Vers une capitalisation des processus d'analyse de traces

Alexis Lebis

Vers une capitalisation des processus d'analyse de traces

Alexis Lebis
1ere année

Sorbonne Universités, UPMC Univ Paris 06, CNRS, LIP6 UMR 7606, 4 place Jussieu, 75005 Paris, France
Université de Lyon, CNRS,
Université Lyon 1, LIRIS, UMR5205, F-69622, France
alexis.lebis@univ-lyon1.fr

Résumé

L'exploitation des données issues de dispositifs d'e-learning représente un enjeu majeur à l'heure actuelle pour la communauté EIAH. Malgré cela, la communauté dispose d'un écosystème hétérogène et disparate de données, de processus d'analyse de traces d'interactions et d'outils d'analyse. Peu de travaux traitent de l'échange ou de la réutilisation des processus d'analyse. Ce manque de capitalisation est un frein pour la communauté, l'obligeant à redéfinir et décliner l'existant. Cet article propose une approche pour capitaliser les processus d'analyse de traces indépendamment des plateformes d'analyses.

1 Introduction et État de l'Art

L'e-learning peut être défini par l'utilisation d'un réseau informatique organisé autour d'outils pédagogiques, de tuteurs, de ressources, d'interactions, pour renforcer les compétences des apprenants (Rosenberg, 2001). Ces dispositifs sont susceptibles de générer des données atteignant parfois le statut de *Big Data*. Elles peuvent relater l'interaction des utilisateurs entre eux (messages privés, forum, *etc*.) ainsi que les activités avec les ressources du dispositif. Nous parlons dès lors de traces d'interactions, qui sont considérées comme réservoirs de connaissances. Ces connaissances peuvent être découvertes par l'analyse desdites traces. Nous nous intéressons dans cet article à la capitalisation de ces analyses de traces. Dans les sections suivantes, nous présentons l'existant relatif à la problématique de l'analyse de trace. Ensuite, nous proposons notre approche pour la capitalisation des processus d'analyse, suivie des premiers résultats expérimentaux. Nous concluons en analysant ces résultats et en présentant les perspectives de recherche soulevées par ce travail.

1.1 Processus d'Analyse de Traces

Un processus d'analyse de traces est l'application de techniques et de méthodes pour obtenir des données sensées (Fayyad *et al.*, 1996) comme des modèles (Davis, 1989) ou des indicateurs (Dimitrakopoulou, 2006). Ces opérations sur les données sont appelées opérateurs et leur succession constitue ledit processus (Mandran *et al.*, 2015). Les acteurs de cette analyse peuvent être des statisticiens, des analystes, des chercheurs... Concevoir un processus d'analyse de traces est une tâche souvent complexe et fastidieuse : le besoin doit être cerné précisément et les données, ainsi que les connaissances du domaine, correctement communiquées. La publication des résultats est également cruciale afin que ceux-ci puissent être utilisé convenablement (Volle, 1984 ; King, 1986).

Actuellement, les processus d'analyse sont dépendants des outils utilisés et de la représentation des données. Cela masque la démarche intellectuelle de réponse au besoin et le sens des données qui y sont manipulées. De fait, en pratique, pour un même besoin, un processus d'analyse peut être décliné sur différents outils d'analyse en fonction du format des données concernées : la capitalisation est inexistante.

Pour mettre en œuvre ces processus d'analyse, plusieurs techniques ont été proposées. Parmi celles-ci, l'Educational Data Mining (EDM) est l'adaptation de la fouille de données aux données pédagogiques complexes dans le milieu éducatif (Romero et Ventura, 2007). L'état de l'art de Baker et Yacef (2009) nous propose une classification des analyses spécifiques aux EIAH. Les auteurs réforment la paire classique « analyse prédictive » et « analyse descriptive » avec cinq éléments : (1) *prediction*, (2) *clustering*, (3) *relationship mining*, (4) *distillation of data for human judgment,* et (5) *discovery with models*. Ce dernier point consiste en l'utilisation de modèles préalablement développés comme composants d'autres analyses, ce qui résonne avec notre ambition de capitalisation. De manière analogue, le Learning Analytics (LA) est une discipline spécifique aux Environnements Informatiques pour l'Apprentissage Humain (EIAH), se basant sur plusieurs disciplines (statistique, recherche opérationnelle...) (Lias et Elias, 2011 ; Ferguson, 2012). Comparativement à l'EDM, il donne un rôle beaucoup plus important aux acteurs, les incluant dans la boucle itérative de conception, d'analyse et de révision (Siemens et Baker, 2010).

