



HAL
open science

Aide à la navigation dans les parcours d'apprentissage par reconnaissance de procédés et recommandations à base de traces

Joffrey Leblay

► **To cite this version:**

Joffrey Leblay. Aide à la navigation dans les parcours d'apprentissage par reconnaissance de procédés et recommandations à base de traces. Rencontres Jeunes Chercheurs Environnements Informatiques pour l'Apprentissage Humain, Jun 2016, Montpellier, France. hal-01342324

HAL Id: hal-01342324

<https://hal.science/hal-01342324v1>

Submitted on 5 Jul 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Aide à la navigation dans les parcours d'apprentissage par reconnaissance de procédés et recommandations à base de traces

Joffrey Leblay
1^{re} année

Université de La Rochelle, L3I, Pôle Sciences et Technologie Avenue Michel Crépeau 17042 La Rochelle Cedex 1
joffrey.leblay@univ-lr.fr

Résumé

Cet article est un positionnement des travaux en cours sur « *l'aide à l'utilisation des applications interactives* ». Ceux-ci sont centrés sur l'utilisateur par reconnaissance de procédés types, en fonction des comportements et des exécutions passées, dans le domaine des EIAH. Nous présentons l'état de l'art des méthodes dans différents domaines dont nous souhaitons combiner les techniques et les résultats. L'objectif est de montrer en quoi ces travaux peuvent être intéressants pour le domaine d'application des EIAH. Pour cela, nous présenterons des contextes possibles d'applications et nous détaillerons quelques-uns des travaux importants. Ils regroupent plusieurs domaines de recherche comme les Systèmes à Base de Traces (SBT), les Systèmes de Recommandation (SR), et la fouille de procédés ou *process mining*. Notre approche visera ainsi à proposer une contextualisation des traces en vue de l'optimisation du processus global en s'appuyant sur des stratégies d'identification et de recommandation de processus.

Introduction

Actuellement, dans le processus d'apprentissage, certains apprenants abandonnent ou échouent dans leurs parcours à cause de leurs qualifications insuffisantes pour suivre l'apprentissage entrepris. Nos travaux cherchent à identifier les processus d'apprentissage menant à un de ces échecs (l'abandon étant considéré comme une forme d'échec) pour trouver le(s) point(s) commun(s) entre ceux-ci. Une fois qu'ils sont identifiés, il est nécessaire de les reconnaître quand un apprenant les entame. Pour finir, il faut recadrer l'apprenant avec des recommandations adaptées à son profil (le parcours entrepris, les enseignements déjà suivis, etc.).

Lors de la découverte de trajectoire inadaptée, le système notifiera l'apprenant ou le formateur sur cette trajectoire d'éventuel échec. Ou pourra aussi intervenir automatiquement en indiquant au système l'action ou l'enchaînement d'actions qui permettra à l'apprenant ou au formateur de reconfigurer les séquences d'activités pédagogiques. Afin de résoudre ce problème, le système s'appuiera sur plusieurs domaines d'expertise.

Dans le but d'intervenir dans le cas de l'exécution d'un processus d'apprentissage inadapté, il est nécessaire de les reconnaître. Pour cela, le domaine du *process mining*, ou fouille de procédés, a été choisi pour identifier les processus représentant les trajectoires des apprenants dans l'application. Cependant, l'identification de ces processus nécessite une base de connaissances. Pour ce faire, la collecte et l'analyse des traces d'apprentissage permettront de récupérer les informations depuis les expériences

passées des apprenants afin d'en récupérer cette base de connaissances. Ceci permettra au système d'identifier les processus pouvant aboutir à un échec. De plus, en utilisant un réseau constitué de l'entrelacement des processus prévus et représentant les trajectoires nominales, il devient possible de découvrir des anomalies de tous types. Grâce à l'observation de ces anomalies, il devient possible de découvrir de nouvelles méthodes pour certains traitements (comme des raccourcis ou des méthodes alternatives) ; à noter que toutes les méthodes ne sont pas forcément meilleures que l'initiale. Pour finir, afin d'intervenir auprès de l'apprenant, dans le but de le recadrer, le domaine de la recommandation sera utilisé. La recommandation pourra aussi répondre à d'autres objectifs. Elle permettra l'identification des processus à échec en cours, de manière dynamique. Puis, elle permettra de recommander à l'apprenant, au formateur ou directement au système, les processus de correction pour éviter les parcours finaux inadaptes.

