus opérationnels et de gestion des ressources d'entreprise entrent dans cette catégorie. Par conséquent, le PAIS est un type particulier de système d'information qui intègre explicitement ou implicitement des schémas de processus opérationnels.

En assurant la flexibilité des processus dans le PAIS, une plus grande importance est accordée au pouvoir humain de prendre des décisions, mais aussi à la capacité humaine de comprendre le contexte du processus et d'utiliser les connaissances du processus. Les processus flexibles impliquent soit que les humains peuvent adapter les processus quand ils le veulent selon leur raisonnement, soit qu'ils doivent prendre des décisions différentes liées aux processus lorsque c'est requis. Un participant expérimenté au processus qui connaît très bien le processus opérationnel est plus en mesure de prendre une décision quant à l'action à exécuter ou à la façon de modéliser un fragment de processus au moment de l'exécution, compte tenu de la situation actuelle. En revanche, cela peut être très difficile pour un participant moins bien informé qui pourrait uniquement se fier aux définitions prescriptives des processus ou pour un participant qui fait face à un environnement très dynamique (Schonenberg 2008) (Van der Aalst 2009).

Les PAIS peuvent offrir un soutien pour découvrir les processus de façon ascendante en recueillant des logs d'événements pendant leur exécution. Cela est rendu possible par la fouille de processus, une discipline qui a évolué sans cesse au cours des dernières années et dont le principal objectif est de « découvrir, surveiller et améliorer les processus réels » en transformant les données de l'événement en connaissances précieuses (Van der Aalst 2011).

Principe de l'approche FlexPAISSeer

L'approche est composée de deux algorithmes, IntentMiner, qui permet de faire de la découverte de modèles de processus intentionnels, et IntentRecommender, qui permet de faire de la recommandation à la volée.

L'algorithme IntentMiner développé dans cette approche se déroule en cinq phases, comme indiqué dans la figure suivante.

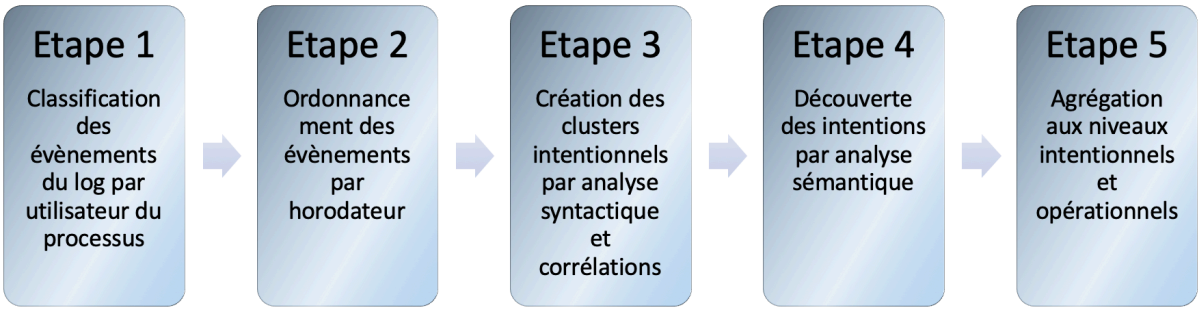


Figure 30. Algorithme IntentMiner

Étape 1 : L'algorithme commence par classer les événements par participant au processus.

Étape 2 : Pour chaque série d'événements appartenant à un participant, les événements doivent être classés par horodatage. Cela implique que les intentions seront également ordonnées dans le temps (ce qui est toutefois un cas simplifié de la réalité).

Étape 3. Les événements transformés (classés par participant au processus et classés par horodatage) sont regroupés dans des groupes intentionnels, eux-mêmes classés (Figure 31). Ce regroupement se fait par l'identification d'une fonction de coût qui mesure la dissimilarité de deux événements. Cependant, si on compare cette approche aux algorithmes traditionnels de clustering où la fonction de coût est appliquée pour chaque paire appartenant à l'espace entier des événements, dans cet algorithme de fouille la fonction est appliquée seulement pour deux événements consécutifs. Par conséquent, cette fonction quantifie la corrélation intentionnelle de deux événements consécutifs. La corrélation intentionnelle dépend des informations connues et incorporées dans les événements : le trace id, le type d'entité et l'instance d'entité spécifique identifiée par son id, stockées dans les informations contextuelles de l'événement. En outre, ces valeurs ne devraient pas contribuer également à la fonction. Par exemple, deux événements consécutifs qui font référence à la même instance d'entité devraient avoir une corrélation beaucoup plus forte que deux événements qui font référence au même type d'entité. La corrélation intentionnelle peut également être considérée comme une analyse syntaxique des événements, car elle quantifie les informations structurelles des actions.

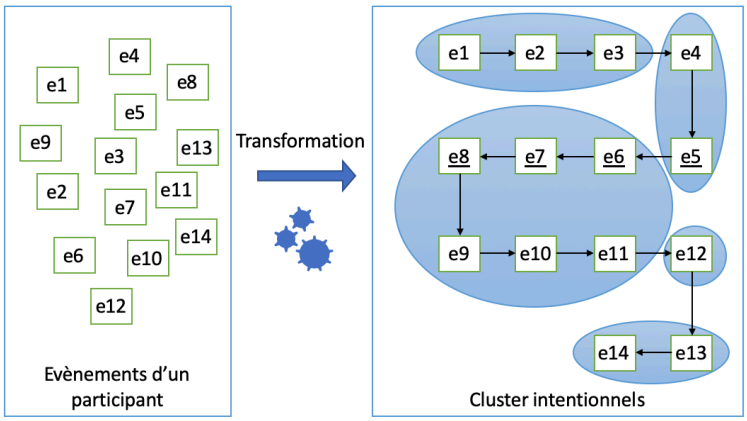


Figure 31. Transformation d'évènements dans des groupes intentionnels

L'étape suivante est la normalisation des valeurs de corrélation calculées par la fonction. Par exemple, si la séquence d'événements appartient à la même trace, alors la valeur de corrélation minimale trouvée dans cet ensemble est « `traceCorrelationFactor * ReferenceUnit` ». Mais, puisque cette corrélation comparée aux autres valeurs de corrélation est la plus basse, de toutes les valeurs est soustraite ce minimum de sorte que la valeur de corrélation minimale devient nulle (voir (Epure 2013) pour plus de détails sur le calcul de corrélation).

Étape 4. Maintenant que les groupes intentionnels sont identifiés, il faut affecter une intention à chaque groupe. L'analyse sémantique des groupes intentionnels consiste à comprendre la structure des actions du processus en interprétant les champs « Opération » et « Type d'événement ». Le modèle d'entité doit être connu à l'avance (il est défini à l'aide du Web-based Application Designer et il est fourni comme entrée au générateur), l'ensemble des opérations est fini et peut également être identifié.

Étape 5. L'étape finale est l'agrégation qui se déroule à deux niveaux différents :

- L'agrégation de l'instance de processus intentionnel dans le modèle de processus global. Si de nouvelles intentions ou de nouvelles connexions entre les intentions sont découvertes, elles sont ajoutées à la carte correspondant au processus.
- L'agrégation des groupes intentionnels correspondant à une intention spécifique dans un modèle de flux de contrôle composé d'événements.

Afin de construire le processus complet, les stratégies doivent également être identifiées. Les stratégies définissent de multiples possibilités de réalisation d'une intention. Par conséquent, ils peuvent être extraits des groupes intentionnels par l'identification des chemins dans le modèle de flux de contrôle constitué d'événements. Cependant, le premier problème est que les chemins multiples ne représentent pas nécessairement des stratégies différentes et le deuxième problème est d'identifier le nom de la stratégie à partir des informations connues. On considère ici qu'une seule stratégie relie deux intentions.

Illustration

L'entreprise test est une société de logiciels située aux Pays-Bas. Le produit principal de l'entreprise est une plate-forme qui soutient le développement de manière agile. La plate-forme permet aux clients de concevoir, prévisualiser, re-concevoir et déployer une grande variété d'applications commerciales. Fournir aux clients un système qui peut être adapté au contexte organisationnel changeant est un aspect clé de la vision de l'entreprise. La flexibilité des entreprises est assurée de deux façons. D'une part, les applications commerciales du client peuvent être générées en réduisant considérablement le temps de développement, et les extensions supplémentaires sont réalisées en suivant les principes agiles. D'autre part, même si certains des processus de l'application commerciale résultante peuvent être automatisés, la plupart d'entre eux intègrent de la flexibilité en soutenant une implémentation centrée sur les données et axée sur l'homme. Dans une approche centrée sur les données, les éléments qui influencent la promulgation du processus sont les entités, les attributs d'entité et les relations d'entité. Une transition pendant l'exécution du processus est déclenchée par un changement dans l'état de l'entité au moyen de formulaires utilisateur (Reichert 2012).

L'algorithme *IntentMiner* a été appliqué sur une branche particulière de la compagnie utilisé par plusieurs crèches aux Pays-Bas. Le système de garde d'enfants prend en charge tous les processus principaux et se compose de deux composants logiciels : une application back-office et un portail Parents, tous deux basés sur le Web. Les principaux modules de processus sont : le module de relation client incluant l'enregistrement des parents, des lieux et des enfants ; le module d'administration des contrats incluant l'enregistrement, le stockage et le traitement automatique des contrats entre les parents et les crèches ; le module de planification, y compris la planification de l'enfant en fonction des groupes, des emplacements et des périodes ; et le module de comptabilité, y compris la facturation, les fonctionnalités de gestion du crédit et l'intégration à un système de comptabilité.

Le processus le plus complexe et le plus souple est l'organisation du planning de l'enfant. Il est déclenché par le processus d'enregistrement et il communique avec les autres modules également. Sa complexité est représentée par la diversité des scénarios existants en matière de planification et par le fait que, même si le système fournit des renseignements contextuels, le participant décide de la façon de mettre en œuvre le processus en conséquence. La planification dépend de l'âge de l'enfant, de l'emplacement et de la préférence des parents d'avoir leur enfant dans un groupe horizontal ciblant le même âge, un groupe vertical composé d'enfants d'âges différents, mais dans la même proportion et un groupe variable. À l'exception de la planification standard, il existe également d'autres types non standards : planification alternée lorsque les groupes, les emplacements ou/et les créneaux horaires sont alternés pour un enfant, planification occasionnelle qui est un soin accessoire, non répétitif, et la planification des vacances. Le processus de planification comprend également la gestion d'une liste d'attente et la gestion de la profession de groupe. Selon la planification choisie, les autres processus peuvent également changer, comme le flux financier nécessite des actions spécifiques du participant au processus dans le cas de la planification accessoire ou de vacances, mais pas nécessairement les mêmes.

Bien que le système de garde d'enfants ait un degré élevé de flexibilité, il a été remarqué lors de son utilisation que cela peut également créer des problèmes. Les participants inexpérimentés de ce processus adoptent souvent une manière de procéder inefficace ou font des erreurs pendant la mise en œuvre en raison de la complexité du scénario et du manque de soutien. De plus, les propriétaires de processus ont remarqué que même si on s'attendait à ce qu'un processus soit mis en œuvre d'une façon précise, la réalité était très différente. Par conséquent, la fouille de processus peut offrir des renseignements très utiles sur la façon dont le système est utilisé et pourrait également constituer la base des recommandations formulées au cours du processus. Toutefois, si le système suggère des activités comme « changer la valeur de l'attribut A de l'entité E » ou « créer un lien entre l'entité E1 et l'E2 », cela serait trop vague pour aider réellement le participant à prendre la décision. Toutefois, recommander les intentions possibles comme « Mettre à jour le contrat » ou « Créer un nouveau groupe pour l'emplacement » et les stratégies possibles pour atteindre une intention déjà choisie peuvent améliorer le processus décisionnel. Décider en termes d'intentions et de stratégies est plus proche du raisonnement humain et de la compréhension. Une fois la décision prise au niveau intentionnel, le système peut en effet suggérer au participant au processus l'action à exécuter ensuite.

La première étape pour pouvoir faire des recommandations efficace est de comprendre le processus utilisé par les utilisateurs. Pour ce faire, le processus intentionnel a été identifié de manière manuelle. Dans la Figure 32, le principal processus soutenu par le système de garde d'enfants est modélisé à l'aide du méta-modèle MAP (Rolland 1999). Les intentions qui ont été identifiées par l'analyse du système et les entrevues avec les intervenants de la compagnie sont les suivantes : « Demander garde d'enfant », « Planifier garde d'enfant » et « Facturer garde d'enfant». Toutes ces sections incorporent des processus complexes, ils peuvent donc être décomposés dans d'autres cartes.

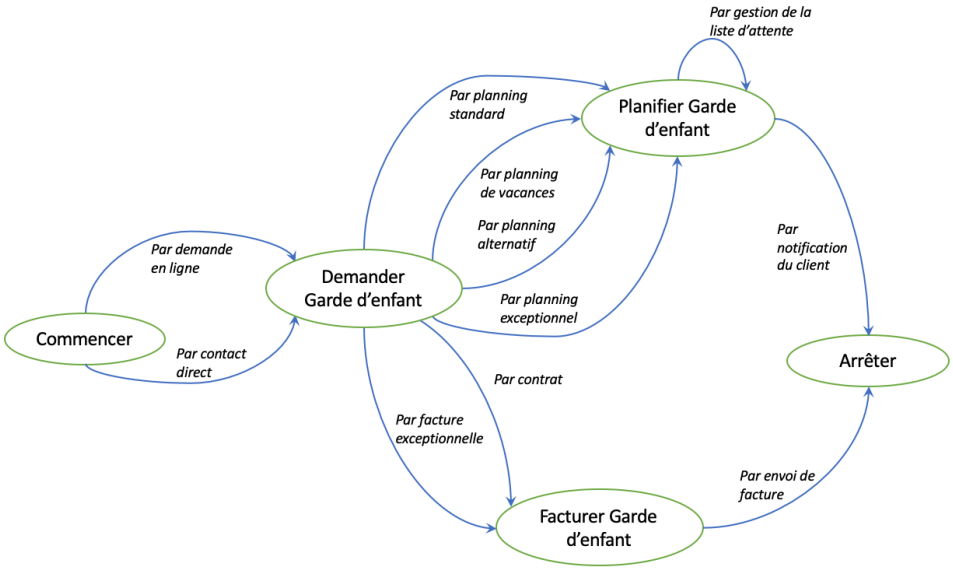


Figure 32. Carte principale du processus de la crèche

Afin d'atteindre l'objectif « Demander garde d'enfant », il existe deux stratégies possibles : « Par demande en ligne » (le parent soumettra une demande de services de garde au moyen du portail) et « Par contact direct » (le parent communique directement avec l'organisme de garde d'enfants et présente une demande, le participant responsable du processus de communication créera la demande directement dans le système).

Comme nous l'avons déjà mentionné, l'intention « Planifier garde d'enfants » est la plus complexe et la plus flexible. Une fois que l'enfant et le parent sont inscrits dans le système et qu'une demande a été soumise, la planification de l'enfant sera effectuée par le participant au processus au moyen de la demande sur le back office. Une ou plusieurs stratégies selon les critères précisés dans la demande sont appliquées pour atteindre cette intention : « Par planification standard », « Par planification de vacances » et « Par planification alternative ». Si la planification requise n'a pas pu être réalisée en respectant tous les critères du parent, la demande est inscrite sur une liste d'attente et le parent est contacté. Plus tard, cette demande sera traitée soit lorsque le parent fait une nouvelle demande avec d'autres critères ou lorsqu'un créneau libre est identifié (un nouveau groupe est créé dans le lieu de garde ou un autre enfant quitte un groupe existant, etc.). Le parent pourrait également communiquer avec l'organisation après la planification permanente, en demandant une planification exceptionnelle (pour plusieurs jours ou une période précise).

La première fois que la demande de planification permanente est faite, le participant au processus génère un contrat. Ensuite, il l'envoie par le biais de la demande au back office aux

parents qui sont en mesure de l'accepter ou de ne pas utiliser le portail. La facture est gérée pour la première fois dans le cas de la planification permanente « Par contrat », ou lorsqu'une planification est modifiée et donc le contrat, ou lorsque des frais de garde accessoires sont demandés (une facture spéciale est émise en plus de la facture permanente).

L'intention Stop est atteinte à partir de l'intention « Planifier garde d'enfant » par la stratégie « Par notification du client » et/ou de l'intention « Facturer garde d'enfant » par la stratégie « Par envoi de facture ».

De plus, l'intention « Demander garde d'enfant » avec la stratégie « Par contact direct » est affinée à l'aide d'une autre carte (Figure 33).

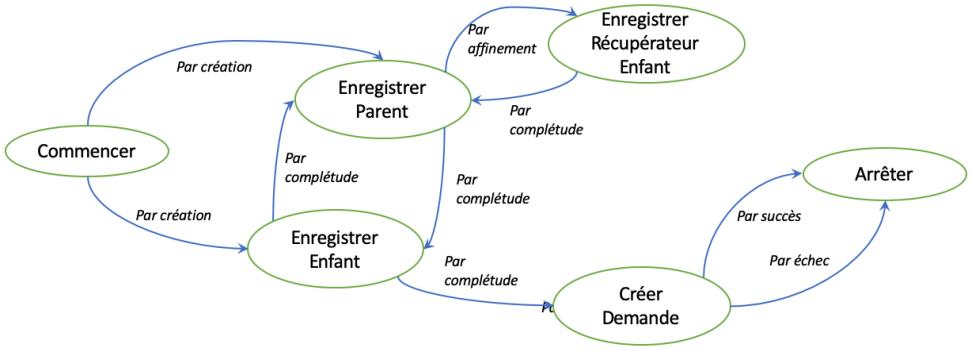


Figure 33. Sous-carte pour la section <"Demande de Garde d'enfant ", "Par contact direct">

Lorsque la demande est faite par contact direct, toutes les opérations peuvent être faites à partir de l'application back-office et cela dépend de la préférence du participant au processus : inscrire le parent, puis l'enfant, puis le « récupérateur » de l'enfant (facultatif) ou alors inscrire le parent, puis le récupérateur de l'enfant, puis l'enfant, etc. Après l'enregistrement de l'enfant est réalisé une demande de garde d'enfant à partir du portail (par le parent) ou de la demande sur le back office (par le participant au processus). Enfin, selon les critères mentionnés et les détails techniques, la demande de services de garde se termine par un succès ou un échec pour différentes activités intégrées aux stratégies correspondantes « Par succès » et « Par échec ».

L'algorithme de fouille de IntentMiner est utilisé pour le processus de « demande de services de garde », car elle intègre la flexibilité par l'existence de multiples stratégies et l'ordre multiple des intentions. Cependant, étant moins complexe que le processus de planification, il peut être analysé en profondeur plus facilement.

La façon d'adopter les processus dans le système de garde d'enfants est dictée par des artéfacts. Plus spécifiquement, les transitions de leurs états composées par les valeurs des attributs définissent le flux de contrôle. Ainsi, une analyse des entités impliquées dans le processus de « demande de garde d'enfants » devient obligatoire.

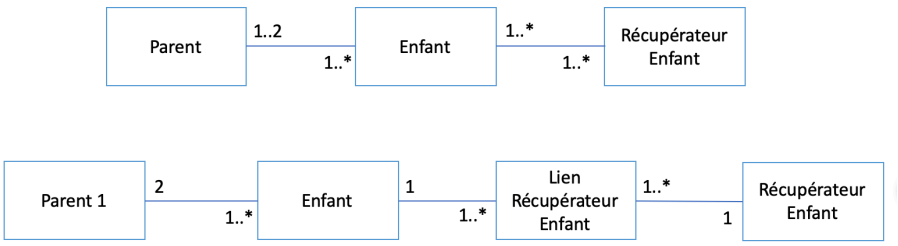


Figure 34. Entités impliquées dans le processus d'enregistrement

La Figure 34 présente un instantané des entités pertinentes pour l'enregistrement. Certains attributs et relations avec d'autres entités externes sont omis par souci de simplicité. Le modèle conceptuel définit trois entités de type Enfant, Récupérateur Enfant et Parent et leurs relations connexes. Cependant, ce modèle n'est pas celui utilisé lors de la génération de l'application en raison d'une contrainte imposée par le générateur : les relations plusieurs-à-plusieurs ne sont pas supportées. Par conséquent, l'entité Enfant a rattaché deux relations « Parent 1 » et « Parent 2 » au même type d'entité, Parent. Avec Enfant et Récupérateur Enfant, la transformation est différente en raison de la multiplicité inconnue de la relation d'association. Pour résoudre ce problème, une nouvelle entité de type Lien Récupérateur Enfant est introduite.

Une partie du scénario concernant l'inscription pourrait être : Inscrire Parent 1, Inscrire Enfant, Inscrire Parent 2 et Inscrire Récupérateur Enfant. La manipulation des entités via les opérations CRUD (« Créer », « Lire », « Mettre à jour » et « Supprimer ») composera le processus intentionnel de bas niveau (représenté comme un flux intentionnel dans la Figure 35).

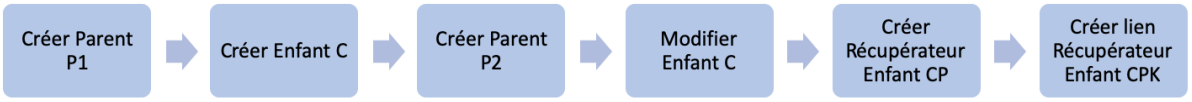


Figure 35. Intentions de bas niveau dans le processus d'enregistrement

(a) Extraire les logs des participants et trier par horodatage	$e_1 \rightarrow e_2 \rightarrow e_3 \rightarrow e_4 \rightarrow e_5 \rightarrow e_6 \rightarrow e_7$ e_1: lire la liste des entités Enfants e_2: Lire l'entité Enfant avec Id C1 e_3: Modifier l'entité Enfant avec Id C1	E_4: Lire la liste des entités Enfant E_5: Lire l'entité Enfant avec Id C2 E_6: Lire la liste des entités Parent E_7 : Lire la liste des entités de récupérateurs d'enfants
(b) Appliquer l'analyse syntaxique	$F(e_1,e_2)=0,5*0+0,3*1+0,2*0=0,3$; $F(e_2,e_3)=0,5*1+0,3*1+0,2*0=1$; $F(e_3,e_4)=0,5*0+0,3*1+0,2*0=0,3$;	$F(e_4,e_5)=0,5*0+0,3*1+0,2*0=0,3$; $F(e_5,e_6)=0,5*0+0,3*0+0,2*0=0$; $F(e_6,e_7)=0,5*0+0,3*0+0,2*0=0$;
(c) Appliquer l'analyse de flux	$C_{I1}=\{e_1,e_2,e_3\}$. $C_{I2}=\{e_4,e_5\}$. $C_{I3}=\{e_6\}$. $C_{I4}=\{e_7\}$ e_1 et e_2 ont une plus grande corrélation que 0 et sont groupés dans C_{I1}. La corrélation de e_3 avec e_2 est plus grande que sa corrélation avec e_4, donc e_3 est ajouté à C_{I1} également. Le premier changement est identifié (la décroissance de 1 à 0,3) donc C_{I2} est construit avec e_4. De plus, la corrélation de e_5 avec e_4 est plus grande que sa corrélation avec e_6 donc e_5 est ajouté à C_{I2}. Le changement (de 0,3 à 0 marque la création de C_{I3} avec e_6. Finalement, puisque la corrélation de e_6 avec e_7 est nulle, on construit C_{I4} avec e_7.	
(d) Appliquer l'analyse sémantique	I_1 (modifier entité Enfant) $\rightarrow$ I_2 (lire entité Enfant) $\rightarrow$ I_3 (lire entités Parent) $\rightarrow$ I_4 (lire entité Récupérateur Enfant)	

Figure 36. Exemple d'application de l'algorithme IntentMiner

Après l'analyse du back-office Childcare, les opérations suivantes ont été identifiées : Créer entité, Modifier entité, Lire entité, Supprimer entité, Modifier champs, Créer relation, Modifier relation, Lire entités, Chercher entité, Chercher entités, Montrer rapport. De plus, la composition d'intention qui constitue la base de connaissances prédéfinie pour l'analyse sémantique est la suivante.

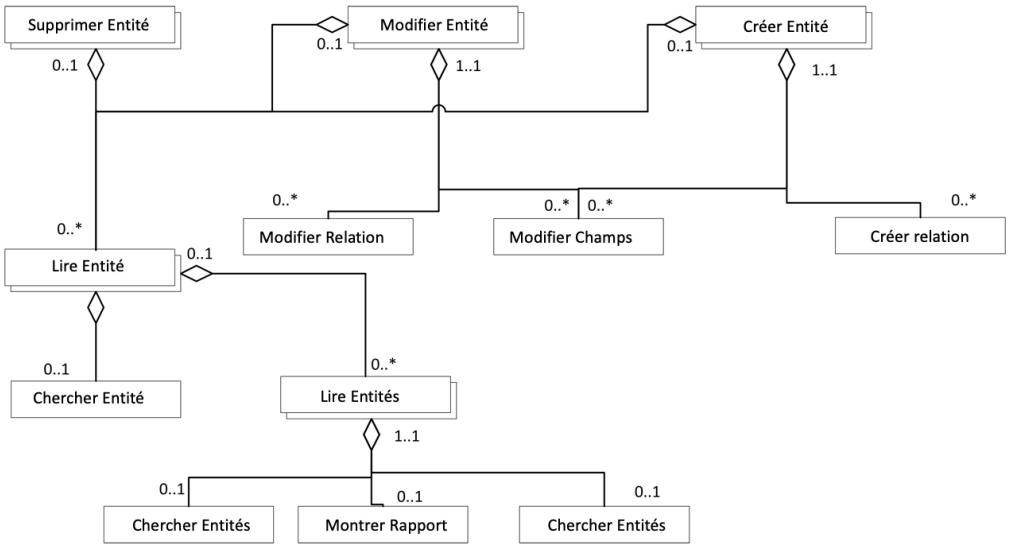


Figure 37. Composition des intentions pour l'analyse sémantique

La Figure 37 montre qu'il y a quatre types d'intentions qui pourraient être identifiées à l'avance pour chaque type d'entité : Créer entité, Modifier entité, Supprimer entité, Lire une entité et Lire entités. En outre, il identifie une liste d'actions intentionnelles qui ne pouvaient pas être considérées comme des intentions de bas niveau autonomes mais comme des actions appartenant à certaines intentions : Modifier relation, Modifier champ, Créer relation, Chercher entité, Chercher entités, Montrer rapport. Il y a deux types de relations de composition : une pour marquer qu'une action appartient à une intention et une autre pour marquer qu'une intention pourrait appartenir à une autre intention. Cela signifie que certaines intentions sont le but final du participant au processus ou qu'elles sont atteintes tout comme la réalisation d'autres intentions : Lire entités, Lire entité. L'exploration de l'intention intégrera ces connaissances pour extraire l'intention d'un cluster intentionnel en appliquant l'analyse sémantique. Une fois que de nouvelles intentions ou de nouvelles actions sont identifiées, la connaissance interne de la technique de fouille d'intentions doit également être mise à jour.

Cet algorithme a permis de tester l'utilisation des intentions de bas niveaux, le travail a été continué par la suite pour permettre de faire de la recommandation aux utilisateurs pendant leur exécution du processus (section III.2.2).

II.3.2. Définir un modèle de processus intentionnel en se basant sur les traces utilisateurs : l'approche Map Miner Method (MMM)

Cette approche a été développée dans le cadre de la thèse de Ghazaleh Khodabandelou (Khodabandelou 2014c).

MMM utilise des techniques de fouille de processus pour trouver le modèle de comportement des utilisateurs. Cependant, contrairement aux méthodes habituelles de la fouille de processus, MMM modélise les comportements des utilisateurs en termes d'attentes et de stratégies. Il adopte un modèle de processus axé sur l'intention pour concevoir les processus, tandis que les approches de fouille habituelles précisent les comportements des utilisateurs en termes de séquences de tâches et de branchements.

Principe du Map Miner Method

D'un point de vue technique, l'approche MMM comporte trois étapes. Tout d'abord la modélisation des activités des utilisateurs : MMM utilise des modèles de Markov caché (MMH) (Rabiner 1989) pour modéliser les activités des utilisateurs en une topologie à plusieurs niveaux des activités des utilisateurs et des stratégies correspondantes. Ensuite, en utilisant différentes techniques d'apprentissage, les stratégies des utilisateurs peuvent être estimées. L'exploration des intentions nécessite la conception de nouveaux algorithmes et outils pour générer des modèles de processus Map. Dans cette perspective, deux algorithmes ont été développés : Deep Miner (à partir des stratégies estimées des utilisateurs, l'algorithme identifie les intentions des utilisateurs et, par conséquent, il découvre des modèles de processus intentionnels avec des intentions de bas niveau, ce qu'on appelle des pseudo-cartes) et Map Miner (à partir des pseudo-cartes, l'algorithme Map Miner structure les modèles de processus Map en regroupant les intentions de bas niveau en intention de haut niveau par rapport à la définition de l'intention dans le formalisme Map).

L'ensemble de l'approche a été appliqué et validé sur des ensembles de données pratiques, d'abord dans un contexte de laboratoire où un exemple a été réalisé avec des étudiants en Master de l'Université de Paris 1 Panthéon-Sorbonne pour la création de diagrammes Entité-Relation. Deuxièmement, MMM a été appliqué dans une expérience à grande échelle, sur les logs d'événements des développeurs d'Eclipse UDC (Usage Data Collector) (Eclipse 2013) pour démontrer la mise à l'échelle de MMM. Les modèles de processus Map qui en résultent fournissent une compréhension précieuse des processus suivis par les développeurs. Une expérience qualitative a été menée qui permet d'évaluer l'étude de cas. Cette évaluation a fourni les commentaires des développeurs sur la perception, l'efficacité et la facilité d'utilisation de la MMM dans la pratique. Un outil, appelé Map Miner Tool, a également été conçu et développé pour automatiser MMM. Il permet de pratiquer l'approche proposée pour obtenir le modèle de processus Map personnalisé uniquement à partir des traces des utilisateurs en ajustant certains paramètres.

L'élément d'entrée de MMM, comme dans la majorité des approches de fouille de processus, est l'ensemble temporel des activités des utilisateurs – les interactions des utilisateurs avec un outil de systèmes d'information pendant une certaine période de temps. Les séquences d'activités ainsi enregistrées sont une trace d'activités pour cet utilisateur. Dans MMM, les traces complètes des utilisateurs qui se sont produites au cours d'une exécution de processus sont analysées. Il convient de noter qu'un processus dans le contexte des systèmes d'information est défini comme une séquence d'activités liées les unes aux autres par un objectif commun (Rozinat 2010). Par conséquent, il existe une forte corrélation et dépendance entre les traces des utilisateurs et elles ne peuvent pas être considérées comme une entité unique, indépendante et non corrélée. Une séquence d'activités contient plus de renseignements enrichis sur l'intention des utilisateurs qu'une seule activité, et ce à

plusieurs niveaux. En effet, l'analyse d'une séquence d'activités permet de déterminer les intentions de haut niveau (comme l'objectif organisationnel), tandis que l'analyse d'activités individuelles mène à des intentions de bas niveau moins informatives, aussi appelées intentions de base ou intentions d'action, qui sont plus proches des activités que des intentions. Les méthodes de fouille de processus ou de fouille d'intentions sont appliquées pour les logs/traces d'un ou de plusieurs utilisateurs. MMM analyse les traces d'un groupe d'utilisateurs réalisées lors de la mise en œuvre d'un processus.

Les approches de fouille de processus utilisent de nombreux algorithmes, des techniques de classification ou de regroupement, des techniques basées sur des ontologies, etc. Les méthodes de fouille d'intentions utilisent principalement des techniques de classification pour classer une seule trace dans une classe d'intention. Le choix des techniques de classification est intéressant dans certains contextes puisque ces travaux tentent d'inférer une intention liée à une requête, puis de la classer dans une catégorie. Ils ne tiennent pas compte des intentions des utilisateurs dans le cadre de l'exécution d'un processus. Alors qu'en raison de la structure multi-niveaux du MMM (activités, stratégies et intentions des utilisateurs), il doit y avoir un modèle mathématique pour mettre en place, au moins, une topologie à deux niveaux pour les activités et les stratégies des utilisateurs.

Le modèle mathématique utilisé dans MMM est le modèle de markov caché (Hidden Markov Models - HMMs) (Rabiner 1989). Parmi les techniques utilisées pour modéliser différents aspects du comportement humain (Gray 1992), les modèles de Markov caché se sont avérés appropriés pour modéliser les processus du monde réel, en particulier les processus cognitifs inobservables (Hayashi 2003) (Hoey 2007). Les HMM sont des automates finis stochastiques et un type particulier de réseau bayésien (Friedman 1997). Ils ont été utilisés dans une grande variété de contextes et se sont avérés utiles dans divers domaines, comme la reconnaissance vocale (Juang 1991) (Gales 1998) (Rabiner 1989), la prévision des données financières (Zhang 2004), le traitement des signaux (Kil 1996), le regroupement de données génériques (Li 1999) et plus tard appliqué largement dans le domaine de la bio-informatique (Martelli 2002) (Delorenzi 2002).

Les phases de MMM sont décrites comme suit :

- Estimation des paramètres : cette phase vise à estimer les paramètres du HMM. Les paramètres d'un HMM doivent être estimés par un apprentissage supervisé ou non supervisé :
    - L'apprentissage supervisé permet d'estimer les paramètres d'un HMM par Maximum Likelihood Expectation. Il prend comme éléments d'entrée les activités des utilisateurs avec les stratégies connexes et comme éléments de sortie les matrices de transition et d'émissions. Par conséquent, les besoins de ce type d'apprentissage sont couteux, puisque les utilisateurs doivent labelliser leurs activités. Cependant, l'apprentissage supervisé présente certains avantages : il permet de valider l'hypothèse proposée d'un point de vue pratique. L'apprentissage supervisé permet également de vérifier la conformité des modèles prescrits et des modèles découverts.
    - L'apprentissage non supervisé permet d'estimer les paramètres d'un algorithme HMM by Baum-Welch. Il prend comme éléments d'entrée les activités des

utilisateurs et le nombre de stratégies et comme éléments de sorties les matrices de transition et d'émissions. L'apprentissage non supervisé présente certains avantages : il ne nécessite pas les traces de stratégies liées aux traces d'activités. Cela en fait une technique facile à utiliser et à automatiser. En outre, l'apprentissage non supervisé est plus efficace que l'apprentissage supervisé. En d'autres termes, les résultats obtenus par l'apprentissage non supervisé sont plus susceptibles de refléter les relations réelles entre les traces que ceux obtenus par apprentissage supervisé.

Les HMM utilisent des matrices de transition et d'émissions pour modéliser les activités observées en termes de stratégies cachées.

- Algorithme Deep Miner : cet algorithme est développé pour découvrir les sous-intentions et ainsi construire les pseudo-cartes. L'algorithme Deep Miner utilise une métrique de fitness et de précision, qui a la propriété intéressante de prendre en compte à la fois la forme et la précision pour optimiser les modèles de processus Map. Bien entendu, Deep Miner utilise le formalisme Map pour définir la topologie d'une Map.
- Algorithme Map Miner : cet algorithme regroupe les sous-intentions en intentions de haut niveau et reconstruit ainsi le modèle de processus Map. Pour ce faire, l'algorithme Map Miner utilise un algorithme de clustering, K-means (Hartigan 1979) pour regrouper les sous-intentions dans les intentions. Comme expliqué plus haut, cette topologie multi-niveaux est due à l'architecture profonde du cerveau humain et au fait que les humains organisent leurs idées et concepts hiérarchiquement. Un algorithme est également développé pour reconstruire le modèle de processus Map à partir des sous-intentions classifiées.

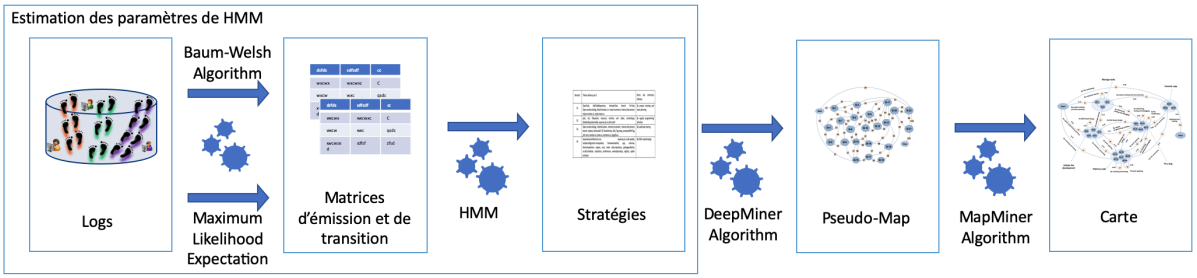


Figure 38. Fonctionnement détaillé de MMM

Illustration

Eclipse Usage Data Collector (UDC) (Eclipse 2013) est un système qui recueille des informations sur la façon dont les développeurs utilisent la plateforme Eclipse. La Fondation Eclipse fournit ces données pour aider les participants et les organisations à mieux comprendre comment la communauté utilise Eclipse (UDC 2013). Dans cette perspective, cette étude de cas vise à modéliser les activités des développeurs UDC en termes d'intentions et de stratégies en construisant automatiquement le modèle de processus Map qui est effectivement suivi par les développeurs. Le modèle de processus Map découvert est utile pour analyser les comportements des développeurs (Khodabandelou 2014b).

Les logs d'événements de cette étude de cas sont des logs de développeurs ayant exécuté des activités dans le Usage Data Collector d'Eclipse (Eclipse 2013). Les logs sont téléchargés sur des serveurs hébergés par la Fondation Eclipse et contiennent 1048576 logs d'événements de développeurs qui ont accepté de renvoyer leurs données à la Fondation Eclipse. Ces données regroupent les activités de plus de 10000 développeurs Java entre septembre 2009 et janvier 2010. Les activités sont enregistrées par horodatage pour chaque développeur, ce qui permet de savoir quand, et par qui, les activités ont été exécutées.

Afin d'appliquer le MMM, il est important de préparer les logs. Le nombre d'activités de développeurs uniques par mois dépasse 500 activités. Ce chiffre contient à la fois les activités récurrentes et les activités non récurrentes (c.-à-d. les activités qui ne sont pas fréquemment exécutées par les développeurs). Les activités non récurrentes ne sont pas représentatives des caractéristiques de comportement des développeurs parce qu'elles n'ont pas été assez répétées pour pouvoir figurer dans un comportement type. Pour cette raison, et aussi pour des raisons de lisibilité, l'étude de cas est limitée aux 130 activités les plus fréquentes réalisées par les développeurs. La Figure 39 contient la liste de ces activités. Certaines de ces activités sont des commandes exécutées directement par les développeurs, d'autres sont des plug-ins ou des fonctionnalités intégrées d'Eclipse. Pour des raisons de lisibilité, le préfixe org.eclipse des activités est supprimé. Les plug-ins et les frameworks sont affichés en caractères gras et les activités associées sont entre crochets. Pour chaque développeur qui a soumis sa trace d'activités à UDC, la trace contenant seulement des activités parmi les 130 plus fréquentes est extraite. De cette façon, les logs d'événements sont préparés pour appliquer MMM.

