Découverte supervisée de Modèles de processus intentionnels basée sur les Modèles de Markov Cachés

Ghazaleh Khodabandelou, Charlotte Hug, Rébecca Deneckère, Camille Salinesi

Centre de Recherche en Informatique
Université Paris 1 Panthéon-Sorbonne
90 rue de Tolbiac,
75013 Paris, France

Ghazaleh.Khodabandelou@malix.univ-paris1.fr, {Charlotte.Hug, Rebecca.Deneckere, Camille.Salinesi}@univ-paris1.fr

RÉSUMÉ. *Cela fait plusieurs décennies que la communauté des Systèmes d'Information (SI) s'intéresse à la découverte 'automatisée' des modèles de processus. Certaines approches se basent sur les activités séquentielles (traces) effectuées par les acteurs du SI pour identifier les modèles de processus. Cependant, ces approches ne portent que sur les activités et les modèles identifiés sont donc orientés-activités. Les modèles de processus intentionnels se concentrent sur les intentions qui ont entraîné les activités plutôt que sur les activités elles-mêmes. Malheureusement, les approches de fouille de processus existantes ne tiennent pas compte de l'aspect caché des intentions derrière les activités. Nous pensons pouvoir découvrir les modèles de processus intentionnels à l'aide de techniques de fouille d'intention. Le but de cet article est de proposer l'utilisation de modèles probabilistes - les Modèles de Markov Cachés (MMC) - pour évaluer les intentions les plus probables à partir des traces. Cet article se concentre sur une approche supervisée pour découvrir les intentions sous-jacentes aux traces d'activités des utilisateurs et de les comparer au modèle de processus intentionnel initial.*

ABSTRACT. *Discovering process models is a subject of interest in the Information System (IS) community. Approaches have been proposed to recover process models, based on the sequential tasks (traces) of IS's actors. However, these approaches only focus on activities and the models identified are, in consequence, activity-oriented. Intentional process models focus on intentions rather than activities. Unfortunately, existing process-mining approaches do not consider the hidden intentions behind the activities. We think we can discover the intentional process models underlying user activities by using Intention mining techniques. Our aim is to propose the use of probabilistic models to evaluate the most likely intentions behind activities, namely Hidden Markov Models (HMMs). We focus here on a supervised approach that allows discovering the intentions behind the user activities traces and to compare them to the prescribed intentional process model.*

MOTS-CLÉS : *fouille d'intention; processus; apprentissage supervisé; découverte de processus.*

KEYWORDS: *intention mining; process modeling; supervised approach; process discovery.*

1. Introduction

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 *et al.,* 2006) (van der Aalst *et al.,* 2004). 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 *et al.,* 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.

Ce travail se concentre sur la découverte des intentions cachées dans les logs et sur la construction du modèle de processus intentionnel. Pour reconstruire le modèle de processus (le modèle suivi qui génère des traces d'activités) et évaluer les intentions, nous utilisons les Modèles de Markov Cachés (MMC) (Baum *et al.,* 1996) et la Carte, un formalisme permettant de définir des modèles de processus intentionnels (Rolland *et al.,* 1999). Nous proposons une méthode en deux étapes. La première étape, dans laquelle les intentions doivent être connues, consiste en une phase d'entraînement pour estimer les paramètres du MMC. La deuxième étape consiste à appliquer les paramètres estimés sur un nouvel ensemble de données et de montrer qu'il est possible de récupérer les intentions de manière automatisée. Dans cet article, nous décrivons une étude de cas qui consiste à analyser des traces d'activités d'étudiants concevant des diagrammes entité-association.

Nous présentons les travaux connexes en section 2. La contribution technique basée sur MMC est décrite en section 3. Les résultats de l'étude de cas sont présentés en section 4 et la section 5 conclue cet article.

2. Travaux connexes

Les MMC sont appliqués dans de nombreux domaines pour reconstruire les états cachés de processus observés, comme la reconnaissance vocale (Juang *et al.,* 1991) et la bio-informatique (Enright *et al.,* 2002). Un type de problème similaire se pose dans le domaine de la fouille de processus, dont l'objectif est d'extraire des séquences d'activités observées à partir de logs d'événements afin de récupérer le workflow d'origine produisant les logs. Dans les diverses approches de fouille de processus, l'objectif est généralement de découvrir la séquence d'activités sous-jacentes aux processus, à partir des logs d'événements, à l'aide d'algorithmes et de techniques variées telles que la classification et les techniques d'apprentissage (van der Aalst *et al.,* 2011). Une approche pour découvrir des processus est l'apprentissage automatique (Herbst *et al.,* 1998) (Eibe *et al.,* 2005). Il existe différentes approches d'apprentissage automatique : supervisé, non supervisé, semi-supervisé et de renforcement. L'apprentissage supervisé est basé sur deux étapes : (a) dans l'étape d'entraînement, un algorithme d'apprentissage est appliqué sur des observations (variables indépendantes) pour entraîner un classificateur, (b) dans l'étape de prédiction, les performances de ce classificateur peuvent être testées avec un jeu d'observations. De plus, le classificateur peut classifier de nouvelles données.

L'objectif de cette approche dans le domaine de la fouille de processus consiste à classifier des séquences d'activités en classes par la recherche de similitudes entre elles. Les approches de *trace clustering* (Minseok *et al.,* 2009) et *sequence clustering* (Ferreira *et al.,* 2007) permettent de classer les traces d'activités. Basées sur la nature des algorithmes de fouille, plusieurs approches ont émergées en fouille de processus (Ferreira *et al.,* 2007) telles que l'α-algorithme (van der Aalst *et al.,* 2004) les graphes acycliques orientés (Agrawal *et al.,* 1998), le clustering hiérarchique (Greco *et al.,* 2005), les algorithmes génétiques (van der Aalst *et al.,* 2005), les graphes d'instance (van Dongen *et al.,* 2004) ou l'acquisition inductive de workflow (Herbst *et al.,* 1998). Tous ces algorithmes ont besoin de logs offrant d'autres informations, comme l'identifiant d'instance de processus qui doit être connu pour chaque trace, ou encore le seuil (paramètre d'algorithme, qui limite la reconstruction des relations causales sous une certaine probabilité et empêche le bruit). Ce type d'algorithmes offre de bonnes performances mais l'aspect intentionnel est complètement ignoré.