Différents travaux décrivent les étapes d'analyse de manière différente (Fayyad *et al*., 1996 ; Stamper *et al*., 2011 ; Volle, 1984), mais trois étapes sont récurrentes : le prétraitement des données, l'analyse et le post-traitement. D'autres étapes sont plus atypiques et reflètent les spécificités et besoins du domaine éducatif. Stamper *et al.* (2011) suggèrent ainsi trois étapes, encourageant nos travaux : une de publication des résultats, une de réutilisation de données et une d'archivage. Cette dernière étape concerne uniquement l'archivage des données raffinées et des métadonnées associées, et non l'archivage des processus d'analyse.

1.2 Plateformes d'Analyse de Traces

La communauté EIAH dispose aujourd'hui d'une grande diversité d'outils d'analyse. Ils sont issus de divers horizons et répondent à des besoins d'analyse

précis. R[1], par exemple, est un langage et un environnement pour le calcul statistique. SPSS[2] est un outil proposé par IBM pour la fouille de données. Des travaux de recherches sur les traces d'interaction dans le domaine des EIAH ont débouché sur des solutions plus spécialisées. PSLC DataShop est une plateforme de stockage et d'analyse – prédéfinie – de traces d'interactions (Koedinger *et al.*, 2010) qui a eu un impact important dans la communauté. Plus atypique, le kernel Trace Based System (kTBS) est une implémentation du paradigme de systèmes à base de traces modélisées (Settouti *et al.*, 2006) qui offre de nouvelles perspectives de raisonnement sur les données. UnderTracks (UT) implémente le paradigme DOP8 qui représente le cycle de vie des données et des opérateurs dans l'analyse (Mandran *et al.*, 2015). Usage Tracking Language (UTL) permet de décrire l'obtention d'un indicateur à partir de données primaires issues de traces (Choquet et Iksal, 2007).

De manière générale, les outils d'analyse peuvent être classés selon trois catégories, comme le laisse sous-entendre Verbert *et al.* (2012), complété par Mandran *et al*. (2015). Ces catégories sont (1) le stockage de données, (2) l'analyse de données (comme R) et (3) les deux simultanément (DataShop, kTBS, UT, UTL). Cette catégorisation nous permet de fixer la limite de nos travaux : nous ne considérons dans nos travaux que les outils permettant l'analyse.

1.3 Capitalisation

La capitalisation est la faculté de disposer de l'existant pour en tirer profit. Pour un processus d'analyse cela concerne notamment sa réutilisation, son adaptabilité, sa modification, son enrichissement, pour tout le spectre des outils d'analyse disponibles.

La volonté de capitaliser les données manipulées par un processus d'analyse se heurte au problème de leur représentation. Les outils d'analyse ne sont que très peu permissifs concernant le format des données utilisées, notamment car l'implémentation des opérateurs d'analyse est assez rigide. Des travaux proposent d'effectuer un *mapping* vers un formalisme plus générique, comme Caliper Analytics[3] ou UTL (Choquet et Iksal, 2007). Cependant, cette correspondance n'est pas destinée à être réadaptée ensuite pour d'autres outils d'analyse. Nous pensons que ces démarches ne font que répondre en partie au problème de capitalisation : elles ne font que reporter le problème d'échange des analyses. Actuellement, la capitalisation des opérateurs d'analyse n'est principalement qu'interne à l'outil d'analyse qui les implémente. Fondamentalement, une fonction R écrite par un utilisateur peut être considérée comme capitalisable. Mais des travaux sont plus ambitieux. C'est le cas d'UT qui considère les opérateurs comme des éléments susceptibles d'être créés, échangés et réutilisés par les différents acteurs. Par exemple un filtre temporel peut être considéré comme la déclinaison spécifique d'un filtre plus générique n'opérant que sur des données temporelles ; il y a là un gain sémantique notable et il est pertinent de le partager. Pour la capitalisation externe vers d'autres outils d'analyse, on constate qu'il s'agit principalement de l'intégration de scripts. Cette approche est dépendante de la tolérance à l'évolution des outils et n'est par conséquent pas toujours envisageable, surtout pour des solutions propriétaires.