Une fois les traces prêtes, BWA estime la matrice de transition (stratégies des développeurs). Le nombre de stratégies obtenues par la méthode heuristique pour cette étude de cas est de 10. Les stratégies sont représentées dans la Figure 39 avec leurs groupes d'activités correspondants. Comme mentionné précédemment, ε exprime le niveau de granularité pour une carte. Un expert peut choisir la valeur de ε par rapport au niveau de granularité attendu. Dans cette étude de cas, la valeur de ε est fixée à 0,06 pour avoir un bon compromis entre avoir une probabilité avec une valeur relativement élevée et un nombre raisonnable de sections. Enfin, les cartes obtenues par les algorithmes Deep Miner et Map Miner sont illustrées à la Figure 41 et à la Figure 42 respectivement.

En ce qui concerne la carte obtenue, les 22 sous-intentions sont regroupées en 7 groupes d'intentions de haut niveau. Notez que MMM peut découvrir rapidement le début et la fin d'un processus ; ainsi, les intentions de début et de fin sont clairement déterminées sur la carte obtenue. Les probabilités de transition d'une intention à une stratégie sont annotées sur les flèches. Ces valeurs correspondent aux probabilités que les développeurs aient choisi une stratégie à partir d'une intention donnée. Les valeurs sur les boucles indiquent les probabilités que les développeurs aient continué à effectuer les activités liées aux stratégies réflexives.

MMM découvre les stratégies et leurs intentions respectives. En d'autres termes, MMM reconstruit la topologie du modèle de processus Map à partir de traces ; les noms des stratégies et des intentions ne sont pas entièrement générés automatiquement. Néanmoins, il est possible de déduire les noms des stratégies et des intentions à partir de la matrice d'émissions E. En effet, E précise les activités associées à chaque stratégie découverte par le MMM (voir la Figure 39).

Numéro	Nom
S1	**mylyn.tasks.ui.commands.**[OpenTask, Ad- dTaskRepository, ActivateTask, SearchForTask], **my-lyn.context.ui.commands**.[Open.context.dialog, Attach- Context, interest.Increment, interest.Decrement], **my- lyn.monitor.ui, mylyn.team.ui**
S2	**core**.[jobs, net, filesystem, resource, runtime, variables, content- type, databinding.observable], **equinox.p2.ui.sdk.install**
S3	**mylyn.context.ui.comman**ds.[Open.context.dialog, AttachContext, interest.Increment, inter-est.Decrement],**team.cvs.ui**.[branch, replace, GenerateDiff, ShowHistory, Add, Tag,merg, compareWithTag], **jsch.core, mylyn**.[monitor.ui, team.ui, commons.ui, bugzilla.ui]
S4	**pde.ui.**EquinoxLaunchShortcut.run, **equinox.p2**.ui.sdk.update, **equinox**.[ds, simpleconfigurator.manipulator, frameworkad- min, app, common, directorywatcher, engine, core, meta- data.repository, garbagecollector, ui.sdk.scheduler, repository, preferences, exemplarysetup, registry, updatechecker]
S5	**core**.[databinding.observable, core.net, core.filesystem, core.resource, core.runtime, core.variables, core.contenttype], **debug.ui.commands**.[RunLast, Debuglast, eof, StepOver, TerminateAndRelaunch, execute, AddBreakPoint, Tog- glebreakPoint], **jdt.debug**.ui. [commands.Execute, commands.Inspect], **jdt.junit**.[junitShortcut.rerunLast, go- toTest,junitShortcut.debug], **ltk.ui.refactoring.commands**. [deleteResources, renameresources, moveResources], com- pare.selectPreviousChange
S6	**ui.edit**.[delete, paste, copy undo, text.goto.lineEnd, text.contentAssist.proposals, text.goto.wordNext], **ui.file.save**
S7	cdt.ui.editor, **jdt.junit**.[junitShortcut.rerunLast, gotoTest, ju- nitShortcut.debug], **team.cvs.ui**.[CompareWithRevision, Com- pareWithLatestRevisionCommand, CompareWithWorkingCopy-Command],**ui.edit**.[delete, paste, copy undo, text.goto.lineEnd, text.contentAssist.proposals, text.goto.wordNext]
S8	**team.ui**.[synchronizeLast, TeamSynchroniz- ingPerspective, synchronizeAll, applyPatch], **ltk.core.refactoring.refactor**.[create.refactoring.script, show.refactoring.history]
S9	**mylyn.monitor.ui, mylyn.context.ui, mylyn.commons.ui**, **team.cvs.ui.**[commitAll, Commit, CompareWithRemote, Sync]
S10	**mylyn.monitor.ui, mylyn.bugzilla.core, mylyn.bugzilla.ui, team.cvs.ui**.[commitAll, Commit, CompareWithRemote, Sync]

Figure 39. Index des stratégies et activités reliées dans le cas Eclipse

Par conséquent, sur la base des noms des activités regroupées dans une stratégie, il est alors possible d'inférer manuellement les noms des stratégies par une analyse sémantique de leurs propriétés et interrelations. De la même façon, puisque les stratégies mènent à des intentions, les noms des intentions peuvent être inférés en analysant les stratégies menant à chaque intention. Par exemple, les principales activités regroupées dans la stratégie S5 sont « refactoring.commands », « jdt.junit », « debug.ui.commands », etc. De ces activités, on peut déduire que les développeurs voulaient déboguer, refacturer et tester le code ; Ainsi, le nom inféré pour cette stratégie est *par refactoring, testing et débogage*. De plus, les principales activités de la stratégie S6 sont « delete », « paste », « copy », « undo », etc. Cela signifie que les développeurs voulaient modifier un code ou un fichier; le nom inféré pour cette stratégie est *par modification de fichier*. Puisque les deux stratégies mènent à une intention, il est possible de déduire que les développeurs qui ont exécuté S5 et S6 avaient l'intention de corriger un bug. En appliquant cette procédure, les noms de toutes les stratégies (indiqués sur les flèches) et les intentions sont inférés. L

a Figure 40 représente les thèmes obtenus par E et les noms de stratégies inférées. Les noms inférés des intentions sont *Démarrer*, *Lancer le développement*, *Gérer les tâches*, *Corriger un bug*, *Améliorer le code*, *Valider le code*. Ce protocole de nommage reste à être entièrement automatisé en construisant des ontologies métier plus pertinentes.

Numéro	Thème obtenus par E	Noms des intentions inférées
S1	OpenTask, AddTaskRepository, ActivateTask, Search- ForTask, Open.context.dialog, AttachContext, in- terest.Increment, interest.Decrement, mylyn.monitor.ui, mylyn.team.ui	By project tracking and team planning
S2	jobs, net, filesystem, resource, runtime, vari- ables, contenttype, databinding.observable, equinox.p2.ui.sdk.install	By regular programming activities
S3	Open.context.dialog, AttachContext, interest.Increment, interest.Decrement, branch, replace, GenerateD- iff, ShowHistory, Add, Tag,merg, compareWithTag, jsch.core, monitor.ui, team.ui, commons.ui, bugzilla.ui	By code/task sharing
S4	EquinoxLaunchShortcut.run, equinox.p2.ui.sdk.update, simpleconfigurator.manipulator, frameworkadmin, app, common, directorywatcher, engine, core, meta- data.repository, garbagecollector, ui.sdk.scheduler, repository, preferences, exemplarysetup, registry, updat- echecker	By OSGI- based design
S5	databinding.observable, core.net, core.filesystem, core.resource, core.runtime, core.variables, core.contenttype, RunLast, Debuglast, eof, StepOver, TerminateAndRelaunch, execute, AddBreakPoint, TogglebreakPoint, debug.commands.Execute, com- mands.Inspect, junitShortcut.rerunLast, gotoTest, ltk.ui.refactoring.commands. deleteResources, renamere- sources, moveResources, compare.selectPreviousChange	By refactoring, testing and debugging
S6	delete, paste, copy, undo, text.goto.lineEnd, text.contentAssist.proposals, text.goto.wordNext, ui.file.save	By file modifica- tion
S7	cdt.ui.editor, junitShortcut.rerunLast, gotoTest, delete, paste, copy, undo, text.goto.lineEnd, text.contentAssist.proposals, junitShortcut.debug, CompareWithWorkingCopyCommand, team.cvs.ui.CompareWithRevision, CompareWith- LatestRevisionCommand	By review- ing and testing
S8	synchronizeLast, TeamSynchronizingPerspective, syn- chronizeAll, applyPatch, create.refactoring.script, show.refactoring.history	By patch applying
S9	mylyn.monitor.ui, mylyn.context.ui, mylyn.commons.ui, team.cvs.ui.commitAll, Commit, CompareWithRemote, Sync	By CVS committing
S10	mylyn.monitor.ui, mylyn.bugzilla.core, my- lyn.bugzilla.ui, team.cvs.ui.commitAll, Com- mit, CompareWithRemote, Sync	Updating issue tracking

Figure 40. Index des stratégies, thèmes et nom des stratégies inférées pour le cas Eclipse

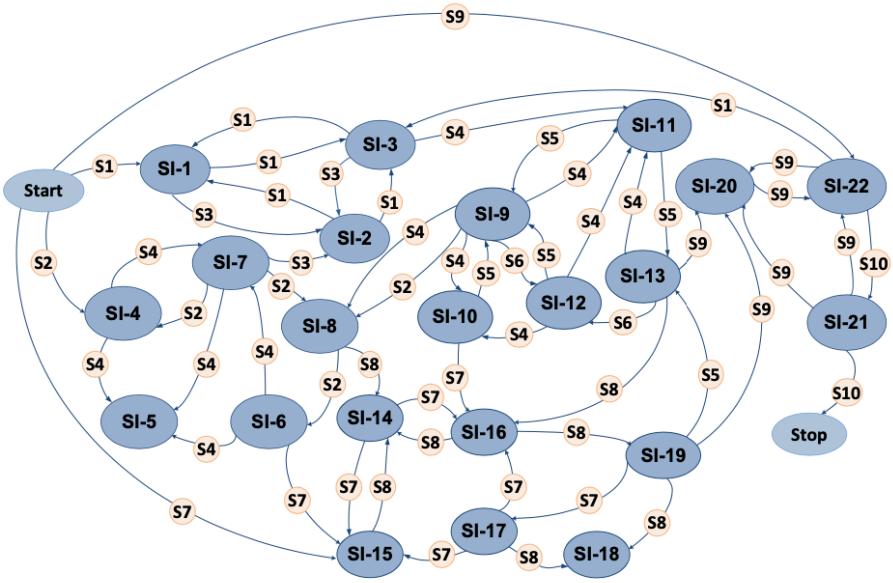


Figure 41. Pseudo carte obtenue par l'algorithme Deep Miner pour le cas Eclipse

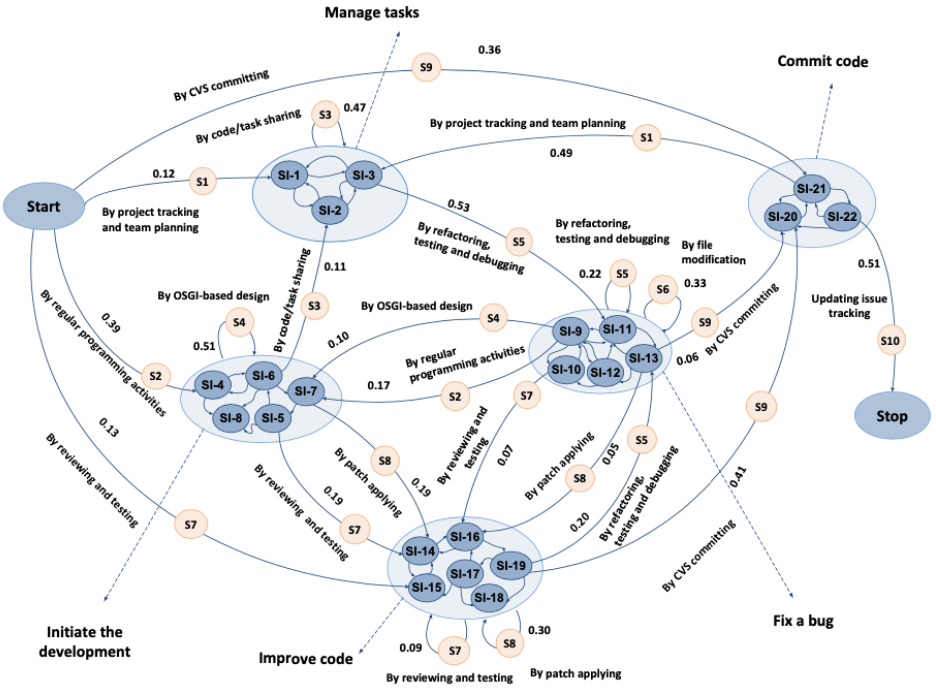


Figure 42. Carte de haut niveau découverte par l'algorithme Map Miner pour le cas Eclipse

Découvrir la carte des développeurs d'Eclipse UDC permet de comprendre les comportements des développeurs pendant le processus de développement. Comme le montre la Figure 42, ils ont choisi différents chemins (séquences de stratégies) avec des probabilités différentes pour réaliser leurs intentions. Un expert peut analyser ces comportements afin de comprendre comment, pourquoi et avec quelles probabilités les développeurs utilisent les différents composants ou plug-ins d'Eclipse, où ils suivent les meilleures pratiques des projets de développement logiciel et où ils s'écartent de ces règles, quels composants ou plug-ins sont plus impliqués que les autres, quels chemins sont plus / moins pris, ou encore où sont les problèmes, etc. La carte peut également être utilisée pour fournir des recommandations aux développeurs afin de choisir le meilleur chemin pour remplir ses intentions. Dans cette étude de cas il n'y a pas de modèle antérieur à comparer avec la carte obtenue mais avons pu établir certaines observations qui donnent des informations intéressantes sur les usages des développeurs (ces observations sont détaillées dans (Khodabandelou 2014b)).

Des interviews ont été faites avec sept experts développeurs sur Eclipse pour valider le résultat obtenu et la carte Eclipse concorde avec les expériences des développeurs, qui l'ont trouvée pertinente dans leurs activités quotidiennes. Les noms des stratégies et les intentions ont généralement été validés par les développeurs, bien que certains d'entre eux aient des suggestions pour les améliorer. Ils auraient besoin d'un outil efficace pour modéliser automatiquement les processus intentionnels et aimeraient avoir un modèle interactif à chaque étape du développement. Le niveau d'abstraction (sous-intentions) devrait être disponible pour chaque intention puisque les développeurs doivent voir quelles sont les composantes des intentions et mieux comprendre la nature profonde d'un processus. Beaucoup de développeurs ont mentionné qu'ils aimeraient voir les interactions et les feedbacks des clients. Par conséquent, il serait intéressant de tracer ces interactions avec les

traces de développement et de générer un modèle de processus Map à partir de ces deux traces.

II.3.3. Définir un modèle intentionnel par adaptation en fonction du comportement des utilisateurs : l'approche MMM

Une fois que l'on a découvert le modèle de processus intentionnel à partir des logs des utilisateurs, il est facile de comparer le modèle prescrit (si l'on en a un) à celui-ci, cela s'appelle le contrôle de conformité (Figure 43). Ce contrôle permet d'identifier les éléments posant problèmes et de modifier le modèle prescrit en fonction de ces informations.

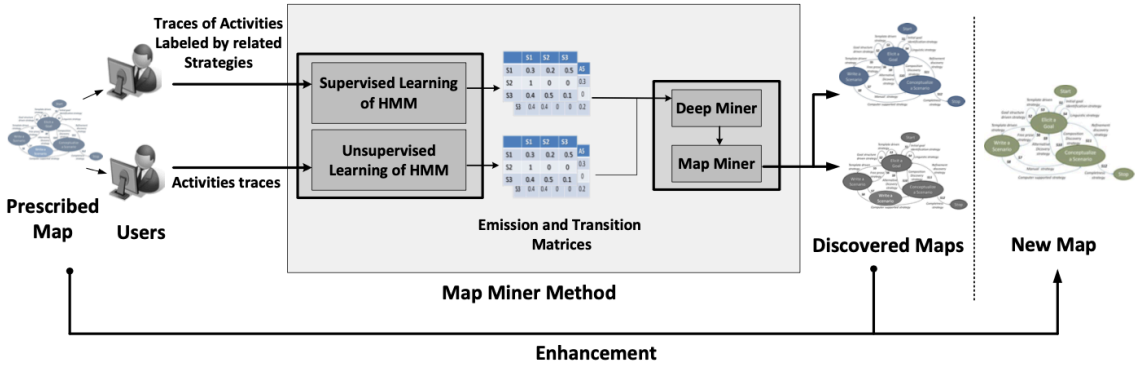


Figure 43. Contrôle de conformité avec MapMinerMethod

Nous montrons ci-dessous deux exemples d'application de modification d'un processus existant suite à la fouille d'intentions faite avec l'approche MMM (section II.3.2), l'une avec apprentissage supervisé (Deneckère 2014b) (Khodabandelou 2013b) (Khodabandelou 2013c) et l'autre avec apprentissage non supervisé (Khodabandelou 2014b) (Khodabandelou 2014a).

Exemple d'application de MMM avec apprentissage non supervisé

Nous avons appliqué MMM (voir la section II.3.2) à un cas d'étude de traces d'utilisation de diagrammes E/R, traces générées par des étudiants de MIAGE lors de l'élaboration de leurs diagrammes. Le modèle prescrit était celui indiqué en Figure 44. Les étudiants pouvaient sélectionner dix stratégies pour remplir quatre intentions, Commencer, Spécifier une entité, Spécifier une association et Stop. Pour choisir une stratégie, les étudiants devaient effectuer l'une des 12 activités prescrites. La Figure 45 donne le nom de toutes les activités et montre en détail le lien entre les stratégies et les activités connexes dans la carte prescrite.

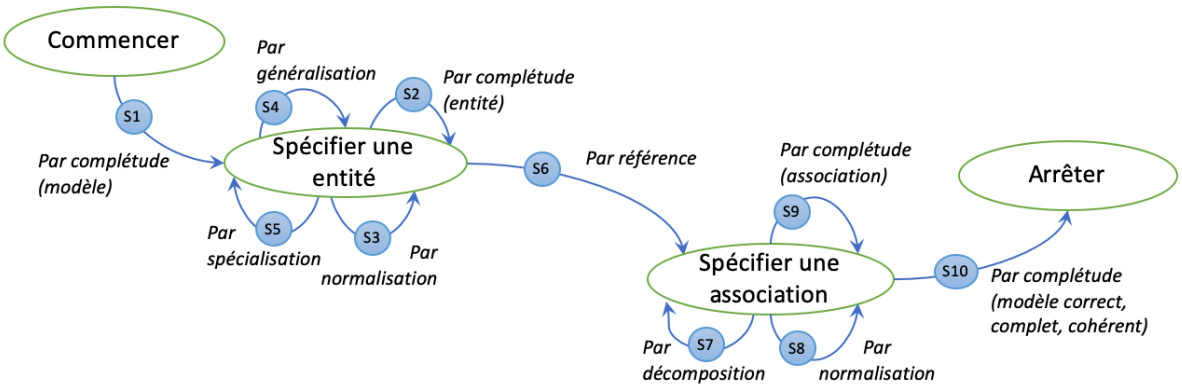


Figure 44. Processus intentionnel de la construction d'un modèle E/R (d'après (Assar 2000))

Section de la carte		Trace des activités associées	Activités
S1	Spécifier une entité par complétude (modèle)	Créer une entité	A1
S2	Spécifier une entité par complétude (entité)	Lier un attribut à une entité	A2
S3	Spécifier une entité par normalisation	Supprimer le lien d'un attribut à une entité Supprimer une entité Définir une clé primaire	A6 A5 A7
S4	Spécifier une entité par généralisation	Créer une entité Créer un lien de généralisation	A1 A3
S5	Spécifier une entité par spécialisation	Créer une entité Créer un lien de spécialisation	A1 A4
S6	Spécifier une association par référence	Supprimer le lien d'un attribut à une entité Créer une entité Créer une association Associer une association à une entité	A8
		Créer une association Associer une association à une entité	A9
S7	Spécifier une association par décomposition	{Créer une association Associer une association à une entité}	A9
S8	Spécifier une association par normalisation	{Supprimer une association Supprimer le lien d'un attribut à une association}	A10
S9	Spécifier une association par complétude (association)	Associer un attribut à une association	A11
S10	Arrêter par complétude (modèle correct, complet, cohérent)	Vérifier le modèle	A12

Figure 45. Sections et traces des activités associées

Premièrement, l'apprentissage supervisé est appliqué sur les traces d'activités pour obtenir les paramètres du modèle. Notez que les traces doivent être compatibles avec l'approche d'apprentissage supervisé. En effet, ils doivent comporter des traces d'activités et des traces correspondantes de stratégies. Comme c'est nous qui avions mis en place l'expérience, nous étions capables de tout enregistrer. Cette procédure est appelée un processus de labellisation. Après modélisation des traces d'activités par HMM et l'application de l'apprentissage supervisé, les paramètres du modèle sont estimés.

L'algorithme Deep Miner utilisant la matrice de transition ainsi calculée, génère une pseudo-carte (Figure 46) avec un seuil ajusté à 0.05. Ce seuil est obtenu par heuristique, c'est une valeur arbitraire qui peut être déterminé par l'expert.

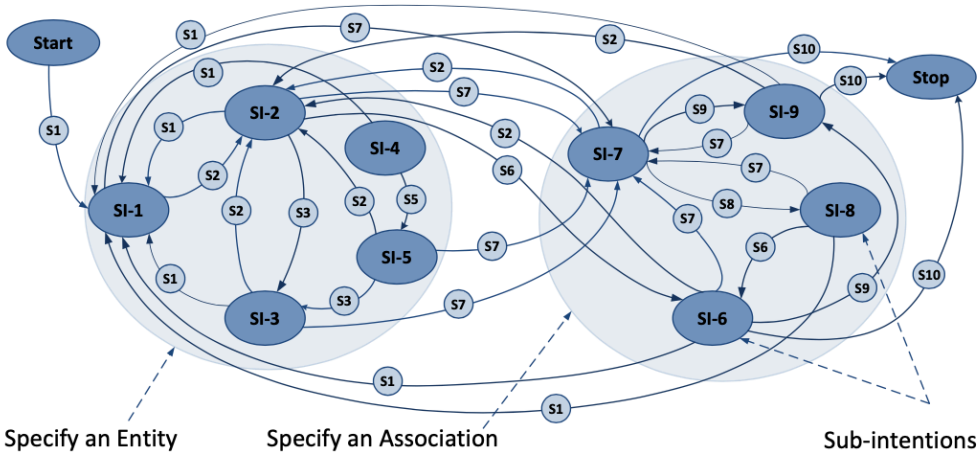


Figure 46. Pseudo carte obtenue par l'algorithme Deep Miner (supervisé)

Dans la Figure 47, les stratégies découvertes par l'algorithme Map Miner qui correspondent aux sections de la carte prescrite sont représentées par des flèches continues. Cependant, MMM détecte également que les étudiants ont dévié de la carte prescrite, puisque certaines stratégies ne sont pas utilisées comme elles ont été prescrites.

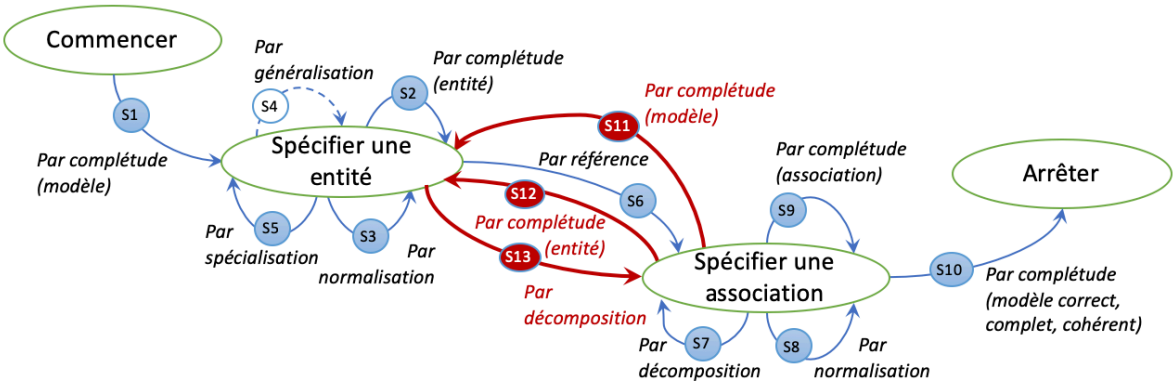


Figure 47. Carte de haut niveau découverte par l'algorithme Map Miner par apprentissage supervisé

En analysant et en comparant la carte découverte avec la carte prescrite, les observations sont les suivantes :

- **Correspondances entre la carte prescrite et la carte découverte.** Les étudiants ont choisi la stratégie S1 pour atteindre l'intention *Spécifier une entité* et ils ont continué à essayer d'atteindre cette intention en choisissant les stratégies S2, S3 et S5. Pour remplir l'intention *Spécifier une association*, ils ont choisi la stratégie S7, S8 et S9. Enfin, ils ont choisi la stratégie S10 pour *Arrêter le processus*.
- **Décalage entre la carte prescrite et la carte découverte.** Les étudiants n'ont jamais utilisé la stratégie de généralisation et la stratégie par décomposition n'est pas choisie correctement En effet, cette stratégie S7 ne doit être sélectionnée que dans la section <Spécifier une association, Spécifier une association, Par décomposition>. Cependant, cette stratégie est également sélectionnée entre les intentions *Spécifier une entité* et *Spécifier une association* (S13).

- **Certaines stratégies soulèvent certaines questions.** La section S9 n'est pas cohérente car l'intention cible diffère complètement de ce qui est induit par la stratégie. Cette section n'est pas représentée dans la Figure 44. La section S11 a été conservé car les utilisateurs peuvent compléter le modèle en créant une nouvelle entité après avoir spécifié une association. Dans ce cas, la situation prend en compte le produit modifié (le diagramme E/R), au contraire de la section S11 qui n'implique aucun produit existant à prendre en compte. Cependant, il n'est pas possible de savoir à partir des traces si les utilisateurs ont bien suivi ces stratégies en fonction de la situation du produit (schéma E/R existant ou non). Avec le même raisonnement, la section S12 permet aux utilisateurs d'ajouter des attributs à une entité de leur diagramme après avoir spécifié une association.

Toutes ces réflexions peuvent amener à modifier le processus prescrit en conséquence, soit en supprimant les stratégies non utilisées ou en ajoutant de nouvelles stratégies à la carte.

Exemple d'application de MMM avec apprentissage supervisé

En appliquant l'apprentissage non supervisé aux mêmes activités, on découvre un modèle de processus Map différent. Nous rappelons que dans ce cas, aucun processus de labellisation des stratégies n'est nécessaire pour exécuter l'algorithme d'apprentissage. Par conséquent, seules les traces d'activités sont utilisées comme entrées de BWA. Cependant, comme mentionné précédemment, l'APB doit être initialisée avec un certain nombre de stratégies possibles. Ici, la méthode heuristique permet de trouver le bon nombre de stratégies. L'APB produit une matrice d'émissions avec plusieurs stratégies identiques lorsque le nombre de stratégies possibles est trop élevé. Par conséquent, 10 est le nombre de stratégies pour le modèle ayant la plus grande probabilité de générer les traces observées. Une fois que le nombre de stratégies est fixé, le BWA peut être exécuté pour apprendre les paramètres du modèle.

L'algorithme Deep Miner génère la pseudo-carte en maximisant la forme et la précision métriques. En effet, la matrice de transition permet de connaître la position de chaque stratégie sur la carte, i.e. quelle est l'intention source et l'intention cible de chaque stratégie. La carte obtenue par l'algorithme Deep Miner avec un seuil de 0,05 est illustrée à la Figure 48.

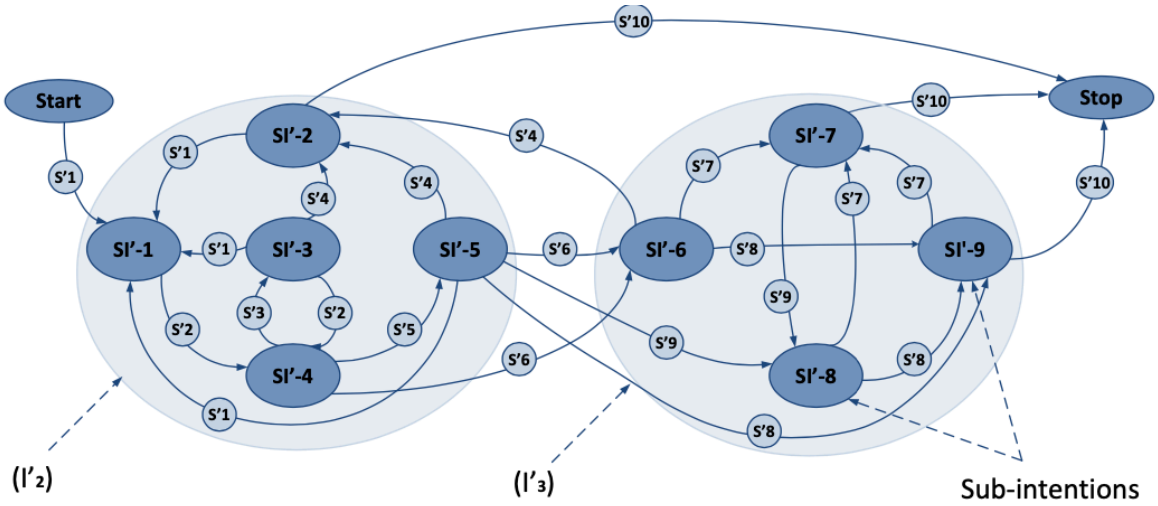


Figure 48. Pseudo carte obtenue par l'algorithme Deep Miner (non supervisé)

Pour obtenir un modèle de processus sous forme de carte, l'algorithme Map Miner regroupe les sous-intentions en quatre intentions générales et un ensemble de stratégies.

Contrairement à l'apprentissage supervisé, puisqu'aucune information préalable sur les stratégies n'est disponible, les noms des stratégies et des intentions ne sont pas connus. Cependant, à partir de la matrice d'émissions E, chaque stratégie découverte par MMM peut être associée à certaines activités. Sur la base des noms des activités, il est alors possible de déduire les principaux thèmes des stratégies à travers une analyse sémantique. De la même façon, les noms des intentions peuvent être inférés en analysant les stratégies menant à chaque intention.

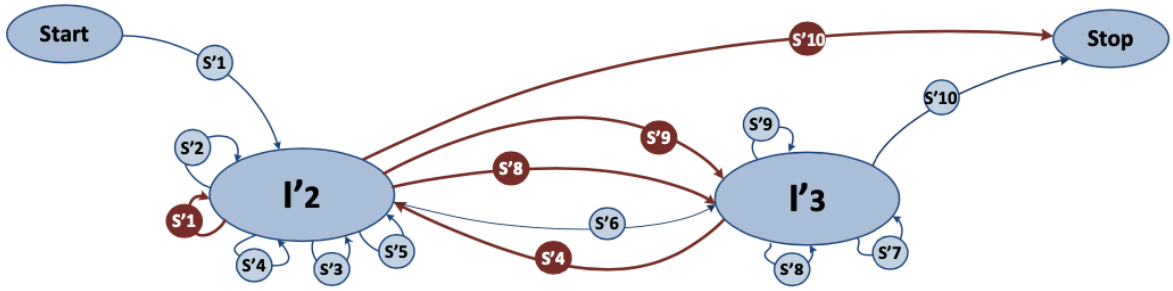


Figure 49. Carte de haut niveau découverte par l'algorithme Map Miner par apprentissage non supervisé

Carte prescrite				Carte découverte		
Intentions	**Index**	**Nom de stratégie**	**Activités**	**Intentions**	**Index**	**Activités**
$I_1 \rightarrow I_2$	S_1	Par complétude (modèle)	a_1	$I'_1 \rightarrow I'_2$	S'_1	a_1 (0,94)
				I'_2	S'_1	a_1 (0,94)
I_2	S_2	Par complétude (entité)	a_2		S'_2	a_2 (0,88), a_1 (0,09)
	S_3	Par spécialisation	a_6, a_5, a_7		S'_3	a_5 (0,1), a_6 (0,63), a_7 (0,28)
	S_4	Par généralisation	a_1, a_3		S'_4	a_1 (0,11), a_2 (0,54), a_8 (0,25)
	S_5	Par normalisation	a_1, a_4		S'_5	a_2 (0,09), a_3 (0,13), a_4 (0,39), a_5 (0,40)
$I_2 \rightarrow I_3$	S_6	Par référence	a_8, a_9	$I'_2 \rightarrow I'_3$	S'_6	a_1 (0,15), a_2 (0,79)
					S'_8	a_1 (0,09), a_2 (0,81), a_9 (0,08)
					S'_9	a_1 (0,37), a_9 (0,19), a_{11} (0,34)
				$I'_3 \rightarrow I'_2$	S'_4	a_1 (0,11), a_2 (0,54), a_8 (0,25)
I_3	S_7	Par décomposition	a_9	I'_3	S'_7	a_9 (0,83), a_1 (0,05), a_2 (0,05)
	S_8	Par normalisation	a_{10}		S'_8	a_1 (0,09), a_2 (0,81), a_9 (0,08)
	S_9	Par complétude (association)	a_{11}		S'_9	a_1 (0,37), a_9 (0,19), a_{11} (0,34)
$I_3 \rightarrow I_4$	S_{10}	Par complétude (final)	a_{12}	$I'_3 \rightarrow I'_4$	S'_{10}	a_{12} (0,87), a_9 (0,08)
				$I'_2 \rightarrow I'_4$	S'_{10}	a_{12} (0,87), a_9 (0,08)

Figure 50. Stratégies de la carte prescrite (à gauche) et de la carte découverte (à droite)

En analysant et en comparant la Carte prescrite et la pseudo-Carte découverte par un apprentissage non supervisé, nous pouvons faire les observations suivantes.

- **Correspondances**. Il y a un certain nombre de correspondances entre les nouveaux groupes de stratégies et les groupes de stratégies dans la carte prescrite, comme S_1 et S'_1, qui à la fois commencent le processus et sont faites de l'activité a_1. La stratégie S_{10} et S'_{10} sont également similaires dans les deux cartes. Elles mettent fin au processus et sont composées de a_{12}. Une autre correspondance est entre S_2 et S'_2, qui sont faites de l'activité a_2. Ceci était attendu puisque les étudiants avaient pour consigne de suivre la carte prescrite.

- **Différences**. Les quelques inadéquations entre les stratégies prescrites et celles qui sont découvertes indiquent que certaines stratégies prescrites n'ont pas été suivies comme prévu par les étudiants. Par exemple, dans la carte prescrite, l'activité a_2 n'est présente que dans la stratégie S_2 pour l'intention I_2 alors que dans la carte découverte, elle est présente dans de nombreuses stratégies : S'_1, S'_4, S'_6, S'_7, et S'_8 pour les intentions I'_2, et I'_3. Le même phénomène est vrai pour l'activité a_1. Cela signifie que ces activités particulières sont utilisées par les apprenants pendant tout le processus et pas seulement pour une seule intention. Ces connaissances peuvent être utilisées pour (i) améliorer le logiciel utilisé par les étudiants pour rendre le modèle prescrit plus facile à suivre, (ii) modifier le modèle prescrit en tenant compte du processus que les étudiants ont réellement suivi.
- **Activités et intentions**. La plupart des activités sont liées aux mêmes intentions dans les cartes prescrites et découvertes. Ceci est vrai pour les activités a_3, a_4, a_6 et a_7 qui ne sont liées qu'à I_2 et I_4. Ceci est également vrai pour les activités a_9 et a_{11} qui sont liées à I_3 et I'_3. Cependant, a_{10} est très rarement utilisé par les étudiants et n'apparaît pas dans les stratégies découvertes.

III. Utiliser un modèle intentionnel

L'utilisation d'un modèle intentionnel peut se faire de différentes façons :

- Par guidage en fonction du contexte (ingénierie des processus)
- Par guidage en fonction du comportement des utilisateurs (fouille de processus)

La figure suivante montre les approches que nous avons proposé dans cet objectif : l'approche MIG (section III.1.1), l'approche de la théorie des graphes (section III.1.2), l'approche Map-TBS (section III.2.1) et l'approche FlexPAISSeer (section III.2.2).

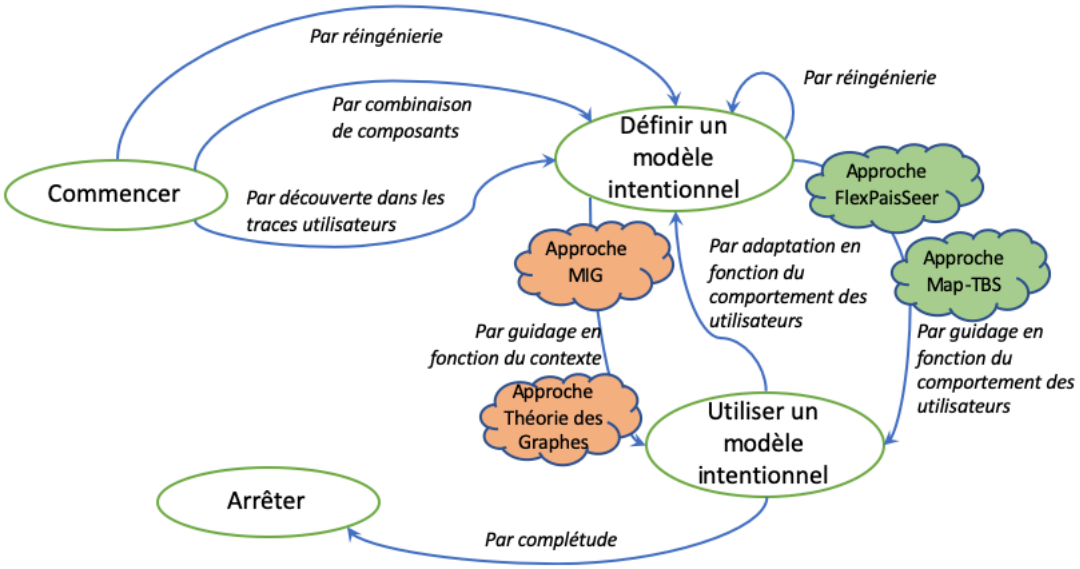


Figure 51. Approches proposées dans le cadre de l'utilisation des processus intentionnels.

III.1. Utiliser un modèle intentionnel par guidage en fonction du contexte

Lorsque les sections de la carte sont identifiées, elles sont associées à un descripteur qui permet de connaître le contexte de réutilisation (quelle situation de départ doit être en cours, quels critères sont à prendre en compte, etc.) et un ensemble de directives (la directive de sélection d'intention et la directive de sélection de stratégie permettent de donner des informations de contexte pour aider le choix). Il est alors possible de comparer la situation et le contexte en cours d'exécution du processus pour pouvoir choisir la bonne section (Figure 52).