Dans ce qui suit, nous commençons par présenter les différents domaines qui vont être utilisés dans nos travaux. Les domaines d'expertise sont décrits dans l'ordre de leur utilisation dans le système que nous souhaitons mettre en place. Ensuite, nous décrivons l'intégration de ces trois domaines pour la résolution de notre problématique. Enfin, nous concluons l'article et donnons des éléments de perspective.

Systèmes à Base de Traces

Les traces présentées dans cette partie sont l'œuvre des travaux du LIRIS de l'université de Lyon par l'équipe TWEAK¹, mais il aurait été aussi possible de citer les travaux de l'équipe EIAH du LIG² de Grenoble ou de IEIAH³ de l'université du Maine.

Le mot traces, selon ce qui est dit dans la thèse (Quiang 2013), peut se définir ainsi : *la trace est une interaction entre deux entités par contact*.

Bien utilisées, les traces fournissent une image des activités du système. Elles sont d'ailleurs le point central de certains systèmes (Mille et Prié 2006). Comprendre les traces permet de pouvoir récupérer des informations d'usages et de les analyser afin de visualiser les interactions entre un ou plusieurs humains et un système d'information (Lund et Mille 2009). Mais, il est aussi possible de récupérer et agréger les informations de plusieurs utilisateurs afin d'obtenir des informations sur les masses (comme dans le crowdsensing, les interfaces utilisatrices intelligentes, etc.) (Bigham et al. 2009). Ce type

¹ <https://liris.cnrs.fr/equipes/?id=75&onglet=resume>

² <http://www.liglab.fr/>

³ <http://www-lium.univ-lemans.fr/fr/content/ingenierie-eiah>

d'agrégation de traces s'effectue dans le cadre de l'étape de transformation (*cf. Transformation*) de traces.

Le principe d'utilisation des traces se décompose en trois étapes :

Collecte

La collecte permet l'observation de l'utilisation d'un système à partir d'une ou plusieurs sources de données, appelées sources de traçage. Elle sert à convertir ou à transformer des informations générées par l'interaction utilisateur/système pendant l'activité en une trace initiale, dite *trace première*. Il y a deux techniques de traçage (Ho 2015). La première est une approche ad hoc qui consiste à tracer tous les observables (l'ensemble des traces enregistrables) ce qui conduit généralement à une grande quantité de traces brutes non directement interprétables par un utilisateur humain (par exemple, les traces collectées par un serveur web ou une application du type Keylogger). La seconde approche fonctionne par instrumentation de l'environnement de travail. Elle consiste à collecter les traces par rapport à un objectif d'observation ainsi que de tracer un ensemble restreint d'observables, jugés pertinents, pendant la session d'observation. Avec cette méthode, il est possible d'associer des labels aux traces pour mieux pouvoir les traiter par la suite. Ces labels s'appellent *obsels*⁴ et sont utilisés dans le cadre des traces modélisées ou *m-traces* (Champin et al. 2013). Elles seront détaillées dans la suite de cette communication. Dans beaucoup de cas, il est plus efficace d'utiliser le système de traçage par instrumentation. En revanche il n'est pas souvent possible de le mettre en place. Souvent, les logiciels utilisés ne peuvent pas être modifiés directement. Il donne donc généralement des traces de l'approche ad hoc ou dont l'instrumentation est inadaptée à l'utilisation voulue. Il est donc nécessaire, dans ces cas-là, de transformer les traces.