Des travaux considèrent toutefois que l'obtention d'une donnée pertinente, comme un indicateur, peut être réalisée en appliquant un ensemble ordonnancé et fixe d'opérateurs (Djouad *et al*., 2009 ; Mandran *et al.*, 2015). De fait, le processus d'analyse a la particularité d'être réutilisable dans d'autres analyses d'une même plateforme (Diagne, 2009). Il ressort de ces travaux une approche où chaque opérateur est vu comme une boîte noire, capitalisable dans l'outil considéré.

Néanmoins, capitaliser un processus d'analyse de traces pour différents outils est une problématique qui, à notre connaissance, n'est pas vraiment traitée par la communauté scientifique en EIAH. Notons tout de même Predictive Model Markup Language (PMML[4]), du Data Mining Group, qui permet l'échange de modèles prédictifs et de *machine learning,* qu'ils soient entraînés ou non, entre différents outils d'analyse. PMML est utilisé par des outils aussi bien libres (e.g. Weka[5]) que propriétaires (e.g. SPSS[2]) : ceci constitue pour nous une preuve que le besoin d'échange et de réutilisation des processus d'analyse est bien réel.

2 Problématique et Proposition

2.1 Questions de Recherche

Afin d'apporter une dimension capitalisable aux processus d'analyse, il faut principalement répondre aux questions de recherche suivantes : (Q1) comment réifier et partager le processus cognitif de conception de l'analyse ? Et donc (Q2) comment modéliser un processus d'analyse pour qu'il soit suffisant ? Nous entendons par suffisant le fait qu'aucun élément le constituant ne se rattache à un outil d'analyse particulier. (Q3) Comment s'affranchir de la dépendance excessive aux données initiales pour favoriser la réutilisation ? Et (Q4) de quelle manière enrichir un processus d'analyse afin d'assister l'acteur dans sa conception et sa réutilisation ?

2.2 Hypothèses

Nos travaux reposent sur trois hypothèses importantes. Tout d'abord, nous estimons que (H1) pour répondre à un besoin d'analyse, le processus cognitif d'analyse associé n'est pas biaisé par les spécificités d'une plateforme. C'est une assertion importante qui nous permet de le considérer comme un ensemble d'actions élémentaires. Ensuite, Rosch (1973) formule le fait que la cognition s'effectue par des catégories qui font office de "point de référence cognitif" plutôt que des

[1] http://www.revolutionanalytics.com/
[2] www.ibm.com/software/fr/analytics/spss
[3] https://www.imsglobal.org/activity/caliperram
[4] http://dmg.org/pmml/v4-1/GeneralStructure.html
[5] http://www.cs.waikato.ac.nz/~ml/weka/index.html

instances élémentaires. Cela nous fait dire que (H2) la manière de penser et de créer l'analyse s'effectue par la manipulation des concepts dégagés par les données, plutôt que par les valeurs de ces données. Et (H3) un processus d'analyse peut être considéré comme une succession, pas nécessairement linéaire, d'opérations ordonnées qui prennent des données en entrée et en produisent en sortie. Cela nous apporte une propriété séquentielle et de consistance.

2.3 Objectif

Notre objectif est d'apporter une solution au problème de la capitalisation des processus d'analyse de traces d'interactions. L'ambition à terme est de disposer d'un entrepôt de processus d'analyse éprouvés qui offre un panorama de l'existant pour les partager, les réutiliser et les enrichir, tout en indiquant les outils d'analyse capables de les réaliser. Pour cela, nous proposons de les décrire (cf. section 2.4) de manière indépendante afin de les rendre génériques.

À la différence de démarches comme celles de Choquet et Iksal (2007), l'objectif n'est pas d'exécuter nous même ce processus d'analyse générique sur des valeurs pour obtenir un résultat concret. Le résultat du processus est juste décrit ; ce seront les outils que la communauté possède qui vont être utilisés pour résoudre ledit processus. C'est un avantage majeur puisque ces outils existent déjà et qu'ils sont potentiellement optimisés. Cela nous permet de nous inscrire dans une démarche d'aide à la création, à l'expression et à la décision, évitant de proposer un nouvel outil de calcul contraignant lui aussi par ses caractéristiques intrinsèques.

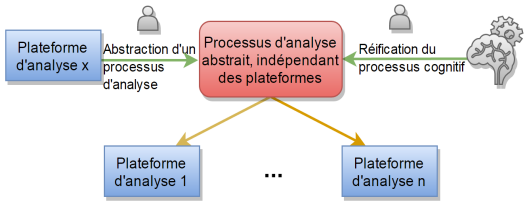

Figure 1. Quand capitaliser un processus d'analyse de traces.