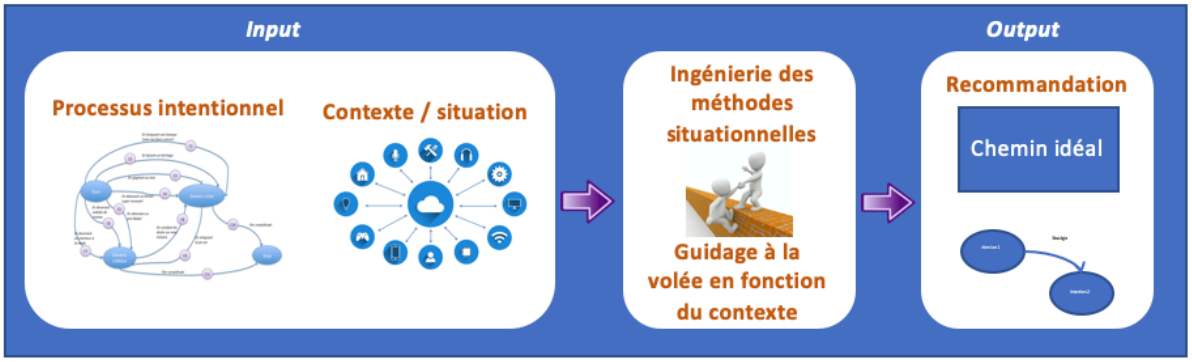


Figure 52. Entrées-Sorties de l'utilisation d'un modèle intentionnel par guidage en fonction du contexte

Nous avons proposé deux approches qui permettent, sur la base d'indicateurs de contexte, d'effectuer les sélections de sections à bon escient.

III.1.1. Utiliser un modèle intentionnel par guidage en fonction du contexte : l'approche MIG (Map Indicator-based Guidance)

Nous avons mentionné que les processus téléologiques contiennent des étapes de prise de décision. Ces décisions peuvent porter sur les modèles à utiliser, actions à effectuer, scénarios possibles, etc. Ce sont notamment les décisions prises au cours de l'exécution des processus téléologiques qui permettent de les adapter au contexte et, par conséquent, d'assurer leur variabilité. Chaque fois qu'un ingénieur doit choisir entre plusieurs possibilités pour continuer son processus, il est confronté à un problème de prise de décision.

Un problème de prise de décision est défini par la présence d'au moins deux alternatives. Le concept d'alternative désigne l'objet de la décision et doivent être identifiées clairement. Ces alternatives sont comparées entre elles selon un ou plusieurs critères. Les critères représentent des informations qui permettent d'évaluer et de comparer les alternatives. Ils peuvent être de nature différente : les caractéristiques intrinsèques des objets de décision, avis des parties prenantes, conséquences potentielles des alternatives, etc. Le problème de prise de décision contient trois éléments de base : la problématique, les alternatives et les critères (Roy, 2005).

Dans le cadre de l'approche MIG (Deneckère 2010a) (Deneckère 2010b) (Kornyshova 2010b) (Deneckère 2011c), chaque étape de prise de décision dans les processus téléologiques est considérée comme un problème de prise de décision. Plus précisément, la problématique est le choix, les alternatives sont des variantes potentiellement disponibles dans un processus sur chacun des points de variation et les critères sont des indicateurs de contexte. L'évaluation de ce problème, et sa résolution, peuvent être envisagées de plusieurs manières. Par exemple, la méthode I* préconise l'utilisation de la notion d'exigences non fonctionnelles en utilisant les 'soft goals' identifiés comme des critères d'évaluation. Les alternatives contribuent alors à différents degrés de satisfaction de ces buts (van Lamswerde, 2009).

Dans l'approche MIG, pour résoudre le problème de prise de décision de notre ingénieur, nous proposons d'utiliser en priorité le contexte du projet. Ce contexte est composé d'un ensemble d'indicateurs permettant de le caractériser (Figure 53).

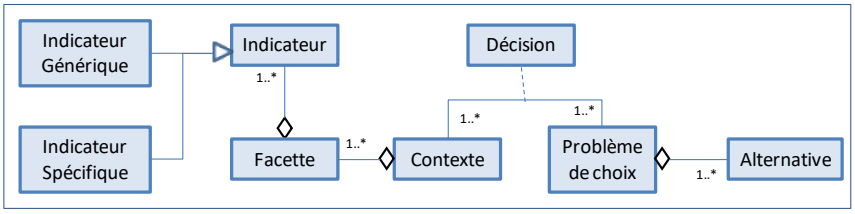


Figure 53. **Sensibilité au contexte des processus téléologiques**

Nous avons défini les indicateurs couvrant les aspects essentiels de l'ingénierie des projets de SI. Nous distinguons ici deux types d'indicateurs : génériques et spécifiques. Les premiers sont communs à la plupart des méta-modèles de processus alors que les autres varient d'un méta-modèle de processus à l'autre.

Indicateurs génériques. Nous avons établi une liste d'indicateurs génériques en nous basant sur le travail de (Kornyshova 2007b) qui propose une typologie d'indicateurs de projet dans le but de guider la recherche et la priorisation de composants de méthodes. Cette proposition est basée sur les travaux de (Mirbel 2006) (Van Slooten 1996) et (Kornyshova 2007b). Elle inclue quatre facettes : la facette organisationnelle, la facette humaine, la facette du domaine d'application et la facette de la stratégie de développement.

Indicateur	Domaine de valeur
Implication de l'équipe de management	{Faible, normal, élevé}
Importance	{Faible, normal, élevé}
Impact	{Faible, normal, élevé}
Urgence	{Faible, normal, élevé}
Limitation des ressources	{Faible, normal, élevé}
Nature des ressources limitées	{ressources financières, ressources temporelles, ressources informatiques}
Taille	{Faible, normal, élevé}
Coût	{Faible, normal, élevé}
Niveau d'innovation	{Faible, normal, élevé}
Nature de l'innovation	{Innovation métier, innovation technologique}

Figure 54. Valeurs de la facette organisationnelle

La facette organisationnelle (Figure 54) met en valeur les aspects organisationnels du développement de projets de système d'information. Par exemple, l'indicateur Implication de l'équipe de management représente l'investissement de l'équipe de direction dans le projet. Les valeurs possibles pour cet indicateur sont 'Faible', 'Normale' et 'Élevée' (i.e. une valeur élevée signifie un grand investissement).

La facette humaine (Figure 55) décrit les indicateurs des personnes impliquées dans le développement du projet. Par exemple, l'indicateur du type de participation des utilisateurs représente les différentes manières qu'ont les utilisateurs de participer au projet, ses valeurs pouvant être 'Réel' ou 'Virtuel'.

Indicateur	**Domaine de valeur**
Résistance et conflit	{Faible, normal, élevé}
Degré d'expertise	{Faible, normal, élevé}
Rôle de l'expert	{Testeur, Développeur, Concepteur, Analyste}
Clarté et stabilité	{Faible, normal, élevé}
Type de participation des utilisateurs	{Réel, Virtuel}
Nombre d'utilisateurs	Nombre

Figure 55. Valeurs de la facette humaine

La facette du *domaine d'application* (Figure 56) inclut les indicateurs caractérisant le domaine du projet de système d'information. Par exemple, l'indicateur *Type de l'application* permet de spécifier la structure organisationnelle du projet. Il peut prendre les valeurs 'Application Intra-organisation', 'Application Inter-organisation' ou encore 'Application Organisation-Client'.

Indicateur	**Domaine de valeur**
Degré de formalisation	{Faible, normal, élevé}
Degré de relations	{Faible, normal, élevé}
Degré de dépendances	{Faible, normal, élevé}
Complexité	{Faible, normal, élevé}
Type de l'application	{Application intra-organisation, Application inter-organisation, Application organisation-client}
Technologie de l'application	{l'application à développer inclut une base de données, l'application à développer est distribuée, l'application à développer inclut une interface utilisateur graphique}
Granularité du projet	{système unique, sous-projets orientés-système, sous-projets orientés-processus, sous-projets hybrides}
Répétition	{Faible, normal, élevé}
Variabilité	{Faible, normal, élevé}
Artéfacts variables	{organisationnel, humain, domaine d'application, stratégie de développement}

Figure 56. Tableau 3. **Valeurs de la facette du domaine d'application**

La facette de la *stratégie de développement* (Figure 57) permet de spécifier différentes informations. Par exemple, l'indicateur représentant l'*origine du système* montre l'origine des éléments réutilisés pouvant être soit du code, soit du domaine fonctionnel, soit encore des interfaces.

Indicateur	**Domaine de valeur**
Origine du système	{réutilisation du code, réutilisation de domaine fonctionnel, réutilisation d'interface}
Organisation de projet	{standard, adaptée}
Stratégie de développement	{d'outsourcing, itérative, de prototypage, orientée phases, orientée composants}
Stratégie de réalisation	{en une fois, de façon incrémentale, de façon concurrentielle, avec overlapping}
Stratégie de livraison	{en une fois, de façon incrémentale, de façon évolutive}
Traçabilité	{Faible, forte}

Nombre d'objectifs	{objectif unique, multiples objectifs}

Figure 57. Valeurs de la facette de la stratégie de développement

Indicateurs spécifiques. Leur identification est basée sur la description de la méthode. L'ingénieur de méthodes les définit en analysant différents aspects organisés en quatre facettes : intentionnelle, satisfactionnelle, décisionnelle et interne, comme dans (Harmsen, 1997).

- La facette intentionnelle concerne les intentions de la méthode.
- La facette satisfactionnelle indique le degré de satisfaction que l'ingénieur a au sujet des résultats de l'application de la méthode.
- La facette décisionnelle représente les processus de prise de décision de la méthode.
- La facette interne concerne les critères connus associés à une gestion de projet spécifique.

Selon le méta-modèle de processus utilisé dans la méthode, les indicateurs qui seront définis dans chacune de ces quatre facettes pourront être différents puisqu'ils seront basés sur les concepts et les règles de ce méta-modèle de processus spécifique.

Typologie des indicateurs. Les indicateurs définis ci-avant peuvent être représentés au sein d'une même typologie. La Figure 58 montre la typologie des indicateurs proposée.

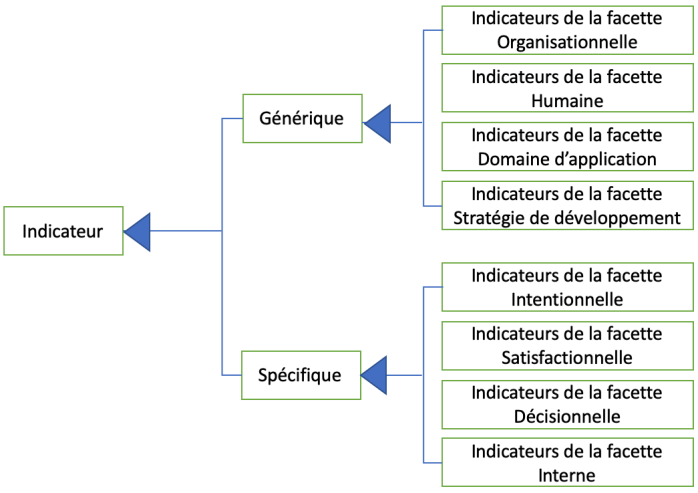


Figure 58. Typologie des indicateurs

Cette typologie indique les principaux indicateurs qui peuvent être définis dans une situation donnée. Elle peut bien entendu être complétée si de nouveaux indicateurs apparaissent, notamment ceux qui découleraient d'une étude plus approfondie des besoins non fonctionnels. Il pourrait également être envisageable de définir les différents liens pouvant exister entre les différents indicateurs ou les différentes facettes.

Les indicateurs ont des valeurs assez informelles mais qui n'empêchent pas l'utilisation de méthodes d'aide à la décision multicritères. Ces valeurs peuvent par exemple être remplacées par des valeurs numériques (les valeurs 'Faible', 'Normal', 'Élevé' peuvent être remplacées par les valeurs '1', '2' et '3'). Une fois quantifiées, les valeurs des différents indicateurs peuvent être agrégées en une valeur générale en utilisant certaines techniques (Roy 1996) (Keeney 1993) (Keeney 1999) pour pouvoir être utilisées comme un critère unique sur lequel

différentes fonctions pourront être utilisées (somme, maximum, minimum, moyenne, somme pondérée…).

Processus d'utilisation

Le processus d'utilisation de cette typologie est appelé IG ('Indicator-based Guidance' ou *Guidage basé sur les Indicateurs*) et est illustré de manière simplifiée dans la Figure 59.

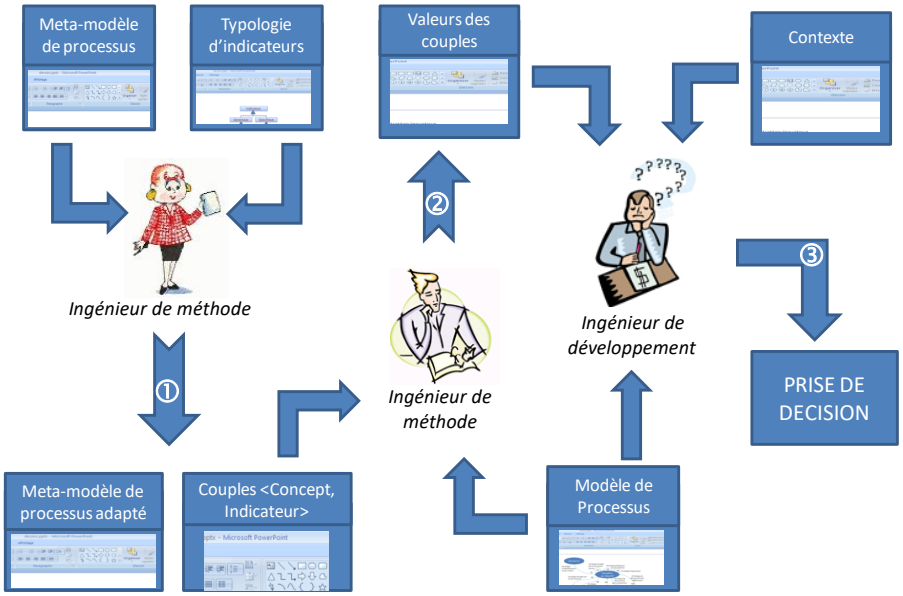


Figure 59. Processus IG

Le processus IG s'effectue à deux niveaux différents : au niveau modèle et au niveau instance. Le premier niveau fait l'objet de la phase préparatoire où l'ingénieur de méthodes va intégrer le concept d'indicateurs dans le méta-modèle de processus que l'ingénieur de développement utilise (niveau modèle – flèche ① à la Figure 59). Le deuxième niveau correspond aux étapes d'utilisation de la typologie d'indicateurs, tout d'abord pour donner des valeurs aux indicateurs (niveau modèle – flèche ② à la Figure 59) avant la phase d'utilisation proprement dite où l'ingénieur de développement va se servir des valeurs d'indicateurs et de la valeur de son contexte de projet pour prendre ses décisions (niveau instance – flèche ③ à la Figure 59).

Processus IG de l'ingénieur de méthodes

L'ingénieur de méthodes est l'acteur qui va utiliser le processus IG au niveau modèle. C'est à cette étape que le méta-modèle de processus utilisé va être étudié. Lorsque cette partie du processus sera terminée, (1) le concept d'indicateur aura été intégré dans le méta-modèle et (2) les concepts impactés par les indicateurs auront été déterminés.

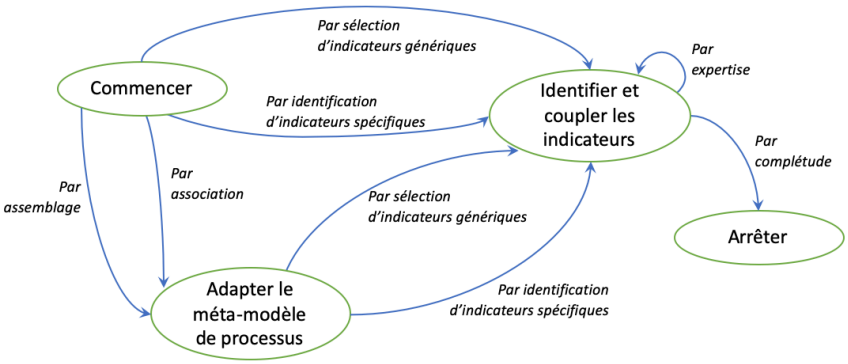


Figure 60. Processus IG de l'ingénieur de méthodes

Au niveau du méta-modèle de processus, l'ingénieur de méthodes doit tout d'abord y intégrer le concept général d'indicateur pour déterminer quels sont les concepts pour lesquels le fait de rattacher des indicateurs de contexte a un sens. Cette intégration se fait selon le principe de l'assemblage de composants de méthode défini par (Ralyté 2001b). Cette approche distingue deux possibilités d'intégration : si le concept d'indicateur est complètement absent du méta-modèle de processus choisi, alors l'intégration de ce concept se fera grâce à une simple association avec les concepts déjà présents (stratégie *Par association*), cependant, si la sémantique des concepts du méta-modèle de processus se chevauche avec celle du concept d'indicateur que l'on veut intégrer, il faut alors modifier le méta-modèle de processus pour pouvoir effectuer l'intégration en conservant la cohérence du méta-modèle (stratégie *Par assemblage*). Il est également possible que le concept d'indicateur, avec la même sémantique que celui qui nous intéresse, soit déjà intégré dans le méta-modèle de processus, dans ce cas, cette étape d'intégration ne sera pas utile et l'ingénieur de méthodes pourra directement passer à l'utilisation de la typologie d'indicateurs.

Il est donc ensuite nécessaire d'identifier quels sont les différents indicateurs génériques s'appliquant à chacun des concepts du méta-modèle de processus. Cette étape se réalise en utilisant la stratégie *par sélection d'indicateurs génériques*. C'est en effet en étudiant le méta-modèle de processus que l'ingénieur de méthodes va pouvoir identifier les différents couples (indicateur générique, concept du méta-modèle) qui seront utiles à l'ingénieur de développement.

La stratégie *par identification d'indicateurs spécifiques* permettra à l'ingénieur de méthodes de déterminer quels sont les indicateurs spécifiques au méta-modèle de processus qu'il peut ajouter dans son intégration. De même que pour la stratégie précédente, il lui faudra former des couples (indicateur spécifique, concept du méta-modèle) pour garantir la cohérence de l'utilisation des indicateurs.

L'ingénieur de méthodes a également la possibilité d'effectuer des ajustements sur les indicateurs utilisés pendant leur intégration. La stratégie *par expertise* lui permettra d'ajouter ou de supprimer un indicateur ou d'en modifier un déjà sélectionné.

Processus IG de l'ingénieur de développement

L'ingénieur de développement est l'acteur qui va utiliser le processus pour prendre des décisions. Il va se servir des modifications effectuées par l'ingénieur de méthodes au niveau modèle pour utiliser au mieux les indicateurs et faire ses choix à bon escient. Pour chacune des décisions que l'ingénieur aura à prendre, le processus IG (1) lui permettra d'évaluer les valeurs des indicateurs pertinents et (2) lui proposera différentes manières de résoudre son problème de prise de décision à l'aide de ces valeurs d'indicateurs. Ce processus est illustré à la figure 5.

Au niveau instance, l'ingénieur de développement doit tout d'abord évaluer la situation. La stratégie *Par expertise du méta-modèle de processus* lui permettra d'affecter une valeur à chaque couple (indicateur, concept) défini par l'ingénieur de méthodes. Cette étape n'est à effectuer qu'une seule fois par méta-modèle de processus. En effet, si le même méta-modèle est utilisé lors d'une autre réalisation du processus IG, il suffira à l'ingénieur de développement de reprendre les valeurs définies lors d'une précédente évaluation. La stratégie *Par expertise des caractéristiques du projet*, quant à elle, permettra à l'ingénieur de développement de définir certaines caractéristiques du projet concerné par son étude.

Ensuite, l'ingénieur de développement peut utiliser ces valeurs de caractérisation du contexte et des indicateurs pour prendre des décisions. L'utilisation de la stratégie *par application de méthodes de choix* lui permet d'utiliser des techniques de décision mono ou multicritères dans ce sens.

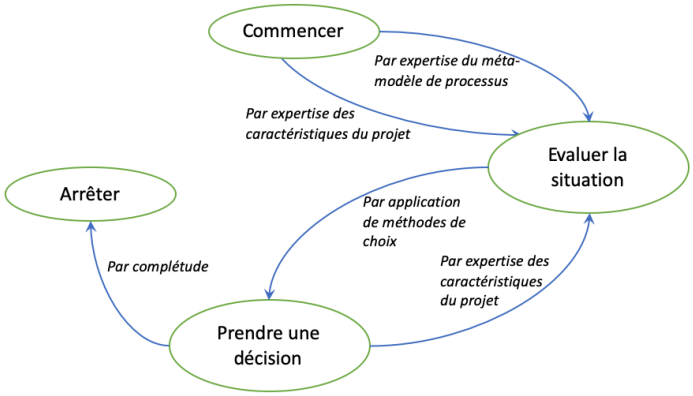


Figure 61. Processus IG de l'ingénieur de développement

Chaque décision prise peut ensuite être prise en compte par l'ingénieur pour effectuer une réévaluation de la situation. La stratégie *Par expertise des caractéristiques du projet* permet ici à l'ingénieur de développement de réévaluer le contexte de son projet avant la prochaine prise de décision.

Ce processus d'utilisation s'arrête lorsque le processus de développement ne comporte plus aucun point de prise de décision.

III.1.2. Utiliser un modèle intentionnel par guidage en fonction du contexte : l'approche de la Théorie des Graphes

Certains travaux ont été réalisés pour combiner le modèle MAP avec un autre type de modélisation, afin d'améliorer l'utilisation pratique des cartes. Par exemple, (Prakash 2006) propose un moyen de transformer les exigences représentées dans une carte en un diagramme de flux de données. Par la suite, la capacité de conception de construction du DFD est disponible pour la mise en œuvre du système. Ils concluent que, même si la notation est à peu près la même, les deux diagrammes sont suffisamment différents les uns des autres car ils abordent des points de vue différents d'un système. Dans deux travaux similaires (Soffer 2005) (Kushnareva 2015), les auteurs combinent la modélisation intentionnelle des cartes avec la modélisation formelle des diagrammes d'états. Cela les a amenés à fournir une formalisation des cartes basées sur les états qui permet son analyse et sa vérification. Malgré ces tentatives, le modèle MAP manque de travaux sur le guidage automatique des processus. Ceci est principalement dû aux difficultés de guidage de la carte au niveau opérationnel afin d'exécuter des processus en lien avec le niveau intentionnel.

La théorie des graphes offre beaucoup de techniques. Par exemple, le problème du plus court chemin permet de trouver un chemin entre deux sommets de sorte que la somme des poids de ses arcs constituants est minimisée. Sur une carte valuée, il peut être utile de trouver un chemin à travers la carte qui satisfera certaines exigences de l'ingénieur d'application (les délais, les coûts, etc.). L'exécution de la carte est alors plus flexible car l'ingénieur a la possibilité de modifier les poids des sections de la carte en fonction du projet en cours (Figure 62). Ce travail est expliqué en détail dans (Deneckère 2009).

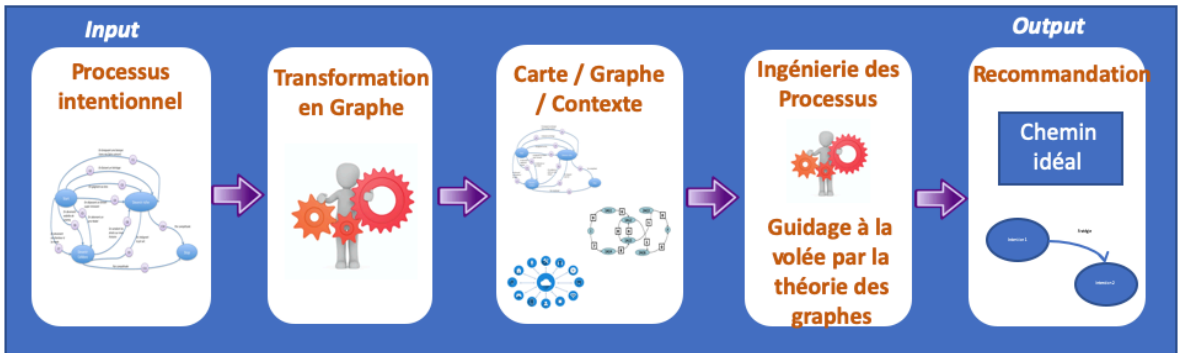


Figure 62. Entrées-Sorties de l'utilisation d'un modèle intentionnel par guidage en utilisant la théorie des graphes

En outre, il y a confusion entre les cartes et les graphes. Bien sûr, les cartes sont visuellement construites sous forme de graphes, mais leur utilisation est complètement différente. Les personnes qui n'ont jamais étudié le modèle MAP ne comprennent pas ces différences, qui sont sémantiquement basées sur les deux niveaux différents utilisés : le niveau intentionnel du modèle MAP et le niveau opérationnel des graphes. Le travail présenté ici montre clairement la différence entre les deux et tente d'utiliser le meilleur de chaque monde pour proposer un guidage pertinent dans les modèles intentionnels.

La théorie des graphes étudie des problèmes tels que la façon de visiter certains endroits uniquement une fois sur une chemin (Euler 1741). En mathématiques et en informatique, la

théorie des graphes est l'étude des graphes, c'est à dire les structures mathématiques utilisées pour modéliser les relations par paires entre les objets d'une certaine collection.

Les graphes sont représentés graphiquement en dessinant un point pour chaque sommet, et en dessinant un arc entre deux sommets s'ils sont reliés. Si le graphe est dirigé, la direction est indiquée en traçant une flèche. Un graphe peut être considéré comme G=(V,E), où V et E sont des ensembles finis disjoints. Nous appelons V l'ensemble des sommets et E l'ensemble des arcs de G (Berge 1985). Un chemin est une séquence alternée de sommets et d'axes. Un chemin commence par un sommet initial et se termine par un sommet final. Un multigraphe ou un pseudographe est un graphe qui est autorisé à avoir plusieurs arcs (également appelés « arcs parallèles » (Balakrishnan 1997)), c'est-à-dire des arcs qui ont les mêmes nœuds d'extrémité. Ainsi deux sommets peuvent être reliés par plus d'un arc. Les cycles sont permis dans ces graphiques. Un cycle est un chemin qui se termine au sommet où il a commencé. Un graphe valué associe une valeur (poids) à chaque arc du graphe.

En ce qui concerne la carte, nous savons que c'est un système de représentation qui a été élaboré pour représenter un modèle de processus en termes intentionnels. Cependant, il n'y a pas de représentation formelle de ce modèle permettant un moyen facile d'automatiser son orientation. Pour sa part, la théorie des graphes a vu le développement d'algorithmes pour manipuler les graphes qui est d'un intérêt majeur pour l'informatique. L'utilisation de ces algorithmes de graphe peut améliorer le guidage des cartes, d'autant plus si le modèle MAP est modifié pour gérer des valuations. En fait, la navigation sur la carte est améliorée par l'utilisation d'une valuation lorsque l'ingénieur prend des décisions fondées sur des critères qualitatifs. Ces critères deviennent une condition préalable à une meilleure orientation. Notre première étape consiste alors à incrémenter le modèle MAP en intégrant le concept de poids. Notre deuxième étape consiste à utiliser les algorithmes de graphes dans les cartes en proposant une correspondance entre les deux modèles.

Les poids sont de deux types différents, statiques ou dynamiques. Le tableau suivant illustre la topologie des indicateurs.

Indicateurs statiques		
Coût	Échelle de 0 à 10	Indique le cout potentiel impliqué par la réalisation de la section
Temps	Échelle de 0 à 10	Montre le temps que passera l'ingénieur à réaliser la section.
…	…	…
Indicateurs dynamiques		
État de l'intention	Échelle de 0 à 10	Donne une évaluation sur la complétude de la réalisation de l'intention
Réalisation de la directive d'intention	Échelle de 0 à 10	Indique le pourcentage de réalisation de la directive de la section.

Les indicateurs statiques sont des valeurs qui sont évalués à l'avance par l'ingénieur de méthode qui crée la carte. Ce sont des critères habituels de la gestion de projet, qui contient des évaluations du coût, du temps et ainsi de suite (Lewis 2005).

Au contraire, les indicateurs dynamiques sont évalués « à la volée ». Par exemple, une exécution de section peut réaliser complètement une intention, ce qui signifie que l'indicateur d'état de l'intention aura une valeur de 10. D'autre part, il peut aussi le réaliser incomplètement et un poids de seulement 5 sur 10 peut indiquer qu'il est nécessaire d'exécuter un cycle sur l'intention, afin de la réaliser plus complètement.

L'indicateur de réalisation de la directive donne une évaluation de la complétude de la section. Suite à la situation du produit en construction, la section peut devoir être exécutée plusieurs fois afin de réaliser complètement la directive.

Prenons l'exemple suivant. La carte contient une section <Démarrer, Stratégie d'identification initiale, Identifier classe> qui sera réalisée par la directive suivante <(Énoncé du problème), Identifier classe par identification initiale>. L'énoncé du problème est, par exemple, « Le client peut avoir plusieurs commandes ». La réalisation de cette directive mènera d'abord à l'identification de la catégorie de clients. Même si l'état de l'intention sera atteint (comme nous avons identifié une classe, ce qui est notre intention cible), la réalisation de la directive ne sera pas complète car l'énoncé du problème contient une autre classe à identifier : la classe de commande. Par conséquent, l'indicateur de l'état de l'intention sera égal à 10, mais l'indicateur de réalisation de la directive sera de 5 sur 10. Il sera alors nécessaire d'exécuter à nouveau la directive afin d'identifier la deuxième classe.

La transformation d'une carte en graphe est assez facile (voir Figure 63 et Figure 64). Chaque carte correspond à un graphe (A). Chaque section de la carte sera ensuite transformée en sommets du graphique (B) et chaque séquence de section identifiée sera représentée sous forme d'arêtes sur le graphe (C). Enfin, nous pouvons identifier la correspondance entre les indicateurs de la MAP et le poids du graphe (D). Les poids des sections doivent ensuite être évalués en un seul indicateur qui peut être appliqué sur les arcs du graphe. Les valeurs des arcs du graphe sont obtenues à partir des valeurs agrégées des sections : le poids de la section devient la même valeur sur tous les arcs entrant pour une section donnée du graphe. Nous supposons que chaque section a la même valeur indépendamment de la section précédemment réalisée. Le modèle MAP permet d'affiner une section avec une autre carte. Ce niveau d'abstraction est conservé avec les graphes car un nœud peut également être affiné comme un autre graphe.

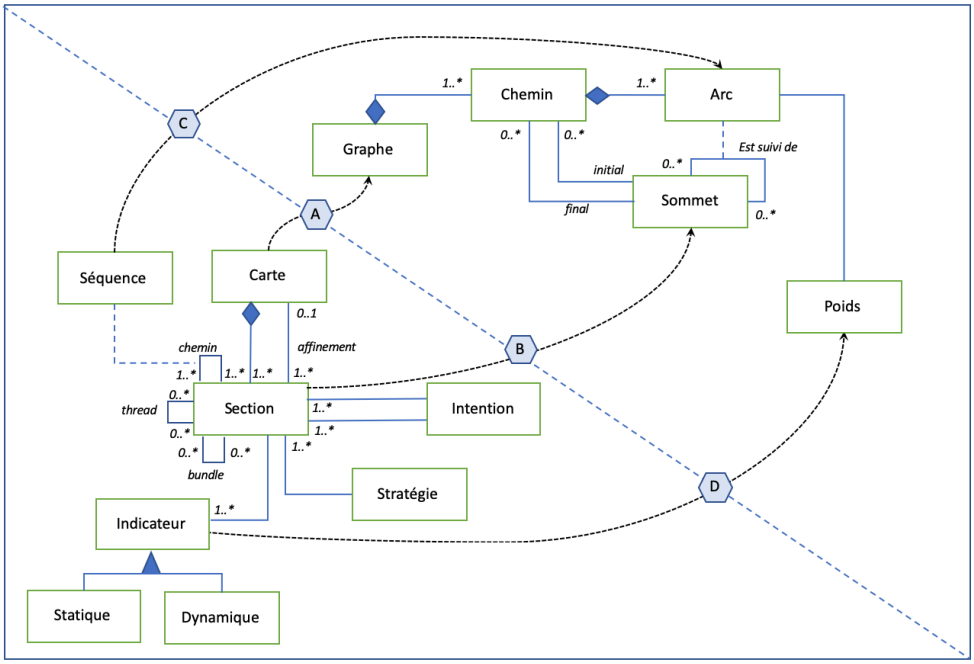


Figure 63. Correspondances entre les concepts de Graphe et les concepts de Carte

Concept de la carte	Concept du graphe	Correspondance
Carte	Graphe	Chaque carte est représentée par un graphe.
Section	Sommet	Le service exécutable dans une carte est la section, qui est représentée comme un sommet dans un graphe.
Séquence	Arc	Le concept permettant la navigation dans une carte est la séquence de sections, qui peut être représentée comme des arcs dans un graphe.
Indicateur	Poids	Les paramètres de guidage appelés indicateurs dans la carte sont les poids dans un graphique.
Chemin	Chemin	L'ensemble des séquences de section (chemin) est également clairement un chemin dans le graphe.
Thread	Ensemble d'arc adjacents	Le thread est la possibilité d'atteindre une intention cible à partir d'une intention source avec plusieurs stratégies. Dans un graphe, la représentation de cet ensemble de sections sera l'ensemble des sommets ayant deux arcs spécifiques, l'un partageant le même sommet de départ et l'autre le même sommet de fin.
Bundle	Ensemble d'arc adjacents	Le bundle est la possibilité d'atteindre une intention cible à partir d'une intention source avec plusieurs stratégies mais avec un OU exclusif ce qui signifie qu'une seule de ces sections peut être utilisée dans la navigation complète. La représentation sur le graphe est la même que pour le thread.

Figure 64. Correspondances entre Graphe et Carte

L'utilisation d'une structure de graphe permet d'utiliser un niveau opérationnel sur notre modèle de processus alors que l'utilisation de la carte est au niveau intentionnel. Ces deux niveaux peuvent être utilisés de manière combinée pour associer leurs propres avantages. Tout d'abord, la théorie des graphes permet l'application d'un certain nombre d'algorithmes spécifiques (chemin complet, chemin le plus court...). Deuxièmement, le modèle MAP permet l'exécution d'un processus à la volée, en suivant la situation en constante évolution du produit pour aider à choisir le bon objectif à atteindre. La Figure 65 illustre l'utilisation combinée des deux niveaux.

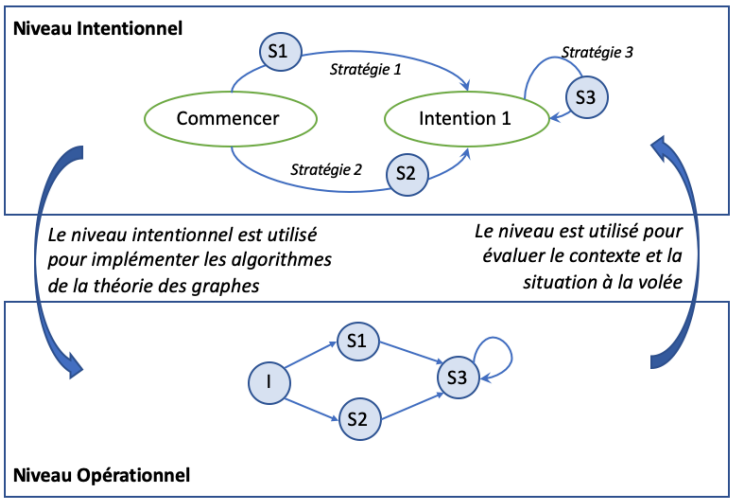


Figure 65. Utilisation des deux niveaux

La façon de faire est la suivante. A partir d'une intention dans la carte, l'ingénieur utilisera le niveau opérationnel du graphe pour appliquer un algorithme de théorie des graphes. Cet algorithme proposera un sommet. Le technicien exécutera ensuite la section correspondante sur le niveau intentionnel de la carte (cette section représente un service exécutable spécifique). Ce processus est répété jusqu'à ce que l'ingénieur atteigne l'intention finale de la carte. L'algorithme proposé est détaillé comme suit.

Avec une carte M et un graphe correspondant G (et toutes les cartes affinées M' ayant un sous-graphe correspondant G'), étant donné un algorithme de théorie des graphes que l'ingénieur souhaite utiliser et étant donné I le nœud initial du graphe et F son nœud final, la Figure 66 présente l'algorithme utilisé dans cette approche.

Current := I Num := 1 Path[1] := I result:= Apply-algo (I, FV)	1
list:= (second-vertex(result))	2
next := list While next != F do	3
Realize (next) If bundle(next) G=Delete-others-vertices-bundle(G, next)	4
end-if G=Evaluate-weight(G)	5
Current := next Num := num+1 Path[num] := current result:= Apply-algo (path[num], F) list:= list + second-vertex(result) next := best-section (list)	6
End-while	

Figure 66. Algorithme utilisé dans l'approche de la théorie des graphes

Les principales étapes de cet algorithme sont expliquées comme suit. Après l'initialisation de plusieurs variables (1), nous appliquons l'algorithme de théorie des graphes choisi du sommet initial au sommet final afin d'obtenir un chemin. Le deuxième sommet de ce chemin est alors mis dans une liste des sections potentielles qui peuvent être atteintes à ce point (2).

L'algorithme contient une boucle while (3) qui sera effectuée jusqu'à ce que l'ingénieur atteigne l'intention finale de la carte.

L'ingénieur atteint le niveau intentionnel en réalisant la section correspondante du sommet. Il est à noter que la réalisation de cette section peut être réflexive, car elle peut être affinée comme une autre carte, et il se peut que nous devions redémarrer cet algorithme avec le sous-graphe correspondant G'.

Si la section réalisée fait partie d'un bundle (4), nous savons que nous ne serons pas autorisés à exécuter une autre section de ce bundle. En conséquence, les sommets correspondants du graphe G sont supprimés pour assurer cette règle.

L'exécution de la section modifiera les poids des indicateurs dynamiques (état des intentions et réalisation des directives). Ils sont évalués et intégrés dans chacun des poids des arcs (5).

Nous pouvons revenir au niveau opérationnel et poursuivre notre chemin un peu plus loin, jusqu'à atteindre le sommet final (6). Notez que le chemin suivi est mémorisé afin de pouvoir revenir en arrière dans la carte. Afin de savoir quelles sections sont atteignables à partir de notre intention, nous appliquons l'algorithme de théorie des graphes choisi de toutes les sections réalisées (tous les sommets du chemin choisi) au sommet final afin d'obtenir les chemins potentiels. Le deuxième sommet de chaque chemin est alors mis dans une liste des sections potentielles qui peuvent être atteintes à ce point. Une évaluation est alors faite entre

tous les éléments de cette liste afin de choisir entre un chemin en arrière ou en avant (cette évaluation prend en compte les indicateurs dynamiques afin de savoir s'il est nécessaire de revenir en arrière dans la carte).

Illustration

Afin d'illustrer notre proposition, nous avons choisi la carte décrivant le processus de construction d'un modèle O* (Rolland 2005c) présentée à la Figure 67.