Transformation

La transformation récupère les traces obtenues à l'issue de la phase de collecte. Ces traces n'étant pas toujours exploitables directement, il est alors nécessaire d'effectuer une ou plusieurs transformations dont le résultat est une nouvelle trace appelée *trace transformée*. Cela permet à l'utilisateur de comprendre les éléments obtenus et permet de faciliter l'analyse de ces éléments selon des besoins différents. Souvent dans le cas des EIAH les traces sont récupérées de système déjà existant et sans le soutien de l'équipe de développement. Aussi, ces traces n'étant pas récupérées par instrumentation, il est nécessaire de les mettre en forme afin qu'elles soient utilisables par les algorithmes qui vont les exploiter.

Exploitation

L'exploitation est l'étape où les traces obtenues sont utilisées. Elles peuvent être exploitées telles quelles ou bien servir à l'élaboration d'indicateurs plus complexes. Il existe plusieurs méthodes d'exploitation des traces. La visualisation nécessite de faire quelques ajustements sur les traces comme la transformation pour les mettre dans le format demandé ou la transformation sous forme de primitives graphiques qui permet la visualisation des données. Une autre méthode

est le calcul d'indicateurs. Un indicateur est une variable à laquelle est attribuée une série de caractéristiques. Les valeurs de l'indicateur peuvent prendre des formes numériques, alphanumériques ou mêmes graphiques. La valeur possède un statut, c'est-à-dire qu'elle peut être brute (sans unité définie), calibrée ou interprétée (Djouad et al. 2010). Il est aussi possible d'utiliser les techniques de *data mining* (ou fouille de données) qui permettront l'analyse de grandes quantités de traces numériques (Loghin 2008). Grâce à l'application de ces techniques de visualisation, il devient alors possible d'identifier des processus types ainsi que les points de blocage.

L'exploitation des traces est plus simple dans le cadre de l'utilisation de traces modélisées (*m-trace*) car, comme précisé plus haut, elles présentent des labels qui facilitent le traitement. Les *m-traces* sont des données plus complexes, car elles sont associées à la notion de modèle de traces. Ces deux points sont décrits ci-après.

M-Trace

Les *m-traces* sont des traces numériques respectant un *modèle de trace*. Elles sont constituées d'une référence à un modèle de *m-trace*, ainsi qu'une origine (date de début) et d'une durée délimitant l'extension temporelle de la *m-trace*, une liste d'*obsels* et d'un ensemble de relations binaires entre ces *obsels*.

Les *m-traces* sont donc utiles pour clarifier les enregistrements mais leur utilisation est impossible sans les modèles de trace auxquelles elles sont associées.

Un **Modèle de m-Trace** permet de définir les champs d'une trace tout comme le ferait une DTD pour un fichier XML, ou pour le définir plus formellement :

« *Un modèle de trace est une sorte d'ontologie décrivant, en plus des types, relations et attributs des observés constituant la trace, le domaine temporel associé à ces éléments* » (Settoui 2011).

Un modèle de traces est généralement constitué d'un domaine temporel définissant la façon dont le temps est représenté, d'un ensemble fini de types d'*obsels*, d'un ordre partiel permettant de les classer, d'un ensemble fini de types de relations permettant de définir quelles relations peuvent exister entre les *obsels* ainsi qu'un ordre de classement, d'un ensemble fini d'attributs décrivant chacun un *obsel* ainsi que de deux fonctions. La première contraint les types de relations et les attributs qu'un type d'*obsel* peut avoir, et la seconde garantit la cohérence et la consistance entre les éléments d'une relation pour respecter la hiérarchie (Settoui 2011).

Grâce aux traces (modélisées ou non), il est alors possible de faire des rétrospectives sur les expériences des apprenants et ainsi constituer une source de connaissances qui pourrait accompagner certains apprentissages pour des contextes définis. Il est envisageable d'utiliser ce principe en lien avec la fouille de procédés (*process mining*) pour construire une base des processus à échec et ainsi grâce à la recommandation de vérifier dynamiquement les traces afin d'identifier quel type de processus est entamé par l'apprenant.