Nous estimons qu'un processus d'analyse peut être décrit de manière générique à deux moments dans son cycle de vie (cf. Figure 1). Premièrement, au moment de la démarche cognitive qu'a un acteur lors de la création d'un processus, pour répondre à un besoin. Il s'agit alors de réifier cette démarche abstraite dans un formalisme indépendant des outils d'analyse. La réification intervient donc avant que le processus ne soit altéré par des spécificités techniques. Deuxièmement, lorsqu'un acteur veut partager son processus d'analyse déjà créé dans un outil. Il s'agit alors de décrire et réifier l'existant grâce au même formalisme indépendant. Cela implique une démarche de rétro-conception au sens où l'acteur doit extraire du processus d'analyse les éléments saillants impliqués et les appareiller à leur concept générique. Dans les deux cas, décrire un processus d'analyse implique de décrire les opérateurs le constituant, d'après (H3). On conviendra alors facilement que pour qu'un processus d'analyse puisse être considéré comme indépendant des outils, chacune de ses opérations constitutives doit aussi l'être. Un opérateur générique est donc l'expression du concept dégagé par les opérateurs qu'il représente. Considérons un filtre temporel : ses techniques d'application ainsi que d'implémentation diffèrent entre les outils (kTBS, R, UT, ...). Malgré cela, le concept sous-tendu par cette opération est d'appliquer un filtre sur le temps : c'est ce que doit représenter l'opérateur générique *Filtre Temporel*, sans prendre en compte les spécificités techniques liées aux outils.

Cette description permet d'assurer l'obtention du résultat du processus d'analyse -son objectif- lorsque les données nécessaires à sa réalisation sont fournies. Il faut cependant noter que par *données nécessaires* nous entendons les *types d'éléments tracés* (*TET*) et non des valeurs (cf. 2.4.1). Cela permet de s'affranchir du problème de représentation des données dans les outils. En conséquence, un processus d'analyse générique peut être résumé en un triplet *(données nécessaires, opérations, données résultats)*, facilitant sa description et sa réutilisation. Par exemple, pour identifier la démarche d'apprentissage d'un apprenant, nous pouvons exploiter comme TET nécessaires les connexions à l'outil, le temps des sessions, les cours lus ainsi que les exercices effectués. La Figure 2 résume les différents concepts présentés jusqu'ici, leur place dans le processus d'analyse générique, ainsi que les interactions qui existent entre eux et avec l'acteur concevant le processus d'analyse.

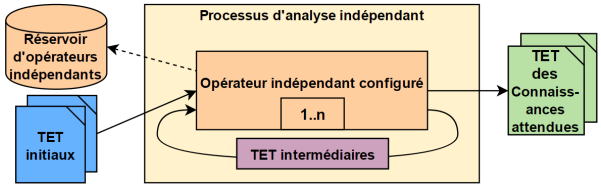

Figure 2. Représentation des différents éléments intervenant dans la description d'un processus d'analyse générique.

2.4 Méta-Modèles de l'Approche

Les méta-modèles présentés ci-après visent à décrire de manière indépendante des plateformes d'analyse un processus d'analyse et ses éléments constitutifs. Ces méta-modèles sont une première proposition pour répondre aux questions de recherche Q1, Q2 et Q3 présentées précédemment. La question Q4 sera traitée plus tard dans la suite du travail.

2.4.1 Méta-Modèle des Données

Pour rappel, nous considérons uniquement le type d'élément tracé (TET), et non ses valeurs. Un TET est le concept exprimé par les valeurs. Concrètement, pour un apprenant ayant un score de 7, **score** est le type d'élément tracé et 7 sa valeur. Cette considération se base sur l'hypothèse H2. De fait, un processus d'analyse générique va être fonction des TET manipulés plutôt que fonction des valeurs ; l'abstraction s'en trouve renforcée. Le méta-modèle *Element* (cf. partie basse de la Figure 3) définit comment représenter un TET. Cela s'effectue principalement par l'intermédiaire d'un nom (*Name*) et

d'un type (*ConstraintType*). Ce sont les seules informations que nous avons conservées pour nous soustraire à la gestion directe des valeurs. Ces deux attributs contribuent à créer une sémantique du TET, interprétée par l'acteur : celui-ci travaille donc uniquement avec des concepts. Le méta-modèle *List* (Figure 3) exprime la notion de conteneur de TET. Il s'agit du seul lien structurel entre eux : chaque TET est affilié à une seule *List* (*IDLinkList*). Cela permet de créer un support sémantique supplémentaire libre à l'interprétation de l'acteur, sans pour autant être excessivement contraignant. Les métadonnées supplémentaires, à savoir *ID*, *Creator*, *DateCreation* et *LastModified*, vont permettre la traçabilité de ces éléments. *Element* et *List* définissent donc les TET qui seront utilisés par les opérateurs génériques, ainsi que leurs relations.

