Focaliser l'extraction d'épisodes séquentiels à partir de traces par le contexte

Béatrice Fuchs

Focaliser l'extraction d'épisodes séquentiels à partir de traces par le contexte.

Béatrice Fuchs[1]

Université de Lyon, UJML3, CNRS, IAE, Laboratoire LIRIS, 69372 Lyon cedex 08, France
beatrice.fuchs@liris.cnrs.fr

Résumé : L'exploration des traces avec l'extraction d'épisodes séquentiels vise à caractériser des utilisateurs en fonction des séquences d'actions qu'ils ont réalisées dans un environnement numérique. Une des difficultés majeures de l'extraction de connaissances est la surabondance de résultats qui rend leur exploitation difficile par un expert humain chargé d'interpréter les résultats. Or, l'utilisation de connaissances *a priori* est souvent efficace d'une part pour limiter le volume de résultats produits et d'autre part pour focaliser l'analyse sur des résultats potentiellement intéressants. Nous proposons d'utiliser le contexte des actions de la trace pour limiter les résultats de la fouille, qui s'exprime sous la forme d'une contrainte qui permet d'obtenir des épisodes signifiants du point de vue du contexte. Des expérimentations dans le cadre applicatif du jeu sérieux Tamagocours pour l'apprentissage des règles juridiques de diffusion de documents montrent la pertinence de cette contrainte pour filtrer de nombreux épisodes sans intérêt avec un rappel et une précision intéressants.
Mots-clés : contraintes, découverte de connaissances, analyse de traces, contexte

1 Introduction

Des volumes importants de traces sont accumulés par les apprenants ou les joueurs dans leur environnement numérique d'apprentissage. Les traces numériques se présentent sous la forme de séquences d'actions contextualisées et temporellement situées décrivant des parcours d'utilisateurs dans un environnement numérique. L'analyse des ces traces qui témoignent de leur activité est importante afin de mieux comprendre leurs difficultés en vue de leur proposer une assistance adaptée en conséquence ou d'améliorer les outils disponibles. Une façon d'étudier ces traces est l'extraction de connaissances à partir de données (ECD), qui vise à extraire des connaissances à partir de données dans un processus *interactif* et *itératif* (Frawley *et al.*, 1992). Une des méthodes les plus adaptées à l'exploration des traces est l'extraction d'épisodes séquentiels qui prend en compte les dimensions événementielle et temporelle des traces et vise à mettre en évidence des séquences typiques d'actions réalisées par des utilisateurs afin de caractériser leur parcours ou leur comportement. Néanmoins plusieurs problèmes se posent lors de la mise en œuvre de l'ECD.

Le premier problème récurrent en fouille de données est que celle-ci produit le plus souvent une quantité très importante de résultats avec une forte redondance dont la plus grande partie est sans intérêt et occulte les résultats intéressants. L'introduction de contraintes pour contrôler la fouille et limiter les résultats de la fouille s'avère indispensable. L'expérience a montré que l'utilisation de mesures d'intérêt *objectives*, telles que le support ou la longueur (Béchet *et al.*, 2014) sont utiles pour contrôler efficacement la fouille, mais elles restent insuffisantes lorsqu'il y a beaucoup de redondance combinatoire (van Leeuwen, 2014). De nombreux travaux ont déjà abordé le sujet pour l'exploration des motifs ensemblistes, qui reste néanmoins ouvert pour les séquences qui sont relativement moins étudiées.

Le deuxième problème rencontré lors de l'exploration des traces est la prise en compte des dimensions représentées dans les traces. L'exploration d'épisode séquentiels restreint l'analyse à deux dimensions : le type d'action (ou type d'événement,)[1] et l'ordre des actions, éventuellement associées à une information temporelle plus précise telle que la date et

1. terme généralement utilisé en fouille de données

l'heure. Ces dimensions sont importantes et doivent être prises en compte mais le contexte dans lequel une séquence d'actions s'est déroulée est également important et celui-ci n'est pas pris en considération par la méthode de fouille d'épisode séquentiels. Par exemple, on s'intéresse non seulement au moment et à l'action réalisée mais également à l'utilisateur qui a réalisé une action et sur quel objet l'utilisateur a agi, *etc.* : le contexte exprime un lien fort entre les différentes actions en lui donnant un sens et exprime une continuité des actions pour la réalisation d'un objectif précis voulu par l'utilisateur.

Ces difficultés ont été mises en évidence lors de l'étude des traces du jeu Tamagocours, un jeu sérieux pour l'apprentissage des règles juridiques de diffusion de documents dans un cadre éducatif. Pour cela nous avons expérimenté l'introduction d'une contrainte supplémentaire qui n'était pas prise en charge par l'algorithme de fouille. Cette contrainte est opérationnelle au moment du pré-traitement et prend en compte le contexte sous la forme d'un ensemble d'attributs afin de ne sélectionner que les occurrences d'épisodes pertinents au sens de ce contexte.