Fouille de Procédés

La fouille de procédés, ou *process mining*, est utilisée pour la découverte et le suivi des processus métiers

⁴ *Obsel (observed element)* est un néologisme désignant un élément observé. Ce mot construit sur le même principe que le mot pixel (*picture element*) permet d'alléger le discours lorsqu'il est souvent question d'éléments observés. (Champin et al. 2013)

(*business process*). Les processus sont créés à partir des traces qui sont récupérées depuis un système. Le *process mining* analyse des traces/log et permet de trouver le modèle des processus. Le résultat de l'analyse est un schéma de modélisation de processus comme par exemple les réseaux de Petri ou un diagramme BPMN. Il est aussi possible, comme cité plus haut, de créer ces données à partir des traces d'un système. Le *process mining* a pour but d'identifier des processus particuliers comme un enchaînement d'actions non prévu par le système ou un processus bloquant ralentissant le travail des autres ressources. Pour la construction du graphe permettant ces recherches, il est possible d'utiliser les algorithmes de création (*Discovery Algorithm*) comme l'algorithme Alpha (Van Der Aalst et al. 2007). Il serait aussi possible d'utiliser les algorithmes de *data mining* comme l'algorithme GSP (*Generalized Sequential Pattern algorithm*) (Srikant et Agrawal 1996). En revanche, ceux-ci ont été développés spécifiquement pour la création de données propres au *process mining* et ont, naturellement, été implémentés dans les logiciels permettant leur visualisation et leur traitement. Ils seront donc préférés aux autres algorithmes potentiels.

L'algorithme Alpha, est un de ces algorithmes de création (*discovery*) qui vise à reconstruire les enchaînements des suites d'événements et à les assembler dans un graphe. Il fait cela en se servant d'une clé primaire comme lien entre les événements : si deux événements se suivent directement sur la même valeur de clé primaire alors ils sont liés par un lien de causalité. Si deux événements étant liés par un lien de causalité se retrouvent en succession directe dans le sens opposé au sens qui a engendré leur lien alors ils sont en parallèle.

Le domaine est encore jeune mais il existe déjà plusieurs outils pour modéliser les réseaux de processus comme ProM (Van Dongen et al. 2005) et Disco qui permettent de faire de la visualisation à partir de traces. Pour une utilisation plus simple de ces logiciels, la norme XES (*Extensible Event Stream*) à base de XML permet de faciliter la communication entre les outils de création et les outils de recherche dans le *process mining* (Verbeek 2010).

Une fois que le *process mining* s'est chargé de créer l'enchaînement des processus, il devient possible de suivre les processus des apprenants depuis les traces observées par le système. Puis, grâce au système de recommandation, il deviendrait possible d'anticiper les futures actions entreprises et éventuellement proposer des processus alternatifs.

Systèmes de Recommandation

Les systèmes de recommandations (SR) ont été développés en parallèle du web. Ils recueillent des informations sur les préférences des utilisateurs pour construire un ensemble d'items (par exemple, des films, des chansons, des livres, *etc.*). L'information peut être acquise explicitement (typiquement en recueillant les évaluations des utilisateurs) ou implicitement (généralement en surveillant le comportement des utilisateurs, tels que les chansons écoutées, applications téléchargées, les sites Web visités et les livres lus) (Lee et al. 2010). Les SR peuvent inclure des caractéristiques démographiques des utilisateurs (comme l'âge, la nationalité, le sexe, *etc.*) ainsi que des

informations sociales comme les *likes*, *followers*, *followed*, *tweets*, et *posts*. On observe une tendance grandissante pour l'utilisation des informations de l'Internet des objets (par exemple, des localisations GPS, RFID, *etc.*) permettant de faire des rapprochements aidant à la recommandation (Levandoski et al. 2012). Les SR essaient d'équilibrer les facteurs tels que l'exactitude, la nouveauté, la dispersion et la stabilité dans leurs recommandations. L'adaptation de ces facteurs se fait grâce à des méthodes de filtrage.