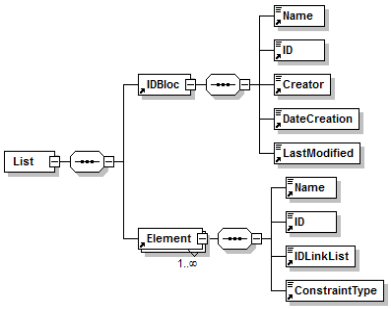

Figure 3. Méta-modèle d'un type d'élément tracé (TET) pour un processus d'analyse générique.

2.4.2 Méta-Modèle d'Opérateur Générique

Dans notre démarche de création d'un opérateur générique, il nous a fallu identifier les opérateurs spécifiques qui lui sont équivalent : ceux qui possèdent les mêmes sens et objectif. Leurs éléments constitutifs récurrents ont été conservés pour former l'opérateur générique. Il est important de préciser qu'il n'en découle pas pour autant une propriété de bijection entre les opérateurs spécifiques et l'opérateur générique. Le méta-modèle que nous proposons pour représenter un opérateur générique est issu de cette démarche (cf. Figure 4). Nous l'avons obtenu après avoir étudié des opérateurs dans UnderTracks/Orange, Weka et kTBS.

Pour conserver l'aspect évolutif et dynamique des données, un opérateur générique se doit d'être exécutable. Au contraire de ceux définis dans les outils de la communauté, il ne peut raisonner que sur les TET, pas sur les valeurs. Le résultat de l'application d'un opérateur générique est régi par la définition de son objectif *via* des règles comportementales à appliquer sur les TET en entrée (*OutputSheet*) et, éventuellement, de son paramétrage. Par exemple, on peut classer les apprenants entre actif et passif, en fonction de leur participation aux cours et aux exercices, par l'utilisation d'un cluster. Chaque apprenant se voit attribué une classe. Cela revient à dire que le cluster crée un nouveau TET *classe* représentant la classe d'un individu, bien qu'aucune valeur ne soit attribuée. Comme la Figure 4 le montre, nos opérateurs génériques ne sont définis que par quelques propriétés générales : le nombre d'entrées (*NbInputs*), de sorties (*NbOutputs*), de paramètres (*NbParameters*) et des contraintes sur ces paramètres (*Constraint*). Le champ descriptif *Description* permet d'enrichir textuellement l'opérateur en y indiquant par exemple son fonctionnement ou encore son utilité. *TargetPlateform* renseigne sur les outils implémentant l'opérateur générique en question. Cela permet d'offrir des indications quant aux plateformes d'analyse susceptibles de réaliser concrètement un processus d'analyse générique.

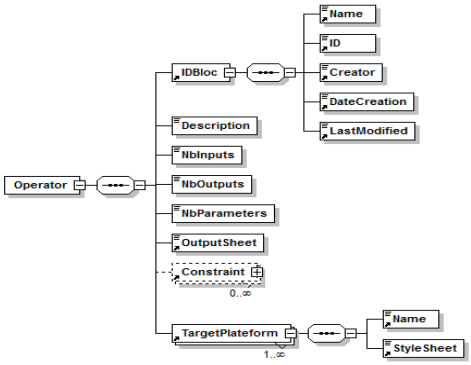

Figure 4. Méta-modèle d'un opérateur générique.

2.4.3 Méta-Modèle du Processus d'Analyse Générique

Un processus d'analyse générique est composé d'opérateurs génériques qui s'enchaînent (H3). Le méta-modèle associé (cf. Figure 5) repose sur les éléments présentés en 2.4.1 et 2.4.2. Un *ConfiguredOperator* représente une étape dans le processus d'analyse générique. Il est assimilable à un triplet (entrée - *Input*, opération - *Operator*, sortie - *Output*). *Input* représente les TET qui seront alors traités par l'*Operator,* produisant éventuellement des *Outputs*, régis par les *OutputSheet* desdits opérateurs.