La suite de l'article est organisée comme suit : la section 2 synthétise les principales contributions aux problématiques, puis la section 3 présente le contexte scientifique et le cadre applicatif des propositions de recherches. Enfin notre proposition est exposée au paragraphe 4 et est suivie par une expérimentation qui permet de d'évaluer sa pertinence. Enfin nous discutons et concluons sur ce travail avec quelques perspectives.

2 État de l'art : exploration de données temporelles

La fouille de données assure le traitement automatique de gros volumes de données pour trouver des *motifs* correspondant à des régularités dans les données. Après interprétation par un utilisateur, ces motifs permettent de construire un modèle d'un phénomène étudié. Les traces sont des données multidimensionnelles et caractérisées par deux dimensions importantes : le type d'action et à quelle moment l'action a eu lieu, sous forme d'une estampille temporelle. Plusieurs méthodes ont été conçues afin de prendre en compte ces deux dimensions. La fouille de séquences tient compte de la disposition ordonnée des données dans les méthodes d'exploration et dans la forme des résultats produits. La découverte de motifs séquentiels dans des bases de séquences s'appuie sur une base de transactions comportant une séquence d'itemsets chacune, et l'extraction consiste à y rechercher les sous-séquences fréquentes d'itemsets (Agrawal & Srikant, 1995). L'inconvénient de cette approche pour l'exploration des traces est qu'elle ne prend pas en compte la dimension *événementielle* comme dimension prioritaire : si elle est présente, elle est considérée à rang égal avec les autres dimensions. Par ailleurs, les traces ne se présentent pas sous la forme d'une base de séquence mais plutôt sous la forme d'une ou plusieurs séquences d'actions où chaque action est associée à un ensemble d'attributs et à une information temporelle précise, et tous les types d'action ne possèdent pas nécessairement les mêmes attributs. Une méthode alternative, la découverte d'épisodes fréquents introduite par (Mannila *et al.*, 1997) exploite des données sous la forme d'une séquence d'événements où chaque événement est associé à une information temporelle précise. Cette méthode est plus adaptée à l'exploration des traces car les actions composant la trace peuvent aisément être transposées en événements dont le type correspond au type d'action et l'estampille à l'information temporelle présente dans la trace (ou au moins l'ordre). Néanmoins les attributs caractérisant le contexte des actions de la trace ne sont pas pris en compte et des traitements complémentaires s'avèrent nécessaires en amont et/ou en aval de l'étape de fouille. Si quelques variantes ont été proposées dans la littérature (Cram, 2010), aucune d'elles n'est satisfaisante compte tenu des caractéristiques des traces.

Par ailleurs, la nature multidimensionnelle des données rajoute de la difficulté au problème

de la surabondance des résultats. Les mesures d'intérêt *objectives* à la base des stratégies de fouille permettent de réduire le temps et l'espace nécessaires à l'exploration des données, notamment grâce à leur propriétés. Elles s'appuient uniquement sur les données explorées et ne présupposent aucune connaissance supplémentaire sur les données à traiter. Parmi les mesures objectives les plus classiques on peut citer le support et la confiance qui sont connues pour être insuffisants si l'on veut extraire des informations utiles et intéressantes (Geng & Hamilton, 2007). La compacité des résultats avec la propriété de fermeture des motifs a montré sa capacité à limiter fortement les résultats mais reste toutefois toujours insuffisante. Conjointement aux mesures objectives, il est également possible d'exploiter les connaissances *a priori* que l'utilisateur possède sur les données sous la forme de mesures d'intérêt *subjectives*. Si les mesures d'intérêt subjectives s'avèrent souvent très efficaces pour réduire considérablement le volume de résultats extraits par la fouille, elles ne sont pas toujours représentables simplement. Elles peuvent être difficiles à prendre en main par l'utilisateur qui doit apprendre comment formuler ses connaissances sous une forme appropriée afin de les transposer en une ou plusieurs mesures (Geng & Hamilton, 2007). Par ailleurs elles ne sont pas toujours transposables facilement dans un autre domaine d'application ou manquent parfois de généricité si bien qu'il est délicat de les prendre en compte dans l'algorithme de fouille. On peut ajouter que, si on trouve dans la littérature de nombreux travaux qui traitent des règles d'association, il est en revanche plus rare de trouver des travaux qui s'intéressent aux épisodes séquentiels, si ce n'est pour des besoins spécifiques (Perer & Wang, 2014).

Dans le cadre de l'exploration de traces, nous proposons d'une part d'exploiter l'extraction d'épisodes séquentiels et conjointement d'utiliser les connaissances *a priori* sur les traces afin de mieux préparer les données au moment du pré-traitement. Ces connaissances s'expriment de façon très simple et générique, et permettent de segmenter une trace en plusieurs séquences avant la fouille. Nous présentons dans la section suivante le cadre de ce travail.