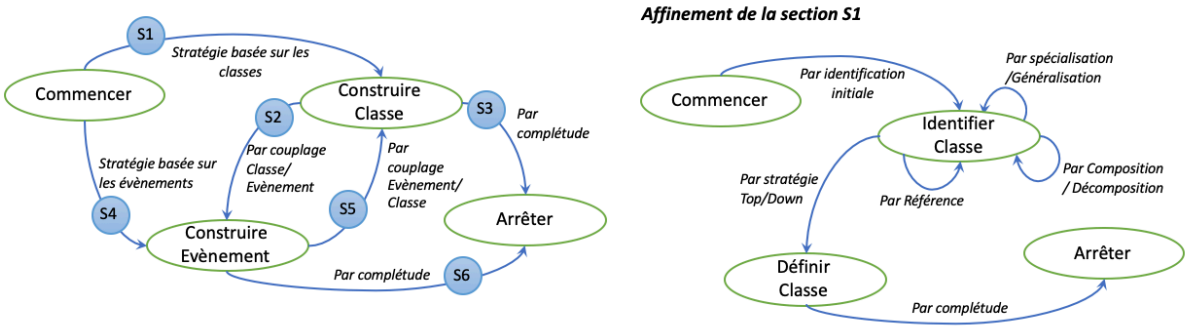


*Figure 67. Carte de la construction d'un modèle O**

Il y a deux façons de lancer la construction d'un modèle O*. D'une part, l'ingénieur peut construire les classes du modèle (par identification initiale, par composition, par héritage, par référence...) avec la stratégie basée sur les classes (section S1). D'autre part, il peut également choisir d'établir le modèle par la construction des événements (par l'identification initiale, par la stratégie descendante ou ascendante...) avec la stratégie basée sur les événements (section S4).

Ces deux sections sont affinées par l'exécution de deux orientations stratégiques qui sont elles-mêmes représentées sous forme de cartes. Par exemple, la carte de droite de la Figure 67 représente l'affinement de la section S1 <Démarrer, Stratégie basée sur les classes, Construire classe>.

À partir du moment où une classe a été construite, l'ingénieur peut alors la compléter, avec la stratégie de couplage classe/événement (section S2) ou par l'identification d'événements connexes - soit les événements qui déclenchent les opérations de cette classe ou un événement interne possible (événement déclenché par une modification spécifique d'un état de classe). De la même manière, la stratégie de couplage Évènement/Classe (section S5) permet à l'ingénieur d'identifier les classes associées (soit la classe impactée par les opérations déclenchées par l'événement, soit une classe à partir de laquelle une modification d'état a déclenché cet événement particulier). Ces deux sections sont également affinées par deux autres cartes.

La stratégie de complétude permet à l'ingénieur de terminer le processus par une vérification du modèle produit obtenu, soit après avoir créé une classe (section S3) ou un événement (section S6). Notez que la complétude ne sera obtenue que s'il y a eu un couplage de ces concepts. L'ingénieur n'aura alors pas la possibilité de terminer la carte sans coupler les événements créés aux classes créées (et vice versa).

Le créateur de la carte a défini les valeurs suivantes pour les indicateurs de poids. L'état de l'objectif et les valeurs de réalisation des directives sont toujours égales à 0 avant toute exécution du processus, car ce sont des valeurs dynamiques qui seront évaluées « à la volée ».

Section		Valeur Statique agrégée
S1	Commencer – Stratégie basée sur les classes – Construire Classe	9
S4	Commencer – Stratégie basée sur les évènements – Construire Évènement	7
S2	Construire Classe – Par couplage Classe / Évènement – Construire Évènement	6
S5	Construire Évènement – Par couplage Évènement / Classe - Terminer	8
S3	Construire Classe – Par complétude – Terminer	1
S6	Construire Évènement – Par complétude – Terminer	1

La valeur agrégée statique des poids de section est calculée avant toute navigation sur la carte. Il représente la complexité des services exécutables correspondants. Dans cet exemple, certaines sections peuvent être affinées par d'autres cartes qui sont elles-mêmes des directives complexes. Le poids de la section représente la complexité de la hiérarchie de la carte.

Selon les règles présentées ci-dessus, nous décrivons la carte 0* comme un graphe (Figure 68).

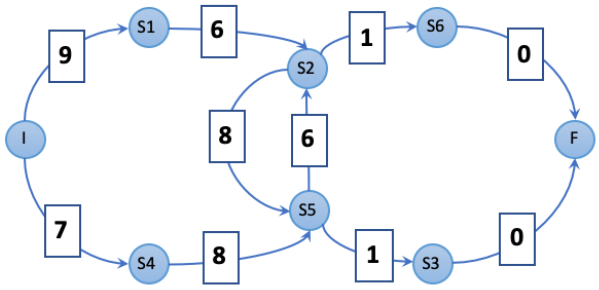


*Figure 68. Le graphe valué correspondant au processus de construction d'un modèle O**

Supposons que ce graphe soit décrit dans un langage interprétable. Il existe différentes façons de stocker des graphes dans un système informatique (soit des listes ou des matrices, soit les deux). Par exemple, la représentation proposée par van Rossum, dans laquelle une table de hachage est utilisée pour associer chaque sommet à un tableau de sommets adjacents, peut être considérée comme une instance de ce type de représentation. Une autre façon de représenter ce graphe sera d'utiliser la structure XGMML (eXtensible Graph Markup and Modeling Language)[1]. Nous avons choisi la représentation matricielle dans cet exemple.

Notre exemple de graphe peut être décrit avec les ensembles E,V et G suivants.

```
V = {I,S1,S2,S3,S4,S5,S6,F}
E = {(I,S1,9), (I,S4,7), (S1, S2,6), (S2,S5,8), (S2,S6,1), (S3,F,0), (S4,S5,8),
(S5,S2,6), (S5,S3,1), (S6,F,0)}
G={'I': ['S1, 9', 'S4, 7'], 'S1': ['S2, 6'], 'S2': ['S5, 8', 'S6, 1'], 'S3': ['F,
0'], 'S4': ['S5, 8'], 'S5': ['S2, 6', 'S3, 1'], 'S6': ['F, 0'], 'F': []}
```

[1] http:// www.cs.rpi.edu/~puninj/XGMML

Cette section illustre l'utilisation de cette double vue d'un processus avec un exemple simple. Nous voulons construire un modèle O* pour la description suivante du projet : « Le client téléphone à l'entreprise pour obtenir une réservation. Il donne des informations sur lui-même (nom, adresse et téléphone) et sur la réservation (date de début, date de fin). Le client peut également annuler une réservation « OK ». »

L'ingénieur choisit d'appliquer systématiquement l'algorithme du trajet de poids minimal. La procédure « Apply-algo (A, B) » choisira alors un chemin entre les sommets A et B qui sera celui avec le poids minimal.

Affectations initiales

```
Current = I
Num=1
Path[1]= I
result:= Apply-algo (I, F)
list:=(second-vertex(result))= S4
next := S4
```

Première iteration

```
Realize (S4) G=Evaluate-weight(G)
Current := S4
Path[2] := S4
Result = apply-algo (S4, F)
second-vertex(result) = S5
list = (S4, S5)
Next = best-section (S4,S5) = S4
```

La première itération de l'algorithme permettra de réaliser la section S4 qui créera l'événement « Demande de réservation ».

Après cette itération, l'ingénieur analyse la situation et établit que l'intention est réalisée à 50 %. La ligne directrice de cette section n'est pas complète car il reste un événement à identifier.

Seconde iteration

```
Realize (S4) G=Evaluate-weight(G)
Current := S4
Path[3] := S4
Result = apply-algo (S4, F)
second-vertex(result) = S5 list = (S4, S5)
Next = best-section (S4, S5) = S5
```

La deuxième itération effectue une deuxième fois la section S4, afin de créer l'événement « Annuler une réservation ». À ce stade, l'ingénieur établit que l'intention donnée est remplie. Sur la base de ces résultats, l'ingénieur continue d'appliquer l'algorithme de S4 sans revenir au sommet initial (I).

L'application d'algorithme identifie que la section suivante à réaliser est S5, il est donc nécessaire de coupler les événements trouvés aux classes.

Troisième itération

```
Realize (S5)
G=Evaluate-weight(G)
Current := S5
Path[4] := S5
Result = apply-algo (S5, F)
second-vertex(result) = S3
list = (S4, S5, S3)
Next = best-section (S4,S5,S3) = S3
```

La troisième itération ira plus loin dans la carte afin de réaliser la section S5 pour identifier les classes couplées à l'événement « Demande de réservation », qui sont d'abord la classe réservation, mais aussi la classe client (que nous trouvons par une étude des liens de référence de la classe de réservation).

La section suivante à réaliser est S3, pour tester l'exhaustivité du produit.

Quatrième itération

```
Realize (S3)
G=Evaluate-weight(G)
Current := S3
Path[5] := S3
Result = apply-algo (S3, F) second-vertex(result) = F
list = (S4, S5, S3, F)
Next = best-section (S4,S5,S3,F) = F
```

La quatrième itération effectue la section S3 qui teste l'exhaustivité du produit souhaité.

Au niveau intentionnel, l'ingénieur a atteint l'intention Terminer ; par conséquent, la navigation sur la carte est finie. Au niveau opérationnel, le processus passe au sommet final F, ce qui met fin à la navigation dans le graphe.

Cet exemple montre l'utilisation de l'algorithme du plus court chemin sur une carte exprimée sous forme de graphe. L'orientation a été améliorée pour faciliter les décisions de l'ingénieur, car chaque section lui a été proposée au cours du processus, en minimisant la complexité du processus.

III.2. Utiliser un modèle intentionnel par guidage en fonction du comportement des utilisateurs

Les techniques de fouille de processus permettent, non seulement de découvrir des modèles de processus en étudiant les logs d'évènements des utilisateurs, mais aussi de pouvoir faire de la recommandation à la volée. Nous avons proposé deux approches en ce sens : l'approche MAP-TBS et l'approche FlexPAISSeer.

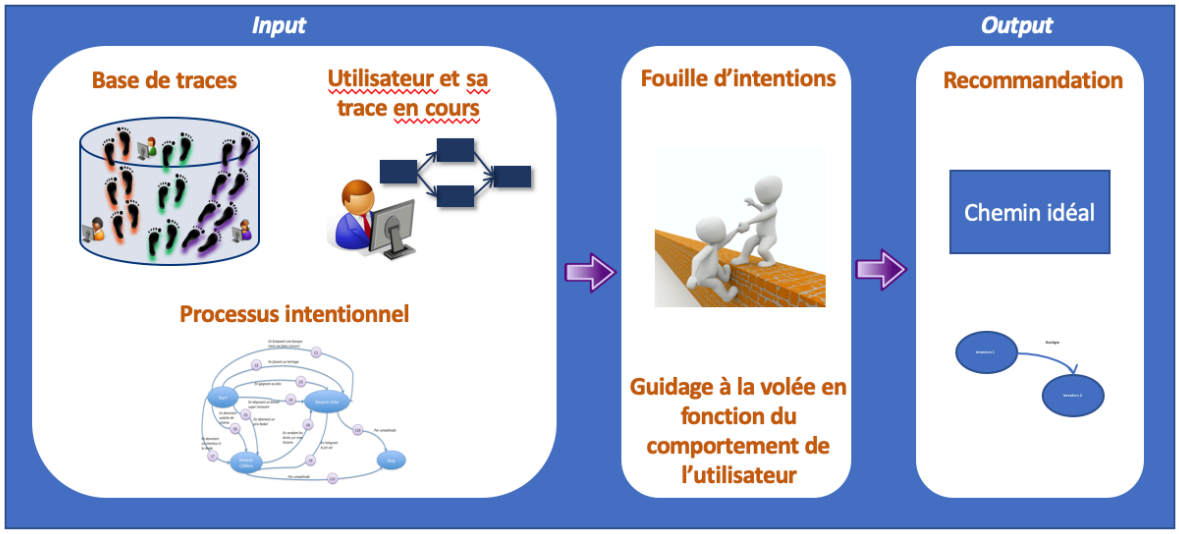


Figure 69. Entrées-Sorties de l'utilisation d'un modèle intentionnel en fonction du comportement des utilisateurs

III.2.1. Utiliser un modèle intentionnel par guidage en fonction du comportement des utilisateurs : l'approche Map-TBS

Un Système basé sur les traces (Trace-Based-System - TBS) est un système qui permet d'enregistrer les traces, de les transformer et de les visualiser. Le premier objectif du TBS est de permettre à l'utilisateur de réfléchir à ses activités. Visualiser nos propres traces enregistrées à l'aide d'un outil informatisé peut nous permettre de nous rappeler et de mieux comprendre ce qui a été fait et ce qui aurait pu être fait pour être plus efficace par exemple. Le deuxième objectif des systèmes de traces est de permettre à l'utilisateur de partager et de réutiliser sa trace. Partager une trace signifie que la trace peut être lue et comprise par l'utilisateur pour maintenir l'activité ou par tout autre utilisateur du système pour réutiliser la trace. Tout en faisant quelque chose de nouveau dans le système, il est utile de savoir ce que les autres utilisateurs ont fait dans le même contexte.

Nous avons défini Map-TBS (Hug 2012) pour gérer les traces modélisées de nos processus intentionnels. Une trace modélisée résulte de l'observation des interactions entre l'utilisateur et le système, c'est-à-dire de l'observation de la navigation à travers le processus par la sélection de sections spécifiques. La Figure 70 présente une vue de haut niveau du modèle Map-TBS. Une trace modélisée est composée de plusieurs évènements, chacun correspondant à une sélection et réalisation de section dans la carte. Chaque élément Trace contient deux attributs : une valeur temporelle estampillant l'heure exacte de la sélection et

une annotation. (Kort 2005) a souligné qu'un défi permanent est l'annotation qualitative des logs d'évènements. En fait, identifier une trace n'est pas suffisant pour trouver sa justification. Un élément trace implique plusieurs sections modélisées : l'une est sélectionnée pour aller plus loin dans le processus (section sélectionnée) sur plusieurs alternatives candidates (sections candidates).

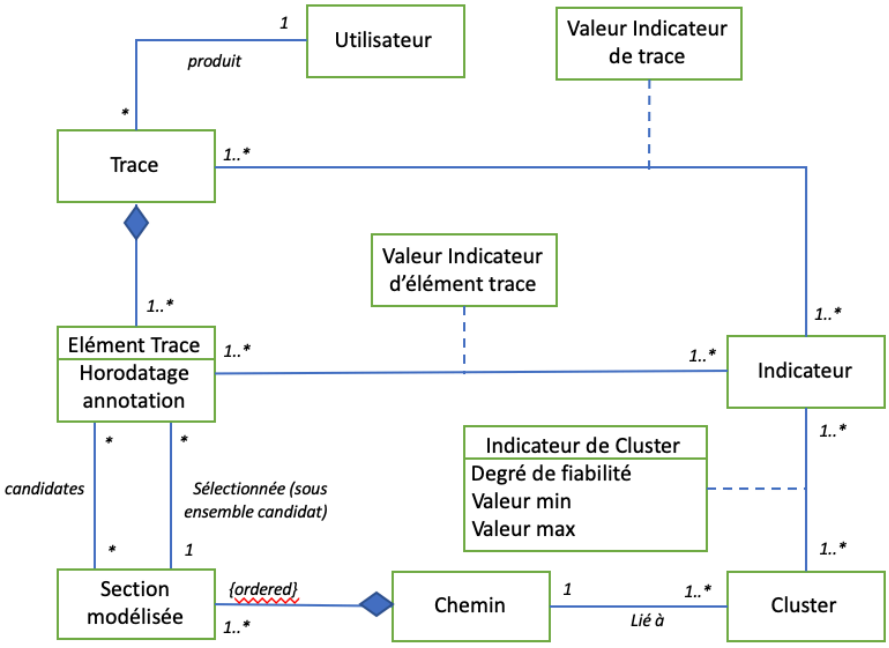


Figure 70. Modèle Map-TBS

Chaque trace correspond à un certain contexte. La situation dans laquelle la trace a été enregistrée est conservée dans un ensemble de valeurs d'indicateurs de traces. Ces indicateurs concernent l'utilisateur qui met en œuvre le processus (âge, sexe), son expertise et son expérience concernant le projet, la nature innovante du projet, etc. Les valeurs des indicateurs précisent le contexte particulier dans lequel une section a été réalisée. Ils peuvent se référer à la complexité du produit en cours d'élaboration, au niveau d'expertise de l'utilisateur pour atteindre l'intention, etc. Les indicateurs utilisés ici sont définis dans (Kornyshova 2011a). Un cluster est un ensemble d'indicateurs génériques de groupe. Il définit un type spécifique de contexte qui peut être réutilisé plus tard pour fournir des recommandations aux utilisateurs. Les indicateurs de cluster ont plusieurs attributs : une valeur minimale et une valeur maximale, qui déterminent la plage de valeurs prises en compte par le profil, et un degré de fiabilité de cet intervalle. Par exemple, un cluster peut être défini pour les utilisateurs de 20 à 30 ans, mais avec une faible fiabilité. Cela signifie que, si les autres valeurs des indicateurs correspondent aux valeurs du cluster, il est possible d'accepter un utilisateur de 19, même s'il n'est pas dans la plage d'âges prédéfinie. Un groupe est lié à un cheminement précis dans un modèle de processus intentionnel qui correspond à la recommandation.

La Figure 71 présente les différentes étapes de la méthode pour fournir des recommandations aux utilisateurs tout en mettant en œuvre des modèles de processus intentionnels.

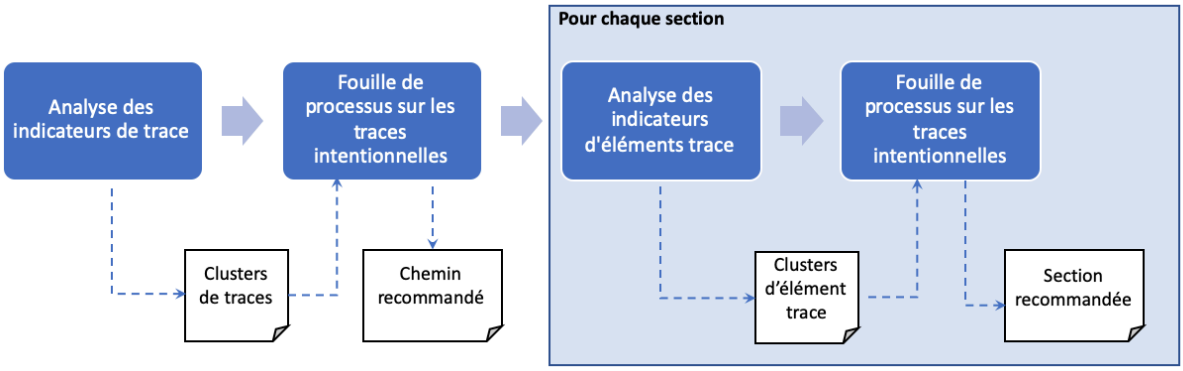


Figure 71. Méthode prescrite pour offrir des recommandations avec Map-TBS

Les étapes « Analyse des indicateurs de trace » et « Analyse des indicateurs d'éléments trace » de la Figure 71 utilisent les indicateurs et les techniques de clustering pour détecter différents types de groupes homogènes dans les nombreux cas d'un modèle de processus intentionnel. La valeur de l'indicateur de trace permet de caractériser le projet et l'utilisateur qui adopte le modèle de processus de la carte : rôle, âge, sexe, expérience, expertise, taille du projet, durée, coût, type d'application sont des exemples d'indicateurs de trace. Ils permettent de détecter des groupes d'utilisateurs homogènes et de types de projets. Notre objectif est de fournir des recommandations aux utilisateurs d'une carte qui peuvent être des ingénieurs en élicitation des exigences, des gestionnaires de projet ou des analystes par exemple. Il y a ensuite différents profils d'utilisateurs selon un modèle de processus intentionnel, appelé « cluster de traces » à la Figure 71. En utilisant l'analyse par correspondance multiple (Multiple Correspondence Analysis - MCA), nous sommes en mesure de réduire le nombre d'indicateurs à deux dimensions.

Ensuite, différents groupes peuvent être définis en appliquant l'algorithme HCPC sur l'ensemble d'individus obtenus. Les valeurs des indicateurs d'éléments traces sont semblables aux indicateurs de traces, sauf qu'elles sont définies au niveau de la section. Pour chaque section réalisée d'une carte, les valeurs des indicateurs sont définies. Les indicateurs d'éléments traces sont, par exemple, la taille du produit, la qualité du produit, le degré de complexité du produit ou le degré d'innovation du produit. Les groupes d'éléments traces sont calculés en fonction des indicateurs d'éléments traces et de la section à laquelle ils correspondent. Ces clusters nous aideront à catégoriser un utilisateur lorsqu'il atteint une section particulière du modèle de processus intentionnel mis en œuvre. Comme dans l'étape « Analyse des indicateurs de trace », les algorithmes MCA et HCPC peuvent être utilisés pour détecter les groupes d'éléments de trace. Les indicateurs permettent de définir des groupes au niveau de la carte (indicateurs de traces) et au niveau de la section (indicateurs d'éléments traces).

Les deux étapes « Fouille de processus sur les traces intentionnelles » utilisent des techniques de fouille de processus pour fournir des recommandations aux utilisateurs en fonction de leur groupe (groupe de traces ou groupe d'éléments traces). Les modèles de séquence sont découverts à l'aide de techniques qui détectent des séquences similaires de sections sélectionnées dans les traces intentionnelles.

Ensuite, chaque cluster de trace est associé à un chemin potentiel qui correspond à une trace fréquente. La combinaison du clustering et de la fouille de processus nous aide à

suggérer une séquence de sections en fonction des valeurs des indicateurs de traces. De plus, les groupes d'éléments traces sont associés à une section particulière. Nous pouvons proposer la section suivante la mieux adaptée à suivre selon le groupe en utilisant des techniques de fouille. L'algorithme de fouille de séquence CSpade permet de calculer les séquences les plus fréquentes de sections qui apparaissent dans les traces. L'algorithme CSpade analyse les traces composées de l'id de séquence (l'id de trace, qui identifie chaque trace d'une manière unique), de l'horodatage, qui correspond à l'horodatage de l'élément trace, du nombre de sections atteintes en ce moment, la liste correspondante des sections atteintes en calculant la fréquence des séquences.

Illustration

Nous allons maintenant illustrer comment Map-TBS peut être utilisé avec la méthode d'ingénierie des exigences CREWS-L'Ecritoire (Rolland 1998b), visant à découvrir et à obtenir des exigences (Figure 72).

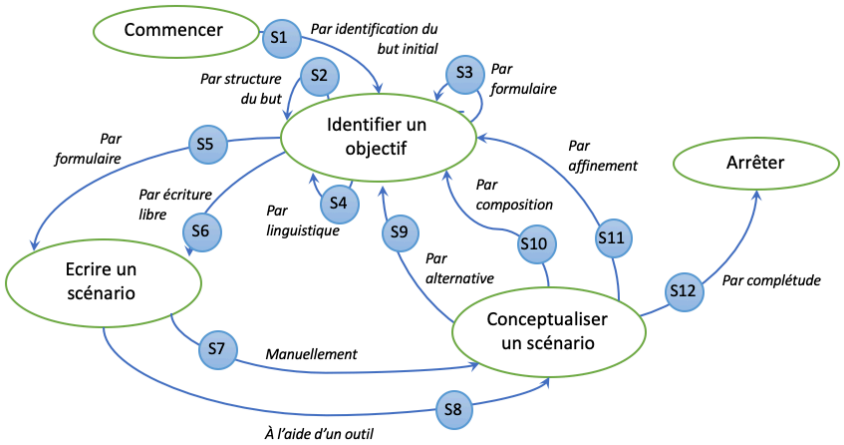


Figure 72. Carte de définition des exigences CREWS-L'Ecritoire (CE)

Prenons l'exemple fictif de Bob en charge de l'analyse des besoins pour un nouveau site de recherche d'emploi comprenant des innovations commerciales. Bob utilise la carte pour obtenir les exigences. La première étape consiste à fournir des valeurs pour les indicateurs de trace, pour identifier le cluster de trace auquel Bob appartient. Il est un gestionnaire de projet de 33 ans avec une faible expérience dans les cartes ; le projet durera 12 mois et a un coût moyen. Nous simulons 75 observations de valeurs d'indicateurs de traces d'autres utilisateurs de la carte. Comme certaines données sont classifiantes (rôle, sexe, expérience…) et que d'autres sont numériques (âge et durée), nous effectuons une analyse par correspondance multiple (AMC) et transformons les données numériques en données classifiantes : l'âge sera défini comme {junior, confirmé, senior, expert} pour les valeurs : âge<30, 31<âge<35, 36<âge<45, âge>46, respectivement. La durée sera changée en {faible, moyenne, élevée} pour la durée en mois de : durée <= 12, 12<durée<=24, durée>24, respectivement.

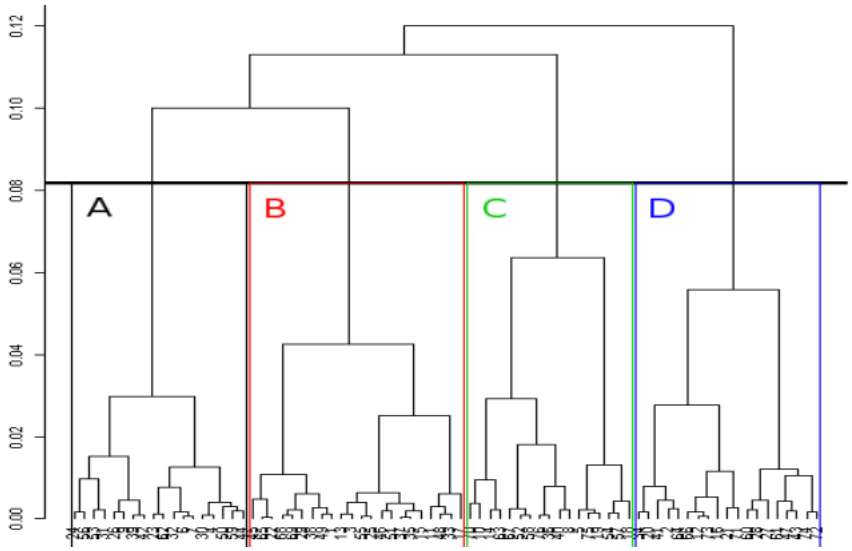


Figure 73. Classification hiérarchique des 75 observations

Quatre groupes ont été définis en fonction des valeurs des indicateurs de traces. La Figure 73 présente le regroupement hiérarchique des observations des autres utilisateurs de la carte. Ce diagramme a été obtenu en utilisant les fonctions MCA (Husson 2011) et HCPC avec l'outil open source R. La fonction HCPC permet de déterminer les caractéristiques significatives de chaque cluster : le groupe A correspond aux juniors féminines gérant des projets de faible durée et de grande taille, le groupe B correspond à des chefs de projet confirmés avec une expertise moyenne gérant des projets de durée et de coût élevés, le groupe C contient des experts masculins travaillant dans des projets de taille et de coût moyens, et enfin, le groupe D est composé d'ingénieurs des besoins chevronnés impliqués dans des projets intra-organisationnels, à coût élevé, de faible taille et de durée moyenne. Nous savons maintenant à quel groupe appartient Bob. En analysant les traces des utilisateurs du groupe C, nous sommes en mesure de fournir une recommandation de la voie qu'il pourrait prendre.

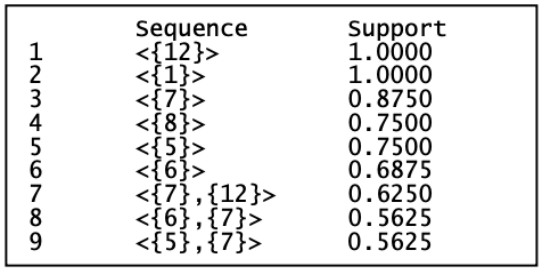
TraceID	Element TraceID	Horodatage	Annotation Utilisateur	Sections candidates	Section sélectionnée
343	1	2022-06-07 11:12		S1	S1
343	2	2022-06-07 11:20	J'ai pensé qu'il était plus simple de définir un objectif en utilisant un formulaire	S2, S3, S4, S5, S6	S3
343	3	2022-06-07 11:23	J'avais cette impression bizarre que c'était la bonne chose à faire.	S2, S3, S4, S5, S6	S5

Figure 74. Exemple de traces

La Figure 74 présente une trace de la réalisation de la carte. Il commence par le choix de la section S1, car il n'y a pas d'autre candidat. Ensuite, l'utilisateur a choisi d'obtenir un nouvel objectif en utilisant un modèle (S3), et à côté d'écrire un scénario en utilisant des modèles (S5). Grâce aux annotations, nous pouvons comprendre que l'utilisation de modèles peut aider les utilisateurs de la carte tout en définissant les objectifs et les scénarios du futur SI par exemple. Nous n'affichons pas les indicateurs d'éléments traces correspondants pour des raisons de lisibilité.

Pour recommander un chemin aux utilisateurs de la carte, nous exécutons l'algorithme CSpade dans l'outil R en utilisant le paquetage aRulesSequences (Buchta 2010). Cet algorithme

calcule les séquences de sections les plus fréquentes qui apparaissent dans les traces. Nous avons généré au hasard 75 traces et appliqué l'algorithme CSpade aux traces du groupe C. La Figure 75 présente un extrait des résultats retournés par CSpade.



	Sequence	Support
1	<{12}>	1.0000
2	<{1}>	1.0000
3	<{7}>	0.8750
4	<{8}>	0.7500
5	<{5}>	0.7500
6	<{6}>	0.6875
7	<{7},{12}>	0.6250
8	<{6},{7}>	0.5625
9	<{5},{7}>	0.5625

Figure 75. Extraction de la séquence trouvée dans les traces

Les deux premiers résultats sont évidents : chaque trace commence par la section S1 et se termine par la section S12, donc leur support est 1 (probabilité de 100%). On voit que la section S7 apparaît souvent dans les traces (support 0,875 ou 87,5%). Les séquences des sections S6-S7 et S5-S7 sont présentes dans 56% des traces. En unifiant les résultats de CSpade, nous pourrions considérer que les chemins recommandés pour le groupe C des utilisateurs de la carte pourraient être comme indiqué dans la Figure 76.

Intention Source	**Intention Cible**	**Séquence**	**Support**
Commencer	Identifier un objectif	<{S1}>	1.0000
Identifier un objectif	Identifier un objectif	<{S2}>	0,5625
Identifier un objectif	Identifier un objectif	<{S3}>	0,5625
Identifier un objectif	Identifier un objectif	<{S4}>	0,5000
Commencer	Identifier un objectif	<{S1},{S2}>	0,3750
Identifier un objectif	Identifier un objectif	<{S3},{S2}>	0,3125
Identifier un objectif	Identifier un objectif	<{S2},{S3}>	0,3125
Identifier un objectif	Écrire un scenario	<{S5}>	0,7500
Identifier un objectif	Écrire un scenario	<{S6}>	0,6875
Identifier un objectif	Écrire un scenario	<{S4},{S5}>	0,3750
Identifier un objectif	Écrire un scenario	<{S3},{S5}>	0,3125
Identifier un objectif	Écrire un scenario	<{S3},{S6}>	0,3125
Identifier un objectif	Écrire un scenario	<{S2},{S6}>	0,3125
Écrire un scenario	Conceptualiser un scenario	<{S7}>	0,8750
Écrire un scenario	Conceptualiser un scenario	<{S8}>	0,7500
Identifier un objectif	Conceptualiser un scenario	<{S6},{S7}>	0,5625
Identifier un objectif	Conceptualiser un scenario	<{S5},{S7}>	0,5625
Identifier un objectif	Conceptualiser un scenario	<{S5},{S8}>	0,5625
Conceptualiser un scenario	Identifier un objectif	<{S11}>	0,5625
Conceptualiser un scenario	Identifier un objectif	<{S10}>	0,4375
Écrire un scenario	Identifier un objectif	<{S8},{S11}>	0,4375
Conceptualiser un scenario	Terminer	<{S12}>	1,000
Écrire un scenario	Terminer	<{S7},{S12}>	0,6250
Écrire un scenario	Terminer	<{S8},{S12}>	0,3750

Figure 76. Chemins recommandés

La Figure 76 doit être lue comme suit. Les utilisateurs du groupe C commencent habituellement la carte en sélectionnant la section S1, ou ils sélectionnent S1 suivie de la section S2 dans 37,5 % des traces. Ensuite, si l'intention est d'obtenir un but, les utilisateurs du groupe C sélectionnent habituellement les sections S2, S3, S4 ou S3 suivies de S2 et vice versa. Si l'intention est d'écrire un scénario, ils sélectionnent S5 (75 %) ou S6 (68,75 %) la plupart du temps. Le tableau présente les séquences pour lesquelles le support est le plus

significatif. Maintenant, imaginez que Bob vient d'atteindre l'objectif « Identifier un objectif » et qu'il a besoin de recommandations sur ce qu'il devrait faire ensuite, puisqu'il est un utilisateur novice de cette carte. Il n'a réalisé que la première section de la carte ; les valeurs des indicateurs d'éléments traces sont faibles : la taille et la qualité des produits sont faibles, le degré d'importance est normal. Les fonctions MCA sont exécutées avec HCPC pour déterminer les groupes parmi les 16 indicateurs d'éléments traces des utilisateurs du groupe C. Les indicateurs d'éléments traces sont générés aléatoirement. Nous obtenons 2 groupes : le premier groupe est caractérisé par un degré de complexité produit élevé et le second groupe par un degré de complexité produit faible. Les autres variables ne sont pas significatives. Bob appartient alors au groupe 2.

Grâce aux valeurs des indicateurs d'éléments traces du groupe 2 et des traces associées, nous pouvons fournir une recommandation pour la prochaine section que Bob pourrait sélectionner. La Figure 77 présente la section recommandée pour les utilisateurs du groupe C2 : groupe C déterminé selon les valeurs des indicateurs de traces et groupe 2 selon les valeurs des indicateurs d'éléments traces dans le groupe C. 80% des utilisateurs du groupe C2 ont écrit un scénario après avoir identifié un objectif (S5). 60 % ont identifié un autre objectif en utilisant la stratégie axée sur la structure du but.

Intention Source	Intention Cible	Séquence	Support
Identifier un objectif	Écrire un scénario	<{S5}>	0,8
Identifier un objectif	Identifier un objectif	<{S2}>	0,6
Identifier un objectif	Écrire un scénario	<{S6}>	0,6
Identifier un objectif	Identifier un objectif	<{S3}>	0,4
Identifier un objectif	Identifier un objectif	<{S2},{S3}>	0,4
Identifier un objectif	Écrire un scenario	<{S3},{S5}>	0,4
Identifier un objectif	Identifier un objectif	<{S4}>	0,2
Identifier un objectif	Écrire un scenario	<{S2},{S5}>	0,2

Figure 77. Section recommandée

III.2.2. Utiliser un modèle intentionnel par guidage en fonction du comportement des utilisateurs : l'approche FlexPAISSeer

Cette approche a été développée dans le cadre du mémoire de master de Elena Epure dans lequel elle a développé un algorithme de recommandation appelé *IntentRecommender* (Epure 2013) (Epure 2014b). Cette approche se base sur les logs des utilisateurs pour faire des recommandations à la volée (voir Figure 69).

FlexPAISSeer a trois objectifs.

- Formuler des recommandations au niveau intentionnel en offrant un soutien plus efficace aux participants au processus de prise de décisions (pour l'identification des critères de décision, l'élaboration des alternatives et l'analyse de celles-ci).
- Fournir des recommandations selon le modèle de processus tel quel, découvert par *IntentMiner* (voir section II.3.1) au lieu d'utiliser un modèle de processus prédéfini qui pourrait ne pas être exactement suivi par les participants dans la pratique. De plus, *IntentMiner* transforme la trace partielle d'événements du participant au processus en un

flux d'intentions qui est donné en entrée à *IntentRecommender* et est également utilisé pour mettre à jour le modèle de processus intentionnel.

- Fournir des recommandations qui contiennent des informations sur le comportement d'autres participants au processus dans une situation de mise en œuvre similaire ou identique, au moyen d'un facteur de confiance (Cleger-Tamayo 2013). Le facteur de confiance est une valeur numérique associée à la recommandation, qui quantifie l'appariement et la fréquence des logs des participants au processus actuel en fonction des données connues du processus.

Donner des recommandations à partir d'un flux d'intentions $F = \{I_1 \rightarrow ... \rightarrow I_n\}$, $n \geq 1$ est une question de prédiction, ayant, comme connaissance préalable, le modèle de processus intentionnel. Une recommandation est la prochaine intention prévue, I_{predit}, qui correspond à un certain facteur de confiance $CF_{Iprédit}$. Nous nous concentrons sur la description des deux parties principales *d'IntentRecommender* : la prédiction et le calcul du facteur de confiance.

La prédiction est la détermination des prochaines intentions en fonction du flux d'entrée des intentions, F, et du modèle de processus. *IntentRecommender* comporte trois étapes :

- Découvrir l'ensemble d'intentions, SI_{predit}, qui sont directement atteignables à partir de la dernière intention I_n, $n \geq 1$ du flux F : $SI_{predit} = \{I_{predit} : I_n \rightarrow I_{predit}$, $n \geq 1$ existe dans le modèle de processus intentionnel$\}$.
- Pour chaque $I_{predit} \in SI_{predit}$, créer l'ensemble des prédécesseurs contenant les intentions trouvées dans le flux, triées par temps en ordre décroissant, $P_{Ipredit} = \{I_n ... I_1\}$ $n \geq 1$. Cependant, il y a deux problèmes possibles. Premièrement, le chemin décrit par F ne peut pas être entièrement trouvé dans le modèle de processus intentionnel. Dans ce cas, $P_{Ipredit}$ est modifié pour ne contenir que les intentions qui décrivent un flux existant vers I_{predit} dans le modèle de processus interne : $P_{Ipredit} = \{I_n ... I_k\}$, n, $k \geq 1$ et $I_k \rightarrow ... \rightarrow I_n \rightarrow I_{predit}$ existe dans le modèle de processus intentionnel$\}$. Deuxièmement, une intention pourrait apparaître plusieurs fois dans $P_{Ipredit}$. Dans ce cas, les interprétations pourraient être : (i) une intention figurait parmi sa liste de prédécesseurs, influençant ainsi son occurrence future ; ou (ii) le flux a exposé différentes façons d'atteindre cette intention. En invoquant le rasoir d'Occam (Lee 2011), qui précise qu'un modèle avec des hypothèses simples doit être choisi, nous avons choisi l'interprétation (ii). Cela implique une autre contrainte sur $P_{Ipredit}$: chaque intention dans la séquence des prédécesseurs doit être unique et différente d'I_{predit}.
- Pour chaque $I_{predit} \in SI_{prédit}$, calculer le facteur de confiance $CF_{Iprédit}$ (1) ayant la possibilité d'être ajusté par des coefficients α et β, $0 \leq \alpha, \beta \leq 1$. L'administrateur du processus peut décider que la fréquence d'un certain chemin est plus importante par la valeur de α, ou la correspondance d'un certain chemin est plus important, par la valeur de β.

$$CF_{Ipredit} = \alpha * P(\{I_{predit}\} + P_{Ipredit}) + \beta * L(\{I_{predit}\} + P_{Ipredit}) / L(F) \quad (1)$$

$$P(\{I_{predit} = P_{Ipredit}) = \text{Probabilité d'occurrence de } I_k \rightarrow \ldots \rightarrow I_n \rightarrow I_{predit}\ n,\ k \geq 1 \quad (2)$$

$$L(\{I_{predit}\} / P_{Ipredit}) = n - k + 2 = \text{Longueur de } I_k \rightarrow \ldots \rightarrow I_n \rightarrow I_{predit}\ n,\ k \geq 1 \quad (3)$$

$$L(F) = n = \text{Longueur de } I_1 \rightarrow I_2 \rightarrow \ldots \rightarrow I_n \quad (4)$$

Chaque fois qu'une nouvelle instance de processus est découverte, *IntentMiner* met à jour l'arbre T_I de chaque intention I avec tous les chemins qui y mènent et leurs fréquences. Nous

calculons les probabilités sur la base des données conservées dans T_I. L'arbre a une structure spécifique : une instance de processus entièrement découverte qui décrit un chemin vers I est stockée dans une feuille ; puis ce chemin est décomposé récursivement dans des chemins plus courts vers I en supprimant une intention de la queue jusqu'à ce qu'il n'y ait plus rien à enlever. Par exemple, considérons *IntentMiner* qui découvre l'instance de processus suivante : $I_1 \rightarrow I_2 \rightarrow ... \rightarrow I_k \rightarrow ... \rightarrow I_n$. L'arbre correspondant à l'intention I_k est mis à jour comme suit : le nœud de feuille $n_1 = I_1 \rightarrow I_2 \rightarrow ... \rightarrow I_k$ est créé, puis un nouveau nœud $n_2 = I_2 \rightarrow ... \rightarrow I_k$ est créé et lié à n_1 et ainsi de suite jusqu'à ce que la racine $r = I_k \rightarrow$ null soit atteinte. Pendant la décomposition du chemin, il peut arriver qu'un nœud soit déjà dans l'arborescence, auquel cas seul le lien est créé et la fréquence du nœud est incrémentée. Considérant, $\#T_I$ le nombre total de chemins trouvés qui mènent à l'intention I, nous avons :

$$P(\{I_{predit}\} + P_{Ipredit}) = Fréquence(\{I_{predit}\} + P_{Ipredit}) / \#T_{Ipredit} \quad (5)$$

Nous calculons le facteur de confiance (1) en utilisant (2-5) et créons la recommandation. Le calcul est réalisé pour chaque intention de SI_{predit}.