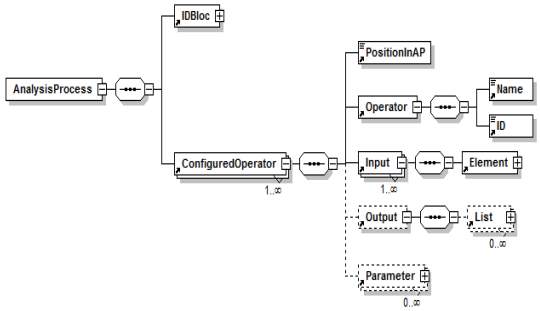

Figure 5. Méta-modèle d'un processus d'analyse générique et indépendant des plateformes d'analyse de traces.

Ainsi, les sorties d'un opérateur peuvent faire partie de l'entrée d'un autre opérateur. Pour pouvoir matérialiser ce chaînage, nous avons muni notre processus d'analyse générique d'une relation d'ordre *viaPositionAP*. De fait, un *ConfiguredOperator* est par extension réflexif, transitif et antisymétrique, ce qui va nous permettre d'organiser l'application des opérateurs de manière fiable. On obtient ainsi une relation de dépendance entre la production finale de notre processus d'analyse et les données nécessaires, sous forme de *TET,* à sa réalisation. De plus, cela permet l'injection d'un processus d'analyse dans un autre.

3 Expérimentation

Nous avons conduit des expérimentations afin de valider ou invalider l'approche proposée et pour éprouver les méta-modèles présentés en 2.4.

3.1 Description et Protocole

Les expérimentations ont été réalisées auprès de 2 informaticiens, 3 statisticiens et 1 cogniticien qui ont tous une expérience en analyse de données EIAH, au travers d'un prototype mettant en œuvre l'approche que nous venons de présenter (cf. Figure 6).

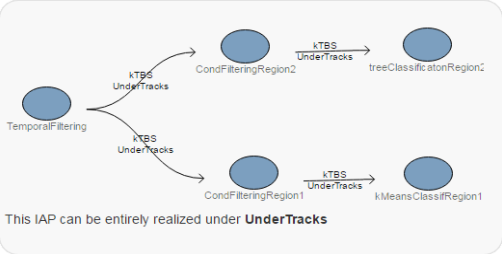

Figure 6. Visualisation d'un processus d'analyse indépendant issu de notre prototype. Les informations sur les arcs représentent les outils d'analyse aptes à réaliser les étapes matérialisées par les nœuds.

La démarche expérimentale se composait de trois parties. La première consistait à matérialiser un besoin d'analyse puis à décrire -textuellement ou graphiquement- le processus d'analyse associé et à évaluer subjectivement sa difficulté. Cette partie permet d'étudier comment un acteur réfléchit et conçoit un processus, exempt de spécificités techniques et de valeurs concrètes : a-t-il besoin des valeurs pour s'exprimer, à quelle granularité opère-t-il sa description… La deuxième partie permet l'appropriation, par la pratique, des concepts de notre approche. Pour cela les sujets ont utilisé notre prototype. Aucune présentation de celui-ci n'avait été effectuée auparavant mais le protocole expérimental présentait son interface. Nous étudions principalement la réussite des sujets à créer une analogie entre sa démarche intellectuelle et l'approche mise en œuvre par le prototype. La troisième partie consistait à transcrire le processus d'analyse décrit lors de la partie 1 dans le prototype. Il s'agit de l'étape de réification d'un processus cognitif décrite dans la Figure 1. Cela nous permet d'étudier une éventuelle divergence sémantique entre les deux processus et d'en étudier les raisons. De plus, cette partie permet d'analyser le comportement du sujet vis-à-vis des concepts abstraits qu'il manipule et la manière dont il opère la description de son processus générique. Tout au long de l'expérimentation, les sujets étaient autonomes. Ils étaient filmés et évalués par un observateur *via* des grilles d'évaluation portant sur le comportement et l'aisance face à différents concepts de l'approche et du prototype. Les deux dernières parties étaient suivies d'un questionnaire. Le prototype ainsi que les documents et productions expérimentales autorisées à être diffusés sont accessibles en ligne[6].

[6] http://liris.cnrs.fr/~alebis/iogap.html

3.2 Résultats

Le Tableau 1 présente une synthèse issue de l'analyse des grilles d'évaluation et des questionnaires remplis par les six sujets lors des expérimentations.