3 Extraction de connaissances à partir de traces

Nous avons développé DISKIT afin d'explorer les traces. DISKIT peut être vu comme un laboratoire d'analyse des traces destiné à être complété par d'autres méthodes en amont et/ou en aval afin d'étudier des problématiques plus précises autour des traces, en particulier (mais pas seulement) des traces d'apprenants. DISKIT met en œuvre les étapes de pré-traitement et de post-traitement du processus d'ECD et encapsule l'étape de fouille assurée par DMT4SP [2] (Nanni & Rigotti, 2007), un prototype d'extraction d'épisodes séquentiels et de règles séquentielles à un conséquent à partir d'une ou plusieurs séquences d'événements. DISKIT prend en charge plusieurs options et contraintes en complément de DMT4SP. Les données de la trace sont collectées soit sous la forme d'un fichier texte soit à partir d'un *système à base de traces* (Champin *et al.*, 2013) qui sert à la fois d'entrepôt destiné au stockage, à la manipulation et la modélisation explicite des traces, mais aussi de base de connaissances afin de mémoriser les interprétations faites de phénomènes étudiés à partir des traces. Nous proposons d'expérimenter avec DISKIT la prise en compte du contexte des actions de la trace pour mieux focaliser la fouille sur des épisodes pertinents. Dans ce travail nous nous sommes appuyés sur les traces du jeu Tamagocours fournies sous la forme d'un fichier au format `csv`. Il s'agit de rechercher des séquences d'actions signifiantes réalisées par les utilisateurs. Nous présentons dans un premier temps le cadre applicatif des travaux qui serviront à illustrer les concepts présentés par la suite.

2. Data Mining Techniques For Sequence Processing,
`http://liris.cnrs.fr/~crigotti/dmt4sp.html`

id	date	actionType	group_id	user_id	grpus_id	Codage	message	help	res_id	item_id	resourceType	mode_of_use	resource_title	creationDate	rightsAgreements	res_size	item_size	reason	game_id	level_id	isWon
5	30/03/2015 11:05:45	fill Cupboard	2													/			2	1	1
7	30/03/2015 11:06:45	help Link	2	3	2_3			Accès aide en ligne								/			2	1	1
8	30/03/2015 11:06:49	tuto	2	3	2_3											/			2	1	1
12	30/03/2015 11:07:42	showItem CUPBOARD	2	3	2_3				43	339	book		Le Grand M	1913	Domaine Public	/	4		2	1	1
16	30/03/2015 11:07:52	showItem CUPBOARD	2	3	2_3				113	529	journal		Le Figaro		Domaine public	/	intégrale		2	1	1
18	30/03/2015 11:07:59	showItem CUPBOARD	2	3	2_3				43	335	book		Le Grand M	1913	Domaine Public	/	intégrale		2	1	1
19	30/03/2015 11:08:02	showItem CUPBOARD	2	4	2_4				108	508	journal		The Washin	1983	Droits d'auteur	/	5 articles		2	1	1
20	30/03/2015 11:08:19	addTo Fridge	2	3	2_3				43	335	book	printedCopies	Le Grand M	1913	Domaine Public	/	intégrale		2	1	1
22	30/03/2015 11:08:38	chat	2	3	2_3	OJ	Quelqu'un sait ce qu'il faut faire??									/			2	1	1
23	30/03/2015 11:08:39	showItem FRIDGE	2	4	2_4				43	335	book	printedCopies	Le Grand M	1913	Domaine Public	/	intégrale		2	1	1
24	30/03/2015 11:08:45	chat	2	6	2_6	OJ	aucune idée!									/			2	1	1
25	30/03/2015 11:08:45	tuto	2	3	2_3											/			2	1	1
26	30/03/2015 11:09:04	feedTamago Good	2	3	2_3				43	335	book	printedCopies	Le Grand M	1913	Domaine Public	/	intégrale		2	1	1
28	30/03/2015 11:09:30	showItem CUPBOARD	2	4	2_4				108	506	journal		The Washin	1983	Droits d'auteur	/	1 article		2	1	1

TABLE *1 – Un extrait du fichier des traces du jeu Tamagocours. Les attributs sont représentés en colonnes, et la colonne* `actionType` *représente le type d'action réalisé par un utilisateur. Chaque ligne est une action repérée par un identifiant (*`id`*) et par une estampille (*`date`*). Tous les attributs ne sont pas définis pour tous les types d'action (Fuchs, 2017).*