Dans cet article, nous appliquons un MMC couplé à un apprentissage supervisé dans la lignée de plusieurs travaux antérieurs.

Dans (van der Aalst *et al.,* 2011), les MMC sont considérés comme versatiles et pertinents pour la fouille de processus, mais les approches non supervisées des MMC sont complexes : (a) en raison de procédures itératives, (b) le nombre d'états d'intentions (en tant qu'entrées de l'algorithme) doit être connu, (c) le résultat des MMC n'est pas très compréhensible pour l'utilisateur final.

Dans (Cook *et al.,* 1998, 1995) , trois algorithmes d'inférence illustrent la découverte de processus à partir des logs d'événements : RNet, Ktail et Markov. RNet

(Das *et al.,* 1994) est une approche statistique qui caractérise un état en fonction des comportements passés - robuste au bruit, mais consommant beaucoup de temps dans la phase d'apprentissage, la taille des réseaux augmente avec le nombre de types de jeton et il faut évaluer de nombreux paramètres. Ktail (Biermann *et al.,* 1972) est une approche algorithmique évaluant l'état actuel en fonction du comportement futur ; lorsque le nombre d'états augmente, on peut contrôler la complexité de l'algorithme par une machine à états finis déterministe mais Ktail n'est pas très robuste au bruit. Markov est une approche hybride, entre les approches statistiques et algorithmiques, qui regarde les comportements proches passés et futurs pour définir l'état futur ; elle est robuste au bruit avec une complexité contrôlable et dispose d'une machine à état finis déterministe mais cet algorithme n'essaie que de découvrir les processus à partir des logs d'événements et ne tient pas compte des intentions des acteurs.

(Rozinat *et al.,* 2008) utilisent un MMC comme une technique de vérification de la conformité en mesurant les similitudes entre des modèles de Markov avec une métrique de distance qui permet d'évaluer la qualité des processus fouillés. Les workflows modélisés dans les réseaux de Pétri sont transformés en MMC mais les états cachés intentionnels des processus ne sont pas pris en compte.

Tous ces travaux considèrent les états cachés du MMC comme des instances du processus alors que nous considérons les états cachés comme des intentions générant les séquences d'activités observées. Nous n'allons pas modéliser un processus comme un ensemble de tâches comme dans les réseaux de Pétri, mais comme des intentions qui sont à l'origine des séquences d'activités. En outre, de nombreuses techniques classiques de classification comme SVM (Support Vector Machine) (Joachims *et al.,* 1997) ne peuvent pas traiter le bruit (données incomplètes ou non pertinentes) et ne prennent pas en compte la variabilité des séquences de données. Elles n'acceptent que des séquences de longueur donnée, alors que les séquences d'activités ont des longueurs variables en fonction des objectifs des acteurs.

3. Contribution

Notre proposition consiste en (i) la modélisation d'un modèle de processus intentionnel utilisant un MMC (adapté au modèle de processus intentionnel car modélisant les états cachés, i.e. les intentions trouvées dans les données observables) ; (ii) l'estimation des paramètres du MMC basée sur les traces obtenues à partir des applications du modèle de processus intentionnel ; (iii) la prédiction des intentions futures et par conséquent les activités futures en utilisant les paramètres estimés de MMC ; et (iv) l'évaluation des intentions associées à une séquence d'activités en utilisant l'algorithme de Viterbi (AV), compte tenu des paramètres du MMC.

Notre travail repose sur l'hypothèse que la réalisation des traces d'activités suit un processus stochastique. Un processus stochastique représente une évolution discrète ou continue d'une variable aléatoire. Ainsi, nous devons choisir un modèle statistique qui permet (a) de connaître les séquences observées, si elles sont importantes ou seulement le résultat d'accidents, (b) d'analyser des séquences observées au fil du

temps, (c) de modéliser des états latents de ces séquences observées, (d) d'extraire les caractéristiques des séquences observées et latentes. Parmi les modèles probabilistes, nous sélectionnons le MMC (Juang *et al.,* 1991) qui satisfait pleinement les critères évoqués. Comme expliqué dans la section précédente, les autres méthodes ont l'inconvénient de ne pas correspondre au modèle intentionnel. Cependant, plusieurs applications de MMC à la fouille de processus (Rozinat *et al.,* 2008) (Herbst *et al.,* 1998) considèrent les états cachés comme étant instances du processus. Nous nous différencions de ces travaux en considérant les états cachés comme étant les intentions des utilisateurs du processus. La contribution de cet article consiste donc à appliquer un MMC pour découvrir les intentions de traces d'activités, ce que nous nommons *fouille d'intention*.

Un MMC est un type particulier de processus stochastique qui capture les relations entre une séquence observable (les activités) et des états cachés (les intentions associées). Plus précisément, le cadre du MMC permet de se concentrer sur plusieurs problèmes tels que a) Quelle est la probabilité d'une séquence d'activités donnée ? (b) Quelles sont les intentions les plus probables liées à une séquence d'activités donnée? Cette dernière tient une place importante dans le cadre de notre travail car nous cherchons à trouver les intentions cachées qui sont associées à une séquence de traces d'activités. Nous présentons dans les sections suivantes un aperçu théorique des MMC.

3.1. *Modèles de Markov Cachés (MMC)*

Un MMC est un formalisme de modélisation statistique du signal qui permet la modélisation d'une séquence comme un nombre fini d'états Les systèmes modélisés par un MMC sont basés sur deux processus de Markov complémentaires. Pour comprendre ce qu'est un processus de Markov, définissons $(X_1, \dots, X_T)$ comme une suite de variables aléatoires de longueur T générées par une chaîne de Markov d'ordre m (ou avec une mémoire m), où m est fini. Le processus de génération s'écrit :

$$\mathbb{P}\left(X_t = x_t \middle| X_{t-1} = x_{t-1}, X_{t-2} = x_{t-2}, \dots, X_1 = x_1\right) = \mathbb{P}\left(X_t = x_t \middle| X_{t-1} = x_{t-1}, X_{t-2} = x_{t-2}, \dots, X_{t-m} = x_{t-m}\right) \quad pour\ t > m. \qquad [1]$$

Cela signifie que pour une chaîne de Markov d'ordre m, le passage à l'état suivant ne dépend que des m états antérieurs. Le choix de m indique à quelle distance dans le passé on doit remonter pour connaître la probabilité de l'état suivant. Dans le contexte d'un MMC, nous avons besoin de définir deux processus de Markov : un modélisant l'état caché du système et le second modélisant les observations du système, sachant qu'il dépend des états cachés. Dans le cadre de notre travail, les états cachés sont les intentions et les observations du système sont les traces d'activités. La structure topologique du MMC permet de définir, selon le contexte, la dépendance entre les états cachés et/ou les observations du passé et les états cachés et/ou les observations du futur.