Le premier constat que nous pouvons faire est que l'approche que nous proposons est comprise par tous les sujets. En effet, chacun d'entre eux a su créer des processus d'analyse cohérents, bien que deux soient incomplets par manque de temps ou d'opérateurs. Ces sujets ont bien pris conscience du fait que les processus d'analyse sont abstraits et indépendants des outils d'analyse ; qu'ils n'ont pas vocation à calculer ; qu'ils sont matérialisés par des concepts d'opération -les opérateurs génériques- ; et que les données nécessaires représentent "[...] le sous-ensemble strictement indispensable pour qu'un processus d'analyse puisse être réalisé" (sujet n°5). En revanche, on distingue un problème d'utilisation et de compréhension sémantique pour la notion de *List* qui n'a pas rempli son rôle de support structurant. Cette notion complexifie l'utilisation des opérateurs génériques, notamment au début des expérimentations. De plus, nous avons assisté à un comportement émergent : l'utilisation d'une seule liste au sein d'un même processus. Notre hypothèse est que les sujets se sont reposés sur leur propre représentation structurelle des données, pour pallier au manque d'apport sémantique de *List*, ce qui leur a permis de décrire correctement les processus d'analyse.

	Compréhension	Utilisation	Limite
Donnée	✓ **Compris** : différent de valeur	✓ **Simple** d'utilisation	~ **Manque** de relation entre les données
Liste	✗ **Confuse** : assimilée aux données	✗ **Complexe** : sémantique nulle	✗ **Manque** de précisions
Opérateur	✓ **Compris** : opérations abstraites, boîte noire	✓ **Simple** : opération à appliquée sur les données	~ **Couverture** : est-ce possible de représenter tous les opérateurs ?
Entrée	✓ **Compris** : données injectées	~ **Peu simple** : difficulté dans le choix des données	~ **Manque** de flexibilité
Sortie	✓ **Compris** : objectif de l'opération	✓ **Simple** : production automatique	✗ **Besoin** de feedback
Paramétrage	✓ **Compris** : influence sur les résultats	✗ **Complexe** : pas de sémantique	✗ **Manque** de sémantique
Processus d'analyse	✓ **Compris** : indépendant et descriptif	✓ **Simple** à réaliser, utile pour connaitre les outils d'analyse disponibles pour un processus donné	~ **Besoin** de sémantique, de flexibilité et de plus d'opérateurs.

Tableau 1. Synthèse des résultats expérimentaux.

Les résultats expérimentaux permettent d'identifier d'autres limitations. Outre le fait que notre prototype n'implémentait pas assez d'opérateurs génériques, nous avons remarqué qu'avoir un retour (*feedback*) sur les données produites est important pour les acteurs. En effet, comme le suggère Fayyad *et al.* (1996), l'analyse est une tâche itérative et il est courant d'affiner le paramétrage des opérateurs après avoir considéré les résultats produits. D'ailleurs, il ressort de l'expérimentation que la granularité d'expression des paramètres doit être plus fine pour que les opérateurs soient correctement paramétrés. Nous avons aussi

remarqué un manque d'expressivité sémantique pour les différents éléments de l'approche. Les sujets avaient ainsi du mal à discerner les objectifs de certaines étapes du processus d'analyse. Un dernier constat est que l'approche est bien reçue par les sujets de l'expérimentation, qui la jugent pertinente et utile pour exprimer, partager et confronter un processus d'analyse générique : le fait de travailler de manière abstraite permet de "mettre en ordre les différentes idées" (sujet n°6) et de "formaliser un cheminement de mise en œuvre" (sujet n°4) dans les différents outils d'analyse.

4 Conclusion, Discussion et Perspectives

Dans cet article, nous avons présenté notre approche pour répondre à la problématique de la capitalisation des processus d'analyse de traces. Nous considérons un processus d'analyse comme une suite de triplets *entrée-actions-résultat* capable d'être décrit grâce à un formalisme abstrait et indépendant des spécificités techniques des différents outils d'analyse. Nous avons mis en œuvre cette approche et mené de premières expérimentations *via* un prototype. Les résultats expérimentaux encouragent fortement l'idée qu'il est possible de décrire indépendamment un processus d'analyse tout en conservant sa structure et ses objectifs, en plus d'informer sur les outils d'analyse capables de le réaliser. Nous avons montré que la réification d'un processus d'analyse issu d'une démarche cognitive était possible. Nous devons maintenant expérimenter la réification de processus d'analyse de la littérature, afin de s'assurer que notre approche est complète.