3.1 Les traces étudiées

Tamagocours (Sanchez *et al.*, 2015) est un jeu collaboratif destiné à l'apprentissage des règles juridiques auxquelles est soumis l'usage de ressources numériques dans le cadre éducatif. Les utilisateurs sont répartis en groupes de 2 à 4 joueurs et doivent alimenter un «Tamago» avec des ressources pédagogiques. Les utilisateurs peuvent consulter les caractéristiques des ressources et les associer à un mode d'utilisation puis les donner au Tamago pour le «nourrir». Le Tamago est associé à un score qui évolue au fur et à mesure des réussites (ressource autorisée) ou échecs (utilisation d'une ressource hors du cadre légal) des actions des utilisateurs du groupe. Les traces de deux sessions de jeu qui ont eu lieu en 2015 et en 2016 ont été collectées dans deux fichiers au format csv. Elles représentent au total 25 944 lignes pour la session 2015 et 20 752 pour la session 2016, chaque ligne correspondant à une action enregistrée dans le jeu. Les actions sont décrites à l'aide de 24 attributs. Un extrait de ce fichier est montré dans la table 1, et la table 2 contient la liste des types d'actions du jeu.

Dans Tamagocours, une séquence de jeu typique est décrite par la séquence `showItemCUPBOARD,addToFridge,feedTamagoxxxx` qui signifie : l'utilisateur consulte une *ressource* sur l'étagère, range *cette ressource* dans le réfrigérateur puis alimente le tamago avec *cette ressource*. Cette séquence peut être étudiée afin d'observer d'une part son issue qui peut être `xxxx = Good` si l'utilisateur a gagné ou `xxxx = Bad` si l'utilisateur a perdu cette séquence de jeu, et d'autre part les autres actions intercalées dans cette séquence, par exemple utilisation du tutoriel, de l'aide, consultation des autres utilisateurs du groupe par des actions de type «`chat`», etc. Par la suite, nous abrégeons ces deux épisodes respectivement `show,add,Good` et `show,add,Bad`. Il est possible de remarquer que c'est la ressource, associée à un mode d'utilisation, qui relie les trois actions. Néanmoins la fouille avec DMT4SP ne prend pas en compte cette information pour sélectionner les occurrences de motifs, ses choix sont uniquement déterminés par les types d'événement, leur proximité temporelle, et la sémantique d'occurrence minimale que nous décrivons dans la section suivante.

addToFridge	ajouter une ressource dans le frigo
chat	envoyer un message
feedTamagoBad	Nourrir le Tamago avec une bonne
feedTamagoGood	ou une mauvaise ressource
fillCupboard	Remplissage de l'étagère avec des ressources
helpLink	affichage de l'aide
removeFromFridge	supprimer une ressource du frigo
showItemCUPBOARD	examiner une ressource placée sur l'étagère
showItemFRIDGE,	examiner une ressource dans le frigo,
showItemLEVEL	examiner une ressource dans le tableau de fin de niveau,
showItemTAMAGO	examiner une ressource dans le Tamago.
showItemSTOMACH	examiner les ressources dans l'estomac du Tamago.
tuto	Consultation du tutoriel

TABLE *2 – Les différents types d'action utilisateur dans le jeu Tamagocours.*

3.2 Extraction d'épisodes séquentiels

DMT4SP est un prototype d'extraction d'épisodes séquentiels et de règles séquentielles à un événement conséquent à partir d'une ou plusieurs séquences d'événements, conformément à une *sémantique d'occurrence minimale* adaptée de (Mannila *et al.*, 1997). DMT4SP prend en entrée une ou plusieurs séquences, un ensemble de paramètres et produit un ensemble d'épisodes séquentiels avec pour chacun d'eux : la fréquence, le nombre de séquences, éventuellement la confiance pour les règles séquentielles, ainsi que, pour chaque occurrence de l'épisode, l'intervalle de temps et le numéro de séquence pour la localiser. Les épisodes séquentiels extraits par DMT4SP sont conformes à la définition suivante :

Définition 1 (Séquence, événement, type d'événement)
Une ***séquence*** $s = \langle (e_1, t_1), (e_2, t_2), \dots (e_n, t_n) \rangle$ *est une suite ordonnée d'****événements*** *où*
$(e_i, t_i)_{i=1\dots n}$ *est un événement,*
$e_i \in E$ *est un* ***type d'événement****, et* E *est l'ensemble des types d'événements, et*
$t_i \in \mathbb{N}$ *est une estampille associée à* e_i *telle que* $\forall i, t_i \in \mathbb{N}$ *et* $t_i < t_{i+1}$.