Illustration

Nous illustrons cette approche sur un exemple correspondant à un outil de gestion de garde d'enfants. Les entrées de l'algorithme comprennent le modèle de processus intentionnel et les arbres associés à chaque intention (couvert par *IntentMiner*). Lorsque le participant au processus invoque *IntentRecommender*, le tracé d'entrée est extrait. Il n'est pas nécessaire que l'instance de processus corresponde au modèle de processus, car notre objectif est de découvrir le processus et non de vérifier la conformité (Van der Aalst 2011).

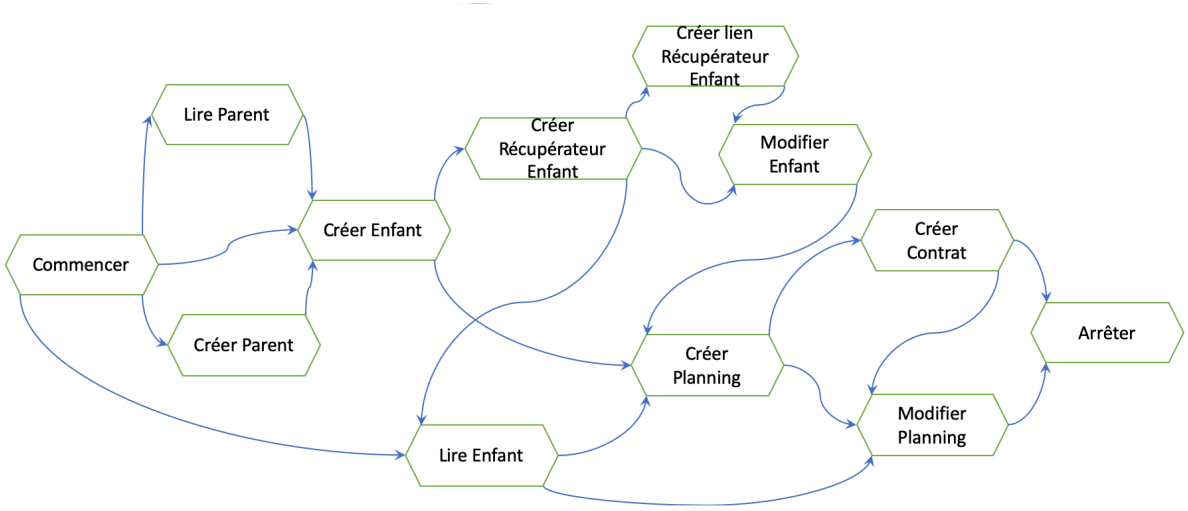


Figure 78. Intentions de bas niveau découvertes avec IntentMiner

(a)	Instance du processus intentionnel du participant : F: I_{t1} (Lire la liste des parents) → I_{t2} (Lire le parent) → I_{t3} (Créer l'enfant) → I_{t4} (Créer le récupérateur d'enfant)
(b)	Découvrir l'ensemble des intentions directement atteignables à partir de la dernière intention I_{t4}: $SI_{prédit}$={I_{p1} (créer le lien pour le récupérateur d'enfant), I_{p2} (modifier l'enfant), I_{p3} (lire l'enfant)}
(c)	Calculer le facteur de confidence pour I_{p3} (lire l'enfant): Pip2 = {It_4, I_{t3}, I_{t2}}

CF_{Ip3}=0,5*P({I_{p3}}+P_{ip3})+0,5*L({I_{p3}}+P_{ip3})/L(F)
= 0,5*P(I_{p3} ← I_{t4} ← I_{t3} ← I_{t2})+0,5*3/4
Où P(I_{p3} ← I_{t4} ← It_3 ← I_{t2}) est calculé selon la formule 5, considérant que l'information extraite de T_{ip3} (fréquence (I_{p3} ← I_{t4} ← I_{t3} ← I_{t2}) et #T*ip3*)

Figure 79. Exemple d'application de l'algorithme IntentRecommender

La première étape de l'algorithme consiste à identifier la dernière intention du participant au processus : I_{t4}. De plus, les intentions directement atteignables à partir de I_{t4} sont indiquées dans le modèle ($SI_{prédit}$ dans la figure). Le chemin vers I_{p3} est formé en fonction du tracé d'entrée et, ensuite, la sous-séquence la plus longue de ce chemin trouvé dans le modèle est extraite (P_{Ip3}). Sur la base de cette séquence maximale, le facteur de confiance est calculé et la première recommandation R_3 (I_{p3}, CF_{Ip3}) est formulée. Nous répétons l'étape c) pour les autres intentions de gauche – I_{p1} et I_{p2}, de la même manière.

Plus de détails sur cet exemple sont donnés dans (Epure 2013) et (Epure 2014a).

IV. Perspectives et Conclusion

Même si de multiples techniques ont déjà été définies dans la définition et dans le guidage des processus intentionnels, de nombreux défis restent encore à résoudre, notamment dans la fouille de processus intentionnels. Je vais vous présenter quelques-unes des perspectives identifiées dans les trois domaines impactés par l'ingénierie des processus intentionnels dans mes recherches.

IV.1. Perspectives en Ingénierie des méthodes situationnelles

Selon (Seligman 1989), une méthode d'ingénierie des systèmes d'information a quatre pilliers : « a way of thinking, a way of modeling, a way of working and a way of supporting ». La notion de famille de méthodes satisfait les trois premiers : (1) un paradigme ou une philosophie de pensée (la construction de méthodes à la volée par assemblage de composants), un ensemble de modèles (typologie des indicateurs, descripteur des composants, structure des composants) et une multi-démarche d'utilisation (les cartes de processus permettant à l'ingénieur de méthodes de construire sa méthode à la volée ou à l'utilisateur de naviguer dans les cartes).

Cependant nous ne nous sommes pas encore réellement occupés de fournir de supports outillés pour cette approche mais avons proposé uniquement l'utilisation d'un catalogue de composants (Kornyshova 2011). Il est indispensable de proposer maintenant des outils qui soutiendront la conception, le développement et l'utilisation des familles de méthodes.

IV.2. Perspectives en Ingénierie des processus

IV.2.1. Guidage avec des données de contexte plus hétérogènes

Le guidage d'un modèle intentionnel se fait en fonction du contexte (contexte du projet, contexte de l'utilisateur) et les descripteurs de composants permettent d'effectuer ce guidage. Les données IoT de senseurs peuvent ajouter beaucoup d'informations qui permettraient d'ajouter des connaissances contextuelles plus nombreuses permettant un guidage plus dynamique. Il serait donc intéressant d'identifier quelles données pourraient avoir un impact dans différents cas et de tester une approche intégrant ces données dans le guidage des utilisateurs.

IV.2.2. Populariser les modèles de processus intentionnels

Les modèles de processus intentionnels ne sont pas toujours d'une définition très facile. Il est plus complexe de réfléchir au niveau stratégique qu'au niveau opérationnel. Cette difficulté engendre des réticences à l'utilisation. Il faut continuer à promouvoir ce type de modèle en multipliant les cas d'études, en offrant un outil de modélisation plus facile d'utilisation que Power Point et en se rapprochant plus des entreprises pour leur offrir cette vision stratégique indispensable.

IV.3. Perspectives en Fouille d'intention

IV.3.1. De la difficulté d'avoir des traces d'activité utilisables

Ce domaine souffre du même maux que sa grande sœur, la fouille de processus : l'obtention de logs d'activités utilisables. Il est en effet assez difficile d'obtenir des logs : lorsqu'il y en a, beaucoup d'entreprises les gardent de manière confidentielle, mais il est surtout difficile d'obtenir des logs bien construits, les logiciels ou outils utilisés dans le processus des utilisateurs n'ayant en général pas été conçus en ce sens. Il est donc intéressant de s'intéresser également aux sources hétérogènes d'enregistrement d'activités pour en déduire une base de logs pouvant être exploitable.

Nous avons donc à développer des techniques permettant d'obtenir cette source indispensable à la fouille. La Figure 80 illustre les différentes stratégies pour la collecte des logs et les trois approches que nous avons déjà proposées dans ce sens.

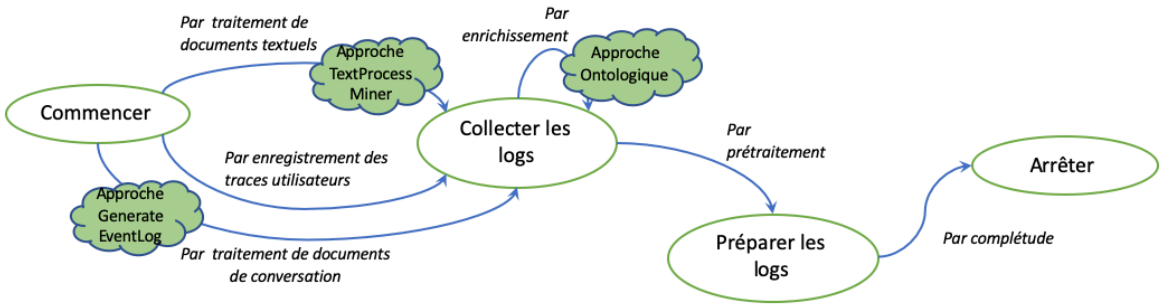


Figure 80. Fonctionnement de l'obtention de logs pour la fouille de processus intentionnels

Les logs d'évènements ne sont pas les seules sources qui peuvent donner des indications sur les activités exécutées. Des processus sont parfois renseignés de manière complètement textuelle mais il est tout de même possible d'extraire les activités dans ces documents. Nous avons travaillé sur deux approches permettant de découvrir des logs d'activités à partir de documents textes (Approche TextProcessMiner - annexe V.1) et de documents de conversation (Approche GenerateEventLog - annexe V.2).

Les sources où l'on peut trouver des séquences d'activités sont multiples et beaucoup de possibilités s'offrent pour identifier ces éléments dans des sources complètement hétérogènes. Il est donc également possible d'enrichir les logs d'activités avec des

informations de contexte qui permettent d'avoir des informations plus intéressantes et permettre d'extraire un meilleur processus intentionnel. Nous sommes en cours d'étude d'une approche pour ajouter ces informations aux logs d'évènements dans le but de faire de la fouille de processus intentionnels. En effet, une quantité énorme de données d'événements et de capteurs est enregistrée quotidiennement tandis que la propagation des technologies de l'information et de la communication (TIC) progresse dans le monde entier dans tous les domaines. Les données sont maintenant disponibles à grande échelle, et il est intéressant de pouvoir utiliser ces données pour enrichir les logs d'évènements. Il faut transformer ces données en valeur afin d'obtenir des renseignements, d'accroître la productivité, de produire de meilleures performances et d'économiser les coûts (Van der Aalst 2016). Dans le cadre de la thèse de Ramona Elali nous travaillons sur une approche ontologique permettant de combiner les logs d'activités à des données de senseurs à l'aide d'ontologies métier (Approche ontologique - Annexe V.3).

IV.3.2. Labellisation des intentions

La fouille de processus intentionnels se compose de plusieurs étapes dont la dernière est la labellisation des intentions. En effet, faire des regroupements d'activités (comme dans l'approche MMM – section II.3.2) ne permet que de faire des groupes mais ceux-ci ne sont pas étiquetés. Dans MMM, cette labélisation est faite de manière manuelle mais si l'on souhaite effectuer cette tâche de manière automatique, il faut alors faire un peu de traitement automatique du langage pour pouvoir retrouver le bon label pour le bon groupe, ce qui permet d'avoir le nom de l'intention le plus pertinent pour pouvoir obtenir un modèle compréhensible pour tout un chacun.

Par exemple, si l'on identifie un ensemble de stratégies permettant de réaliser une même intention (non étiquetée) appelées *par débugage*, en *testant*, *par refactorisation*, *par modification*. Il est manuellement possible de déduire que les développeurs qui ont exécuté ces stratégies ont l'intention de *Corriger un bug*. Il est toutefois bien plus difficile d'identifier le label de cette intention de manière automatique.

IV.3.3. Recommandations à la volée

L'un des objectifs de la fouille de processus intentionnels est la recommandation à la volée. Pour l'instant les approches proposées restent assez générales mais différentes possibilités existent pour réussir à guider l'utilisateur dans son processus intentionnel.

Celle-ci peut se faire uniquement avec une base de traces, le modèle intentionnel prescrit et la trace en cours d'exécution de l'utilisateur mais il est aussi possible d'ajouter d'autres informations telle que les données de contexte.

Ramona Elali est en cours d'élaboration d'une approche pour pouvoir faire ce type de recommandation, comme l'illustre la Figure 81 (Elali 2021) (Elali 2022b) (Elali 2022a).

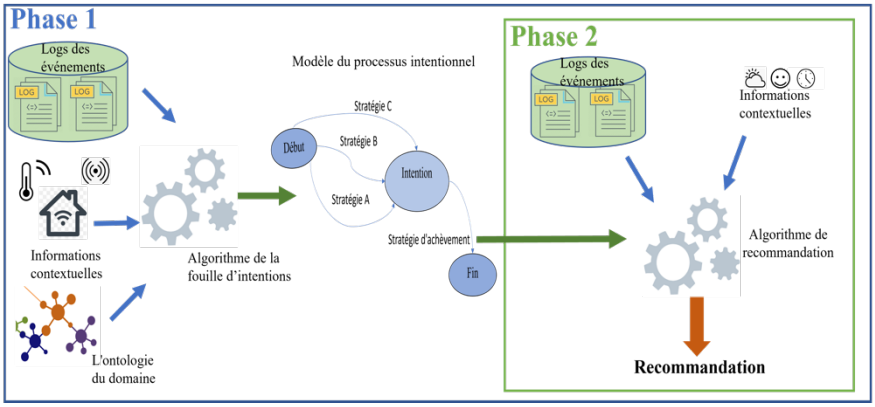


Figure 81. Approche proposée

- Dans la phase 1, différentes sources de logs (logs des événements, logs de capteurs, etc.) avec leurs informations contextuelles sont collectées. Ensuite, un algorithme de fouille d'intentions construit un modèle intentionnel. Cet algorithme découvrira les intentions et les stratégies du modèle intentionnel à partir des logs d'activités, avec l'aide de l'ontologie du domaine.

- Dans la phase 2, le modèle de processus intentionnel construit dans la phase 1 et les logs des activités actuelles de l'utilisateur avec ses informations contextuelles sont donnés comme paramètres d'entrée à l'algorithme de recommandation. Ensuite, après avoir traité les paramètres d'entrée, l'algorithme de recommandation fournira une recommandation appropriée pour l'utilisateur en fonction de son activité actuelle.

IV.4. Conclusion

Différentes contributions ont été présentées dans ce document. Celles-ci s'étendent de 1998, date à laquelle j'ai commencé à publier, à 2022 et mes dernières publications. La majorité de mes publications ont pour centre l'ingénierie des processus intentionnels, avec une forte concentration aux croisements de ce domaine avec celui de l'ingénierie des méthodes situationnelles et celui de la fouille de processus. Il y a encore beaucoup à faire et nous continuons à travailler sur les défis identifiés, notamment sur l'utilisation de sources hétérogènes pour la génération de logs et la recommandation à la volée.

V. Annexes

V.1. Annexe 1 : Génération de logs à partir de documents textes - l'approche TextProcessMiner

Ce travail a été réalisé dans le cadre de la thèse de Elena Epure (Epure 2018c) (Epure 2015).

Bien que les sciences humaines aient de plus en plus adopté la recherche numérique, il existe toujours une résistance importante à la modification des méthodes de recherche traditionnelles fondées principalement sur la production manuelle de sources textuelles et d'information qualitative (Chrysanthi 2012) (Gold 2012). Cette question est également alimentée par un mauvais alignement des systèmes d'information existants pour gérer, analyser et exploiter adéquatement le type particulier d'information découlant de la recherche en sciences humaines (Martin-Rodilla 2013) (Gonzales-Perez 2011) (Gonzales-Perez 2012) (Hug 2011).

Au contraire, d'autres domaines ont favorisé pléthore de recherches à cet égard, conduisant à l'émergence de plusieurs disciplines autonomes telles que la gestion des connaissances (Tiwana 2000), l'ingénierie des méthodes (Brinkkemper 1996) et la fouille de processus (Van der Aalst 2011) ainsi que leurs diverses solutions. Nous nous concentrons en particulier sur la fouille de processus, puisque les travailleurs en sciences humaines tiennent un registre textuel des processus suivis. En exploitant ces sources textuelles pour faire de la fouille de processus, nous pouvons faciliter et améliorer le travail d'équipe et le partage des connaissances. De plus, nous pouvons permettre d'établir un terrain d'entente pour la comparaison, la validation et la centralisation des processus et méthodologies appliqués. Ce sont des avantages déjà validés de la fouille de processus (Van der Aalst 2011).

La technique proposée, TextProcessMiner, génère la base de logs à partir du texte puis découvre un modèle d'instance de processus. Comparé à d'autres ouvrages connexes (Thorne 2013) (Soares 2013) (Goncalves 2010) (Ghose 2007) (Friedrich 2011), TextProcessMiner est entièrement non supervisé et utilise des techniques de traitement du langage naturel en mettant l'accent sur la sémantique du verbe.

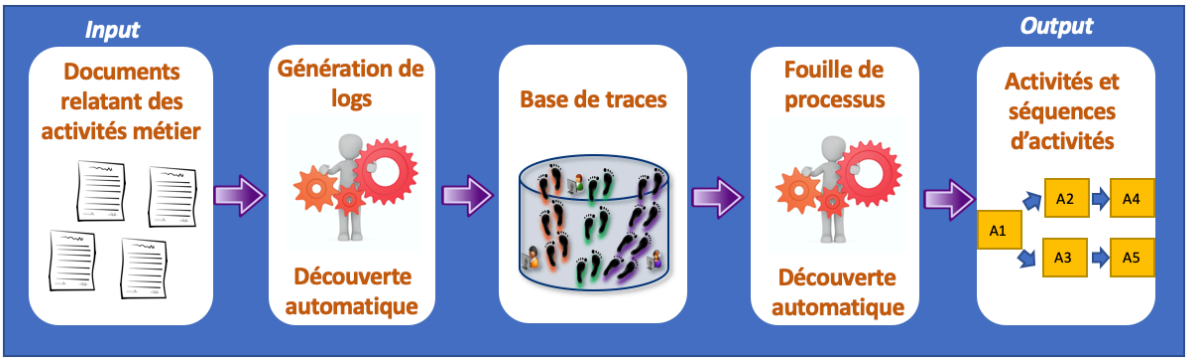


Figure 82. Entrées-Sorties de la construction de logs d'activités et de processus à partir de texte

L'approche TextProcessMiner a trois composantes: *TextCleaner* (nettoyage et préparation de la section textuelle concernant le processus suivi), *ActivityMiner* (extraction des activités du texte), *ActivityRelationMiner* (extraction des relations entre les activités, pour découvrir l'instance du processus).

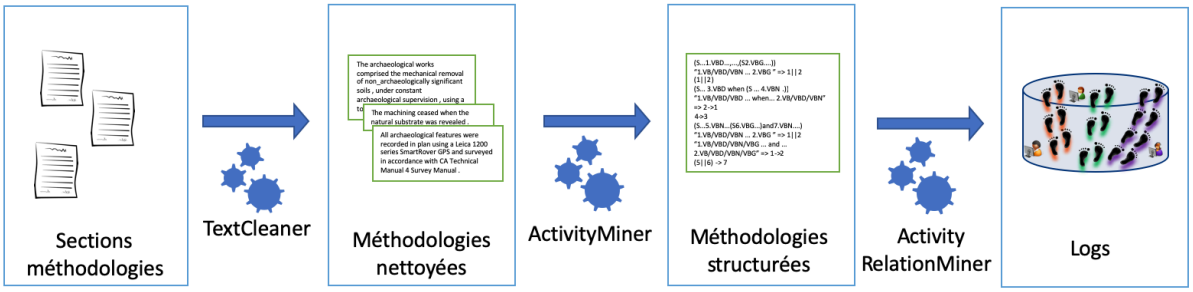


Figure 83. Processus suivi par TextProcessMiner

TextCleaner

L'entrée de *TextCleaner* est la section méthodologie du rapport textuel. Le traitement du texte consiste en l'extraction de la section, en la segmentation des phrases et en des actions de nettoyage. Ces mesures sont les suivantes : Remplacer « - » par « _ » ; Ajouter un espace avant et après chaque signe de ponctuation ; Retirer « » des noms possessifs singuliers ; Supprimer les commentaires introduits par des parenthèses si aucun verbe n'a été trouvé dans ces commentaires.

Nous supprimons également les phrases qui contiennent des négations parce que nous supposons que les activités sont introduites par des structures positives : elles représentent quelque chose qui a été fait. Après avoir analysé notre corpus, nous avons remarqué que les négations peuvent être liées à un nom : « no » + nom, « not » + nom, « none of » + nom, ou peuvent apparaître en relation avec un verbe : « not » + verbe, « n't » + verbe. Nous avons également relevé plusieurs exceptions, lorsqu'un mot-clé de négation peut apparaître dans une expression qui n'est pas une négation (p. ex., « not only... but also »). Dans cette situation, nous ne rejetons pas la phrase.

La sortie de *TextCleaner* est un fichier texte contenant une phrase par ligne. Ce fichier texte est ensuite introduit dans TextProcessMiner en entrée et utilisé par ses sous-composants : *ActivityMiner* et *ActivityRelationMiner*.

ActivityMiner

Le but *d'ActivityMiner* est d'exploiter des activités. Nous formulons les exigences suivantes *d'ActivityMiner* : l'algorithme doit identifier les verbes transitifs qui sont très probablement liés à une activité et, pour chaque verbe transitif découvert, identifier son ou ses objets.

Tout d'abord, *ActivityMiner* produit le treebank pour chaque phrase en utilisant les balises Stanford et NLTK, ainsi que l'analyseur Stanford (les détails de ce choix sont donnés dans (Epure 2015). L'analyseur de Stanford est capable de produire un treebank soit directement à partir d'un texte brut ou d'une phrase étiquetée - une liste de lignes contenant les mots de la phrase et leur partie des balises de discours. Le processus est donc le suivant : (1) utiliser

l'analyseur de Stanford avec du texte brut, (2) utiliser l'analyseur de Stanford avec le texte déjà étiqueté avec le taggeur de Stanford, (3) utiliser l'analyseur de Stanford avec le texte déjà étiqueté avec le tagueur NLTK.

Nous utilisons le résultat du tagger de Stanford comme entrée de base de l'analyseur mais nous le modifions artificiellement et automatiquement en privilégiant les balises NLTK dans les situations suivantes : (1) si le marqueur NLTK identifie le mot comme nom (NN, NNS) et si le marqueur de Stanford identifie le mot différemment, mais pas comme verbe dans la forme « ing » (VBG), suivi d'une préposition (IN), et non comme verbe dans la forme passée (VBD), et non comme verbe dans la forme antérieure du participe (VBN), (2) si le tag NLTK identifie le mot comme verbe dans la forme de base (VB) et si le tag Stanford identifie le mot différemment mais pas comme verbe dans toute autre forme (VBG, VBD, VBN) et non comme nom (NN, NNS).

Nous nous concentrons ensuite sur l'identification des verbes transitifs. Identifier les verbes dans une phrase est une tâche simple : nous cherchons dans le treebank pour les sous-arbres VP ou ADJP pour identifier ses enfants qui commencent par VB, le cas échéant. Cependant, une classification automatique des verbes comme extra-connaissance transitive et intransitive est nécessaire. Deux sources - VerbNet (Kipper 2004) et WordNet (Miller 1995) - ont été utilisées pour compiler un dictionnaire de verbes ayant comme clé le verbe dans la forme infinitive et comme valeur un drapeau booléen pour la transitivité. Bien que les sources soient différentes, elles partagent toutes deux des informations communes sur les verbes : les cadres. Un cadre illustre comment un verbe peut être utilisé dans une phrase simple (Kipper 2004). La transitivité dans VerbNet est considérée comme vraie si la trame « NP V NP » se trouve parmi les trames du verbe et si la première NP a le rôle sémantique d'Agent (Kipper 2004). Un agent est une entité active et intentionnelle qui exerce l'activité introduite par le verbe (Kipper 2004). Avec la condition que certains des verbes transitifs, qui sont plutôt des états que des actions, ont été exclus. De même, nous avons analysé le corpus WordNet (Miller 1995) et ajouté d'autres verbes au dictionnaire. Dans ce cas, la transitivité a été jugée selon que le cadre « quelqu'un verbe quelque chose » faisait partie des cadres du verbe. On peut remarquer que « quelqu'un » est l'équivalent d'un élément NP avec le rôle sémantique d'agent. Deux verbes ont été changés artificiellement pour être intransitif : « être » et « avoir », parce qu'ils ne représentent pas des activités. De plus, lors des expériences, lorsque des verbes transitifs qui n'étaient pas dans le dictionnaire ont été découverts, nous les avons ajoutés manuellement à notre fichier stockant le dictionnaire des verbes.

Les verbes peuvent être actifs ou passifs. Par conséquent, l'objet d'un verbe peut apparaître comme objet ou sujet. La forme du verbe n'est pas vérifiée. Au lieu de cela, nous vérifions d'abord si le verbe a un objet. Dans le cas contraire, nous considérons que le sujet est l'objet du verbe. Il s'agit en général d'une hypothèse valide puisque nous travaillons avec des verbes transitifs et que la voix passive est souvent utilisée dans les rapports. Pour trouver l'objet après le verbe, nous recherchons un élément NP étant le premier frère droit du sous-arbre VB. Pour trouver l'objet avant le verbe, nous cherchons le premier frère gauche du sous-arbre VP ou ADJP ou de l'un de leurs ancêtres. Un verbe peut avoir plusieurs objets dans une énumération ou un objet composé de plusieurs noms. L'algorithme *ActivityMiner* découvre les deux cas.

Le résultat de *ActivityMiner* est une liste de lignes où le premier élément de la ligne est le verbe et le deuxième élément de la ligne est la liste des objets. Les verbes et les noms sont lemmatisés à l'aide du lemmatizer WordNet (Lommatzsch 2014). Le pseudo-code de l'algorithme principal d'extraction des activités à partir d'une phrase est résumé dans l'algorithme à la Figure 84.

Algorithm 1 Function to mine activities from a sentence

Require: *sent* the input sentence and *Verbs* the dictionary of verbs
Ensure: *Activities* the list of tuples (*verbs*, *objects*)

```
1: function MINE_ACTIVITIES(sent, Verbs)
2:     Activities ← []
3:     tagged ← TAG_SENTENCE(sent)
4:     treebank ← PARSE_TAGGED(tagged)
5:     for all subtree ∈ subtrees(treebank) with node label VP or ADJP do
6:         verbs ← FIND_VERBS(subtree, Verbs)
7:         if #verbs = 0 then continue
8:         end if
9:         objects ← FIND_OBJS_AFTER_VB(subtree)
10:        if #objects = 0 then
11:            objects ← FIND_OBJS_BEFORE_VB(subtree)
12:            if #objects = 0 then continue
13:            end if
14:        end if
15:        activities ← FORM_ACTIVITIES(sent, verbs, objects)
16:        extend Activities with the newly discovered activities
17:    end for
18:    return Activities
19: end function
```

Figure 84. Algorithme principal d'extraction des activités

ActivityRelationMiner

Le but de *ActivityRelationMiner* est d'explorer les relations entre les activités découvertes. Comme nous l'avons déjà mentionné, il existe trois types de relations : la séquence, le parallélisme et l'exclusion mutuelle.

Dans la section de la méthodologie du rapport archéologique, l'auteur rapporte les événements dans l'ordre dans lequel ils se sont produits. Par conséquent, la relation par défaut entre deux activités consécutives est la séquence. En outre, nous considérons également que deux phrases sont séquentielles par défaut. Cependant, si l'ordre est différent, il y a des indices dans le texte qui annoncent le changement, par exemple des structures temporelles comme « last year », « in 2011 » ou des prépositions / conjonctions comme « before », « after », « and », « or », « in order to ».

Nous utilisons *ActivityRelationMiner* sur une base de règles. La base de connaissances représente un ensemble de règles définies après analyse du corpus. Certaines règles sont présentées dans la Figure 85. Les activités extraites par *ActivityMiner* pour chaque phrase ainsi que la phrase elle-même est en entrée de l'algorithme. Premièrement, la phrase est transformée en ne conservant que les balises des verbes trouvés entre les activités, les balises S et leurs parenthèses correspondantes et d'autres structures clés comme les conjonctions, ponctuation, prépositions, etc. Tous les autres mots sont remplacés par « ... ». Nous en montrons deux exemples :

- La phrase « The excavations were structured to accommodate the requirements of the developer. » devient « ... 1.VBN à 2.VB ... . »;

– La phrase « Particular areas were targeted by the second machining, including sondages across ditches and enclosure interiors. » devient « 1.VBN ... , 2.VBG ... et ... . ».

Les nombres sont utilisés pour relier les verbes transitifs aux activités découvertes.

... 1.VB/VBN/VBD ... , ... 2.VBG ...	1 -> 2 (independent clause)
... 1.VB/VBN/VBD 2.VBG ...	1 || 2 (dependent clause)
... where VB/VBD/VBN 2.VB/VBD/VBD ...	1? -> 2 (decision)
... 1.VB/VBN/VBD ... 2.VB/VBN/VBD ... or 3.VB/VBN/VBD ...	1 x 2 x 3 (branches)
... in order to 1.VB ... , ... 2.VB/VBN/VBD ...	2->1(sequence)

Figure 85. Règles pour découvrir les relations entre les activités

Les règles sont toujours appliquées pour une paire de verbes. Pour extraire les paires, nous prenons en considération les clauses de la phrase et les dépendances entre les verbes. Après avoir découvert la relation pour chaque paire de verbes, le fragment de processus pour la phrase complète est composé. Enfin, les fragments de processus obtenus de toutes les phrases sont assemblés pour obtenir l'instance de processus. Nous considérons que les fragments du processus sont dans l'ordre, sans ordonner les phrases et les articles indépendants en fonction des structures temporelles.

Nous considérons également la situation des verbes ayant des objets multiples. L'activité est transformée en plusieurs activités parallèles, une pour chaque objet, si les objets sont énumérés avec « and ». L'activité est transformée en plusieurs activités mutuelles exclusives, une pour chaque objet, si les objets sont énumérés avec « or ». L'algorithme correspondant est présenté à la Figure 86.

Algorithm 2 Function to mine the process instance fragment for a sentence
Require: *sent* the input sentence and *Activities* its list of activities
Ensure: *ProcessFragment* a string with the activities and their relationships represented with the symbols from Table I

```
1: function MINE_PROCESS_FRAGMENT(sent, Activities)
2:     transformed ← TRANSFORM_SENT(sent, Activities)
3:     pairs ← EXTRACT_VB_PAIRS(transformed)
4:     relationships ← []
5:     for all pair = (v1, v2) ∈ pairs do
6:         r ← APPLY_RULES(pair, transformed)
7:         add r to relationships
8:     end for
9:     ProcessFragment ← BUILD_STRING(relationships, transformed)
10:    return ProcessFragment
11: end function
```

Figure 86. Algorithme d'identification de l'instance d'un processus pour une phrase

Nous montrons ici qu'il est possible de partir de documents textuels pour obtenir des séquences d'activités qu'il sera plus tard possible d'utiliser en tant que source pour définir un modèle de processus intentionnel.

Illustration

Nous avons décidé de nous concentrer sur l'archéologie dans cette expérimentation, compte tenu d'une collaboration établie que nous avions avec cette communauté. Nous considérons la section sur la méthodologie du rapport archéologique comme une trace de

processus textuelle où l'archéologue décrit le processus suivi par l'équipe au cours de son travail. Cette section peut apparaître sous des titres légèrement différents - méthodologie, méthodologie d'enquête, méthodologie d'excavation, méthodologie d'évaluation - mais elle a le même objectif. Des exemples de rapports archéologiques se trouvent à (Archaeopress 2014), le dépôt public britannique. Nous avons validé cette technique à l'aide d'un fragment extrait de la section méthodologie du rapport « Land at Station Road Honeybourne Worcestershire Archaeological Excavation ». Dans ce cas la section sur la méthodologie décrit les fouilles et toutes les autres activités connexes, comme l'analyse des découvertes archéologiques, leur enregistrement et leur documentation. Par conséquent, une section de méthodologie est une trace de processus qui saisit les activités réalisées pendant le processus et leur commande.

Nous avons remarqué en analysant manuellement les rapports archéologiques que parfois la section de méthodologie contenait une description du même processus (l'excavation), mais pour différentes zones. Normalement, le texte faisant référence à chaque instance de processus (l'excavation de chaque zone distincte) devrait être identifié. Chaque instance de processus devrait être exploitée séparément et tous les modèles d'instance de processus résultants devraient être agrégés pour obtenir le modèle de processus final. Bien que nous reconnaissions cette situation et que nous ayons identifié les exigences de la solution finale plus générale, nous avons décidé de sélectionner parmi les rapports archéologiques seulement ceux dont les méthodologies contenaient un seul exemple de processus. Dans la Figure 87, un fragment de la méthodologie archéologique est présenté avant et après le nettoyage.

1.14 The archaeological works comprised the mechanical removal of non-archaeologically significant soils, under constant archaeological supervision, using a toothless ditching bucket. The machining ceased when the natural substrate was revealed. All archaeological features were recorded in plan using a Leica 1200 series SmartRover GPS and surveyed in accordance with CA Technical Manual 4 Survey Manual (2012). 1.15 (...) no deposits were identified that required sampling. (...)
The archaeological works comprised the mechanical removal of non_archaeologically significant soils , under constant archaeological supervision , using a toothless ditching bucket . The machining ceased when the natural substrate was revealed . All archaeological features were recorded in plan using a Leica 1200 series SmartRover GPS and surveyed in accordance with CA Technical Manual 4 Survey Manual .

Figure 87. Exemple d'un fragment de méthodologie – Avant et après le nettoyage

Au début, le texte est traité et nettoyé. Plus précisément, les mesures suivantes sont prises :

- Supprimer « 1.14 » et « 1.15 », la numérotation précédant chaque paragraphe ;
- Remplacer « - » par « _ » dans « non-archaeologically » ;
- Écrire chaque phrase sur une ligne distincte ;
- Préfixer tous les signes de ponctuation avec un espace ;
- Supprimer le texte entre parenthèses « (2012) » parce qu'il ne contient aucun verbe ;
- Supprimer la phrase "no deposits were identified that required sampling" parce qu'elle contient la négation sous forme de « no » + nom. Cependant, conservez la phrase avec "non-archaeologically" parce que la négation sous forme d'adverbe « no » + n'induit pas une négation dans la sémantique de la phrase entière.

De plus, l'objectif est de découvrir les logs d'activités. Au cours de la présélection initiale des sections sur la méthodologie, nous avons remarqué que les activités étaient introduites la plupart du temps par des verbes et, dans quelques rares cas, par des noms dérivés des verbes correspondants (p. ex., « the removal of spoil »). Une activité par définition implique que quelqu'un fait quelque chose. Par conséquent, deux parties composent une activité : le verbe et son(ses) objet(s). La propriété du verbe de se lier à des objets est appelée transitivité.

Les travaux existants dans le traitement du langage naturel permettant l'identification du verbe et de ses objets dans le texte sont les parseurs. Un analyseur de langage naturel est capable d'analyser grammaticalement la structure d'une phrase et de produire un treebank, représentation d'une phrase en tant qu'arbre avec des annotations à différents niveaux : niveau de clause, niveau de phrase et niveau de mot (Marcus 1993). Au niveau de la clause, par exemple, la balise S marque une clause déclarative simple tandis que la balise SBAR marque une clause introduite par une conjonction subordinatrice. Au niveau de la phrase, les plus intéressants pour notre objectif sont NP - qui marque une phrase de nom, VP - qui marque une phrase de verbe et ADJP - qui marque une phrase adjective. Enfin, au niveau du mot, chaque mot est marqué d'une balise de parole : NN pour le nom singulier, NNS pour le nom pluriel, VB pour le verbe, VBG pour le verbe en forme « ing », VBN pour le verbe en forme de participe passé, etc. Une liste complète de toutes les balises se trouve dans (Ling 2015). Les arbres sont présentés dans la Figure 88.