Filtrage Collaboratif

Le Filtrage Collaboratif (FC) est un algorithme de recommandation populaire qui fonde ses prédictions et recommandations sur les notes ou le comportement d'autres utilisateurs dans le système. L'hypothèse fondamentale est que les opinions des autres utilisateurs peuvent être sélectionnées et regroupées de manière à fournir une prédiction raisonnable des opinions des utilisateurs traités. Le principe est simple. Si plusieurs utilisateurs, qui ont un profil ou une liste d'opinions similaires avec l'utilisateur traité, ont une opinion sur un item (musique, film, livre ou même un autre utilisateur) que l'utilisateur traité n'a pas encore vu alors il est possible que celui-ci ait la même opinion que les autres. Le FC peut aussi être utilisé avec les items plutôt que les utilisateurs, on appelle cela le *FC à base d'items*. Le principe est simple, les items sont considérés comme approchant si un utilisateur les a appréciés. Dans ce cas, la similarité entre items est basée uniquement sur le jugement des utilisateurs (Linden et al. 2003). Pour le FC à base d'items, la représentation d'un item peut se limiter à son identifiant. Si on prend un exemple de recommandation de livres, le SR ne connaît pas quels sont les genres de livres que sont « Les Hauts de Hurlevent » et « Anna Karenine ». Il sait juste que les utilisateurs ont apprécié ces deux livres, ce qui permet d'inférer que les deux livres sont voisins. Les FC sont des méthodes qui jouent un rôle important dans les SR bien qu'ils soient, bien souvent, mixés avec d'autres techniques comme le filtrage à base de contenu (Alchiekh Haydar 2014).

Filtrage à Base de Contenu

Le Filtrage à Base de Contenu est une approche où chaque item est défini par un ensemble attributs, et les items ayant des valeurs proches dans leurs attributs sont considérés approchants (similaires). Quand un utilisateur attribue une bonne note à un item, un item approchant sera considéré comme une recommandation potentielle (Pazzani et Billsus 2007).

Nous pouvons constater une dichotomie entre les deux familles de techniques précédentes : le filtrage à base de contenu et le filtrage collaboratif à base d'items. L'un se sert des descriptions des items pour les rapprocher et l'autre se sert des liens entre les utilisateurs et les items pour le faire. Plusieurs recherches ont été menées pour les associer en vue de mettre en place des techniques dites d'hybridation.

Filtrage Hybride

Le Filtrage Hybride n'est pas une approche indépendante, mais plutôt la combinaison de plusieurs approches de recommandation. Normalement, pour prédire une note, on calcule un score de similarité entre

les items ce qui permet de quantifier leur rapprochement. En filtrage hybride, le calcul de similarité est effectué par toutes les approches de l'hybridation, ce résultat est appelé « score local ». Puis le résultat est fusionné pour donner le « score final ». Cela étant, il est aussi possible d'appliquer la méthode d'hybridation à la fonction de similarité. Malgré le grand nombre de méthodes, il est possible de les rassembler en sept groupes distincts (Burke 2007).

Weighted ou Pondérée, est une technique qui combine numériquement les scores venant de deux techniques de recommandation (comme les méthodes de calcul de similarité). Elle est le moyen le plus simple pour une hybridation. Elle est flexible, car il suffit d'ajuster les paramètres de pondération pour quantifier la contribution de chaque approche.

Switch ou Alternance, est une technique un peu plus compliquée. Le but est que le système, à l'aide des paramètres qui lui sont donnés comme le profil de l'utilisateur ou les valeurs descriptives d'un produit, puisse alterner entre plusieurs techniques de recommandation et qu'il choisisse la plus efficace dans le contexte donné.

Mixed ou Mixte, est une technique qui utilise simplement chaque méthode qui lui est associée pour générer une liste d'items propices à être recommandés. Puis la liste passe dans chaque système mixé pour obtenir différentes listes classées. Enfin, ces listes sont fusionnées pour obtenir un classement des items potentiellement appréciés.

Feature Contribution ou Assimilé, est une technique dont le but est de fusionner les SR. Chaque attribut qui n'est normalement pas pris en compte par le SR sera pris en compte par un autre SR, ce qui permet de prendre en compte un panel d'items beaucoup plus large qu'avec un seul SR.