Définition 2 (épisode séquentiel, occurrence minimale, fréquence)
Soit $S = \{s_k\}_k$ *un ensemble de séquences.*
Un ***épisode séquentiel*** $p = \langle e_1, e_2, \dots, e_m \rangle, e_i \in E$ *est une séquence de types d'événements de longueur* m.
Une ***occurrence*** o *de l'épisode séquentiel* p *est une séquence d'estampilles distinctes* $\langle t_1, t_2, \dots t_m \rangle$ *telles que* $\exists k, (e_i, t_i) \in s_k$ *et* $\forall\, i < j \in [1, m], t_i < t_j$.
Une occurrence o *de l'épisode* p *est dite* ***minimale*** *si elle n'inclut pas une autre occurrence du même épisode dans* s_k*, c'est-à-dire s'il n'existe pas d'occurence* $o' = \langle t'_1, t'_2, \dots, t'_m \rangle$ *telle que* $(t_1 < t'_1$ *et* $t'_m = t_m)$ *ou* $\exists\, i \in [2, m], (t_1 = t'_1$ *et* $t'_i < t_i)$.
Si $O = \{o_i\}_i$ *est l'ensemble des occurrences de l'épisode* p *dans* S*, la* ***fréquence*** *d'un épisode séquentiel* p *est définie par* $\sigma(p) = |O|$.
Un épisode séquentiel p *est* ***fréquent*** *si* $\sigma(p) \geqslant \sigma_{min}$*, où* σ_{min} *est le support minimum choisi par l'utilisateur. La fouille retourne un ensemble* P *d'épisodes fréquents tels que* $P = \{p_i\}, \forall i\ \sigma(p_i) \geqslant \sigma_{min}$.

Les définitions ci-dessus appellent plusieurs remarques. Tout d'abord il est possible de remarquer qu'une occurrence d'épisode est définie par des estampilles distinctes, elle ne peut donc pas contenir plusieurs événements distincts ayant la même estampille. Par exemple

dans la séquence suivante :

Exemple 1 :

		Good	
S	show	add	Good
t_i	1	2	3

DMT4SP ne sélectionne pas l'occurrence $\langle$(show, 1),(add, 2),(Good, 2)$\rangle$ mais sélectionne l'occurrence $\langle$(show, 1),(add, 2),(Good, 3)$\rangle$.

Ensuite, la documentation de DMT4SP, précise que la définition d'une occurrence minimale est adaptée par rapport à la définition de (Mannila *et al.*, 1997) : elle impose que les types d'événements intermédiaires et terminal e_2 à e_m doivent se produire «le plus tôt possible». Soient par exemple l'épisode show,add,Good, et trois occurrences de cet épisode resituées ci-dessous dans un extrait de séquence :

Exemple 2 :

o_1		show	add			Good
o_2	show		add			Good
o_3		show			add	Good
S	show	show	add	remove	add	Good
t_i	1	2	3	4	5	6

Dans cet exemple, la seule occurrence minimale au sens de DMT4SP est o_1, qui correspond aux estampilles $\langle$2, 3, 6$\rangle$. DMT4SP fournit donc en sortie l'intervalle [2, 6] pour une seule occurrence. La définition est équivalente à celle de (Mannila *et al.*, 1997) où une occurrence minimale est définie come un intervalle de temps contenant (au moins) une occurrence minimale : définir une occurrence minimale comme un intervalle ne tient pas compte du fait qu'il peut y avoir plusieurs occurrences de l'épisode séquentiel dans cet intervalle. Ceci a une conséquence sur le support (ou fréquence [3]) qui est calculé à partir des intervalles des occurrences minimales et ne reflète pas le nombre réel d'occurrences minimales à l'intérieur de ces intervalles. En réalité, même si la définition de DMT4SP est un peu différente de (Mannila *et al.*, 1997), elle est équivalente, car les occurrences sorties par DMT4SP sont définies par l'intervalle de temps contenant (au moins) une occurrence d'épisode, et la fréquence est définie comme le nombre d'intervalles de temps où l'épisode est trouvé. Les conséquences de ces remarques sur les résultats seront abordées dans la section suivante.

Il existe par ailleurs dans DMT4SP une multitude de contraintes sur les épisodes séquentiels qui s'avèrent utiles afin de limiter les résultats. Dans DMT4SP, deux définitions du support sont prises en compte. La première correspond à celle de la définition 2, c'est-à-dire le nombre d'occurrences minimales trouvées dans toutes les séquences. La deuxième définition est le nombre de séquences qui contiennent au moins une occurrence minimale. Il est possible de spécifier un seuil minimum pour ces deux supports, et seuls les épisodes qui satisfont simultanément ces deux seuils sont sélectionnés. Des contraintes temporelles permettent de limiter l'étalement des épisodes et des événements dans le temps. La fenêtre temporelle est l'intervalle de temps maximal entre le premier et le dernier événement des occurrences d'un épisode. L'intervalle de temps (min/max) entre deux événements consécutifs d'un épisode peut également être précisé. Il est également possible de contraindre la longueur des épisodes (min/max), et leur imposer un préfixe composé d'une séquence d'événements consécutifs par lesquels les épisodes doivent débuter, ou un suffixe constitué d'un seul événement terminal.