En termes de perspectives, nous comptons faire évoluer le méta-modèle des données en un graphe sémantique pour répondre aux lacunes de sémantique identifiées lors de l'expérimentation. Nous pressentons que cela permettra d'assouplir la réutilisation des processus d'analyse dans d'autres contextes et éventuellement de dégager des propriétés intéressantes sur les opérateurs génériques. Une autre perspective est de pouvoir instancier un processus d'analyse générique dans les outils d'analyse. Cette instanciation pourra prendre la forme d'une notice de réalisation, à défaut d'être automatique. L'intérêt est double : répondre au besoin de *feedback* et créer un lien entre les différents outils existants.

Remerciements : Ce travail a été financé dans le cadre du projet Hubble (ANR-14-CE24-0015).

Références

Baker, R. S. J. D., and Yacef, K. 2009. The State of Educational Data Mining in 2009 : A Review and Future Visions. Journal of EDM1(1):3–16.

Choquet, C., and Iksal, S. 2007. Modélisation et construction de traces d'utilisation d'une activité d'apprentissage: une approche langage pour la réingénierie d'un EIAH. Revue STICEF 14.

Davis, F. D. 1989. Perceived usefulness, perceived ease of use, and user acceptance of information technology. 319-340. MIS quarterly.

Diagne, F. 2009. Instrumentation de la supervision par la réutilisation d'indicateurs: Modèles et Architecture. Ph.D. diss, Université Grenoble I.

Dimitrakopoulou, A. ; Petrou, A. ; Martinez, A. ; Marcos, J. A. ; Kollias, V. ; Jermann, P. ; and Bollen, L. 2006. State of the art of interaction analysis for Metacognitive Support & Diagnosis.(D31.1.1) EU Sixth Framework programme priority 2, Information society technology, Network of Exc.

Djouad, T. ; Mille, A. ; Reffay, C. ; and Benmohamed, M. 2009. Ingénierie des indicateurs d'activités à partir de traces modélisées pour un Environnement Informatique d Apprentissage Humain. Revue STICEF 16.

Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. 1996. From data mining to knowledge discovery in databases. AI magazine 17(3):37.

Ferguson, R. 2012. The state of learning analytics in 2012: a review and future challenges. Technical Report KMI-12-01.

King, G. 1986. How not to lie with statistics: Avoiding common mistakes in quantitative political science. American Journal of Political Science:666-687.

Koedinger, K. R. ; Baker, R. S. ; Cunningham, K. ; Skogsholm, A. ; Leber, B. ; and Stamper, J. 2010. A data repository for the EDM community: The PSLC DataShop. Handbook of educational data mining 43.

Lias, T. E. ; and Elias, T. 2011. Learning Analytics: The Definitions, the Processes, and the Potential. learninganalytics.net

Mandran, N. ; Ortega, M. ; Luengo, V. ; and Bouhineau, D. 2015. DOP8: merging both data and analysis operators life cycles for technology enhanced learning. In Proceedings of LAK 2015, 213-217. ACM.

Rosch, E. H. 1973. Natural categories. Cognitive psychology 4(3), 328-350.

Rosenberg, M. J. 2001. E-learning: Strategies for delivering knowledge in the digital age. New York: McGraw-Hill.

Settouti, L. S. ; Prié, Y. ; Mille, A. ; and Marty, J. C. 2006. Systèmes à base de trace pour l'apprentissage humain. In Proceedings of TICE 2006.

Siemens, G. ; and Baker, R. S. J. 2010. Learning Analytics and Educational Data Mining : Towards Communication and Collaboration.

Stamper, J. C. ; Koedinger, K. R. ; Baker, R. S. J. D. ; Skogsholm, A. ; Leber, B. ; Demi, S. ; Spencer, D. 2011. Managing the educational dataset lifecycle with datashop. Lecture Notes in Computer Science6738 LNAI, 557–559.

Verbert, K. ; Manouselis, N. ; Drachsler, H. ; Duval, E. ; Wolpers, M. ; Vuorikar, i R. ; and Vuorikari, R. 2012. Dataset-Driven Research to Support Learning and Knowledge Analytics. Educational Technology & Society 15(3), 133–148.

Volle, M. ; and Malinvaud, E. 1984. Le métier de statisticien. Economica.