Feature Augmentation ou Contributeur, est une technique où le but est d'ajouter des SR à un autre, qui sera désigné comme principal, pour l'aider dans sa décision. L'idée est de pondérer les valeurs d'entrées appréciées des contributeurs pour les exposer à la critique du SR principal.

Cascade, est une hybridation faite pour les SR utilisant un score de sortie probabiliste. Le SR principal va déterminer la probabilité d'appréciation des items et les SR secondaires vont se suivre les uns les autres pour affiner les résultats. Les SR secondaires ne peuvent pas changer le classement du SR principal.

Meta-Level, sans réelle traduction française, est une approche assez semblable à l'approche contributeur. Plusieurs SR s'enchaînent pour créer une liste de valeurs acceptables pour être recommandées. Puis, le SR final récupère cette liste pour faire sa recommandation. La contribution des SR secondaires est donc faite lors de la sélection.

Il est à noter que chaque méthode d'hybridation agit soit sur la méthode de recherche, soit sur le score obtenu par les calculs de similarité pour lesquels plusieurs méthodes existent.

Méthodes de Calcul de Similarité

Une méthode de Calcul de Similarité est une approche permettant de créer un score permettant de quantifier la

ressemblance entre deux items sur les valeurs observées. Par exemple, en calculant la similarité entre deux apprenants, il est possible de savoir quels peuvent être les processus à échec que l'un d'eux doit éviter, compte tenu de ses faiblesses. Il existe plusieurs méthodes pour trouver ce score.

Les réseaux Bayésiens (Bayesian Networks), permettent de calculer la probabilité qu'un item soit le suivant en fonction de ce que l'utilisateur a déjà traité. Il s'agit d'un graphe pondéré possédant la capacité de voir les actions déjà effectuées. L'avantage des réseaux bayésiens est qu'ils permettent de recommander des items en prenant en compte les tendances de l'utilisateur. Par exemple, dans l'hypothèse où le processus d'apprentissage peut être réduit à un seul processus, si un apprenant suit un processus dont la continuité, selon son profil, a le plus de chance d'être une trajectoire inadaptée, alors le système notifiera l'apprenant pour le recadrer afin qu'il revienne sur un processus adapté. Il existe une forme de ces réseaux qui ne prend en compte que le dernier item traité, on parle alors de réseaux markoviens.

L'algorithme de Pearson, est une méthode permettant de trouver la corrélation entre deux variables. Il s'agit juste d'un calcul de covariance sur le produit des variances et permet d'obtenir un score de similarité compris entre 1 et -1, avec 1 pour une ressemblance parfaite, 0 aucune ressemblance et -1 une totale différence. L'algorithme de Pearson est souvent utilisé avec le filtrage collaboratif.

La similarité cosinus (ou mesure cosinus), permet de calculer la similarité entre deux vecteurs à n dimensions en déterminant le cosinus de l'angle entre eux. La similarité entre deux vecteurs s'obtient par le produit scalaire et la norme des vecteurs.

Le TF-IDF (Term Frequency-Inverse Document Frequency) est une méthode de pondération. Cette mesure statistique permet, par exemple, d'évaluer l'importance d'un terme contenu dans un document en calculant son nombre (ou sa fréquence) d'occurrences.

Dans notre cas, l'utilisation des réseaux Bayésiens qui est la plus pertinente. Elle nous permet de prévoir la finalité des événements et de faire avancer l'apprenant dans ce graphe. Plus l'apprenant progresse dans le graphe plus il précise ces potentielles finalités. Si celui-ci réduit trop ces chances de réussite, le système de recommandation lui proposera les activités pour réussir à son processus d'apprentissage.