Choisir le bon ordre des chaînes de Markov dans un MMC est un défi, car il détermine le comportement du MMC. Nous allons travailler ici avec un modèle *M1M0*. Cela signifie que, parmi les différents ordres de chaînes de Markov, nous nous

appuyons sur deux chaînes de Markov, l'une d'ordre 1 et l'autre d'ordre 0. Une chaîne de Markov d'ordre 1 est un processus pour lequel le passage à l'état suivant ne dépend que de l'état actuel (appelé aussi propriété de Markov). Une chaîne de Markov d'ordre 0 est un processus pour lequel le passage à l'état suivant ne dépend d'aucun état.

Définition 1 : Processus caché. A l'instant t, l'état caché du système I_t ne dépend que de l'état caché du système au moment précédent I_{t-1}. Cette définition s'applique aux intentions (modèle *M1*). Soit une séquence d'intentions désignée par $I_{1:T} = (I_1, \dots, I_T) \in S^T$, S étant l'ensemble des intentions et T étant la longueur de la séquence. Une chaîne de Markov homogène, dont le paramètre est désigné par q, modélise le processus caché des intentions :

$$q(u, v) = \mathbb{P}(I_{t+1} = v | I_t = u), \forall u, v \in S, \quad [2]$$

et

$$q(u) = \mathbb{P}(I_1 = u), \forall u \in S \quad [3]$$

Le paramètre q en (3) contient les probabilités d'intention à l'état initial et dans (2), les probabilités de transition pour les intentions suivantes.

Définition 2 : Processus observé. A un instant t donné, l'observation A_t ne dépend d'aucune séquence précédente observée. Cette définition se rapporte à la séquence d'activités (modèle *M0*). On note une séquence d'activités des acteurs par $A_{1:T} = (A_1, \dots, A_T) \in R^T$, R étant l'ensemble des activités. La probabilité d'émission p d'une observation $a \in R$ pour une intention donnée $u \in S$, est donnée par:

$$p_u(a) = \mathbb{P}(A = a | I = u) \quad [4]$$

P et q sont les paramètres de MMC (probabilités de transition et d'émission). Les probabilités de transition sont les probabilités d'un état caché au temps t pour atteindre un autre état caché à l'instant $t + 1$ (ou de rester dans le même état). Les probabilités d'émission sont la distribution de probabilité conditionnelle des variables observées dans un état caché donné à l'instant t. Ce modèle est un modèle *M1M0* puisque le processus caché est une chaîne de Markov d'ordre 1 et le processus observé est une chaîne de Markov d'ordre 0. Autrement dit, cela signifie que le choix de l'exécution d'une activité pendant la mise en œuvre d'un modèle de processus n'est pas isolée, mais en corrélation avec d'autres activités préalables.

A titre d'exemple, considérons un cas à trois intentions cachées $\{I_1, I_2, I_3\}$ et quatre activités observées $\{a_1, a_2, a_3, a_4\}$. Le modèle *M1M0* associé est décrit par la loi de probabilité de l'intention initiale et par deux matrices: une matrice 3×3 pour les probabilités de transition d'intentions cachées - I [5] - et une matrice 3×4 représentant les probabilités d'émission des activités observées pour chaque intention - A [6].

$$I = \begin{pmatrix} q(I_1, I_1) & q(I_1, I_2) & q(I_1, I_3) \\ q(I_2, I_1) & q(I_2, I_2) & q(I_2, I_3) \\ q(I_3, I_1) & q(I_3, I_2) & q(I_3, I_3) \end{pmatrix} \quad [5] \qquad A = \begin{pmatrix} p_1(a_1) & p_1(a_2) & p_1(a_3) & p_1(a_4) \\ p_2(a_1) & p_2(a_2) & p_2(a_3) & p_2(a_4) \\ p_3(a_1) & p_3(a_2) & p_3(a_3) & p_3(a_4) \end{pmatrix} \quad [6]$$

3.2. *Estimation des paramètres*

Il existe deux approches pour estimer les paramètres du MMC : *supervisée* et *non supervisée*. L'approche supervisée est utilisée lorsque la segmentation de séquences d'activités en intentions est connue alors que l'approche non supervisée est utilisée si aucune segmentation de séquence d'activités n'est connue. Nous adoptons l'approche *supervisée* dans cet article. L'approche supervisée comporte deux phases : une phase d'apprentissage pour entraîner l'algorithme avec des séquences d'activités suffisamment longues pour trouver les paramètres du MMC et une phase pour l'évaluation des intentions relatives à des séquences d'activités données.

L'estimation des paramètres du MMC consiste à trouver la répartition de probabilité des traces : le couple (p, q) défini dans [2] et [4]. Si une séquence d'activités $A_{1:T}$ est disponible et les intentions correspondantes connues, le calcul des estimations pour (p, q) consiste à utiliser l'estimation du maximum de vraisemblance (Enders, 20004). Cela permet d'estimer les paramètres $q(u, v)$ et $p_u(a)$ pour qu'ils maximisent la probabilité que les intentions $I_{1:T}$ génèrent les séquences d'activités $A_{1:T}$. Cela consiste à compter le nombre de transitions d'une intention à l'autre et le nombre d'apparitions de chaque activité au cours des intentions, comme indiqué ci-dessous :

$$\hat{q}(u, v) = \frac{Num(u,v)}{\sum_{x \in S} Num(u,x)}, \qquad [7] \qquad \hat{p}_u(a) = \frac{Num(a|u)}{Num(a)}, \qquad [8]$$

où $Num(u, v)$ dans [7] désigne le nombre de transitions de l'intention u vers l'intention v, $Num(a)$ dans [8] désigne le nombre d'apparitions de l'activité a et $Num(a|u)$ désigne le nombre d'apparitions de l'activité a pendant l'intention u. Ces paramètres permettent d'apprendre la répartition des activités et des intentions dans le modèle de processus. Cette phase d'apprentissage est nécessaire pour l'étape suivante afin d'évaluer les intentions les plus probables liées à une séquence d'activités donnée.