3. selon que celui-ci est calculé de façon absolue ou relative

DMT4SP fournit les résultats sous une forme textuelle avec, pour chaque épisode séquentiel satisfaisant les contraintes spécifiées par l'utilisateur : un numéro unique, la liste des types d'événements composant l'épisode, la fréquence (nombre d'occurrences), ainsi que les informations de localisation des occurrences de l'épisode sous la forme d'un intervalle de temps contenant l'occurrence, ainsi que le numéro de séquence.

3.3 DISKIT

DISKIT prend en charge les traitements en amont et en aval de la fouille. En pré-traitement, DISKIT construit une ou plusieurs séquences à partir d'une trace en associant chaque type d'action à un type d'événement et inversement au cours du post-traitement. Cette transformation syntaxique est bijective et a pour unique objectif de se conformer au format requis par DMT4SP. Dans la suite, les termes «action» et «événement» seront utilisés comme synonymes de ce fait. Puis DISKIT déclenche la fouille avec les données et paramètres qui lui ont été fournis, récupère les résultats de la fouille et les met en forme afin de les rendre intelligibles. Les résultats sont restitués dans un fichier texte en sortie.

DISKIT effectue par ailleurs d'autres traitements qui ne sont pas pris en charge par DMT4SP. Tout d'abord DISKIT effectue le calcul de la *fermeture* des motifs en post-traitement en s'appuyant sur la fréquence. Deux options permettent, au moment du post-traitement, de sélectionner les épisodes contenant (ou ne contenant pas respectivement) un *pattern* donné sous la forme d'une séquence de types d'événements. Si les types d'événements se retrouvent dans le même ordre sans être nécessairement contigüs dans les épisodes retournés par DMT4SP, ceux ci sont sélectionnés (respectivement éliminés). Dans l'une des expérimentations de la section suivante, nous avons utilisé cette option afin de restreindre l'étude aux épisodes `show,add,Good` et `show,add,Bad` : les trois types d'événements devaient se retrouver dans cet ordre dans les épisodes pour que ces derniers soient sélectionnés. DISKIT permet également de prendre en compte les attributs caractérisant les actions. L'option `split` permet, au moment du pré-traitement, de fractionner une traces en plusieurs séquences en fonction des valeurs des attributs donnés en argument. Ceci permet à DMT4SP de rechercher des épisodes signifiants dans un ensemble de séquences plus petites. Par exemple il est au minimum nécessaire de fractionner la trace en entrée par groupe d'utilisateurs de façon à «isoler» l'analyse des actions des différents groupes, qui travaillent de façon indépendante, dans des séquences séparées. Nous présentons dans la section suivante la mise en œuvre de cette option pour rechercher des occurrences signifiantes d'épisodes séquentiels.

4 Prendre en compte le contexte

Une trace est issue de l'observation d'une *activité* et témoigne de l'existence d'actions passées *situées* qui ont été réalisées par des *acteurs* en interaction avec leur *environnement*. Nous nous intéressons ici aux traces laissées par des utilisateurs dans un environnement numérique et issues d'un processus de collecte sous la forme d'éléments *observés* : des actions associées à des éléments de contexte qui apportent des précisions d'ordres différents allant des *objets* manipulés par l'utilisateur par exemple, ou bien d'autres acteurs avec lesquels il interagit, *etc.* Les actions enregistrées dans une trace doivent être associées à un contexte explicite, qui ont été capturées lors du processus de collecte. Néanmoins, il n'est pas possible de «tout» collecter, soit parce que tous les aspects du phénomène étudié ne sont pas observables, soit parce qu'il n'est pas possible de prévoir toutes les utilisations qui pourront être faites des traces en aval. Cependant, dans le cas des traces numériques, le concepteur instrumente l'application

de façon à collecter le plus d'éléments informatifs possibles qui servent de support dans le but d'en faciliter l'interprétation (Champin *et al.*, 2013).

Dans le cadre du jeu Tamagocours, la prise en compte du contexte permet de focaliser l'analyse sur le but poursuivi par les utilisateurs du jeu : le contexte se situe dans le cadre d'équipes constituées de plusieurs utilisateurs (les *acteurs*, qui peuvent être étudiés soit individuellement, soit dans le cadre d'entités «groupe»), dans une séquence de plusieurs jeux (des *mises en situation* de difficultés croissantes) et portant sur des ressources numériques associées à un mode d'utilisation (les *objets*).