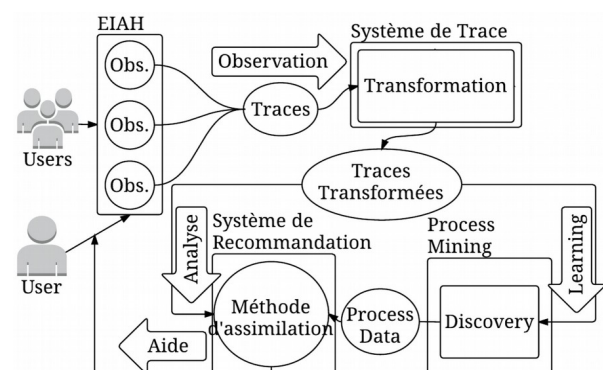


Figure 1: Mise en application

Mise en application

Le contexte d'application auquel nous nous intéressons est celui des EIAH. Notre objectif est d'analyser les parcours des utilisateurs dans l'application et de les utiliser pour : d'une part aider la progression des apprenants et d'autre part améliorer le pilotage global du système (cf. Figure 1). Le parcours désigne le processus suivi par un apprenant et peut avoir une granularité variable : par exemple, à un niveau bas, on peut considérer les activités réalisées et à un niveau haut considérer la réussite ou non de modules entiers de formation. L'orientation de nos travaux nous mène à combiner et à connecter plusieurs domaines d'expertise. Chaque domaine possède son emplacement dans le système. Le domaine des traces collecte et transforme celles-ci pour donner de la matière au *process mining* afin que celui-ci crée une base de connaissances. Le résultat respecte une norme de formatage de traces telle que le XES.

Le système commence par récupérer les traces d'exécution des apprenants. Chaque trace est ensuite traitée en fonction de sa granularité et selon le modèle de traces défini.

Ensuite, grâce à l'algorithme Alpha du *process mining*, un graphe d'actions est créé. Chaque finalité des processus de ce graphe est la réussite ou non de l'apprenant. Une fois chaque donnée associée, le graphe résultant peut être assimilé à un réseau bayésien utilisé dans le domaine de la recommandation. Le but de celui-ci est alors de déterminer le pourcentage de réussite des apprenants ayant pris chaque chemin référencé. Grâce à ce graphe, il sera aussi possible de corrélérer les actions possibles (comme utilisation de document de travail, les cours suivis ou les exercices faits) et les réussites/échecs des apprenants, ce qui permettrait de mesurer la pertinence des documents dans un contexte donné. Il sera également intéressant de corrélérer les processus obtenus au profil de l'apprenant.

Une fois le réseau bayésien créé alors le SR l'utilise pour déterminer les chances de réussite du processus entrepris par l'apprenant. Le système utilise les traces pour suivre chaque apprenant. À chaque avancée de l'apprenant dans le graphe, le système l'informe des actions qui seront les meilleures pour sa réussite.

Conclusion

Nous proposons de combiner trois domaines d'expertise pour aider les apprenants dans leur parcours d'apprentissage. L'intérêt de ce système est qu'il intègre une vision processus, au travers du *process mining*, dans le domaine de l'aide à l'utilisation d'applications interactives. Chaque domaine a une importance dans le système. Dans le cas de l'application aux EIAH, le domaine des traces se charge de la collecte et de la transformation des traces afin de pouvoir les trier et les filtrer, mais l'exploitation est laissée au domaine du *process mining*. Avec l'aide des algorithmes de création du *process mining* nous associons le domaine des traces à celui de la recommandation. Le domaine de la recommandation possède bon nombre de méthodes d'assimilation et de filtrage mais dans le cadre du traitement des processus l'utilisation des réseaux bayésiens est tout indiquée de

par la ressemblance entre les données de sortie du *process mining* et les réseaux bayésien. Grâce à un jeu de données suffisant, comme les inscriptions, notes d'examens et profils des élèves, il serait possible de prouver ces hypothèses afin de mettre en place ce système et, potentiellement, aider à améliorer les processus d'apprentissage des apprenants.

Le positionnement présenté dans le présent article, décrit le début des travaux en cours sur l'aide au parcours à base de traces dans les applications interactives. Ces travaux se poursuivent par la définition du cas d'étude concret issu de données des parcours des étudiants à l'IUT de La Rochelle pour la validation des hypothèses exposées. Il est néanmoins à noter que les résultats pourront être exploités dans d'autres types de systèmes tels que les ERP (*Enterprise Resource Planning*).