3.3. *Evaluation des séquences d'intentions*

Une fois les paramètres estimés, il faut identifier l'ensemble le plus probable d'intentions associées à une séquence d'activités. Soit une séquence d'activités $A_{1:T}$ de longueur T, on peut générer toutes les intentions possibles de longueur T. Ensuite, pour chaque intention $I_{1:T}$, on peut calculer la probabilité $\mathbb{P}(A_{1:T}|I_{1:T})$. Toutefois, il s'agit d'une recherche par force brute et cette technique ne peut être utilisée pour comparer toutes les intentions possibles car trop complexe. Par exemple, si le nombre d'intentions est C, la complexité est C^T, et augmente de façon exponentielle avec *T*.

Au lieu d'une recherche par force brute, l'AV (Forney, 1973) peut être utilisé pour obtenir, à partir d'une séquence observée donnée, la séquence d'intentions cachées la plus probable. Cet algorithme est capable de calculer la probabilité qu'une observation (ou une intention) ai été changée en une autre, et simplifie radicalement la complexité de la recherche de la séquence originale cachée la plus probable. De ce fait, la complexité exponentielle devient linéaire. Pour utiliser l'AV, il est nécessaire de connaître les paramètres estimés. Nous noterons qu'une séquence donnée d'activités, de

longueur T, peut être générée par de nombreuses intentions de la même longueur. Néanmoins, l'une des séquences a la plus forte probabilité d'émergence, c'est la séquence la plus probable d'intentions générant la séquence d'activités associées.

Étant donné une séquence de traces d'activités $A_{1:T}$, les paramètres estimés $\hat{p}_u(a)$ et $\hat{q}(u,v)$ et les probabilités initiales pour chaque intention, l'AV tente de trouver la séquence des intentions cachées associées $\bar{I}_{1:T}$ qui maximise $\mathbb{P}(I_{1:T}|A_{1:T})$. Nous pouvons également écrire le problème comme suit.

$$\bar{\mathrm{I}}_{1:\mathrm{T}} = \arg\max_{\mathrm{I}_{1:\mathrm{T}}} \mathbb{P}(\mathrm{I}_{1:\mathrm{T}}|\mathrm{A}_{1:\mathrm{T}}) \qquad [9]$$

L'AV génère une séquence d'intentions, ce qui permet de prendre en compte l'historique des activités des utilisateurs. Dans la section suivante, nous détaillons la façon dont nous utilisons un MMC dans une étude de cas spécifique.

4. Application du MMC sur des données réelles

Nous avons effectué une expérience afin d'obtenir des traces d'activités de 71 étudiants de Master MIAGE. Le modèle de processus utilisé pour les guider a été spécifié intentionnellement grâce à une Carte. Afin d'obtenir des traces, nous avons enregistré les activités réalisées tout au long de la création d'un diagramme entité-association selon un modèle de processus intentionnel prescrit par un site Web.

Le modèle de processus utilisé (figure 1) est une carte (Rolland *et al.,* 1999) permettant de guider les utilisateurs dans la création de diagrammes entité-association, basé sur (Assar *et al.,* 2000) mais simplifié pour cette expérimentation. Les utilisateurs peuvent sélectionner onze stratégies pour atteindre trois intentions (*Spécifier une entité*, *Spécifier une association* et *Arrêter)*. Une carte est représentée comme un graphe orienté. Chaque flèche représente une stratégie que l'utilisateur peut choisir pour réaliser une intention (définie comme un nœud) en fonction de sa situation. Par exemple, si la situation actuelle est *Démarrer* et l'intention de l'utilisateur est de *Spécifier une entité*, il n'y a qu'une seule stratégie (par complétude) pour réaliser cette intention. Lorsque la situation est *Spécifier une entité*, il y a 4 stratégies (par complétude, par généralisation, par spécialisation, par normalisation) pour atteindre la même intention. Il est possible de progresser dans le processus en choisissant les stratégies qui conduisent à des intentions, mais une fois qu'*Arrêter* est atteint, la mise en œuvre du processus prend fin.

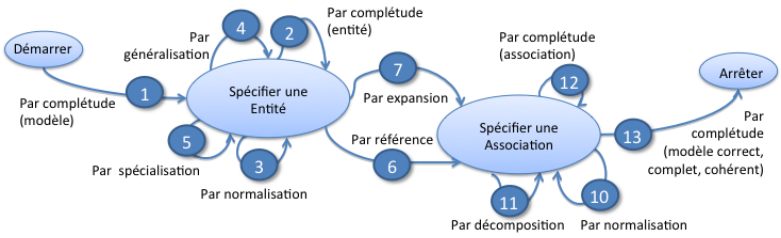

Figure 1. *Modèle de processus intentionnel.*

Pour atteindre une intention en suivant une stratégie, les utilisateurs doivent effectuer des actions. Le terme d'action diffère d'activité : les actions sont effectuées par les utilisateurs, mais les activités résultent de l'enregistrement des actions de l'utilisateur par un outil. Il existe quinze actions liées à la carte : le tableau 1 montre en détail le lien entre chaque section et les actions connexes. Plusieurs actions apparaissent simultanément dans les deux intentions *Spécifier une entité* et *Spécifier une association* comme par exemple la *suppression d'un attribut.* En effet, une même action peut être exécutée afin de réaliser différentes intentions. En conséquence, il n'est pas trivial de connaître les intentions de tous les acteurs lors de l'exécution d'une action. Notre but ici est de fournir une méthode pour trouver les intentions cachées derrière une séquence d'activités.