Lors de l'exploration des traces du jeu, il s'agit donc de retrouver des épisodes séquentiels qui ont du sens du point de vue du contexte d'apparition des d'actions de la séquence. Concrètement, le contexte est représenté dans les traces par un ensemble d'attributs. Lors de l'exploration des traces par DMT4SP, la proximité temporelle est prépondérante, mais il faut néanmoins tenir compte du fait que les sessions du jeu Tamagocours ont été menées avec plusieurs groupes d'utilisateurs en parallèle, et la trace témoigne de cette organisation : les actions des utilisateurs sont organisées séquentiellement avec le seul critère temporel, et les différentes actions des groupes et des utilisateurs se retrouvent «mélangées» dans la trace. Il est donc obligatoire et indispensable d'organiser les données au moment de la préparation en amont de la fouille pour tenir compte d'une part du désordre dans les données et d'autre part de l'indépendance du travail des différents groupes d'utilisateurs, de sorte que la recherche d'épisodes ait lieu au sein séquences «cohérentes».

Soit par exemple l'extrait suivant de la trace de la session 2015 :

Exemple 3 :

id	605	606	608	610	611	612	614	616	618	619
action	show	show	show	add	Good	add	Good	add	help	Bad
group	4	3	3	4	4	3	3	4	4	4
user	13	18	18	12	12	9	9	13	14	13
item	385	363	264	736	736	363	363	385		385
game	19	16	16	19	19	16	16	19	19	19

Si le contexte n'est pas pris en compte, DMT4SP sélectionne les occurrences minimales `(show,608),(add,610),(Good,611)` et `(show,608),(add,610),(Bad,619)`. Les actions `(show,606)` et `(Good,614)` ne peuvent par conséquent pas être sélectionnées dans une occurrence minimale.

Si le numéro de groupe est pris en compte dans le contexte, une occurrence ne peut se situer que dans un même groupe, ce qui revient à analyser les deux séquences suivantes :

Exemple 4 :

id	606	608	612	614	605	610	611	616	618	619
action	show	show	add	Good	show	add	Good	add	help	Bad
group	3	3	3	3	4	4	4	4	4	4
user	18	18	9	9	13	12	12	13	14	13
item	363	264	363	363	385	736	736	385		385
game	16	16	16	16	19	19	19	19	19	19

Dans cette situation, l'occurrence `(show,608),(add,612),(Good,614)` peut être sélectionnée pour le groupe 3, et `(show,605),(add,610),(Good,611),(show,605),(add,610),(Bad,619)` pour le groupe 4. Toutefois, aucune de ces occurrences ne porte sur le même numéro d'item (un item est l'association d'une ressource et d'un mode d'utilisation), la seule occurrence qui ait du sens est `(show,605),(add,616), (Bad,619)`, mais celle ci n'est pas considérée comme

minimale par DMT4SP qui ne prend pas en compte le contexte. Il est par conséquent nécessaire de contextualiser davantage les épisodes à l'aide d'autres attributs comme le numéro de jeu car les différents jeux qui se succèdent sont indépendants, mais également sur le numéro d'item. Une meilleure façon d'analyser la trace consisterait donc à rechercher les épisodes dans les séquences suivantes :

Exemple 5 :

	item 385				item 363			item 264	item 736		
id	605	616	618	619	606	612	614	608	610	611	618
action	show	add	help	Bad	show	add	Good	show	add	Good	help
group	4	4	4	4	3	3	3	3	4	4	4
user	13	13	14	13	18	9	9	18	12	12	14
item	385	385		385	363	363	363	264	736	736	
game	19	19	19	19	16	16	16	16	19	19	19

Dans cet ensemble de séquences, DMT4SP trouverait les deux occurrences minimales : `(show,605),(add,616),(Bad,619)` pour l'item 385 et `(show,606),(add,612),(Bad,614)` pour l'item 363. Il est possible de remarquer dans cette dernière occurrence que des utilisateurs différents ont collaboré pour la réalisation de la séquence de jeu. Il est également possible de remarquer que l'action `help` n'est associée à aucun item, c'est la raison pour laquelle elle apparait dans toutes les séquences du même groupe et du même jeu car il n'est pas possible de *décider* si – et quel – item est concerné par la consultation de l'aide. Ce point sera abordé dans la section 4.1.

4.1 Le contexte dans FINEPIO

Nous avons développé un algorithme appelé FINEPIO [4] et qui recherche toutes les occurrences minimales d'un épisode séquentiel en prenant en compte le contexte. FINEPIO prend en entrée une trace, un ou plusieurs épisodes séquentiels dont on souhaite rechercher toutes les occurrences, un contexte sous la forme d'un ensemble d'attributs, et recherche dans la trace toutes les occurrences des épisodes dans le contexte spécifié. On peut remarquer que la fréquence d'un épisode recherché avec FINEPIO peut être supérieure à celle calculée par DMT4SP, car il est possible de trouver plusieurs occurrences d'épisode dans chaque intervalle de temps qui n'est compté qu'une fois par DMT4SP. Les définitions associées à FINEPIO prennent en compte cette particularité dans les définitions suivantes.