Références

- Alchikh Haydar, C. 2014. Les systèmes de recommandation à base de confiance. Thèse de Doctorat, Université de Lorraine.
- Bigham, J. P., Lau, T., & Nichols, J. 2009. TrailBlazer: Enabling blind users to blaze trails through the Web. in Proceedings of the 14th International Conference on Intelligent User Interfaces Sanibel Island, FL, USA : 177–186.
- Burke, R. 2007. Hybrid web recommender systems. The adaptive web: Springer. LNCS 4321 : 377–408.
- Champin, P.-A., Mille, A., & Prié, Y. 2013. Vers des traces numériques comme objets informatiques de premier niveau : une approche par les traces modélisées. *Intellectica*, (59) : 171–204.
- Djouad, T., Settouti, L. S., Prié, Y., Refay, C., & Mille, A. 2010. Un Système à Base de Traces pour la modélisation et l'élaboration d'indicateurs d'activités éducatives individuelles et collectives. *Mise à l'épreuve sur Moodle. Technique et Science Informatique* 29(6) : 721–741.
- Ho, H. N. 2015. Trace-Based Multi-Criteria Decision Making in Interactive Application for Adaptive Execution. Thèse de Doctorat, Université de La Rochelle.
- Lee, S. K., Cho, Y. H., & Kim, S. H. 2010. Collaborative filtering with ordinal scale-based implicit ratings for mobile music recommendations. *Information Sciences*, 180(11) : 2142–2155.
- Levandoski, J. J., Sarwat, M., Eldawy, A., & Mokbel, M. F. 2012. LARS: A Location-Aware Recommender System. in Proceedings of IEEE 28th International Conference on Data Engineering (ICDE'2012) : 450–461. Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, USA
- Linden, G., Smith, B., & York, J. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1) : 76-80.
- Loghin, G.-C. 2008. Observer un Environnement Numérique de Travail pour réguler les activités qui s'y déroulent. Thèse de Doctorat, Université de Savoie
- Lund, K., & Mille, A. 2009. Traces, traces d'interactions, traces d'apprentissages : définitions,

modèles informatiques, structurations, traitements et usages. Dans *Analyse de Traces et Personnalisation Des EIAH* : Lavoisier-Hermès, 21–66.

Mille, A., & Prié, Y. 2006. Une théorie de la trace informatique pour faciliter l'adaptation dans la confrontation logique d'utilisation/logique de conception. *Rencontres Interdisciplinaires Sur Les Systèmes Complexes Naturels et Artificiels*, Rochebrune, Megève : 183–196.

Pazzani, M. J., & Billsus, D. 2007. *Content-Based Recommendation Systems. The adaptive web: Springer. LNCS 4321* : 325–341.

Settouti, L. 2011. *Systèmes à Base de Traces Modélisées : Modèles et Langages pour l'exploitation des traces d'Interactions*. Thèse de Doctorat, Université Claude Bernard Lyon 1.

Srikant, R., & Agrawal, R. 1996. Mining sequential patterns: generalizations and performance improvements. In: *Proceedings of the 5th International Conference on Extending Database Technology*, 1(5) : 3–17.

Van der Aalst, W. M. P., Reijers, H. A., Weijters, A. J. M. M., Van Dongen, B. F., Alves de Medeiros, A. K., Song, M., & Verbeek, H. M. W. (2007). Business process mining: An industrial application. *Information Systems*, 32(5) : 713–732.

Van Dongen, B. F., Varbeek, H. M. W., Weijters, A. J. M. M., & Van Der Aalst, W. M. P. ; (2005). *The ProM Framework: A New Era in Process Mining Tool Support*. New York. Springer.

Verbeek, H. M. W., Buijs, J. C. A. M., Van Dongen, B. F., & van der Aalst, W. M. P. 2010. XES, XESame, and ProM 6. In *Information Systems Evolution*, Springer Berlin Heidelberg : 60–75.