Phrase 1	Phrase 2	Phrase 3
(ROOT (S (NP (DT The) (JJ archaeological) (NNS works)) (VP **(VBD comprised)** (NP (NP (DT the) (JJ mechanical) (NN removal)) (PP (IN of) (NP (ADJP (RB nonarchaeologically) (JJ significant)) (NNS soils)))) (, ,) (PP (IN under) (NP (JJ constant (JJ archaeological) (NN supervision))) (, ,) (S (VP **(VBG using)** constant) (JJ (NN (NP (DT a) (NN toothless) (**VBG ditching)** (NN bucket))))) (. .)))	(ROOT (S (NP (DT The) (NN machining)) (VP **(VBD ceased)** (SBAR (WHADVP (WRB when)) (S (NP (DT the) (JJ natural) (NN substrate)) (VP (VBD was) (VP (VBN revealed))))))) (. .)))	(ROOT (S (NP (DT All) (JJ archaeological) (NNS features)) (VP (VBD were) (VP (VP **(VBN recorded)** (PP (IN in) (NP (NN plan))) (S (VP **(VBG using)** (NP (DT a) (NNP Leica) (NNP 1200) (NN series) (NNP SmartRover) (NNP GPS))))) (CC and) (VP **(VBN surveyed)** (PP (IN in) (NP (NN accordance))) (PP (IN with) (NP (NNP CA) (NNP Technical) (NNP Manual) (CD 4) (NNP Survey) (NNP Manual)))))) (. .)))

Figure 88. Treebank pour le fragment de texte

Tous les sous-arbres VP et ADJP sont vérifiés s'ils contiennent au moins un verbe transitif. Pour identifier les verbes, nous extrayons les valeurs des feuilles des sous-arbres VB, VBD, VBN et VBG. Les valeurs « was » et « were » ne sont pas considérées parce que le verbe « be », bien que transitif, n'est pas un verbe d'action et, dans cette situation, il a un rôle auxiliaire dans la voix passive.

Ensuite, on découvre les objets de chaque verbe en vérifiant si le sous-arbre qui introduit le verbe a, comme frère droit, un sous-arbre NP. C'est le cas pour : Phrase 1 - « (VBD comprised) », « (VBG using) »; Phrase 3 - « (VBG using) ». Pour tous les autres, il est vérifié si leur sous-arbre VP ou ADJP correspondant ou l'un de leurs ancêtres a un sous-arbre NP comme frère gauche. La première clause NP trouvée est prise. De plus, si un noeud de type S est rencontré, la recherche s'arrête. C'est le cas pour : Phrase 2 - « (VBD ceased) », « (VBN revealed) »; Phrase 3 - « (VBN recorded) », « (VBN surveyed) ». Pour « (VBG ditching) » dans la phrase 1, aucun objet n'est trouvé selon la méthode définie, par conséquent, il n'est pas considéré comme une activité. Les verbes et les noms dans les éléments NP sont extraits et lemmatisés, obtenant les activités dans la Figure 89.

Phrase 1	Phrase 2	Phrase 3
1.comprise removal_of_soil 2.use toothless_bucket	3.cease machining 4.reveal substrate	5.record feature 6.use leica_series_smartover_gps 7.survey feature

Figure 89. Activités extraites du fragment

Enfin, nous découvrons les relations entre les activités en appliquant un ensemble de règles à partir d'une base de connaissances que nous définissons. Les symboles spéciaux que nous utilisons et le processus final sont expliqués dans la Figure 90.

Symbole	Utilisation	Exemple
-	Pour les noms d'activité : lorsque l'objet est composé de plusieurs noms ou lorsque le verbe a une particule	area_of_trench carry_out
->	Pour la relation d'activité : lorsqu'une activité suit une autre activité (séquence)	excavate trench -> inspect soil
||	Pour la relation d'activité : lorsque deux ou plusieurs activités sont exécutées dans le même temps ou se chevauchent (parallélisme)	excavate trench || collectfind
x	Pour la relation d'activité : lorsqu'il y a une décision entre deux ou plusieurs activités (exclusion mutuelle)	take photograph x draw plan
()	Pour la relation d'activité : en influençant la préséance des relations	(excavate trench || collect find) -> draw plan

Figure 90. Symboles dans la représentation du processus

Les phrases sont transformées : (1) en remplaçant les verbes des activités par leurs balises ; (2) en conservant les structures clés comme prépositions, conjonctions, ponctuations ; et (3) en remplaçant tout le reste par l'espace réservé « ... ». Les résultats du prétraitement sont présentés à la première ligne de chaque phrase de la Figure 91. Les règles appliquées sont présentées à la deuxième ligne et le résultat après application des règles à la troisième ligne. L'instance du processus final est présentée à la dernière ligne.

Phrase 1	(S...1.VBD...,...,(S2.VBG....))
	"1.VB/VBD/VBN ... 2.VBG " => 1||2
	(1||2)
Phrase 2	(S... 3.VBD when (S ... 4.VBN .))
	"1.VB/VBD/VBD ... when... 2.VB/VBD/VBN" => 2->1
	4->3
Phrase 3	(S...5.VBN...(S6.VBG...)and7.VBN....)
	"1.VB/VBD/VBN ... 2.VBG " => 1||2 "1.VB/VBD/VBN/VBG ... and ... 2.VB/VBD/VBN/VBG" => 1->2
	(5||6) -> 7
(1.comprise removal_of_soil || 2.use toothless_bucket) -> 4.reveal substrate -> 3.cease machining -> (5.record feature || 6.use leica_series_smartover_gps) -> 7.survey feature	

Figure 91. Instance du processus découvert dans le frargment

Lorsque nous trouvons le motif « , ... », entre deux verbes dans le texte, nous le considérons comme une explication, une addition ou un détail. C'est la raison pour laquelle nous le remplaçons par « ... » dans l'Énoncé 1 avant d'appliquer la règle elle-même.

Les parenthèses et la balise S sont ignorées lors de l'application des règles. Nous les conservons néanmoins car elles influencent le regroupement des activités. Dans les phrase 1 et Phrase 3, les activités qui sont extraites des clauses dépendantes seront regroupées : (1||2) et (5||6). De plus, les parenthèses et la balise S nous permettent de décider de la paire de verbes pour laquelle les règles sont cochées. L'appariement par défaut prend deux verbes consécutifs tels qu'ils apparaissent dans la phrase. Dans la phrase 3, la deuxième règle est appliquée à 5.VBN et 7.VBN à la place en raison du raisonnement sur les clauses. Pour cette illustration, nous n'avons pas de verbe avec plusieurs objets. Dans ce cas, l'activité aurait été divisée en plusieurs activités parallèles ou mutuelles exclusives, qui sont également regroupées.

En utilisant TextProcessMiner sur le texte méthodologique du cas archéologique, nous avons obtenu 34 activités. Un fragment du journal est présenté à la Figure 92. En interrogeant l'auteur du document, nous avons identifié 4 faux positifs, 3 activités avec des noms erronés et 1 activité manquante.

19. record characteristic	24. find deposit	29. recover topography_of_surface
20. record position_of_context	25. draw section	30. take cloud_of_point
21. photograph context	26. show shape_of_cut	31. keep find
22. excavate context	27. record sequence	32. take sample_of_occupation_layer
23. use appropriate_tool	28. reconstruct section	33. require it

Figure 92. Fragment du log découvert

Ce log d'activité peut ensuite servir à faire de la fouille de processus classique comme dans l'exemple illustré à la Figure 92 ou servir de point d'entrée à des techniques de fouille de processus intentionnels.

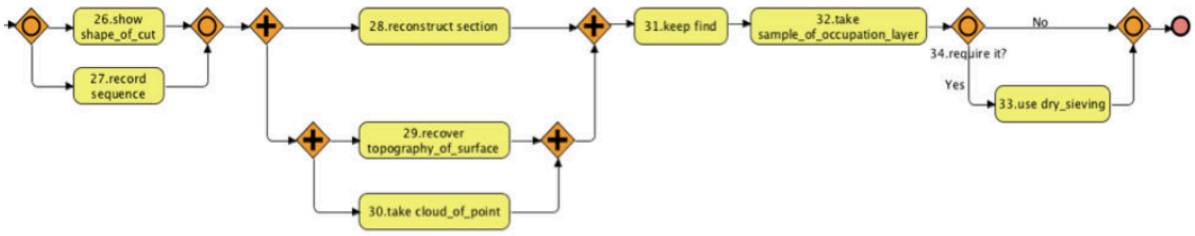


Figure 93. Fragment du modèle de processus découvert

V.2. Annexe 2 : Génération de logs à partir de documents de conversations - l'approche GenerateEventLog

Il est également possible de générer des logs à partir de documents de conversation. Il n'y a qu'un autre travail connu qui utilise la fouille de processus pour analyser les conversations (Wang 2011). La première différence avec la solution ci-dessous est que les classes utilisées

pour annoter leur corpus sont adaptées de (Kim 2010), inspirées d'actes de dialogue et adaptées aux forums de questions et réponses. De plus, l'unité d'annotation est le tour et la stratégie d'annotation est une étiquette unique. Dans cette situation, la correspondance des traces de processus sur les fils de conversation est le choix naturel (Wang 2014). Nous présentons ici une nouvelle façon de modéliser les conversations avec les modèles de processus, comme des séquences de processus internes sur les intentions de discours. Ce travail est basé sur la thèse d'Elena Epure (Epure 2018c) (Epure 2018a) (Epure 2017b) (Epure 2018b) et utilise une taxonomie d'intentions de discours présenté dans (Compagno 2018).

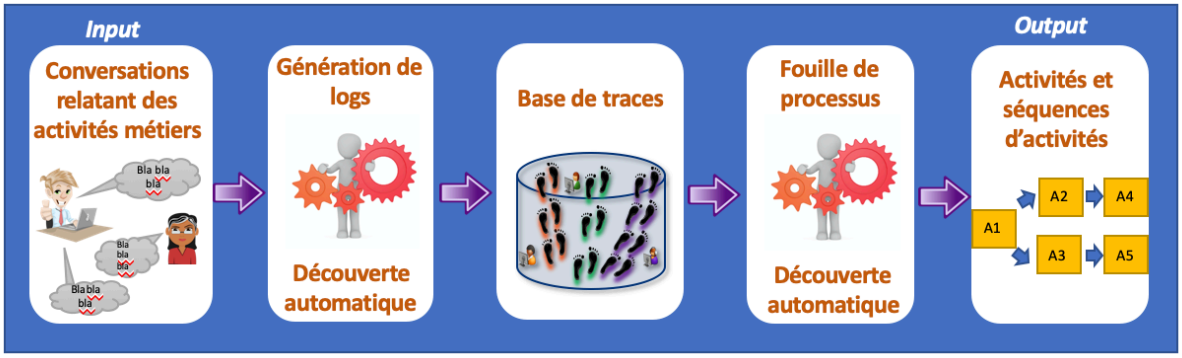


Figure 94. Entrées-Sorties de la construction de logs d'activités et de processus à partir de conversations

La première chose est de déterminer quelle unité de conversation (par exemple un tour ou une phrase) est considérée comme un événement. Le choix des événements influe sur le résultat et l'interprétation du modèle. L'objectif ici est de révéler les relations entre les intentions de parole. Par conséquent, les intentions de discours sont le choix le plus approprié pour les activités dans l'application de la fouille de processus. Les intentions de parole sont utilisées pour caractériser les énoncés individuels. Considérant qu'un événement est interprété comme une occurrence d'une activité, un parallèle similaire peut être établi entre les déclarations et les actes de parole : une déclaration fait ressortir une intention de parole (ou plus).

Par conséquent, les classes d'intention de parole ou les types d'acte de parole sont indiquées sur les activités de processus et les énoncés sont associés aux événements. Appliquer de la fouille de processus sur ces données est équivalent à identifier des modèles récurrents d'intentions de discours dans les messages ou les conversations. Ces régularités de discours sont conceptualisées à travers les intentions de discours proposées et leurs relations exprimées comme relations dans le processus. Par exemple, cette approche pourrait permettre l'identification automatique de paires de contiguïtés fréquentes dans les dialogues en ligne (questions et réponses). Cependant, plusieurs aspects importants de la construction des logs doivent encore être abordés : qu'est-ce qu'une trace de processus devrait être et comment assurer la relation entre l'événement et l'activité, compte tenu du fait que les énoncés peuvent contenir plusieurs étiquettes ; par conséquent, un événement est lié à plusieurs activités en même temps.

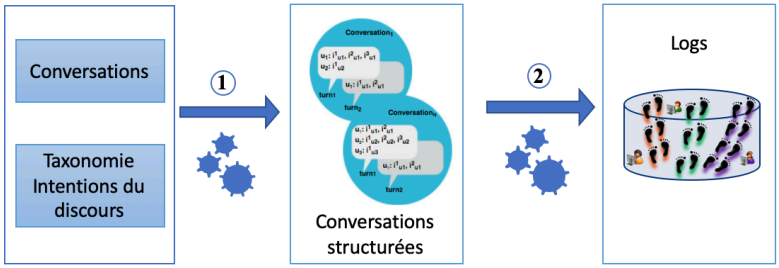


Figure 95. Approche GenerateEventLog

Les conversations asynchrones sont souvent composées de fils simultanés où différents utilisateurs peuvent répondre au même message initial. Chacune de ces réponses peut se développer dans un fil individuel, donnant une structure arborescente à toute la conversation. Une conversation ne peut pas être transformée directement en une trace car il n'y a pas d'échange linéaire de tours (qui pourrait être transformé en un flux séquentiel d'événements). Cependant, les fils de conversation sont linéaires et nous supposons plutôt que chaque fil correspond à une trace de processus.

Chaque tour individuel est composé d'une séquence d'énoncés et, par souci de simplicité, nous supposons que chaque énoncé est étiqueté avec un seul acte de parole. Par exemple, en considérant que $Conversation_1$ et $Conversation_N$ dans la Figure 95 ont chacun un fil unique et que chaque énoncé est caractérisé par la première intention de parole seulement, alors deux traces de processus sont générées $\tau_1 = \{e_{C1\text{-}t1\text{-}u1}, e_{C1\text{-}t1\text{-}u2}, e_{C1\text{-}t2\text{-}u1}\}$ et $\tau_N = \{e_{CN\text{-}t1\text{-}u1}, e_{CN\text{-}t1\text{-}u2}, e_{CN\text{-}t1\text{-}u3}, e_{CN\text{-}t2\text{-}u1}\}$, où $e_{Cj\text{-}tk\text{-}ul}$ est un événement associé à la première intention de parole u_l du tour t_k dans la conversation C_j. Si toutes les séquences présentées dans ces deux traces sont correctes, les séquences entre deux tours différents ne sont pas complètes. Entre deux tours consécutifs, seule la séquence allant du dernier énoncé du premier au premier énoncé du second est capturée (par exemple $e_{C1\text{-}t1\text{-}u2}$,$e_{C1\text{-}t2\text{-}u1}$ in τ_1). Cependant, d'un point de vue linguistique, les conséquences directes devraient se situer entre les intentions de parole de tous les énoncés du premier tour et les intentions de parole de tous les énoncés du prochain tour, puisque les tours entiers forment une séquence. Dans une certaine mesure, ces relations peuvent encore être capturées comme des séquences indirectes répétitives. Néanmoins, celles-ci sont moins prioritaires par rapport aux séquences directes, conduisant à un traitement inégal des énoncés dans les algorithmes de fouille de processus. De plus, les modèles de processus générés de cette façon saisiraient en réalité des relations mixtes entre les intentions de parole : dans un tour (p. ex., $e_{C1\text{-}t1\text{-}u1}$, $e_{C1\text{-}t1\text{-}u2}$) ou entre les tours (p. ex., $e_{C1\text{-}t1\text{-}u2}$, $e_{C1\text{-}t2\text{-}u1}$), qui semblent équivalents mais qui ne le sont pas et qui engendrent un biais potentiel pour l'étude des conversations.

On pourrait penser que, dans cette situation, on génère plutôt une trace pour chaque combinaison unique d'intentions de parole (p. ex., pour C_1, il y aurait deux traces de processus $\tau_{C1\text{-}1} = \{e_{C1\text{-}t1\text{-}u1}, e_{C1\text{-}t2\text{-}u1}\}$ et $\tau_{C1\text{-}2} = \{e_{C1\text{-}t1\text{-}u2}, e_{C1\text{-}t2\text{-}u1}\}$). Cependant, la complexité augmente d'autant : en supposant que m tourne avec une moyenne de l déclarations par tour, alors, juste pour une conversation, des traces de processus l^m sont générées. Bien sûr, en réalité, une conversation peut ne pas avoir plus de 50 tours et un tour peut ne pas avoir plus de 30 énoncés, ce qui signifie que la complexité précédente est plus élevée. Cependant, la valeur est encore très haute. En outre, la fréquences des activités est une donnée importante pour les techniques de fouille de processus. Par conséquent, l'utilisation de cette stratégie pour la

génération de logs augmente les fréquences liées aux tours plus près de la racine dans l'arbre de conversation. Ces tours sont communs à plusieurs fils simultanés : par exemple, chaque prononciation d'un tour étant la racine dans l'arbre de conversation serait répétée l^{m-1} fois.

L'approche choisie est de "couper" les arbres de conversation en niveaux situés à une distance égale des racines et de considérer une trace de processus comme un tour. En conséquence, la technique de fouille génère d'abord des modèles de processus conversationnels à tour de rôle. Deuxièmement, pour obtenir une perspective globale des conversations, ces modèles de processus sont placés dans une séquence, d'abord le modèle obtenu avec les messages initiateurs, ensuite le modèle généré à partir des commentaires de premier niveau et ainsi de suite. Cette approche est vaste parce qu'elle révèle à la fois des modèles de processus de construction de tour et des connaissances sur les conversations en général. Cependant, des limites sont également identifiées : les relations entre les intentions de parole d'un tour à l'autre ne sont supposées être que des dépendances potentielles et aucune information sur la signification de la relation ou sur des intentions de parole interreliées spécifiques d'un tour à l'autre (par exemple directes et engagées) n'est disponible. Néanmoins, cette limitation est acceptée comme un compromis afin de bien dissocier les relations entre les intentions de parole.

Jusqu'à présent, la méthode pour obtenir les logs est la suivante : les intentions de discours sont des activités de processus ; chaque énoncé, automatiquement annoté avec une intention de discours, représente un événement; la séquence des énoncés dans un tour est transformée en une séquence d'événements et représente une trace de processus; toutes les traces de processus générées pour chaque tour dans le corpus, appartenant au même niveau dans l'arbre de conversation, composent un log. Enfin, des techniques de fouille de processus sont appliquées sur ces logs et des modèles visuels de stratégies de processus pour construire des tours sont obtenus. Cependant, un défi supplémentaire existe en raison de l'annotation multi-étiquettes.

Pour mieux exposer les défis posés par l'annotation multi-étiquettes dans la génération de logs, commençons par l'exemple suivant. Un énoncé est annoté avec *assert* et *suggest*. Ceci pourrait être interprété comme : a) l'énoncé est à la fois affirmation et suggestion (concurrence), ou b) l'énoncé est affirmation ou suggestion (décision), ou c) l'énoncé contient d'abord une affirmation puis une suggestion ou vice-versa (séquence, possible dans les énoncés complexes). Une approche de discrimination entre ces interprétations aurait été de demander aux annotateurs d'indiquer explicitement laquelle des situations susmentionnées s'appliquait. Néanmoins, l'annotation manuelle aurait exigé plus d'efforts, alors que l'objectif était de concevoir une procédure aussi simple que possible. En outre, des algorithmes d'apprentissage automatique dédiés auraient été nécessaires pour identifier automatiquement ces relations. De plus, comme les énoncés correspondent à de multiples intentions de discours, une correspondance individuelle entre les énoncés et les événements est remise en question. En fait, il semble plus correct de transformer un énoncé en plusieurs événements, un pour chaque intention de discours associée. Cependant, on pose des hypothèses, premièrement, sur la nature des relations—qu'il s'agit toujours de séquences directes et, deuxièmement, sur l'ordre des activités dans une relation—quelle intention de parole des trois étiquettes devrait être considérée comme le premier, le deuxième ou le troisième événement?

Considérons encore le post suivant avec les intentions de discours associées par la prononciation entre parenthèses : u_1 : "Bonjour, j'ai de grandes nouvelles!" (*rejoice*, *greet*) u_2 : "Vous pouvez maintenant vous abonner à ma chaîne Youtube pour de nouvelles vidéos sur l'abandon du tabagisme" (*assert*, *request*). Si les relations entre les intentions de parole de chaque énoncé sont manuellement déduites, alors *greet* et *rejoice* sont dans l'ordre direct, tandis que *assert* et *request* sont concomitantes. Cependant, un log pourrait contenir simplement des séquences d'événements groupés dans des traces de processus ; par conséquent, des relations plus complexes ne peuvent pas être injectées.

Une solution heuristique est créée afin d'assurer une correspondance individuelle entre les énoncés et les événements. La stratégie est de choisir au hasard l'une des intentions du discours d'énonciation, en donnant moins de priorité. Cette stratégie vise à obtenir la cartographie la moins biaisée. Ainsi, il maintient une haute précision de ce qui pourrait être observé si les conversations étaient interprétées manuellement. De plus, en suivant cette stratégie, les séquences directes dans le log sont correctes parce que toute intention de parole d'un énoncé est suivie de toute intention de parole de l'énoncé suivant (par exemple, *greet* et *rejoice* sont suivies par *assert* et *request* dans l'exemple précédent). Cependant, une limitation existe également : certaines informations connues sont rejetées, ce qui donne une représentation sous-maximale. Ici, l'exhaustivité n'est pas un critère indispensable dans le processus d'annotation manuelle, contrairement à l'exactitude, car les annotateurs humains ne sont pas tenus de trouver toutes les intentions de parole possibles réalisées dans un énoncé, mais les plus évidentes. L'algorithme permettant de générer des logs à partir de conversations asynchrones annotées est présenté dans l'algorithme présenté à la Figure 96.

Pour N conversations, avec chaque conversation ayant un maximum de tours T et chaque tour avec un maximum de prononciations U, la complexité de l'algorithme est O(NTU). De plus, on suppose que la structure de données de l'arbre de conversation tient à jour des listes de tours par niveau. Par conséquent, obtenir la profondeur maximale de tous les arbres est O(N) et obtenir les tours à un certain niveau est O(1).

D'autres stratégies pour créer des logs sont proposées. La première conduit à une représentation séquentielle précise, semblable à celle présentée dans la section précédente. L'énoncé est cartographié sur un seul événement, mais l'activité est la concaténation de ses intentions de discours dans l'ordre alphabétique. L'avantage de cette approche est que toutes les informations connues sont exploitées. Néanmoins, même si les étiquettes sont classées par ordre alphabétique, le nombre d'activités possibles dans le processus est très élevé. Pour une taxonomie d'intention proposée avec 18 classes, le nombre d'activités possibles atteint des centaines de possibilités, rendant le modèle de processus irréalisable pour l'exploration visuelle. Si le but de l'analyse est de capturer autant d'informations que possible et que le nombre d'intentions possibles est plutôt faible (ou réduit en utilisant un paramètre de coupe élevé dans le filtrage d'activité), alors cette stratégie peut être utilisée.

Cependant, un autre aspect doit être pris en compte dans l'analyse des modèles. Deux activités, telles que "assert greet" et "assert greet rejoice", devraient être des activités séparées, bien que, pragmatiquement, elles se chevauchent. Dans ce cas, l'application d'un groupement d'activités basé sur leurs étiquettes avant l'analyse du modèle ou l'utilisation

d'une technique d'extraction de processus qui prend déjà en charge le groupement de nœuds peut être envisagée.

Algorithm Generate event logs from a list of conversation trees, one file per tree level.

Require:

convTrees—a list of conversation trees; each tree node stores a linked list of *utterances* and its *level* in the conversation; each utterance object has a property to get its *labels*.
outputFolder—the folder where the event logs are serialized to files.

procedure GENERATEEVENTLOGS(*convTrees*, *outFolder*)
    maxTreeDepth ← GETMAXDEPTH(*convTrees*) ▷ get the maximum depth of the trees.
    for each *level* = 1 to *maxTreeDepth* **do**
        eventLog ← GENERATEEVENTLOG(*convTrees*, *level*)
        SAVEEVENTLOG(*eventLog*, *outFolder*) ▷ serialize event log object to csv file.

function GENERATEEVENTLOG(*convTrees*, *level*)
    eventLog ← []
    for each *convTree* ∈ *convTrees* **do**
        turns ← GETTURNS(*convTree*, *level*) ▷ return the turns found at the specified level in the tree.
        for each *turn* ∈ *turns* **do**
            pTrace ← GENERATEPROCESSTRACE(*turn*)
            add *pTrace* to *eventLog*
    return *eventLog*

function GENERATEPROCESSTRACE(*turn*)
    pTrace ← []
    traceId = GUIDGENERATOR( ) ▷ generate a global unique identifier.
    for each *utterance* ∈ *turn.utterances* **do**
        event ← GENERATEEVENT(*utterance.labels*, *traceId*)
        add *event* to *pTrace*
    return *pTrace*

function GENERATEEVENT(*labels*, *traceId*)
    activity ← GETACTIVITY(*labels*)
    eventId = GUIDGENERATOR( ) ▷ generate a global unique identifier.
    return (*traceId*, *activity*, *eventId*)

function GETACTIVITY(*labels*)
    if *labels.length* = 1 **then**
        activity ← labels[1]
    else
        if "assert" ∈ *labels* **then**
            remove "assert" from *labels*
        activity ← RANDOM(*labels*) ▷ select a random element from the list.
    return *activity*

Figure 96. Algorithme GenerateEventLog

Les stratégies qui correspondent à un énoncé de plusieurs événements dans le log sont décrites. Étant donné que toutes ces données introduisent certaines hypothèses concernant les relations entre les événements, il convient de veiller à interpréter les modèles de processus. L'une consiste simplement à transformer les étiquettes d'énonciation en événements consécutifs en tenant compte de l'ordre des étiquettes comme dans l'annotation ou par ordre alphabétique. Ainsi, les relations sont capturées entre les intentions de parole, mais la direction de la relation n'est pas nécessairement correcte. Une deuxième stratégie est dérivée de la précédente : un ordre "correct" possible des événements pourrait être créé en utilisant le contexte et la connaissance pragmatique. Par exemple, un énoncé marqué avec (assert, greet) au début d'un post pourrait être transformé en greet $>_L$ assert. Cependant,

créer ces règles est un biais et formuler des hypothèses sur les relations d'intention de discours à partir de modèles de processus dérivés de ces stratégies est moins conseillé.

Illustration

L'étude de cas consistait en des observations directes et indirectes avec un chercheur en linguistique. Les logs d'événements pour les trois premiers niveaux de conversations ont été générés à l'avance à partir du même corpus Reddit.

Le chercheur a analysé les conversations, afin d'en extraire des modèles de processus intéressants et de comparer les modèles lorsqu'il utilise des intentions de parole ou des types d'actes de parole comme activités. Le chercheur a exprimé son désir d'avoir accès à deux logs supplémentaires : l'un correspondant à tous les tours existant dans le corpus et l'autre correspondant à tous les commentaires du corpus. Ces documents ont été fournis et le chercheur a poursuivi l'analyse. Comme la deuxième partie de l'étude de cas consistait en des observations indirectes, les résultats de l'analyse des conversations ont principalement été analysés pour évaluer la pertinence. En outre, une discussion ouverte finale a été organisée avec le chercheur pour en savoir plus sur la pertinence perçue et l'utilité de l'approche pour l'analyse de conversation.

L'algorithme permettant de générer des logs à partir de conversations annotées a été implémenté en tant que script Python. L'entrée requise est un fichier de valeurs séparées par des virgules (csv) avec chaque ligne contenant un énoncé, l'étiquette d'énoncé (jusqu'à trois) et l'identificateur unique du tour auquel l'énoncé appartient. Ainsi, l'approche complète est automatique.

La découverte des relations entre les intentions de parole se réalise en deux étapes. Premièrement, les conversations asynchrones sont annotées avec des intentions de discours. Deuxièmement, les logs saisissant les aspects pertinents du comportement conversationnel sont générés à partir des conversations annotées. Bien que l'évaluation ait été essentiellement qualitative, les résultats étaient encourageants. Des connaissances déjà attendues sur l'utilisation des forums de santé ont été découvertes, ce qui a validé l'approche pour générer des logs et découvrir des modèles de processus d'intentions de discours avec la fouille de processus.

L'étude avec un chercheur en linguistique a validé la pertinence des résultats pour l'analyse des conversations. Les intentions détaillées du discours ont donné lieu à des observations plus riches et plus précises, qui auraient pu être incomplètes ou trompeuses en utilisant uniquement les types d'actes de parole comme activités. Enfin, le chercheur linguistique a exprimé son souhait d'analyser aussi les processus transversaux, afin d'avoir des informations encore plus pertinentes pour l'analyse des conversations.

V.3. Annexe 3 : Enrichir les logs d'évènements avec des informations de contexte - l'approche ontologique

Il est possible d'enrichir les logs d'activités avec des données de contexte (Figure 97), la difficulté étant de relier correctement les données récupérées par les senseurs pour les associer de manière pertinente aux activités présentes dans les logs.

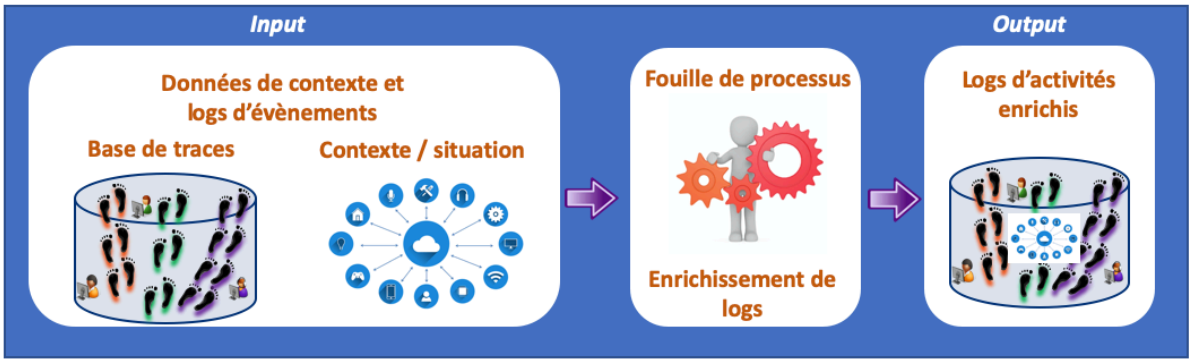


Figure 97. Entrées-Sorties de l'enrichissement de logs d'activités avec des données de contexte

Une approche possible est d'utiliser des ontologies pour combiner les données des capteurs et les logs d'activité. Tout d'abord, toutes les données acquises à partir des capteurs et toutes les activités du log peuvent être interprétées sémantiquement dans une ontologie spécifique au domaine. Deuxièmement, tous les liens entre les concepts ontologiques (en particulier les liens entre les éléments de capteurs et les éléments d'activités) doivent être définis. Par exemple, une valeur spécifique de données de capteur peut déclencher le début ou la fin d'une activité.

Pour combiner les données des capteurs et les logs d'activité à l'aide de l'ontologie de domaine, nous devons procéder en trois étapes : (1) créer l'ontologie des capteurs à partir des données des capteurs, (2) créer l'ontologie d'activité à partir du journal d'événements et (3) affiner l'ontologie du domaine en créant les liens entre les deux ontologies précédentes. Cette approche est étudiée dans le cadre de la thèse de Ramona Elali (Elali 2022a) (Elali 2021).

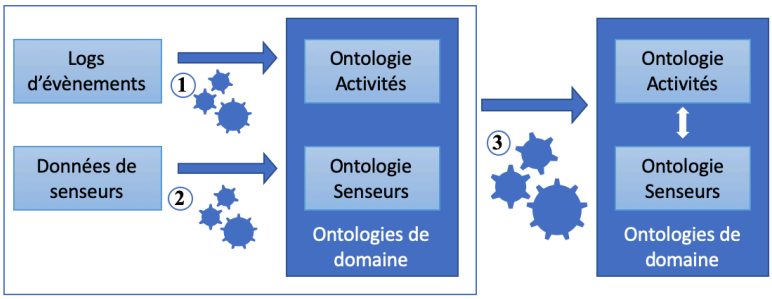


Figure 98. Approche Ontologique

Si le log et les données du capteur sont liés à des ontologies existantes, nous pouvons passer directement à l'étape 3 pour identifier les liens entre elles.

Nous avons pour l'instant utilisé deux techniques pour essayer de définir ces liens : l'horodatage et les règles d'associations.

Horodatage. L'horodatage est considéré comme le seul attribut permettant de trouver la relation entre l'événement d'activité et le capteur. Nous avons développé un algorithme Python qui prend comme entrée le journal des événements et le journal du capteur. Les deux journaux doivent être triés en fonction du champ d'horodatage. Cet algorithme est composé de deux étapes principales. Dans la première étape, le début et l'achèvement de chaque activité doivent être identifiés. Ensuite, le champ horodatage est utilisé pour faire pivoter le log en fonction du début et de la fin de chaque activité. Ensuite, dans la deuxième étape, une jointure a été faite entre les deux logs en utilisant le champ de l'horodatage pour créer le lien entre les données du capteur et les données de l'événement.

Règles d'association. L'extraction de règles d'association est la technique de recherche de relations entre différents éléments. Il est couramment utilisé dans l'analyse des paniers de consommation, où il identifie les articles fréquemment achetés ensemble à des fins de marketing. Il s'agit de trouver des tendances, des associations, des structures causales et des corrélations fréquentes dans différents ensembles de données existant dans différents types de bases de données, comme les bases de données transactionnelles, les bases de données relationnelles ou toute autre forme de bases de données (Zaki 2014) (Sunita 2012). Le but de l'extraction de règles d'association est de trouver les règles qui nous permettent de deviner l'apparence d'un élément spécifique après les occurrences d'autres éléments dans la transaction. C'est l'un des principaux concepts des techniques basées sur des règles d'apprentissage automatique qui découvrent des relations intéressantes entre les éléments dans de grandes bases de données en indiquant les règles des associations fréquentes qui sont composées d'un antécédent et qui en découlent (Zaki 2014). Les règles d'association permettent de découvrir les relations cachées entre les éléments qui sont fréquemment utilisés ensemble. Il est représenté par des déclarations if-then (un article antérieur et un article consécutif). Apriori (Agrawal 1994) est l'un des meilleurs algorithmes standard pour l'exploration d'objets fréquents et l'apprentissage des règles d'association. Dans ce travail, nous avons utilisé les techniques d'extraction de règles d'association puisque nous devions trouver des règles entre les capteurs et les activités. Pour appliquer l'algorithme Apriori, il faut d'abord transformer l'ensemble de données en un journal de transaction supporté par l'algorithme. Nous avons étiqueté chaque log par date de jour, puis nous avons combiné les deux logs et les avons triés en fonction du jour et de l'horodatage. Nous avons ensuite transformé le log combiné en un log de transaction supporté par Apriori. En plus du journal des transactions, Apriori a besoin de saisir plusieurs paramètres tels que le support minimum et la confiance minimum. Le support indique la fréquence à laquelle un détail est fréquent dans les transactions. La confiance indique la fréquence à laquelle la règle (si alors) est trouvée vraie.

Illustration

Dans notre expérience, nous avons utilisé l'ensemble de données fourni par le challenge BP-Meets-IoT 2021 (Koschmider 2021). Cet ensemble de données comprend deux logs d'activités de personnes en plus des registres de capteurs enregistrés pendant 21 jours consécutifs entre le 16 mars 2020 et le 6 avril 2020. En étudiant l'ensemble de données, nous avons trouvé 53 activités et 16 capteurs. Les deux registres contenaient des données sur le

bruit; certains enregistrements ont été dupliqués, en plus de certains enregistrements qui ne représentent aucun type d'activités ou de capteurs. Nous avons nettoyé l'ensemble de données du bruit en supprimant ces lignes, et nous avons obtenu 23166 enregistrements dans le journal des événements et 39304 enregistrements dans le journal des capteurs. Chaque enregistrement dans les journaux d'événements est identifié par le nom de l'activité tandis que chaque enregistrement dans les logs du capteur est identifié par le nom du capteur ; les deux logs sont catégorisés par l'horodatage et l'ID de ressource.

Nous avons défini les ontologies, de domaine et de capteurs, en étudiant les données fournies.

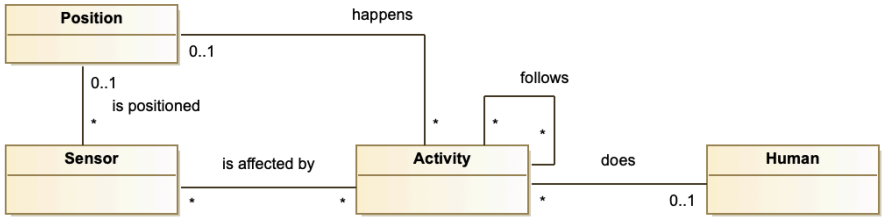


Figure 99. Liens entre les Ontologies.

Nous pouvons identifier différentes relations entre les ontologies définies (Figure 99). Chaque humain fait des activités. Les capteurs affectent les activités et peuvent être positionnés à une position spécifique. Chaque activité peut être suivie d'une ou de plusieurs activités (qui peuvent être faites séquentiellement ou non). Chaque activité peut se produire à une position particulière à la maison et être affectée par des capteurs. Chaque position peut être liée à des capteurs et des activités.

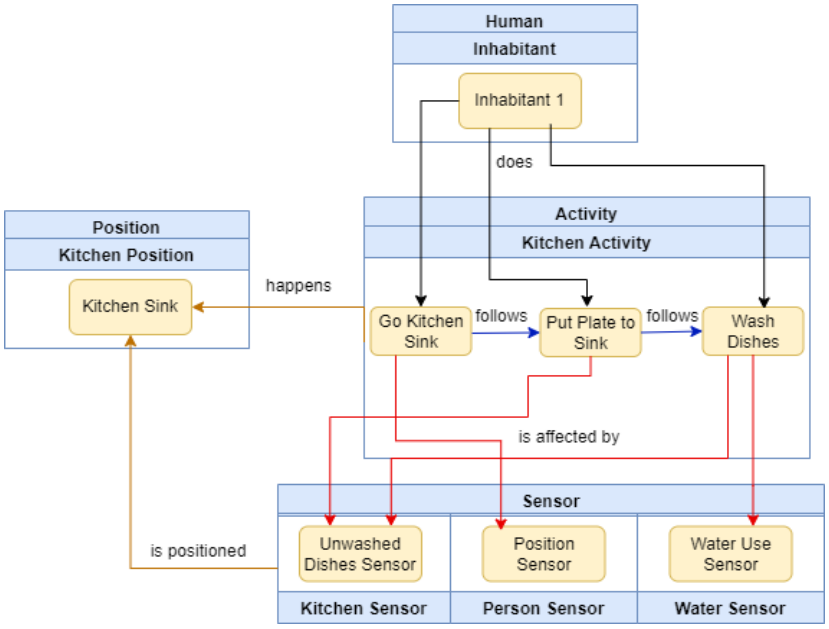


Figure 100. Exemple de liens entre les différentes ontologies définies

La Figure 100 illustre un exemple des liens entre les ontologies. Lorsque l'intention est de laver la vaisselle, l'ensemble des activités liées à cet objectif aura lieu à l'intérieur de la

cuisine ; l'habitant ira à l'évier de la cuisine, mettra l'assiette à l'intérieur de l'évier et commencera à laver la vaisselle. Dans l'intervalle, le capteur de plats non lavés augmentera sa valeur en fonction des plaques qui sont lavées. Le capteur d'utilisation de l'eau augmentera sa valeur et le capteur de position indiquera que la position humaine est à côté de l'évier de cuisine.