Section de la carte		Trace des Actions Associées	Codes des Actions
1	Spécifier une entité Par complétude (modèle)	Créer une entité	A1
2	Spécifier une entité Par complétude (entité)	Créer un attribut Lier un attribut à une entité	A3, A4
3	Spécifier une entité Par normalisation	Supprimer attribut Supprimer le lien d'un attribut à une entité	A10, A15
		Supprimer une entité Supprimer un attribut*, (Supprimer une association, Supprimer un attribut *) *	A11, A10*, (A12, A10*)*
		Définir une clé primaire	A9
4	Spécifier an entité Par généralisation	Créer une entité Créer un lien de généralisation	A1, A7
5	Spécifier an entité Par spécialisation	Créer une entité Créer un lien de spécialisation	A1, A8
6	Spécifier an association Par référence	Supprimer un attribut Créer une entité Créer une association lier une association à une entité, lier une association à une entité	A10, A1, A2, A6, A6
7	Spécifier an association Par expansion	Créer une association	A2
10	Spécifier an association Par normalisation	Supprimer une association (Supprimer un attribut, Supprimer le lien d'un attribut à une association)*	A12, (A10, A15)*
		Supprimer un attribut	A10
11	Spécifier an association Par décomposition	Créer association Associer une association à une entité Associer une association à une entité	A2, A6, A6
12	Spécifier une association Par complétude (association)	Créer un attribut Associer un attribut à une association	A3, A5
13	Arrêter Par complétude (final)	Vérifier cohérence Vérifier la complétude	A13, A14

Tableau 1. Sections et traces des actions associées (* : action itérative).

Pour être en mesure d'enregistrer les traces des acteurs, nous avons développé une application web avec HTML, PHP, JavaScript et MySQL. L'application enregistre les traces des commandes exécutées par les utilisateurs lors de la création de leurs schémas. Chaque événement contient des informations sur l'action qui a été exécutée, l'instance de processus auquel elle appartient, et l'horodatage de l'exécution de l'action. Le modèle utilisé pour stocker les traces peut être trouvé dans (Hug *et al.,* 2012).

Actions Associées	Code
Créer une entité	A1
Créer une association	A2
Créer un attribut	A3
Associer un attribut à une entité	A4
Associer un attribut à une association	A5
Associer une association à une entité	A6
Créer un lien de généralisation	A7
Créer un lien de spécialisation	A8

Actions Associées	Code
Définir une clé primaire	A9
Supprimer un attribut	A10
Supprimer une entité	A11
Supprimer une association	A12
Vérifier le cohérence	A13
Vérifier la complétude	A14
Supprimer un lien	A15

Tableau 2. Actions associées et codes.

4.1. *Estimation des paramètres du MMC*

Les utilisateurs suivaient un modèle de processus intentionnel, il était facile de connaître les intentions cachées derrière les actions. L'ensemble des données est constitué d'une séquence d'actions et des intentions associées. Au total, nous avons enregistré 4141 traces d'actions produites par 71 étudiants. En conséquence, la longueur de la séquence des traces d'actions est de 4141. La connaissance des intentions nous permet de travailler dans le cadre de la fouille d'intention supervisée pour estimer les paramètres du MMC (coefficients des matrices A et I). Comme le modèle comprend 3 intentions et 15 actions, la taille de I est de 3×3 et la taille de A est 3×15. Les coefficients de la matrice de transition I correspondent au nombre de transitions d'une intention à l'autre et les coefficients de la matrice A au nombre de fois où chaque trace d'action apparaît pour chaque intention.

Bien sûr, la qualité des coefficients estimés dépend de la longueur des séquences utilisées pour calculer les estimations. Si la longueur des séquences est trop courte, les séquences ne captureront pas tous les comportements des utilisateurs et les coefficients estimés seront de mauvaise qualité. Au contraire, si la longueur des séquences est suffisamment longue, nous pourrons capturer les comportements typiques des étudiants et les coefficients estimés seront satisfaisants. Ce phénomène se vérifie effectivement avec notre base de données. Premièrement, nous estimons les matrices A et I avec la longueur totale des séquences, c'est à dire, 4141. Puis, pour onze différentes longueurs de séquence (1, 5, 10, 20, 30, 40, 50, 100, 200, 300 et 400), nous estimons les matrices A et I. On obtient onze couples de matrices de différentes qualités. L'erreur d'estimation des coefficients diminue avec la longueur des séquences. Par exemple, pour une estimation sur des séquences de longueur 20, l'erreur est de 0,15 pour les paramètres de

$\hat{A}$ et 0.08 pour les paramètres de $\hat{I}$. Néanmoins, pour des séquences de longueur 400, les erreurs sur les matrices de transition et d'émission diminuent à 0,05 et 0,001 respectivement. Les équations 10 et 11 donnent respectivement la matrice estimée des émissions d'actions pour chaque intention et la matrice estimée de probabilité de transition pour les intentions (pour des séquences d'une longueur de 4141). Ces résultats constituent la meilleure estimation des paramètres.

$$\hat{A} = \begin{pmatrix} 0.1276 & 0 & 0.4020 & 0.4020 & 0 & 0 & 0.0008 & 0.0025 & 0.0068 & 0.0240 & 0.0102 & 0.0102 & 0 & 0 & 0.0138 \\ 0.0436 & 0.4782 & 0.1239 & 0 & 0.1239 & 0.0873 & 0 & 0 & 0 & 0.0768 & 0 & 0.0332 & 0 & 0 & 0.0332 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 & 0 \end{pmatrix} \quad [10]$$

$$\hat{I} = \begin{pmatrix} 0.9438 & 0.0522 & 0.0040 \\ 0.2775 & 0.6771 & 0.0454 \\ 0.4588 & 0 & 0.5412 \end{pmatrix} \quad [11]$$

Dans l'équation 11, il est important d'attirer l'attention sur le coefficient $\hat{I}_{31}$ (0.4588) qui voudrait dire qu'il y a une transition possible entre *Arrêter* et *Spécifier une entité*, ce qui n'est pas possible. La raison pour laquelle ce coefficient n'est pas nul est que nous traitons les traces des différents utilisateurs de manière séquentielle. Par conséquent, lorsque la trace d'un étudiant se termine par l'intention *Arrêter*, elle est suivie par la trace d'un autre étudiant qui commence par *Spécifier une entité*.