Cet exemple nous a permis de valider le fait qu'il était possible d'identifier les liens ontologiques entre les données des capteurs et les activités à l'aide de l'horodatage sur les données et les événements, et d'identifier les liens par des règles d'association.

Il reste encore beaucoup à faire et d'autres techniques à tester pour obtenir les relations les plus efficaces entre les données de capteurs et les activités présentes dans le log.

VI. Bibliographie

(Abad 2010) Abad Z., Sadi M., Ramsin R., Towards tool support for situational engineering of agile methodologies In APSEC 2010, Asia Pacific, 2010

(Abad 2012) Abad Z., Alipour A., Ramsin R., Enhancing Tool Support for Situational Engineering of Agile Methodologies in Eclipse, Studies in Computational Intelligence, Vol 430, pp 141-152, 2012

(Abrahamsson 2002) Abrahamsson P., Salo O., Ronkainen J., and Warsta J., Agile software development methods: Review and analysis, VTT Publications, Espoo, 2002

(Agerfalk 2003) Agerfalk, P.J.: Information systems actability: Understanding Information Technology as a Tool for Business Action and Communication. Doctoral dissertation. Dept. of Computer and Information Science, Linköping University, 2003

(Agrawal 1994) Agrawal, R., Srikant, R.: "Fast algorithms for mining association rules". In Proc. Int'l Conf. Very Large Data Bases (VLDB), pages 487–499, Sept. (1994).

(Agrawal 1998) Agrawal R., Dimitrios Gunopulos and Frank Leymann. Mining process models from workflow logs. Springer, 1998

(Alsayed 2022) Alsayed A., Deneckère R., A Sustainability Matrix for Smartphone Application, Workshop ISESL, International Conference on Advanced Information Systems Engineering, 73-85, 2022

(America 2003) America P., Rommes E., Obbink J. H., Multi-view variation modeling for scenario analysis. In PFE'03, p. 44–65, 2003

(Ammar 2013) Ammar A., Famille de Méthodes d'Analyse en Sciences Humaines et Sociales, Master Thesis, University of Paris 1 Panthéon-Sorbonne, 2013

(Amyot 2002) Amyot D., Mussbacher G., URN: Towards a New Standard for the Visual Description of Requirements, Proceeding of 3rd Int. WS on Telecommunications and beyond : the broader applicability of SDL and MSC, 2002

(Amyot 2009) Amyot D., Horkoff J., Gross D., Muss- bacher G., A lightweight GRL profile for i* modeling. In Advances in Con- ceptual Modeling-Challenging Perspectives, pages 254–264. Springer, 2009

(Anderson 2003) Anderson D., Agile management for software engineering: Applying the theory of constraints for business results, 2003

(Archaeopress 2014) Archaeopress.com, Archaeopress - British Archaeological Reports [Online]. Available: http://www.archaeopress.com/ArchaeopressShop/Public/defaultAll.as p?intro=Home&PublishedDateGT=15+Feb+2014. [Accessed: 15- Feb- 2015].

(Armenise 1993) Armenise P., Bandinelli S., Ghezzi C., Morzenti A., A survey and assessment of software process representation formalisms, Int. Journal of Software Engineering and Knowledge Engineering, vol. 3, n° 3, 1993

(Asadi 2011) Asadi M., Mohabbati B., Gašević D., Bagheri E.. Developing Families of Method- Oriented Architecture. ME 2011, pp. 168-183, IFIP AICT 351, Springer, Heidelberg, 2011

(Assar 2000) Assar S., Ben Achour C., Si-Said S., Un Modèle pour la spécification des processus d'analyse des Systèmes d'Information. In Proceedings of 18ème Congrès INFORSID, pp. 287-301, 2000

(Bachmann 2001) Bachmann F., Bass L., Managing variability in software architectures, In the 2001 Symposium on Software Reusability (SSR '01), ACM Press, p. 126–132, 2001

(Baeza-Yates 2006) Baeza-Yates R., Calderon-Benavides L., Gonzalez-Caro C., The intention behind web queries. In String processing and information retrieval, pages 98–109. Springer, 2006

(Baker 2001) Baker D., Bridges D., Hunter R., Johnson G., Krupa J., Murphy J., Sorenson K., Guidebook to decision-making methods, Developed for the Department of Energy, http://emi- web.inel.gov/Nissmg/Guidebook_2002.pdf, 2001

(Balakrishnan 1997) Balakrishnan, V. K., Graph Theory, McGraw-Hill; 1 edition February 1, 1997

(Baudry 2002) Baudry M., Vincent N., Multicriteria decision making, First annual meeting on health science and technology, Tours, France, 2002

(Beck 2000) Beck K., Extreme programming explained: embrace change, Addison-Wesley, UK, 2000

(Beck 2001) Beck K., Beedle M., van Bennekum A., Cockburn A., Cunningham W., Fowler M., Grenning J., Highsmith J., Hunt A., Jeffries R., Kern J., Marick B., Martin R. C., Mellor S., Schwaber K., Sutherland J., Thomas D., Manifesto for Agile Software Development. Agile Alliance, 2001

(Beck 2005) Beck K., Extreme programming eXplained: embrace change, 2nd ed. The XP Series. Boston, MA, USA, Addison Wesley, 2005

(Benzécri 1973) Benzécri J.-P, L'analyse des données: L'analyse des correspondances, Dunod, Paris. B, 1973

(Benzécri 1982) Benzécri J.-P, Histoire et préhistoire de l'analyse des données, Dunod, Paris, 1982

(Berge 1985) Berge C., Graphes. Gauthier-Villars, 1985

(Berkson 1944) Berkson J, Application of the Logistic Function to Bio-Assay, Journal of the American Statistical Association, vol. 39, pp. 357-365, 1944

(Bessai 2008) Bessai K., Claudepierre B., Saidani O., Nurcan S., Context-aware business process evaluation and redesign, Proceedings of BPMDS'08, 2008

(Bliss 1934) Bliss, C. I., The method of probits, Science, vol. 79, no. 2037, pp. 38–39, 1934

(Boehm 1988) Boehm B., A Spiral model of software development and enhancement, IEEE Computer, 21, 5, p. 61-72, 1988

(Bosh et al., 2001) Bosch J., Florijn G., Greefhorst D., Kuusela J., Obbink H., Pohl K. Variability issues in Software Product lines, 4 international workshop on Product Family engineering (PEE-4), Bilbao, Spain, 2001

(Bouyssou 2001) Bouyssou D., Outranking methods, In Encyclopedia of Optimization, Kluwer, 2001

(Bradley 2005) Bradley N. A., Dunlop M. D., Toward a multidisciplinary model of context to support context-aware computing, Human-Computer interaction, Lawrence Erlbaum Associates, 2005

(Bresciani 2004) Bresciani P., Perini A., Giorgini P., Giunchiglia F., Mylopoulos J., Tropos: An agent-oriented software development methodology. Autonomous Agents and Multi-Agent Systems, vol. 8, no. 3, pages 203–236, 2004

(Brinkkemper 1996) Brinkkemper S., Method engineering: engineering of information systems development method and tools, Information and Software Technology, 38:7, 1996

(Brinkkemper 1998) Brinkkemper S., Saeki M., Harmsen F. Assembly Techniques for Method Engineering. Proceedings of CAiSE 1998, pp. 381-400, LNCS 1413, Springer Verlag, 1998

(Bucher 2007) Bucher T., Klesse M., Kurpjuweit S., Winter R., Situational Method Engineering. On the Differentiation of Context and Project Type, ME 07, Vol. 244, Springer, Boston, pp. 33-48, 2007

(Buchta 2010) Buchta C., Hahsler M., arulesSequences: Mining frequent sequences, R package v.0.1-11, http://CRAN.R-project.org/package=arulesSequences, 2010

(Bunt 1997) Bunt H., Context and dialogue control, Proceedings of CONTEXT'97, 1997

(Canning 2014) Canning, J., Statistics for Humanities, Available from http://statisticsforhumanities.net, 2014

(Cayla 2008) Cayla D., Organizational Learning: A Process Between Equilibrium and Evolution, Journal of Economic Issues, 42, 2, p. 553-560, 2008

(Chakhar 2006) Chakhar S., Cartographie décisionnelle multicritère: formalisation et implémentation informatique, PhD Thesis, Université Paris Dauphine, France, 2006

(Chen 2002) Chen Z., Lin F., Liu H., Liu Y., Ma W., Wenyin L., User intention modeling in web applications using data mining. World Wide Web, vol. 5, no. 3, pages 181–191, 2002

(Cleger-Tamayo 2013) Cleger-Tamayo S., Fernández-Luna J.M., Huete J.F., Tintarev N., Being Confident about the Quality of the Predictions in Recommender Systems. In: Serdyukov, P., Braslavski, P., Kuznetsov, S.O., Kamps, J., Rüger, S., Agichtein, E., Segalovich, I., Yilmaz, E. (eds.) ECIR 2013. LNCS, vol. 7814, pp. 411-422. Springer, Heidelberg, 2013

(Clements 2001) Clements P., Northrop L.. Software Product Lines: Practices and Patterns. Addison- Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001

(Cockburn 2002) Cockburn A., Agile software development, Addison-Wesley, London, UK, 2002

(Cohn 2005) Cohn M., Agile estimating and planning, Prentice Hall editors, 2005

(Cook 1998a) Jonathan E Cook and Alexander L Wolf. Discovering models of soft- ware processes from event-based data. ACM Transactions on Software En- gineering and Methodology (TOSEM), vol. 7, no. 3, pages 215–249, 1998.

(Coutaz 2002) Coutaz J., Rey G., Recovering foundations for a theory of contextors, 4th ICCADUI, Valenciennes, France, 2002

(Chrysanthi 2012) Chrysanthi A., Papadopoulos C., Murrieta-Flores P., Thinking beyond the tool. Oxford: Archaeopress, 2012

(Compagno 2018) Compagno D., Epure E.V., Deneckère R., Salinesi C, Exploring Digital Conversation Corpora with Process Mining, International Journal of Corpus Linguistics and Pragmatics, Springer International Publishing, 2018

(Crowe 1997) Crowe T.J., Rathi K., Rolfes J.D., Applying a Taxonomy of Business Processes to Identify Reengineering Opportunities, http://www.prosci.com/rathi.htm, 1997

(Dardenne 1993) Dardenne A., Van Lamsweerde A., Fickas S., Goal- directed requirements acquisition. Science of computer programming, vol. 20, no. 1, pages 3–50, 1993

(Delorenzi 2002] Delorenzi M., Speed T., An HMM model for coiled-coil domains and a comparison with PSSM-based predictions. Bioinformatics, vol. 18, no. 4, pages 617–625, 2002

(Deneckère 1998) Deneckère R., Souveyet C., Patterns for extending an OO model with temporal features, OOIS'98, 201-218, 1998

(Deneckère 2001a) Deneckère R., Approche d'extension de méthodes fondée sur l'utilisation de composants génériques. Thèse de doctorat, Université de Paris 1-Sorbonne, 2001.

(Deneckère 2001b) Deneckère R., Souveyet C., Organising and selecting patterns in pattern languages with Process Maps, OOIS 2001, 71-80, 2001

(Deneckère 2002a) Deneckère R., Using meta-patterns to construct patterns, OOIS'02, France, 2002

(Deneckère 2002b) Deneckère R., Construction de patrons à l'aide de métapatrons, Deneckère R., L'Objet, logiciel, base de données, réseaux (RSTI série), vol 7, 4 pp. 83-105, 2002

(Deneckère 2008a) Deneckère R., Ralyte J., Kraiem N, Modélisation modulaire de méthode: le méta-modèle COMET, dans "Ingénierie des méthodes: des nouvelles tendances de développement des applications informatiques" (N. Kraïem & Y. Jamoussi (Ed.)), p.117-149, 2008

(Deneckère 2008b) Deneckère R., Iacovelli A., Kornyshova E., Souveyet C., From method fragments to method services, Proceedings of EMMSAD 2008, Montpellier, France, 2008

(Deneckère 2009) Deneckère R., Kornyshova E., Rolland C., Enhancing the guidance of the intentional model MAP: Graph theory application, Actes de la conférence Research Challenges in Information Science (RCIS'09), Fès, Maroc, pp 13-22, 2009

(Deneckère 2010a) Deneckère R., Kornyshova E., Process line configuration: An indicator-based guidance of the intentional model MAP. In Enter- prise, Business-Process and Information Systems Modeling, pages 327–339. Springer, 2010

(Deneckère 2010b) Deneckère R., Kornyshova E., La variabilité due à la sensibilité au contexte dans les processus téléologiques, Actes du congrès INFORSID 2010, Marseille, France, pp 161-176, 2010

(Deneckère 2011a) Deneckère R., Rychkova I., Nurcan S., Modeling the role variability in the MAP process model, Actes de la conférence Research Challenges in information Science (RCIS'11), Guadeloupe, France, pp 180-188 2011

(Deneckère 2011b) Deneckère R., Kornyshova E., Rychkova I., Des lignes de processus aux familles de processus. INFORSID, France. pp.283-298, 2011

(Deneckère 2011c) Deneckère R., Kornyshova E., Processus téléologique et variabilité: Utilisation de la sensibilité au contexte, Ingénierie des systèmes d'information, 16 (1), 61-88, 2011

(Deneckère 2011d) Deneckère R., Kornyshova E., Rolland C., Method Family Description and Configuration, Actes du forum ICEIS, Beijing, Chine, pp 384-387, 2011

(Deneckère 2014a) Deneckère R., Kornyshova E., Ralyte J., Famille de méthodes : la flexibilité au cœur du processus de construction de méthode. Revue des Sciences et Technologies de l'Information - Série ISI : Ingénierie des Systèmes d'Information, Lavoisier, 19/1 - 2014, pp.67-95, 2014

(Deneckère 2014b) Deneckère R., Hug C., Khodabandelou G., Salinesi S., Intentional Process Mining: Discovering and Modelling the Goals behind Processes Using Supervised Learning, International Journal of Information System Modeling and Design (IJISMD), 2014

(Deneckère 2015a) Deneckère R., Hug C., Onderstal J., Brinkkemper S., Method Association Approach: Situational construction and evaluation of an implementation method for software products, Research Challenges in Information Science (IEEE RCIS) 2015, 2015, Athens, Greece, 2015

(Deneckère 2015b) Deneckère R., Kornyshova E., Concept de Famille de Méthodes : qu'en pensent les utilisateurs ?. INFORSID, Biarritz, France. pp.49-64, 2015

(Deneckère 2016a) Deneckère R., Hug C., Jaffal A., Kirsch Pinheiro M., Le Grand B., Mazo R., Rychkova I., Context Management and Intention Mining for Adaptive Systems in Mobile Environments: From Business Process Management to Urban Games?, dans "Digital Interfaces in Situation of Mobility: Cognitive, Artistic and Game devices" (Bernard Guelton (Ed.)), Common Ground Publishing, 2016

(Deneckère 2016b) Deneckère R., Kornyshova E., Iacovelli A., Progressive Integration of Method Components: A Case of Agile IS Development Methods. RCIS: Research challenges in Information Science, Grenoble, France, 2016

(Deneckère 2016c) Deneckère R., Kornyshova E., Évaluation du concept de famille de méthodes et de son application aux méthodes agiles, Ingénierie des Systèmes d'Information 21 (2), 2016

(Deneckère 2020) Deneckère R., Rubio G., EcoSoft: Proposition of an Eco-Label for Software Sustainability, CAISE workshop ISESL, Grenoble, France, 121-132, 2020

(Deneckère 2021) Deneckère R., Kornyshova E., Hug C., A Framework for Comparative Analysis of Intention Mining Approaches, International Conference on Research Challenges in Information Science, 20-37, 2021

(Dey 2001) Dey A., Abowd G., Salber D., A conceptual framework and toolkit for supporting the rapid prototyping of context-aware applications. Human-computer Interaction, 16 2-4 (special issue on context-aware computing), p 97–166, 2001

(Dietz 2004) Dietz J.L.G., Basic Notions Regarding Business Processes and Supporting Information Systems, Proceedings of BPMDS'04, Latvia, Riga, p. 160-168, 2004

(Dowson 1988) Dowson M., Iteration in the Software Process, Proc. 9th Int. Conf. on Software Engineering, 1988

(Dowson 1993) Dowson M., Software Process Themes and Issues, IEEE Conf. on the Software Process , 1993

(Drury 2008) Drury J. L., Scott S. D., Awareness in unmanned aerial vehicle operations, International C2 journal, Geoffrey N. Hone, 2:1, 2008

(Eclipse 2013) Eclipse. Filtered UDC Data. url- http://archive.eclipse.org/projects/usagedata/, 2013.

(Elali 2021) Elali R., An Intention Mining Approach using Ontology for Contextual Recommendations, Proceedings of the Doctoral Consortium Papers Presented at the 33rd International Conference on Advanced Information Systems Engineering (CAiSE 2021), Melbourne, Australia, June 28, 2021

(Elali 2022a) Elali R., Kornyshova E., Deneckère R., Salinesi C., Domain Ontology Construction with Activity Logs and Sensors Data–Case Study of Smart Home Activities, International Conference on Advanced Information Systems Engineering, 49-60, 2022

(Elali 2022b) Elali R., Recommandations contextuelles fondées sur la fouille d'intentions et les ontologies, forum Jeunes chercheurs Jeunes Chercheuses, INFORSID, 2022

(Epure, 2013) Epure E, Intention-Mining: a solution to process participant support in process aware information systems, Master thesis, technical report 2013-020, Utrecht University, 2013

(Epure 2014a) Epure E., Hug C., Deneckère R., Salinesi C., Intention Mining: Discovering Intentional Processes from Ocean's Big Data, Ocean's Big Data Mining, Brest France, 2014

(Epure 2014b) Epure E., Hug C., Deneckère R., Brinkkemper S., What Shall I Do Next? Intention Mining for Flexible Process Enactment. 26th International Conference on Advanced Information Systems Engineering (CAiSE), Thessalonique, Greece, 2014

(Epure 2015) Epure E., Martín-Rodilla P., Hug C., Deneckère R., Salinesi C., Automatic Process Model Discovery from Textual Methodologies, IEEE Ninth International Conference on Research Challenges in Information Science (IEEE RCIS) 2015, Athens, Greece, 2015

(Epure 2016) Epure E., Ingvaldsen J., Deneckère R., Salinesi C., Process Mining for Recommender Strategies Support in News Media, IEEE Tenth International Conference on Research Challenges in Information Science (IEEE RCIS) 2016, Grenoble, France, 2016

(Epure 2017a) Epure E., Kille B., Ingvaldsen J., Deneckère R., Salinesi C., Albayrak S., Modeling the Dynamics of Online News Reading Interests, 25th Conference on User Modeling, Adaptation and Personalization, 363- 364, 2017

(Epure 2017b) Epure E., Zitnik S., Compagno D., Deneckère R., Salinesi C., Automatic analysis of online conversations as processes. Journée Analyse de données textuelles en conjonction avec EDA, Lyon, France, 2017

(Epure 2017c) Epure E., Deneckère R., Salinesi C., Analyzing Perceived Intentions of Public Health-Related Communication on Twitter Preprint, AIME 2017: Artificial Intelligence in Medicine, Jun 2017, Vienna, Austria, 2017

(Epure 2017d) Epure E., Kille B., Ingvaldsen J., Deneckère R., Salinesi C., Albayrak S., Recommending Personalized News in Short User Sessions, Proceedings of the 11th ACM Conference on Recommender Systems, 121-129, 2017

(Epure 2017e) Epure E., Deneckère R., Salinesi C., Kille B., Ingvaldsen J., Devising News Recommendation Strategies with Process Mining Support, Interdisciplinary Workshop on Recommender Systems, May 2017, Paris, France, 2017

(Epure 2018a) Epure E., Compagno D., Salinesi C., Deneckère R., Bajec M., Zitnik S, Process Models of Interrelated Speech Intentions from Online Health-related Conversations, Artificial Intelligence in Medicine, Elsevier, 2018, 91, pp.23-38, 2018

(Epure 2018b) Epure E., Deneckère R., Salinesi C., Compagno D., Bajec M., Zitnik S., Revealing Intention-centered Models of Online Conversations on Health Care , Artificial Intelligence in Medicine journal, 2018

(Epure 2018c) Epure E., Automatically Modeling Conversations as Processes of Interrelated Speech Intentions, These de doctorat, Université Paris 1 Panthéon-Sorbonne, 2018

(Etien 2003) Etien A., Deneckère R., Salinesi C., Extending methods to express change requirements, 1st International Workshop on Engineering Methods to Support Information Systems Evolution (EMSISE'03), Suisse, 2003

(Euler 1741) Euler L., Solutio problematis ad geometriam situs pertinentis, Commetarii Academiae Scientiarum Imperialis Petropolitanae 8, 128-140, 1741

(Firesmith 2002) Firesmith D.G., Henderson-Sellers B., The OPEN Process Framework: An Introduction. Addison-Wesley, London, UK, 330 p, 2002

(Friedman 1997) Friedman N., Geiger D., Goldszmidt M., Bayesian net- work classifiers. Machine learning, vol. 29, no. 2-3, pages 131–163, 1997

(Friedrich 2011) Friedrich F., Mendling J., Puhlmann F., Process Model Generation from Natural Language Text, 23rd International Conference, CAiSE 2011, London, UK, June 20-24, pp. 482- 496, 2011

(Gales 1998) Gales M., Maximum likelihood linear transformations for HMM- based speech recognition. Computer speech & language, vol. 12, no. 2, pages 75–98, 1998

(Gauthier 2022) Gauthier F., Deneckère R., Perception des Méthodes Agiles par les Développeurs Aujourd'hui, INFORSID 2022

(Ghose 2007) Ghose A., Koliadis J. , Chueng A., Process Discovery from Model and Text Artefacts, in IEEE Congress on Services, pp.167-174, 2007

(Gold 2012) Gold M., Debates in the digital humanities. Minneapolis: Univ Of Minnesota Press, 2012

(Goncalves 2010) Gonçalves C., Santoro F., Baião F., A case study on designing business processes based on collaborative and mining approaches, 14th International Conference on Computer Supported Cooperative Work in Design (CSCWD), pp. 611- 616, 2010

(Gonzalez-Perez 2011) Gonzalez-Pérez C., Parcero-Oubiña C., A Conceptual Model for Cultural Heritage Definition and Motivation in Revive the Past: Proceedings of the 39th Annual Conference on Computer Applications and Quantitative Methods in Archaeology. Amsterdam University Press. Beijing, China, pp.234-244, 2011

(Gonzalez-Perrez 2012) Gonzalez-Perez C., Martín-Rodilla P., Parcero-Oubiña C., Fábrega-Álvarez P., Güimil-Fariña A., Extending an Abstract Reference Model for Transdisciplinary Work in Cultural Heritage, Metadata and Semantics Research, Springer Berlin Heidelberg, pp. 190-201, 2012

(Gray 1992) Gray W., John B., Atwood M., The precis of Project Ernestine or an overview of a validation of GOMS. In Proceedings of the SIGCHI conference on Human factors in computing systems, pages 307–312. ACM, 1992

(Greco 2006) Greco G., Guzzo A., Ponieri L., Sacca D., Discovering expressive process models by clustering log traces, IEEE Trans. on Know. and Data Eng. , Vol. 18, n°8, p. 1010-1027, 2006

(Grosz 1997) Grosz G., Rolland C., Schwer S., Souveyet C., Plihon V., Si-Said S., Ben Achour C., Gnaho C., Modelling and Engineering the Requirements Engineering Process: An Overview of the Nature Approach, Requirements Engineering Journal, 2, p. 115-131, 1997

(Guzélian 2007) Guzélian G., Cauvet C., SO2M : Towards a service-oriented approach for method engineering, Proceeding of IKE'07, Las Vegas, Nevada, USA, 2007

(Halmans 2003) Halmans J., Communicating the variability of a software product family to customers, Software and system modeling, Springer-Verlag, 2003

(Harmsen 1997) Harmsen F., Situational method engineering, Moret Ernst & Young, Doctoral dissertation University of Twente. 1997

(Hartigan 1979) Hartigan J., Wong M., Algorithm AS 136: A k- means clustering algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics), vol. 28, no. 1, pages 100–108, 1979

(Hashemi 2008) Hashemi R., Bahrami A., LaPlant J., Thurber K, Discovery of Intent through the Analysis of Visited Sites. In Hamid R. Arabnia and Ray R. Hashemi, editeurs, IKE, pages 417–422. CSREA Press, 2008

(Hayashi 2003) Hayashi M., Hidden markov models to identify pilot instrument scanning and attention patterns. In Systems, Man and Cybernetics, 2003. IEEE International Conference on, volume 3, pages 2889–2896. IEEE, 2003

(Henderson-Sellers 2002) Henderson-Sellers B., Process meta-modelling and process construction: examples using the OPF. Ann. Software Engineering 14(1-4), 341–362, 2002

(Henderson-Sellers 2008) Henderson-Sellers B., Gonzalez-Perez C., Ralyté J., Comparison of Method Chunks and Method Fragments for SME. ASWEC 2008, pp. 479-488, Los Alamitos, CA, USA, 2008

(Hoey 2007) Hoey J., Little J., Value-directed human behavior analysis from video using partially observable markov decision processes. Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 29, no. 7, pages 1118–1132, 2007

(Hotelling 1933) Hotelling H., Analysis of a complex of statistical variables into principal components, Journal of Educational Psychology, vol. 24, pp. 417-441 & 498-520, 1933

(Hug 2011) Hug C., Salinesi C., Deneckère R., Lamasse S., "Process modeling for Humanities: tracing and analyzing scientific processes" in Revive the Past: Proceedings of the 39th Annual Conference on Computer Applications and Quantitative Methods in Archaeology. Amsterdam University Press. Beijing, China, pp.245-255, 2011

(Hug 2012) Hug C., Deneckère R., Salinesi C., Map-TBS: Map process enactment traces and analysis, Proceedings of RCIS'12, International Conference on Research Challenges in Information Science, Valencia, Espagne, pp 204-209, 2012

(Hug 2014a) Hug C., Khodabandelou G., Epure E., Deneckère R., Salinesi C., Intention Mining: from logs to intentional process models, Quintas Jornadas de Ingenieria de Sistemas Informaticos y de Computacion, 2014

(Hug 2014b) Hug C., Deneckère R., Ammar A., Intentional Process Modeling of Statistical Analysis Methods, Computer Applications and Quantitative Methods in Archaeology (CAA), Paris, France, Intentional Process Modeling of Statistical Analysis Methods, 2014

(Husson 2011) Husson F., Josse J., Le S., Mazet J., FactoMineR, R package v. 1.16, http://CRAN.R-project.org/package=FactoMineR, 2011

(Iacovelli 2012) Iacovelli A., Approche orientée service pour la configuration de méthodes outillées. PhD thesis. University Paris 1 Panthéon-Sorbonne, 2012

(ISO 2002) ISO/IEC 15939: Software Engineering - Software Measurement Process, 2002

(Jankovic 2013) Janković M., Bajec M., Khodabandelou G., Deneckère R., Hug C., Salinesi C., Intelligent Agile Method Framework, ENASE'13, France, 2013

(Jarke 1994) Jarke M., Pohl K., Rolland C., Schmitt J. R., Experienced-Based Method Evaluation and Improvement : A Process Modeling Approach, CRIS series: Method and associated Tools for the Information Systems Life Cycle, North Holland, 1994

(Jeyakumaran 2021) Jeyakumaran S., Deneckère R., Rychkova I., Comment la Conformité au RGPD est intégrée dans les Pratiques de Gestion de Processus Métier (BPM) ?, Conférence INFORSID, 2021

(Juang 1991) Juang B., Rabiner L., Hidden Markov models for speech recognition. Technometrics, vol. 33, no. 3, pages 251–272, 1991

(Kang 1990) Kang C., Cohen S., Hess J., Novak W., Peterson S., Feature-Oriented Domain Analysis (FODA) Feasibility Study, 1990

(Karlsson 1997) Karlsson J., Ryan K.: A Cost–Value Approach for Prioritizing Requirements, IEEE Software, 1997

(Karlsson 2004) Karlsson F., Ågerfalk P.J., Method Configuration: Adapting to Situational Characteristics while Creating Reusable Assets, Journal IST, Vol.46 (9), 2004

(Karlsson 2008) Karlsson F., Ågerfalk P. J., Method Configuration: The eXtreme Programming Case, in XP 2008, Limerick, Ireland, pp 32-41, 2008

(Keeney 1993) Keeney R.L., Raiffa H., Decisions with Multiple Objectives: Preferences and Value Trade-Offs, Cambridge University Press, 1993

(Keeney 1999) Keeney R.L., Foundations for Making Smart Decisions, IIE Solutions, 31, No. 5, pp. 24-30, 1999

(Khodabandelou 2013a) Khodabandelou G., Hug C., Deneckère R., Salinesi C., Bajec M., Cots Products To Trace Method Enactment: Review And Selection. 21th European Conference on Information Systems, Utrecht, Netherlands. 2013

(Khodabandelou 2013b) Khodabandelou G., Hug H., Deneckère R., Salinesi C., Supervised Intentional Process Models Discovery using Hidden Markov Models. Seventh International Conference on Research Challenges in Information Science, Paris, France, 2013

(Khodabandelou 2013c) Khodabandelou G., Hug C., Deneckère R., Salinesi C.. Découverte supervisée de Modèles de processus intentionnels basée sur les Modèles de Markov Cachés. XXXIème Congrès INFORSID, Paris, France, 2013

(Khodabandelou 2013d) Khodabandelou G., Hug C., Deneckère R., Salinesi C., Process Mining Versus Intention Mining, EMMSAD'13, Spain, 2013

(Khodabandelou 2014a) Khodabandelou G., Hug C., Deneckère R., Salinesi C., Supervised vs. Unsupervised Learning for Intentional Process Model Discovery. Business Process Modeling, Development, and Support (BPMDS), Thessalonique, Greece. pp.1-15, 2014

(Khodabandelou 2014b) Khodabandelou G., Hug C., Deneckère R., Salinesi C., Unsupervised discovery of intentional process models from event logs, International Working Conference on Mining Software Repositories (MSR) 2014, Inde, 2014

(Khodabandelou 2014c) Khodabandelou G., Mining Intentional Process Models. Thèse de doctorat. Université Panthéon-Sorbonne - Paris I, English, 2014

(Kil 1996) Kil D., Shin F., Pattern Recognition and Prediction with Appli- cations to Signal Characterization. Modern Acoustics and Signal Processing, vol. XVI, page 418, 1996

(Kim 2010) Kim S., Cavedon L., Baldwin T., Classifying dialogue acts in one-on-one live chats, Conf. on Empirical Methods in Natural Language Processing, EMNLP '10, Stroudsburg, PA, USA, ACL, pp. 862–871, 2010

(Kipper 2004) Kipper K., Snyder B., Palmer M., Extending a verb-lexicon using a semantically annotated corpus, Fourth International Conference on Language Resources and Evaluation (LREC 2004). Lisbon, Portugal, May, 2004

(Kirsh-Pinheiro 2008) Kirsch-Pinheiro M., Vanrompay Y., Berbers Y., Context-aware service selection using graph matching, Proceedings of ECOWS 2008, vol. 411, 2008

(Koehler 2006) Koehler K. Thompson S. Organizational Change Management: Embracing New Ways of Thinking. Government Advantage, Crowe Chizek and Co., 2006

(Kornsyhova 2007a) Kornyshova E., Salinesi C. Business Process Priorisation with Multicriteria Methods: Case of Business Process Reengineering, International Conference on Enterprise Information Systems (ICEIS), Funchal, Portugal, 2007)

(Kornyshova 2007b) Kornyshova E., Deneckère R., Salinesi C., Method chunks selection by multicriteria techniques: an extension of the assembly-based approach, Proc. of the International IFIP WG8.1 Conference ME 07, Springer, Geneva, Switzerland, 2007

(Kornsyhova 2008) Kornyshova E., Deneckère R., Salinesi C., Using Multicriteria Decision-Making to Take into Account the Situation in System Engineering, CAISE'08 forum, 25, 2008

(Korsyshova 2009a) Kornyshova E., Salinesi C., Deneckère R., Which Method to Support Multicriteria Decision Making Systematically in IS Engineering?, Systems Research Forum Journal, vol 3 pp 15-24, 2009

(Kornsyhova 2009b) Kornyshova E., Deneckère R., Salinesi C., Improving Software Development Processes with Multicriteria Methods, Atelier MADSI: Méthodes avancées de développement des SI, 2009

(Kornyshova 2010a) Kornyshova E., Deneckère R., Decision-Making Ontology for Information System Engineering, submitted to EMMSAD 2010 Conference, 2010

(Kornyshova 2010b) Kornyshova E., Deneckère R., Claudepierre B. Contextualization of method components, , Actes de la conférence Research Challenges in Information Science (RCIS'10), Nice, France, pp 235-246, 2010

(Kornsyhova 2011) Kornyshova E., MADISE: Method Engineering-based Approach for Enhancing Decision-Making in Information Systems Engineering, PhD thesis, University of Paris 1 Panthéon-Sorbonne, Paris, France, 2011

(Korsnyhova 2011a) Kornyshova E., Deneckère R., Claudepierre B., Towards Method Component Contextualization, Journal IJISMD, vol 2, 4 p. 49-81, 2011

(Korsnyshova 2011b) Kornyshova E., Deneckère R., Rolland C., Method Families Concept: Application to Decision-Making Methods, Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD), London, United Kingdom, 2011

(Kornsyhova 2012a) Kornyshova E., Deneckère R., Using an Ontology for Modeling Decision-Making Knowledge, , Knowledge-Based and Intelligent Information & Engineering Systems (KES'12), San Sebastian, Espagne, 2012

(Kornyshova 2012b) Kornyshova E., Deneckère R., Decision-Making Method Family MADISE: Validation within the Requirements Engineering Domain, Actes de la conférence Research Challenges in information Science (RCIS'12), Valencia, Espagne, pp 120, 2012

(Kornyshova 2013) Kornyshova E., Ralyte J., Deneckère R., Constructing Method Families Based on the Variability Analysis, Research Challenges in Information Science (IEEE RCIS) 2013 Forum, France, 2013

(Kornsyhova 2022a) Kornyshova E., Deneckère R., Sadouki K., Gressier-Soudan E., Brinkkemper S.,Smart Life: Review of the Contemporary Smart Applications, International Conference on Research Challenges in Information Science (RCIS 2022), 302-318, 2022

(Kornsyhova 2022b) Kornyshova E., Deneckère R., A Proposal of a Situational Approach for Enterprise Architecture Frameworks: Application to TOGAF, International Conference on Knowledge-Based and Intelligent Information System (KES 2022), 2022

(Kornsyhova 2022c) Kornyshova E., Deneckère R., Gressier-Soudan E., Murray J., Brinkkemper S., From Smart Life to Smart Life Engineering: a Systematic Mapping Study and Research Agenda, 15th International Baltic Conference on Digital Business and Intelligent Systes (DB&IS 2022), 2022

(Kort 2005) Kort J., de Poot H., Usage Analysis: Combining Logging and Qualitative Methods, Workshop of CHI'05, 2005

(Koschmider 2021) Koschmider A., Leotta F., Serral E., Torres V., BP-Meets-IoT 2021 Challenge Dataset, 2021

(Kuloor 2003) Kuloor C., Eberlein A., Aspect-oriented requirements engineering for software product lines, In ECB'03, p. 98–107, 2003

(Kumar 1992) Kumar K., Welke R.J., Methodology Engineering: A Proposal for Situation Specific Methodology Construction. Challenges and Strategies for Research in Systems Development, Cotterman, W. and J. Senn (eds.), J. Wiley, Chichester, UK, pp. 257-266, 1992

(Kushnareva 2015) Kushnareva E., Rychkova I., Deneckère R., Le Grand B., Modeling crisis management process from goals to scenarios, AdaptiveCM 2015 – 4th International Workshop on Adaptive Case Management and other non- workflow approaches to BPM, Innsbruck, Austria, 2015

(Lee 2011) Lee J., Kim H., Seo W., Kim K., Kim C.H., Condition-based process patterns for model- ing of human processes in Knowledge-intensive Business Services. Journal of Expert Systems with Applications 38, 4025-4038, 2011

(Le Grand 2019) Le Grand T., Deneckère R., COOC: an Agile Change Management Method, IEEE BI 4.0, workshop of CBI, Moscow, Russia, 2019

(Lewis 2005) Lewis J. Project Planning, Scheduling & Control, 4E, McGraw Hill, 2005

(Li 1999) Li C., Biswas G., Finding behavior patterns from temporal data using hidden markov model based unsupervised classification. Proceedings of the 1999 CIMA: Computational Intelligence Methods and Applications (Rochester, NY, June 22-25), pages 266–272, 1999

(Liaskos 2007) Liaskos S., Jiang L., Lapouchnian A., Wang Y., Yu Y., do Prado Leite J., Mylopoulos J., Exploring the Dimensions of Variability: a Requirements Engineering Perspective. In VAMOS'07, Limerick, Ireland, 2007

(Ling 2015) Ling.upenn.edu, PennTreebankP.O.S.Tags, Available: https://www.ling.upenn.edu/courses/Fall_2003/ ling001/ penn_treeban k_pos.html. [Accessed: 15- Feb- 2015], 2015

(Lommatzsch 2014) Lommatzsch A., Real-Time News Recommendation Using Context- Aware Ensembles. In Proceedings of the 36th European Conference on IR Research (ECIR '14). Springer, 51–62, 2014

(Loucopoulos 1995) Loucopoulos P., Kavakli E., Enterprise Modelling and the Teleological Approach to Requirements Engineering, International Journal of Intelligent and Cooperative Information Systems, 1995.