En ce qui concerne les intentions, la figure 2 illustre le modèle de processus obtenu grâce aux probabilités de transitions. Nous pouvons déduire de ces résultats que lorsque les acteurs réalisent *Spécifier une entité*, ils ont une forte probabilité (0,9438) de continuer à exercer cette intention. Ils peuvent passer à *Spécifier une association* avec une probabilité de 0,0522, et ils peuvent aller directement à *Arrêter* avec une probabilité de 0,004. Cette dernière transition est très surprenante car elle n'est pas permise dans le modèle donné sur la figure 1. Cela signifie que certains des étudiants ont dévié du modèle de processus prescrit. Quand ils réalisent *Spécifier une association*, ils continuent principalement à garder la même intention avec une probabilité de 0,6771 ou ils passent à *Spécifier une entité* avec une probabilité de 0,2775. Une fois de plus, cette transition n'est pas présente dans le modèle de processus prescrit, il s'agit donc d'un écart. Enfin, lorsque les étudiants réalisent *Arrêter*, la seule intention suivante possible est *Arrêter,* ce qui est conforme au modèle prescrit puisque deux actions différentes sont effectuées dans cette intention : *Vérifier la cohérence* et la *Vérifier la complétude*. En conséquence, la première action est toujours suivie de *la deuxième*. Une fois qu'un étudiant a effectué ces deux actions, sa trace se termine. Par conséquent, l'intention suivante provient d'un autre étudiant qui commence le processus avec la première intention (*Spécifier une entité*).

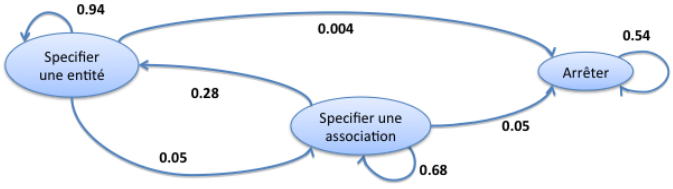

Figure 2. *Modèle du processus découvert, obtenu par les probabilités de transition.*

En ce qui concerne la matrice d'émission, on peut vérifier que les actions $\{A_2, A_4, A_5, A_6, A_7, A_8, A_9, A_{11}, A_{13}, A_{14}\}$ n'apparaissent que pour une intention

unique alors que les autres actions $\{A_1, A_3, A_{10}, A_{12}, A_{15}\}$ peuvent apparaître pour plusieurs intentions. Cela est conforme à la définition des traces d'activités figurant dans la section 4 et signifie que trouver les intentions associées à ces dernières actions est une tâche difficile. Avec une méthode triviale, on ne peut pas distinguer de quelles intentions sont issues ces actions. Pour le comportement global des acteurs, nous avons appris que les actions les plus exécutées pour *Spécifier une entité* sont A_3 et A_4 : *Créer un Attribut* et le *Associer l'attribut à l'entité*. L'action la plus exécutée pour *Spécifier une association* est *Créer une association*. Cette connaissance est utile pour analyser les comportements des étudiants.

4.2. *Récupération des intentions pour des traces d'actions aléatoires*

Maintenant que nous avons estimé les paramètres du MMC, nous pouvons les utiliser pour trouver les intentions cachées derrière n'importe quelle séquence de traces d'actions à l'aide de l'AV. Notre objectif ici est de déterminer la longueur minimale des séquences d'estimation des paramètres du MMC qui permet d'atteindre des performances satisfaisantes avec l'AV. Par conséquent, nous comparons les performances de l'AV avec les onze couples de matrices estimées, obtenues avec les longueurs de séquences de 1, 5, 10, 20, 30, 40, 50, 100, 200, 300, et 400. Le protocole de comparaison est le suivant : pour 1000 séquences d'actions tests de longueur 1500, nous appliquons l'AV, pour chaque couple de matrices. Pour chacune des séquences d'actions tests, nous obtenons onze séquences d'intentions estimées. En comparant les intentions estimées avec les intentions réelles et en moyennant les résultats sur les 1000 réalisations, nous pouvons obtenir le pourcentage d'erreur associé à chaque couple de matrices. Il est intéressant de noter que le nombre d'erreurs de l'AV diminue quand l'estimation des matrices est bonne. Plus précisément, avec des matrices estimées à l'aide des séquences de longueur 200, nous avons un pourcentage d'erreur inférieur à 5%, ce qui est satisfaisant.

4.3. *Validation*

Afin d'évaluer les résultats associés à l'application du MMC sur des données réelles, nous choisissons les mesures de validation suivantes : le rappel, la précision et la F-mesure (une combinaison du rappel et de la précision) (Goutte *et al.,* 2005). Avant d'expliquer ces mesures, il est nécessaire d'introduire quelques notions : le Vrai Positif (VP) représente le nombre d'intentions correctement affectées par l'AV à la bonne classe d'intention ; le Faux Négatif (FN) représente le nombre d'intentions qui n'ont pas été affectés à la bonne classe d'intention ; le Faux Positif (FP) représente le nombre d'intentions incorrectement affectées à la bonne classe d'intention. Ces termes permettent de déterminer si l'estimation est en accord avec les intentions réelles. Les mesures de rappel, précision et F-mesures sont définies comme suit :

$$\text{Rappel} = \frac{\text{VP}}{\text{VP+FN}} \quad [13] \qquad \text{Précision} = \frac{\text{VP}}{\text{VP+FP}} \quad [14] \qquad \text{Fmesure} = \frac{2\times\text{precision}\times\text{recall}}{\text{precision+recall}} \quad [15]$$

Le rappel (13) est le rapport entre le nombre d'intentions correctement identifiées par l'AV et le nombre d'intentions dans le jeu de données. Notons que cela ne prend pas en compte le nombre d'intentions faussement identifiées par l'algorithme. La précision (14) désigne le rapport entre le nombre d'intentions correctement identifiées par l'AV et le nombre d'intentions identifiées par l'algorithme. En général, il est possible d'augmenter la précision afin de réduire le rappel et vice-versa. La F-mesure (15) est une combinaison de la précision et du rappel. Cette mesure est également nommée mesure F-1, car la précision et le rappel sont pondérés de façon égale. La mesure F-1 nous donne l'efficacité de la récupération de l'intention, en considérant une importance identique pour le rappel et la précision.

Nous avons donc calculé le rappel, la précision et la F-mesure pour les trois intentions. La figure 3a montre ces mesures, moyennées sur les 1000 séquences de test pour l'intention 1 : *Spécifier une entité*. Les trois courbes se stabilisent à une longueur de séquence d'estimation de 200. Cela signifie qu'à partir de cette longueur, l'AV fournit des résultats stables. Le résultat du rappel exprime le fait que l'algorithme trouve 99,50% des actions liées à *Spécifiez une entité* - presque toutes les activités liées à cette intention sont identifiées. Mais l'algorithme associe-t-il des actions à une intention alors qu'elles appartiennent en fait à d'autres ? La valeur de la précision apporte la réponse à cette question. Le résultat de la précision se stabilise à 97%, ce qui signifie que 3% des actions sont associées à *Spécifier une entité*, alors qu'elles appartiennent à d'autres intentions. La F-mesure est un compromis entre le rappel et la précision qui, comme mentionné plus haut, attribue une pondération identique à ces deux mesures. Nous obtenons une F-mesure de 0.98 pour l'intention *Spécifier une entité*. En d'autres termes, lorsque l'intention de l'utilisateur est *Spécifier une entité*, l'AV est capable de trouver cette intention avec une excellente exactitude.

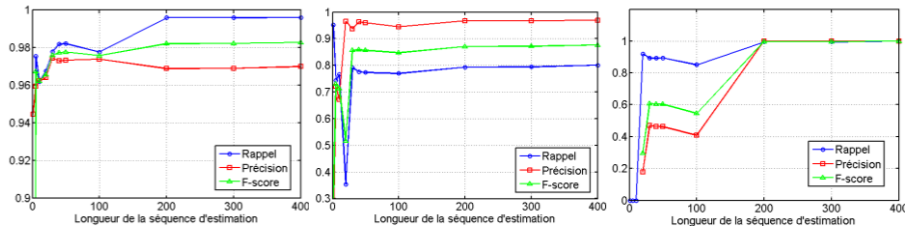

Figure 3. *Rappel, précision et F-mesure pour l'intention 1, 2 et 3.*

Nous avons appliqué les mêmes mesures pour l'intention *Spécifier une association* (figure 3b). Dans ce cas, les trois courbes se stabilisent également autour d'une longueur de séquence d'estimation de 200. À partir de cette longueur, la valeur du rappel est de 0,80. Cela signifie que l'AV retrouve 80% des actions liées à l'intention de *Spécifier une association*. Le rappel est inférieur à celui de l'intention précédente parce que l'AV identifie des actions liées à l'intention 2 comme des actions associées à l'intention 1. C'est pourquoi le rappel de l'intention 2 est de 80%. Cependant, la précision de l'intention 2 est meilleure et se stabilise à 96%. En d'autres termes, 4% des actions sont associées à *Spécifier une association* alors qu'elles appartiennent

probablement à *Spécifier une entité*. La raison de cette confusion est qu'il existe des actions communes à ces deux intentions. L'exactitude de la récupération pour l'intention *Spécifier une association*, F-1, est mesurée à 0,87. En d'autres termes, lorsque l'intention de l'utilisateur est *Spécifier une association*, l'AV est capable de retrouver cette intention avec une très bonne exactitude. En ce qui concerne la troisième intention, *Arrêter*, les trois mesures se stabilisent toutes à 1, comme le montre la figure 7c, car les actions associées à cette intention ne sont pas communes à d'autres intentions. Il est alors trivial d'identifier cette intention. Le tableau suivant montre les résultats obtenus pour les mesures de rappel, de précision et la F-mesure pour chaque intention prédéfinie.

	Rappel	**Précision**	**F-MESURE**
Intention 1	99,50%	97%	0.98
Intention 2	80%	96%	0.87
Intention 3	100%	100%	1

Tableau 3. Rappel, précision et F-mesure pour les intentions.

Ces résultats montrent que nous avons pu trouver les intentions cachées derrière les activités avec une précision satisfaisante. Les résultats de la F-mesure démontrent l'efficacité et les performances du MMC appliqué à notre jeu de données.

5. Perspectives et conclusion

Cet article montre que le MMC est une stratégie de modélisation efficace pour extraire les intentions de traces d'activités. Nous avons utilisé une approche supervisée et les résultats présentés dans notre première expérimentation sont prometteurs. Nous avons réussi à trouver les intentions cachées derrière les activités avec une efficacité, des performances et une exactitude satisfaisantes. Par ailleurs, nous avons également obtenu les probabilités de passage d'une intention à l'autre et les probabilités d'apparition des activités dans chaque intention, première étape pour découvrir le modèle de processus intentionnel.

Une autre contribution de ce travail est la définition d'un nouveau domaine de recherche appelé *fouille d'intention*, en lignée avec des travaux antérieurs de notre équipe sur les modèles de processus intentionnels (Nature (Jarke *et al.,* 1999), Carte (Rolland *et al.,* 1999)), les modèles de processus d'alignement (InStAll (Thevenet *et al.,* 2007)) et les processus d'orientation (Mentor (Grosz, 1994), Crews-L'Ecritoire (Tawbi *et al.,* 1998)). L'objectif principal de la fouille d'intention est de comprendre le comportement des acteurs à partir de logs d'événements afin de leur offrir un meilleur guidage à travers la mise en œuvre des processus.

Dans de futurs travaux, nous allons essayer de faire face à la même problématique de découverte d'intentions en utilisant cette fois une approche non supervisée. Dans ce type de démarche, nous ne connaitrons la segmentation d'aucune séquence d'activités. En d'autres termes, les liens entre les intentions et les séquences d'activités sont inconnus. Nous utiliserons l'algorithme de Baum-Welch (Forney, 1973) qui est une variante plus générale de l'algorithme d'espérance-maximisation (Baum *et al.,* 1970).

Nous avons également l'intention d'étudier la probabilité d'apparition d'une séquence d'activités donnée en supposant que les paramètres de MMC sont connus. Cette perspective est intéressante car différentes intentions peuvent conduire à la même séquence d'activités. Comme le nombre d'intentions possibles augmente exponentiellement avec la longueur de la séquence, le calcul de cette probabilité peut être assez complexe et l'une des solutions pour surmonter ce problème pourrait être d'utiliser les algorithmes d'inférence pour les MMC Forward-Backward (van der Aalst *et al.,* 2004).