(Maiden 2002) Maiden, N.A.M., Pavan, P., Gizikis, A., Clause, O., Kim, H., Zhu, X., Integrating Decision-Making Techniques into Requirements Engineering, REFSQ'02, Essen Germany, 2002

(Malcom 1967) Malcolm N., Explaining Behavior, The Philosophical Review, 76, 1, 1967

(Marcus 1993) Marcus M.P., Marcinkiewicz M.A., Santorini B., Building a large annotated corpus of English: The Penn Treebank Computational linguistics, vol. 19pp. 313-330, 1993

(Maria 2009) Maria V., Werneck B., de Padua Oliveira A., Cesar Sampaio do Prado Leite J., Comparing GORE Frameworks : i-star and KAOS. Anais do WER09 - Workshop em Engenharia de Requisitos, Valparaíso, Chile, 2009

(Martelli 2002) Martelli P., Fariselli P., Krogh A., Casadio R., A sequence-profile-based HMM for predicting and discriminating β barrel membrane proteins. Bioinformatics, vol. 18, no. suppl 1, pages S46–S53, 2002

(Martin 2003) Martín M., Olsina L., Towards an Ontology for Software Metrics and Indicators as the Foundation for a Cataloging Web System, In Proceedings of the First Conference on Latin American Web Congress (LA-WEB 2003), IEEE Computer Society, Washington, DC, USA, 103-113, 2003

(Martin-Rodilla 2013) Martín-Rodilla P., Empirical Approaches to the Analysis of Archaeological Discourse, in Across Space and Time: 41st Computer Applications and Quantitative Methods in Archaeology Conference, Perth, Australia, 2013

(Maurer 2004) Maurer R., Tips for Leading Change. The Journal for Quality and Participation Organizational change, 2004

(Meyer 2014) Meyer B., Agile!: The Good, the Hype and the Ugly, Springer Publishing, 2014

(Miller 1995) Miller G., WordNet: a lexical database for English, Communications of the ACM, vol. 38, no. 11, pp. 39-41, 1995

(Mirbel 2006) Mirbel I., Ralyté J., Situational method engineering: combining assembly-based and roadmap-driven approaches. Requirements Engineering, vol. 11, no. 1, pages 58–78, 2006

(Moisiadis 2002) Moisiadis F., The fundamentals of prioritizing requirements, Systems Engineering: Test and Evaluation Conference, Sydney, October, 2002

(Mustajoki 2005) Mustajoki J., Hamalainen R.P., Salo A., Decision Support by Interval SMART/SWING – Incorporating Imprecision in the SMART and SWING Methods, Decision Sciences, 36(2):317-339, 2005

(Najar 2011) Najar S., Kirsch-Pinheiro M., Souveyet C., Towards semantic modeling of intentional pervasive information systems. In Proceed- ings of the 6th International Workshop on Enhanced Web Service Technolo- gies, pages 30–34. ACM, 2011

(Nehan 2007) Nehan Y.-R., Deneckère R., Component-based situational methods, ME'07, Switzerland, 2007

(Ngo-The 2005) Ngo-The A., Ruhe G., Decision Support in Requirements Engineering, In Engineering and Managing Software Requirements, Ed. By A. Aurum and C. Wohlin 267-286, 2005

(Nurcan 2004) Nurcan S., Business Process Modeling for developing Process Oriented IT Systems, IRMA'04, New Orleans, USA, 2004

(Outmazgin 2013) Outmazgin N., Soffer P., Business Process Workarounds: What Can and Cannot Be Detected by Process Mining. In Enterprise, Business-Process and Information Systems Modeling, pages 48– 62. Springer, 2013

(Pearson 1901) Pearson K, On Lines and Planes of Closest Fit to Systems of Points is Space, Philosophical Magazine Series, vol. 2, no. 6, pp. 559-572, 1901

(Pohl 2005) Pohl K., Böckle G., Van der Linden F., Software Product Line Engineering: Foundations, Principles, and Techniques, Springer Verlag, 2005

(Prakash 2006) Prakash N., Rolland C., Systems Design for requirements expressed as a map, Proc. of the conference IRMA 06, Washington DC, 2006

(Qumer 2007) Qumer A., Henderson-Sellers, Construction of an Agile Software Product-Enhancement Process by Using an Agile Software Solution Framework (ASSF) and Situational Method Engineering, In COMPSAC 2007, Beijing, China, pp 539-542, 2007

(R 2014) R, computer software. Available from: http://www.r-project.org/, 2014

(Rabiner 1989) Rabiner L., A tutorial on hidden Markov models and se- lected applications in speech recognition. Proceedings of the IEEE, vol. 77, no. 2, pages 257–286, 1989

(Rakotomalala 2011) Rakotomalala R, Pratique de la Régression Logistique, Université Lyon 2, 2011

(Ralph 2008) Ralph P., Wand Y., A Teleological Process Theory of Software Development, JAIS Theory Development Workshop, 8, 23, 2008

(Ralyté 1999) Ralyté J., Rolland C., Plihon V., Method Enhancement with Scenario Based Techniques. CAISE'99, Springer-Verlag, Heidelberg, Germany, p. 103-118, 1999

(Ralyté 2001a) Ralyté J., Rolland C., An Approach for Method Reengineering. Proceedings of ER 2001, pp.471-484, LNCS 2224, Springer-Verlag, 2001

(Ralyté 2001b) Ralyté J., Rolland C., An Assembly Process Model for Method Engineering. Proceedings of the 13th Conference on Advanced Information Systems Engineering (CAISE'01), Interlaken, Switzerland, June 2001. K. Dittrich, A. Geppert, M. Norrie (Eds.). LNCS 2068, Springer-Verlag, pp. 267-283, 2001

(Ralyté 2003) Ralyté J., Deneckère R., Rolland, C., Towards a Generic Model for Situational Method Engineering. Proceedings, 2003

(Ralyté 2004) Ralyte J., Rolland C., Deneckère R., Towards a meta-tool for change-centric method engineering: a typology of generic operators, In CAISE'04, Lettonie, 2004

(Ralyté 2005a) Ralyte J., Maiden N., Rolland C., Deneckère R., Map-driven Modular Method Re-Engineering: Improving the RESCUE Requirements Process, CAISE'05, Portugal, 2005

(Ralyté 2005b) Ralyte J., Maiden N., Rolland C., Deneckère R.. Applying modular method engineering to validate and extend the RESCUE requirements process. Conceptual Modeling (ER'05), Klagenfurt, Austria. pp.209-224, 2005

(Ralyté 2008) Ralyte J., Deneckère R., Jamoussi Y., Construction de méthodes par composition, « Ingénierie des méthodes: des nouvelles tendances de développement des applications informatiques" (N. Kraïem & Y. Jamoussi (Ed.)) . 217-288, 2008

(Ralyté 2013) Ralyté J., Towards a Method Family Supporting Information Services Co-creation in the Transdisciplinary Context, IJISMD, Vol. 4(3), IGI-Global, 2013

(Reichert 2012) Reichert M., Weber B., Enabling Flexibility in Process-Aware Information Systems - Challenges, Methods, Technologies. Springer Berlin-Heidelberg, 2012

(Rey 2002) Rey G., Coutaz J., Le Contexteur : une abstraction logicielle pour la réalisation de systèmes interactifs sensibles au contexte. IHM'02, p. 105-112, 2002

(Rodrigues 2013) Rodrigues I., Matos N., Abreu S., Deneckère R., Diaz D., Towards Constraint-Informed Information Systems, Research Challenges in Information Science (IEEE RCIS) 2013, France, 2013

(Rolland 1993) Rolland C., Modeling the requirements engineering process. In Information Modelling and Knowledge Bases V: Principles and Formal Tech- niques: Results of the 3rd European-Japanese Seminar, Budapest, Hungary, May, pages 85–96, 1993

(Rolland 1998) Rolland C., A Comprehensive View of Process Engineering, CAISE'98, Springer-Verlag, 1998

(Rolland 1998b) Rolland C., Souveyet C., Ben Achour C., Guiding Goal Modelling Using Scenarios, IEEE Trans.on Soft. Eng., vol. 4, n°12, 1998

(Rolland 1999) Rolland C., Prakash N., Benjamen A., A Multi-Model View of Process Modelling, Requirements Engineering, Springer-Verlag London Ltd., vol. 4, n° 4, 1999

(Rolland 2000) Rolland C., Nurcan S., Grosz G., A Decision Making Pattern for Guiding the Enterprise Knowledge Development Process, Journal of Information and Software Technology, Elsevier, 42, p. 313-331, 2000

(Rolland 2005b) Rolland C., Salinesi C., Modeling goals and reasoning with them. In Engineering and Managing Software Requirements, pages 189– 217. Springer, 2005

(Rolland 2005) Rolland C, Modelisation of the O* process with MAP, Internal report of the 'Centre de Recherche en Informatique', University Paris 1 Sorbonne, 2005

(Rolland 2007) Rolland C. Capturing system intentionality with maps. In the book Conceptual Modelling in Information Systems Engineering, J. Krogstie, A. Opdahl, S, Brinkkemper (Eds.), 2007

(Rolland 2007 bpmds) Rolland C., Prakash N., On the Adequate Modeling of Business Process Families, BPMD'07, Trondheim, Norway, 2007

(Rolland 2010) Rolland C., Kirsch-Pinheiro M., Souveyet C., An intentional approach to service engineering. Services Computing, IEEE Transactions on, vol. 3, no. 4, pages 292–305, 2010

(Rosemann 2006) Rosemann M., Recker J., Context-aware process design: exploring the extrinsic drivers for process flexibility, CAISE'06, workshops and doctoral consortium, Luxembourg, Namur University Press., p 149-158, 2006

(Rosen 2008) Rosen M. A., Fiore S. M., Salas E., Letsky M., Warner N., Tightly coupling cognition: understanding how communication and awareness drive coordination in teams, International C2 journal, 2:1, 2008

(Rossi 2004) Rossi, M., Ramesh, B., Lyytinen, K., Tolvanen, J-P., Managing evolutionary method engineering by method rationale, Journal of the AIS, 5(9): 356-391, 2004

(Roy 1996) Roy B., Multicriteria Methodology for Decision Aiding, Dordrecht, Kluwer Academic Publishers, 1996

(Roy 2005) Roy B., Paradigms and challenges, Book chapter, In Multiple Criteria Decision Analysis - State of the Art Survey, Springer. editor(s) J. Figueira, S. Greco, M. Ehrgott, 2005

(Royce 1970) Royce W. W., Managing the development of large software systems : concepts and techniques, Proceedings of Wescon, 1970

(Rozinat 2007) Rozinat A., Alves de Medeiros AK., Gunther CW., Weijters AJMM., Van der Aalst W., Towards an evaluation framework for process mining algorithms. Beta, Research School for Operations Manage- ment and Logistics, 2007

(Rozinat 2010) Rozinat A., Process Mining Conformance and Extension. PhD thesis, Technische Universiteit Eindhoven, 2010

(Ruhe 2003) Ruhe G., Eberlein A., Pfahl D., Trade-off Analysis for Requirements Selection, International Journal of Software Engineering and Knowledge Engineering, 13, 4, 345-366, 2003

(Saaty 1980) Saaty T.L, The Analytic Hierarchy Process, NY, Mc Graw Hill, 1980

(Salber 1998) Salber D., Dey K.A., Abowd D.G., Ubiquitous Computing : Defining an HCI research agenda for an emerging interaction paradigm. In Georgia Tech GVU technical report, 1998

(Salifu 2007) Salifu M., Yu Y., Nuseibeh B., Specifying monitoring and switching problems in context. Proc. 15th Int. Conference on Requirements Engineering RE'07, 2007

(Salinesi 2003) Salinesi C. Rolland C., Fitting business models to system functionality exploring the fitness relationship. In Advanced Informa- tion Systems Engineering, pages 647–664. Springer, 2003

(SAS 2014) SAS, computer software. Available from: http://www.sas.com/en_us/software/analytics.html, 2014

(Schilit 1994) Schilit B., Adams N., Want R., Context-aware computing applications, IEEE WMCSA'94, Santa Cruz, CA, US, p. 89-101, 1994

(Schonenberg 2008) Schonenberg H., Weber B., Dongen B., Van der Aalst, W. M. P., Supporting flexible processes through recommendations based on history. Proceedings of the 6th International Conference on Business Process Management (BPM '08) (pp. 51-66). Springer Berlin-Heidelberg, 2008

(Schwaber 2002) Schwaber K. , Beedle M., Agile Software Development with Scrum, Prentice Hall PTR, Australia, 2002

(Seligman 1989) Seligmann P.S., Wijers G.M., Sol H.G., Analyzing the structure of IS methodologies, an alternative approach. In: Proceedings of the First Dutch Conference on Information Systems, Amersfoort, the Netherlands, 1989

(Serena 2007) Serena Software, An Introduction to Agile Software Development, Inc. Serena, Mariner, TeamTrack, http://www.serena.com (accessed in march 2015), 2007

(Shore 2007) Shore J., The art of agile development, O'Reilly Media Editors, 2007

(Simon 1960) Simon, H., The New Science of Management Decision, Harper&Row, 1960

(Soares 2013) Soares D., Santoro F., Baião F., Discovering collaborative knowledge-intensive processes through e-mail mining, Journal of Network and Computer Applications, vol. 36, no. 6, pp.1451-1465, 2013

(Soffer 2005) Soffer P., Rolland C., Combining intention-oriented and state-based process modeling, Lectures notes in Computer Science. Proc. of the ER conference, Klagenfurt, Austria, 2005

(Sokal 1993) Sokal RR, Sneath PHA, Principles of Numerical Taxonomy, W. H. Freeman and Company, San Francisco, 1993

(Souveyet 1998) Souveyet C., Deneckère R., Conception de bases de données: Aspects temporels, Techniques de l'ingénieur. Informatique, H3268. 1-H3268. 27, 1998

(Souveyet 1999) Souveyet C., Deneckère R., Overview of the TOOBIS project: From a temporal extension of ODMG to a Patterns-based approach extending OO model, International Journal of Computer Systems Science and Engineering 14 (6), 1999

(Souveyet 2015) Souveyet C., Deneckère R., Témoignage, ça bouge à la MIAGE Sorbonne !!!, Carine Souveyet, Rébecca Deneckère, Actes de la conférence 2/2 Quels Talents pour l'entreprise numérique ?, Université Paris 1 Panthéon-Sorbonne, 2015

(Stapleton 1995) Stapleton J., DSDM –Dynamic system development method. Addison-Wesley, UK, 1995

(Sunita 2012) Sunita D. A., Lobo L. M. R. J., A Comparative Study of Association Rule Algorithms for Course Recommender System in E-learning. International Journal of Computer Applications. 39. 48-52, 2012

(Taylor 1964) Taylor C., The Explanation of Behaviour, London, Routledge, 1964

(Thevenet 2009) Thevenet L-H., Proposition d'une modélisation conceptuelle d'alignement stratégique : la méthode INSTAL, Thèse de doctorart, Université Paris 1 Panthéon-Sorbonne, 2009

(Thorne 2013) Thorne C., Cardillo E., Eccher C., Montali M., Calvanese D., Process Fragment Recognition in Clinical Documents, in 13th Conference of the Italian Association for Artificial Intelligence (AI*IA 2013), Springer, pp.227-238, 2013

(Tiwana 2000) Tiwana A., The knowledge management toolkit: practical techniques for building a knowledge management system. Upper Saddle River, NJ: Prentice Hall PTR, 2000

(UDC 2013) Eclipse UDC. Filtered UDC Data. url- http://www.eclipse.org/org/usagedata/, 2013

(Van de Ven 1995) Van de Ven A.H., Poole M.S., Explaining Development and Change in Organizations, The Academy of Management Review, vol. 20, n° 3, p. 510-540, 1995

(Van der Aalst 2004a) Van der Aalst, W. Weijters T., Maruster L., Workflow mining: Discovering process models from event logs. Knowledge and Data Engineering, IEEE Transactions on, vol. 16, no. 9, pages 1128–1142, 2004

(Van der Aalst 2004c) Van der Aalst W., Weijters AJMM., Process mining: a research agenda. Computers in industry, vol. 53, no. 3, pages 231–244, 2004

(Van der Aalst 2009) van der Aalst, W.M.P., Process-Aware Information Systems: Lessons to Be Learned from Process Mining. In: Jensen, K., van der Aalst, W.M.P. (eds) Transactions on Petri Nets and Other Models of Concurrency II. Lecture Notes in Computer Science, vol 5460. Springer, Berlin, Heidelberg, 2009

(Van der Aalst 2011) Van der Aalst W., Process mining: discovery, conformance and enhancement of business processes. Springer, 2011

(Van der Werf 2011) Van der Werf J.M.E.M., Compositional Design and Verifica- tion of Component Based Information Systems. 2011

(Van Gurp 2000) Van Gurp J., Variability in Software Systems, the key to Software Reuse, Licentiate Thesis, University of Groningen, Sweden, 2000

(Van Gurp 2001) Van Gurp J., Bosch J., Svahnberg M., On the notion of variability in software product lines, working IEEE/IFIP conference on software architecture, 2001

(Van Lamsweerde 2001) Van Lamsweerde A., Goal-oriented requirements engi- neering: A guided tour. In Requirements Engineering, 2001. Proceedings. Fifth IEEE International Symposium on, pages 249–262. IEEE, 2001

(Van Lamsweerde 2004) Van Lamsweerde A., Letier E., From object orientation to goal orientation: A paradigm shift for requirements engineering. Radical Innovations of Software and Systems Engineering in the Future, 2941(I), 325-340. Springer-verlag Berlin. Retrieved from http://discovery.ucl.ac.uk/97080/, 2004

(Van Lamsweerde 2009) van Lamsweerde A., Reasoning About Alternative Requirements Options, Conceptual Modeling: Foundations and Applications, p. 380-397, 2009

(Van Slooten 1996) Van Slooten K., Hodes B., Characterising IS development projects, IFIP WG8.1 Conference on Method Engineering, 1996

(Veblen 1898) Veblen, T., Why is Economics not an Evolutionary Science?, The Quarterly Journal of Economics, 12, 4, p. 373-397, 1898

(VersionOne 2013) VersionOne, 8th annual state of agile development survey, http://www.versionone.com (accessed in Nov. 2014) , 2013

(Vincke 1989) Vincke P, L'aide multicritère à la decision, Edition Ellipses, Edition de l'Université de Bruxelles, Bruxelles, Belgique (1989)

(Wang 2011) Wang S., Nan B., Rosset B. Zhu J., Random lasso, The Annals of Applied Statistics, 5, pp. 468–485, 2011

(Wang 2014) Wang G., Wang H., Li J., Abrahams A.n Fan W., An analytical framework for understanding knowledge-sharing processes in online q&a communities, ACM Trans. Manage. Inf. Syst., 5, pp. 18:1–18:31, 2014

(Ward 1963) Ward, JH, Hierarchical grouping to optimize an objective function, Journal of the American Statistical Association, vol. 58, pp. 236-244, 1963

(Weiss 1999) Weiss D.M., Lai C.T.R., Software product-line engineering: a family-based software development process. Addison-Wesley, 1999

(Wiegers 1999) Wiegers K., First Things First: Prioritizing Requirements, Software Development, vol. 7, no. 9, 1999

(Wistrand 2004) Wistrand W., Karlsson F., Method components: rationale revealed, Proceedings of CAISE'04, Springer-Verlag. Riga, Latvia, 2004

(Yu 1987) Yu E., What Does It Mean to Say that a Specification is Complete? In Software Specification and Design, Proceeding of IWSSD-4, Fourth In- ternational Workshop on. IEEE, 1987

(Yu 1997) Yu E., Towards modelling and reasoning support for early-phase requirements engineering. Proceedings of ISRE 97 3rd IEEE International Symposium on Requirements Engineering. IEEE Comput. Soc. Press., 1997

(Yu 2009) Yu E., Social Modeling and i., Conceptual Modeling Foundations and Applications, 5600(c), 99-121, 2009

(Yu 2011) Yu E., Modelling strategic relationships for process reengineering. Social Modeling for Requirements Engineering, vol. 11, page 2011, 2011

(Zaki 2014) Zaki J.M., Meira J.W., Data Mining and Analysis: Fundamental Concepts and Algorithms, 2014

(Zhang 2004) Zhang Y., Prediction of financial time series with Hidden Markov Models. PhD thesis, Simon Fraser University, 2004

VII. Curriculum Vitae

Née le 19/12/1970 (51 ans), deux enfants.

Grade : MCf (depuis septembre 2002), Hors classe depuis 2019

Etablissement d'affectation : Université Paris 1 Panthéon-Sorbonne

Section de CNU : 27

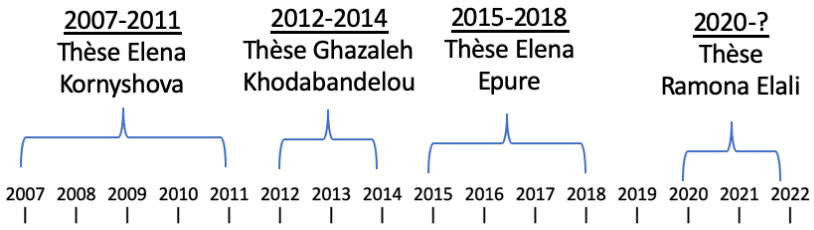
Unité de recherche d'appartenance : EA – 1445 – Centre de Recherche en Informatique (CRI)

Dernier diplôme obtenu : 2001 - Thèse en informatique de l'université Paris 1 Panthéon-Sorbonne

Parcours professionnel

- Promotion Hors Classe en 2019
- Depuis 2002 : Poste de Maître de Conférences : Université Paris 1 Panthéon-Sorbonne
- 2000-2002 : Poste de contractuel PRAG : IUT Fontainebleau
- 1999-2000 : Poste de demi ATER IUT Fontainebleau
- 1998-1999 : Poste de demi ATER Université Paris 1
- 1996-1998 : vacataire IUT Fontainebleau et université Paris 1 Panthéon-Sorbonne
- 1995-1998 : Allocataire de recherche (Université Paris 1 Panthéon-Sorbonne)

VII.1. Encadrements doctoraux



J'ai encadré trois thèses soutenues : Elena Kornyshova, Ghazaleh Khodabandelou et Elena Epure.

Elena Kornyshova

- Titre: « Madise : une approche basée sur l'ingénierie des méthodes pour améliorer la prise de décision dans l'ingénierie des systèmes d'information ».

- Période : 2006-2011 (deux ans d'arrêt du travail doctoral pour des raisons financières et emploi en contrat de travail dans le monde professionnel).
- Taux d'encadrement : 70% (avec Camille Salinesi 30%).
- Avenir du docteur : Elena est maintenant Maitre de conférences au CNAM.

Ghazaleh Khodabandelou

- Titre : « Contextual Recommendations using Intention Mining on Process Traces ».
- Période : 2010-2013
- Taux d'encadrement : 33% (avec Camille Salinesi 33% et Charlotte Hug 33%)
- Financement : Thèse financée pendant un an par un financement d'une autre université.
- Avenir du Docteur : Maitre de conférences à l'université de Créteil

Elena Epure

- Titre : «Intention mining and perception of human behavior from unstructured texts»
- Période : 2014-2018.
- Taux d'encadrement : 60% (avec Camille Salinesi 40%).
- Financement : Thèse avec bourse ministérielle (deux ans en tant que monitrice et un an avec un contrat entreprise) puis un contrat d'ATER.
- Prix de thèse : Université Paris 1 et INFORSID
- Avenir du docteur : Ingénieur de recherche chez Deezer.

J'ai également un encadrement de thèse en cours : Ramona Elali.

Ramona Elali

- Titre : «Recommandations contextuelles fondées sur la fouille d'intentions et les ontologies»
- Période : 2020- ?.
- Taux d'encadrement : 40% (avec Elena Kornyshova 40% et Camille Salinesi 20%).
- Financement : Thèse avec bourse ministérielle.

VII.2. Prix et distinctions scientifiques

- 1 best paper award à la conférence RCIS 2013 (Core B)
- PES de 2013 à 2017
- PEDR de 2017 à 2021

VII.3. Invitations

- 2009 : Participation à une école d'été à Phnom Penh (Cambodge) pour dispenser un cours d'une semaine sur l'architecture des systèmes d'information au Cambodia Institute of Technology.

- 2014 : Séjour de quelques jours à Ljubljana (Slovénie) pour travailler sur la proposition de projet Proteus qui nous a été accordé par la suite.
- 2019 : Tutoriel présenté dans la conférence internationale ER 2019, Brésil
- 2020 : Tutoriel présenté dans la conférence internationale RCIS 2020, Grèce (en visio conférence étant donné la pandémie)
- 2023 : Invitation pour présenter la recherche en Système d'Information au TEC (Technologico de Costa Rica)

VII.4. Participation à des jurys de thèse

- Février 2017 : jury de Houda Zaidi (CNAM). « Amélioration de la qualité des données - Correction sémantique des anomalies inter-colonnes »

VII.5. Membre de Comités de Sélection

- UFR 27 de l'université Paris 1 (2004, 2012, 2018, 2021, 2022)
- UPEM (2018)
- IUT de Belfort (2021)
- Université de Toulouse Capitole (2022)

VII.6. Comités de pilotage de congrès

- **Présidente de l'organisation** de la conférence EDOC 2019
- **Co-Présidente du Programme** d'INFORSID 2012, du forum de CAISE 2013, de RCIS 2014, du forum de RCIS 2018
- **Responsable des actes** de RCIS 2016, CBI 2016, RCIS 2022.
- **Présidente de l'organisation locale** de ME 2011, RCIS 2013, INFORSID 2013 , EDOC 2019
- **Participation à l'organisation locale** de RE 2005.
- **Responsable de la publicité des conférences** CAISE 2011, CAISE 2012, CAISE 2016 et RCIS 2017.
- **Co-responsable de la communication** de ISESL 2021
- **Co-chair** de ISESL 2022

VII.7. Visibilité

L'un des piliers de la recherche au Centre de Recherche en Informatique est la collaboration et très peu de publications sont effectuées seul. Je ne déroge pas à cette tendance et n'ait que 4 publications publiées à titre individuel (dont ma thèse). Je collabore au sein du CRI à 37%, avec des collaborateurs français hors de mon laboratoire à 32% et à l'international à 30% (Figure 101).

	Seule	Intra CRI	Collaborations France	Collaborations internationale		
Journal International	0	2	2	3	7	9,21%
Conférence Internationale	1	12	14	15	42	55,26%
Journal National	1	1	1	0	3	3,95%
Conférence Nationale	0	3	2	0	5	6,58%
Workshop International	0	8	1	1	10	13,16%
Workshop National	0	0	1	1	2	2,63%
Chapitre de Livre	0	1	0	2	3	3,95%
Thèse	1	0	0	0	1	1,32%
Doctoral consortium	0	2	0	0	2	2,63%
Rapport Interne	0	0	0	1	1	1,32%
	3	29	21	23	76	100,00%
	3,95%	38,16%	27,63%	30,26%	100,00%	

Figure 101. Collaborations dans les publications

VII.7.1. Visibilité locale

- **Conseils de l'établissement**

J'ai été membre de la **Commission à la Recherche** de l'université de 2013 à 2016.

Je participe régulièrement aux **jurys de VAE** pour la MIAGE.

- **Conseils de l'UFR 27**

Directrice adjointe de l'UFR 27 depuis 2021 : l'UFR 27 inclue les Mathématiques et l'Informatique. Je m'occupe de la gestion de la partie Informatique de l'UFR contenant le CRI (Centre de Recherche en Informatique) et la formation MIAGE. A ce titre, j'interviens donc également dans l'établissement des cours d'informatique en licence MIASHS L1 et L2 pour avoir une bonne cohérence avec la L3 MIAGE.

Membre du conseil de l'UFR 27 depuis 2014.

Membre du Conseil Consultatif Scientifique de l'UFR 27.

Coordinatrice C2i pour l'UFR 27 (environ 150 étudiants par an) de 2009 à 2015 (création des supports de cours et de TDs, gestion des plannings et des groupes, gestion de l'équipe enseignante, élaboration des examens, etc.).

- **Conseils du laboratoire CRI**

Directrice adjointe pour le laboratoire de recherche CRI depuis 2017. Le CRI ne comptabilise qu'une dizaine de permanents, une dizaine de thésards et quelques stagiaires. J'aide ma collègue directrice dans les tâches inhérentes à la direction du laboratoire.

Membre du laboratoire depuis 2002.

Responsable financière du laboratoire de 2014 à 2017.

Membre des comités de suivis des doctorants du laboratoire (2 en 2019, 8 en 2020, 7 en 2022).

Membre de comités scientifiques des doctorants du laboratoire (1 en 2022).

Organisation de l'événement Poster Madness au CRI en 2014 et 2015 : Cet événement demande à ce que chaque doctorant du CRI présente un poster sur son thème de recherche avec 3 minutes de présentation orale. Un concours est organisé avec trois prix différents : le prix de la meilleure présentation, celui du plus beau poster et celui du coup de cœur. Nous invitons régulièrement d'autres laboratoires à se joindre à nous pendant cet événement.

Participation à l'amélioration de la « vie du labo ». Depuis quelques années, nous organisons régulièrement des évènements sociaux au sein du laboratoire de recherche pour conserver la convivialité au sein de notre équipe. Cela passe par l'organisation d'une fête de Noël (la « CRIsmas party »), d'un pique-nique estival (le « barbeCRI »), de la semaine de la gentillesse, de la participation à des évènements sportifs (comme les 20 km de Paris, au sein de l'équipe de Paris 1) comme par des évènements orientés-recherche comme la Poster Madness. Tous ces évènements sont auto-financés par les participants et permettent de souder notre équipe de manière beaucoup plus forte. En tant que directrice adjointe, mon rôle dans ces événements s'est accéléré ces dernières années pour un pilotage plus efficace de ces évènements.

VII.7.2. Visibilité nationale

Membre titulaire du CNU 27 depuis 2016 (Responsable de la liste Alternative, rang B, pour le dernier mandat).

Organisation du colloque pluridisciplinaire Numérique et Société en 2014 et 2015. Mise en relation, avec présentation et table ronde, de chercheurs en informatique avec des chercheurs en sciences humaines pour leur faire partager leurs points de vue sur une question spécifique (le web, le genre à travers les jeux vidéos, les espaces sonores, les prothèses cognitives, les réseaux sociaux, etc.). https://www.pantheonsorbonne.fr/numerique-et-societe/

Membre du comité exécutif de l'association INFORSID et **secrétaire** de cette association depuis 2020 (association loi 1901 qui rassemble les chercheurs en informatique des organisations et systèmes d'information et qui a pour objectif de promouvoir les recherches effectuées dans ces domaines en faisant intervenir le plus largement possible les utilisateurs et les industriels).

Secrétaire de la CDM (Conférence des directeurs de MIAGE) depuis décembre 2012. Participation à la CDM en tant que responsable de licence MIAGE puis de master MIAGE depuis 2012.

Organisation de la nuit de l'info depuis 2012. Etant responsable de site, je m'occupe également de trouver des partenaires industriels pouvant proposer des défis lors de la nuit. Nous avons entre 30 et 50 étudiants participants à ces évènements, encadrés pendant la nuit par 3 enseignants de la MIAGE. Je suis en général également responsable d'un défi organisé par le laboratoire de recherche et ouvert à tous les participants de la nuit.

Organisation de la participation des étudiants de MIAGE aux **JNM** (Journées Nationales MIAGE) tous les ans.

Publication d'un article sur la MIAGE Sorbonne présenté lors de la conférence *Quels talents pour l'entreprise numérique ?* (Souveyet 2015)

VII.7.3. Visibilité internationale

Membre du groupe de travail 8.1 de l'IFIP (IFIP WG 8.1 : Design and Evaluation of Information Systems).

Co-chair du Task Group « Smart Life » du groupe de travail 8.1 de l'IFIP depuis 2022.

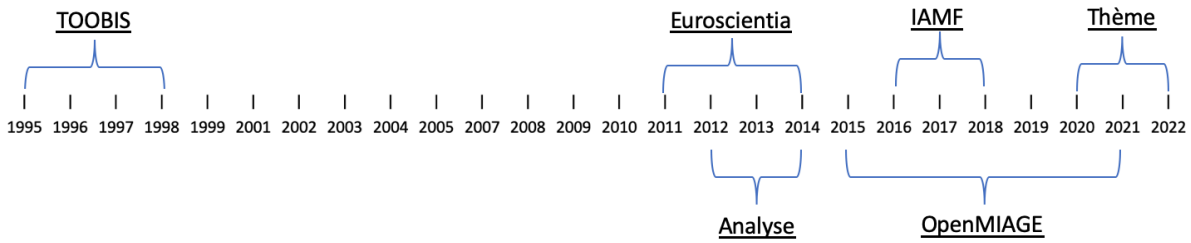
Membre régulier de comités de programmes pour des conférences internationales (CAISE, RCIS, CBI, BigD2M, TEAR, EMMSAD) et nationale (INFORSID), ainsi que ponctuel pour certaines conférences (ME'07, MCIS'08, CMM 11, MCIS 11, ME 11, ACM 12, TEAR 12, PRET-4 12, PRET-5 12, RCIS Forum 12, WorldCIST 13, ECIS 13, ISPCM 13, CBI 13, PRET 13, MCCSIS-ISPCM 14, ECIS 15, QMMQ 15, ANT 16, ANT 18, AMCIS 20) ou revues (SREP'08, IST 11, IJSMD 12, ISI 14, ISI 15, ISI 16, ISI 17, BISE 22).

Responsabilités éditoriales : Edition de 3 numéros spéciaux (IJISMD 2015, SNAM 2015, ISI 2013, CSIMQ 2018) et édition de 5 actes de conférences (forum RCIS 2018, RCIS 2014, forum CAISE 2013, INFORSID 2012, ME 11).

Organisation de la **participation des M1 MIAGE à des conférences internationales de recherche** depuis 2021 (Workshops de EDOC en distanciel en 2021, Workshops de CAISE en présentiel en 2022).

Incitation auprès des tuteurs enseignants pour **diffuser les bons mémoires** des étudiants de M2 sous forme d'articles de recherche, par exemple (Le Grand 2019) (Deneckère 2020) (Jeyakumaran 2021) (Alsayed 2022) (Gauthier 2022).

VII.8. Participation à des contrats



Participation au Projet ESPRIT "**TOOBIS**" (Temporal Oriented Object dataBases for Information Systems) (1995-1998). Ce projet visait à développer un système compétitif de gestion de base de données orientée objet temporel pour soutenir la représentation et l'interrogation de

données bi-temporelles. J'ai participé à ce pojet pendant ma thèse pour développer une méthodologie de modélisation conceptuelle innovante pour les applications temporelles.

Publications liées au projet TOOBIS : (Deneckère 1998), (Souveyet 1998), (Souveyet 1999), (Deneckère 2001a), (Deneckère 2001b), (Deneckère 2002a), (Deneckère 2002b)

Participation au projet '**Analyse'** (2012-2014) projet transdisciplinaire sur les traces et analyse de traces des processus scientifiques, avec les chercheurs en Sciences humaines et en Statistiques de l'université Paris 1. Ma participation à ce projet (avec Camille Salinesi et Charlotte Hug) était de conceptualiser le méta-niveau de l'analyse SHS pour permettre de guider et recommander des actions à un utilisateur SHS voulant analyser des données. https://analyseshs.hypotheses.org/

Publications liées au projet Analyse : (Hug 2011), (Ammar 2013), (Hug 2014b)

Participation au projet '**Euroscientia'** (2011-2014), sur la modélisation conceptuelle des données de chercheurs en Sciences Humaines français et allemands. Le projet est financé par l'ANR et la Deutsche Forschungsgemeinschaft. Ma participation dans ce projet (avec Camille Salinesi et Charlotte Hug) était de construire et valider le schéma de la base de données qui a été utilisée ensuite pour intégrer les données concernant la circulation des savoirs d'états en Europe entre 1750 et 1850. http://euroscientia.univ-paris1.fr/

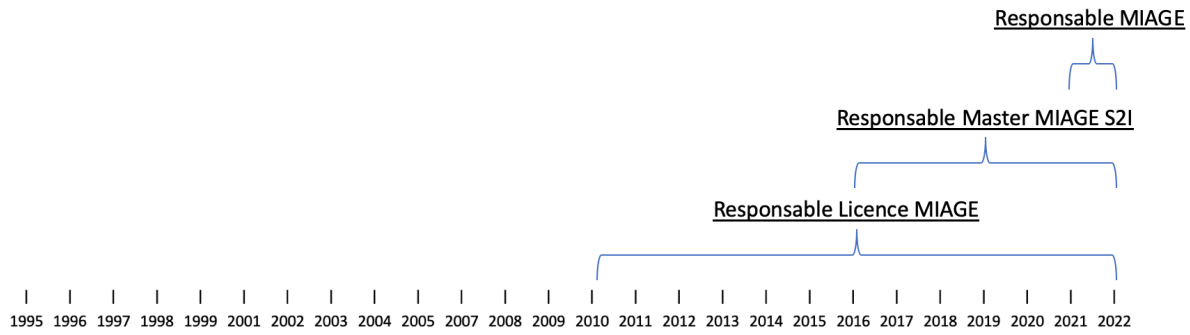
Participation au projet Proteus '**IAMF'** (2016-2018): Traces et analyse de traces pour l'ingénierie des méthodes situationnelles' avec les chercheurs Slovènes de l'équipe de Marko Bajec. Ma participation dans ce projet (avec Camille Salinesi et, notemment avec Elena Epure, notre étudiante en thèse) a été d'apporter notre analyse sur la fouille d'intentions et de proposer des techniques pouvant s'intégrer dans le projet.

Publications liées au projet IAMF : (Jankovic 2013), (Khodabandelou 2013a)

Participation au projet IDEFI-N **OpenMIAGE** (2015-2021) qui vise à mettre en place un dispositif innovant de formation numérique basé sur les compétences recherchées par les entreprises et les salariés. Ce projet est porté par l'ensemble des MIAGE de France et est financé par l'ANR. Je participe à ce projet avec Carine Souveyet (CRI). En collaboration avec le réseau des MIAGE, nous avions entamé la première tâche depuis plusieurs années pour identifier l'ensemble du référentiel de compétences correspondant à cette formation au niveau national. Même si ce référentiel est toujours en évolution et que je participe toujours à cette tâche, je travaille essentiellement dans ce projet sur la compréhension de l'outil par les utilisateurs, sur les retours de ceux-ci et l'impact engendré sur le projet. Nous intervenons également sur la modélisation du processus mis en place dans le projet pour permettre de guider au mieux la finalisation de celui-ci. http://openmiage.univ-lyon1.fr/

Participation au projet NCU « **Theme »** (2020-2022) qui reprends le projet OpenMIAGE pour aller un peu plus loin. Je suis responsable du site Paris 1 pour ce projet et suis conceptrice des modules du thème Ingénierie des Systèmes d'Information.

VII.9. Responsable d'années de formation



Responsable de la formation MIAGE Sorbonne depuis 2021 (entre 150 et 200 étudiants sur 6 groupes : L3 classique et L3 apprentissage, M1 classique et M1 apprentissage, M2 IKSEM et M2 S2I). Gouvernance de la formation dans son ensemble : Encadrement de l'équipe des responsables de parcours (6 groupes), élaboration des maquettes d'enseignement, tenue des réunions pédagogiques / conseils de perfectionnement / journées portes ouvertes, élaboration des plannings, prospection des enseignants vacataires, contacts avec le CFA, gestion du budget, etc.

Responsable du Master M2 MIAGE parcours S2I depuis 2016 (environ 27 étudiants). Pilotage du parcours: étude des dossiers de candidature, tenue des jurys de délibération, encadrement des étudiants, gestion au quotidien, etc.

Responsable de la Licence MIAGE depuis 2010 (un groupe en formation classique et un par voie de l'apprentissage au départ puis uniquement celui de l'apprentissage depuis 2020). Environ 90% des étudiants de L3 Miage continuent leur cursus universitaire en Master (80% dans le Master Miage de notre université). Pilotage du parcours: étude des dossiers de candidature, tenue des jurys de délibération, encadrement des étudiants, gestion au quotidien, etc.

Encadrement de stagiaires et d'apprentis de diverses formations, professionnelles ou de recherche, chaque année.

Entreprenariat : En master MIAGE Sorbonne, il est possible en deuxième année de choisir de faire, non pas un mémoire de recherche mais un projet de création d'entreprise. Celui-ci s'effectue par petits groupes sur un projet innovant et est suivi par des associations partenaires (PIVOD et MIAGE Entrepreneurs). Il s'agit de réaliser la partie préparatoire à une création d'entreprise, sans aller jusqu'au dépôt officiel de celle-ci (business plan, business model, pitchs, enquêtes de marché, etc.). J'encadre ces groupes depuis 2020.