Remerciements. Nous remercions Sébastien Sion pour le développement du site web et nos étudiants du Master MIAGE Sorbonne pour leur participation aux expérimentations.

6. Bibliographie

Agrawal, R., Gunopulos, D., Leymann, F., "Mining Process Models from Workflow Logs", 6th Int. Conf. on Extending Database Technology: Advances in Database Technology, LNCS 1377, 1998, p.469-483.

Assar, S., Ben Achour, C., Si-Said, S., "Un Modèle pour la spécification des processus d'analyse des Systèmes d'Information", *INFORSID, 2000*.

Baum, L. E., Petrie, T., "Statistical Inference for Probabilistic Functions of Finite State Markov Chains", *The Annals of Mathematical Statistics,* Vol. 37, n°6, 1966, p. 1554-1563.

Baum, L. E., Petrie, T., Soules, G., Weiss, N., "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains", *Ann. Math. Statist.,* Vol.41, n°1, 1970, p. 164-171.

Biermann, A.W., Feldman, J.A., "On the Synthesis of Finite States Machines from Samples of Their Behavior", *IEEE Trans. on Computers*, Vol.21, n°6, 1972, p. 592-597.

Cook, J., Wolf, A., "Automating process discovery through event-data analysis", 17th Int. Conf. on *Soft. Eng.*, p. 73-82, 1995.

Cook, J., Wolf, A., "Discovering Models of Software Processes from Event-Based Data", *ACM Trans. on Soft. Eng. and Meth.*, Vol.7, n°3, 1998.

Das, S., Mozer, M.C., "A unified Gradient Descent/Clustering Architecture for Finite State Machine Induction", 6th *Advances in Neur. Inf. Proc. Sys.* 1993, Morgan Kaufmann, 1994.

Eibe, F., Witten, I.H., *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd Ed., Morgan Kaufmann Publishers Inc., 2005, San Francisco, USA.

Enders, W., *Applied econometric time series*, Hoboken, N.J., Wiley, cop. 2004.

Enright, A., van Dongen, S., Ouzounis, C., "An efficient algorithm for large-scale detection of protein families", *Nucleic Acids Research*, Vol.30, n°7, p.1575-1584, 2002.

Ferreira, D., Zacarias, M., Malheiros,M., Ferreira, P., "Approaching Process Mining with Sequence Clustering: Experiments and findings", *BPM 2007*. LNCS, Vol. 4714, 2007, p. 360-374. Springer, Heidelberg.

Forney, G.D., "The Viterbi Algorithm", *IEEE*, Vol.61, n°3, p. 268-278, 1973.

Goutte, C., Gaussier, E., "A Probabilistic Interpretation of Precision, Recall and F-score, with Implication for Evaluation", *27th European Conf. on Inf. Retrieval*, p. 345-359, 2005.

Greco, G., Guzzo, A., Pontieri, L., "Mining Hierarchies of Models: From Abstract Views to Concrete Specifications", *BPM 2005*, 2005.

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, 2006, p. 1010-1027.

Grosz, G., "MENTOR: a Step Forward in Guidance for Information System Development", *5th Workshop on the Next Generation of CASE Tools*, 1994, Utrecht, Pays Bas.

Hug, C., Deneckère, R., Salinesi, C., "Map-TBS: Map process enactment traces and analysis", *RCIS'12*, Valencia, Espagne, 2012.

Herbst, J., Karagiannis, D., "Integrating Machine Learning and Workflow Management to Support Acquisition and Adaptation of Workflow Models", 9th Int. *Workshop on Database and Expert Systems Applications*, 1998.

Jarke, M., C. Rolland, C., Sutcliffe, A., Domges, R., *The NATURE of Requirements Engineering*, Shaker Verlag, 1999, Aachen.

Joachims, T., Text categorization with support vector machines, Technical report, LS VIII, N°23, University of Dortmund, 1997.

Juang, B.H., Rabiner, L.R., "Hidden Markov Models for Speech Recognition", *Technimetrics by American Stat.Assoc. and American Society for Quality*, Vol.33, n°3, 1991, p. 251-272.

Minseok, S., Günther, C.W., van der Aalst, W.M.P., "Trace Clustering in Process Mining Business Process Management Workshops", LNBIP, 2009, Vol.17, p. 109-120.

Rolland, C., Prakash, N., Benjamen, A., "A Multi-Model View of Process Modelling", *RE* 99, Springer-Verlag London Ltd, 1999.

Rozinat, A., Veloso, M., van der Aalst, W.M.P., "Evaluating the quality of discovered process models", 2nd Intl. Workshop on the *Induction of Process Models*, 2008, p. 45-52, Antwerp, Belgique.

Tawbi, M., Souveyet, C., Rolland, C., "L'ECRITOIRE a tool to support a goal-scenario based approach to requirements engineering", *Inf. and Soft. Tech. J.*, Martin Shepperd, Ed., Elsevier Science B.V, 1998.

Thévenet, L.H., Salinesi, C., "Aligning IS to Organization's Strategy: The InStAll Method", Proc. of *CAiSE 2007*, 2007, Trondheim, Norvège.

van der Aalst, W.M.P., Weijters, A.J.M.M., Maruster, L., "Workflow Mining: Discovering Process Models from Event Logs", *IEEE Trans. on Knowl. and Data Eng,* Vol. 16, n°9, 2004, p. 1128-1142.

van der Aalst, W.M.P, Medeiros, A., Weijters, A., "Genetic Process Mining", *Applications and Theory of Petri Nets 2005*, LNCS 3536, Springer, 2005, p.48-69.

van der Aalst, W.M.P., *Process Mining: Discovery, Conformance and Enhancement of Business Processes*, 1st Ed., Springer, 2011, p. 184-352.

van Dongen, B., van der Aalst, W.M.P., "Multi-Phase Process Mining: Building Instance Graphs", *Int. Conf. on Conceptual Modeling*, LNCS 3288, Springer, 2004, p. 362-376.