Définition 3 (Séquence, événement)
Une ***séquence*** s *est une suite ordonnée d'événements :*
$s = \langle (e_1, t_1, d_1, c_1), (e_2, t_2, d_2, c_2), \dots (e_n, t_n, d_n, c_n) \rangle$ *où*
Un ***événement*** *est un quadruplet* $(e_i, t_i, d_i, c_i)_{i=1\dots n}$ *avec :*
$e_i \in E$*, est le* ***type d'événement****,*
$t_i \in \mathbb{N}$ *est une* ***estampille*** *associée à* e_i*,* $\forall\, i < j \in \mathbb{N}, t_i \leq t_j$*,*
$d_i \in \mathbb{N}$ ***identifie*** *tout événement de façon unique :* $\forall j \neq i, d_j \neq d_i$*,*
$C_i = \{(a_i^j, v_i^j)\}$ *est le contexte de l'événement où :*
$A_i = \{a_i^j\}$ *est l'ensemble des attributs associés à l'événement,*
avec $A_i \subseteq A$*,* A *étant l'ensemble de tous les attributs, et*
v_i^j*, est la valeur de l'attribut* a_i^j *pour l'occurrence* i
qui prend ses valeurs dans le domaine de valeurs de l'attribut considéré.

4. FINd EPIsode Occurrences.

Définition 4 (Contexte, contexte valué)
Un contexte $C = \{a^k\}_{k=1...q} \subseteq A$ définit l'ensemble des attributs à prendre en considération dans les événements. C'est un sous-ensemble d'attributs de A dont les valeurs doivent coïncider dans les événements composant les occurrences d'épisodes.
Un ***contexte valué*** *$c_i \subseteq C_i$ est l'ensemble les valeurs prises par les attributs de contexte dans un événement i : c'est un ensemble de couples (attribut, valeur) correspondant à un contexte C pour un événement i :*
$c_i = \{(a_i^j, v_i^j)\}_{i=1...n, j \leq q}$, *tel que* $a_i^j \in C \cap C_i$

D'après cette définition, il est possible, pour un événement i, que $C \not\subseteq A_i$, c'est à dire que certains attributs de C ne soient pas définis pour un événement i, comme c'était le cas pour l'action `help` dans l'exemple 5.

Dans FINEPIO, la définition d'un épisode séquentiel est identique à celle de DMT4SP, mais la définition des occurrences et des occurrences minimales diffère.

Définition 5 (épisode séquentiel, occurrence, occurrence minimale)
Un ***épisode séquentiel*** *p est une séquence ordonnée de types d'événements*
$p = \langle e_1, e_2, \ldots, e_m \rangle, e_i \in E$ *de longueur* m
Une ***occurrence*** $o = \langle (e_1, t_1, d_1, c_1), (e_2, t_2, d_2, c_2), \ldots (e_m, t_m, d_m, c_m) \rangle$ *de l'épisode séquentiel p dans le contexte C est une sous séquence d'une séquence s telle que :*
$(e_i, t_i, d_i, C_i)_{i=1...m} \in s, \forall\, i < j \in [1, m], t_i < t_j$ *et* $\forall\, i \neq j \in [1, m], d_i \neq d_j$ *et*
$\exists (e_k, t_k, d_k, C_k) \in o$ *tel que* $C \subseteq A_k$ *et* $\forall i \neq k, c_i \subseteq c_k$
Cette dernière contrainte oblige l'existence d'au moins un événement dans l'occurrence qui possède tous les attributs du contexte afin de pouvoir établir l'inclusion.
Une occurrence $o = \langle (e_i, t_i, d_i, c_i) \rangle_{i=1,m}$ de l'épisode p est ***minimale***
si toute occurrence o' de p, $o' = \langle (e'_i, t'_i, d'_i, c'_i) \rangle_{i=1,m}$ est telle que
$t_1 < t'_1$ *et* $t_m < t'_m$ *ou* $t_1 > t'_1$ *et* $t_m > t'_m$ *ou* $t_1 = t'_1$ *et* $t_m = t'_m$
On note $O = \{o_i\}_i$ l'ensemble des occurrences de l'épisode p dans S.
La ***fréquence*** *d'un épisode séquentiel p est le nombre d'occurrences minimales de cet épisode dans S. Elle est notée $\sigma(p) = |O|$.*

La définition des occurrences d'épisodes impose qu'il existe au moins un événement dans l'occurrence dont le contexte valué contient tous les attributs du contexte, c'est-à-dire un événement k tel que $C \subseteq A_k$. Les contextes valués des autres événements de l'occurrence d'épisode doivent être soit inclus dans ce dernier, soit égal. L'absence de valeur dans un contexte valué ne signifie pas que le contexte est différent, mais qu'il est incomplet, et il est alors *possible* que l'événement correspondant soit pertinent dans un contexte, mais il n'est pas possible de *décider* dans quel contexte précisément. Comme dans l'exemple 5 précédent, une action de type `chat` ou `help` ne comporte pas d'item, mais si ces actions interviennent dans un même groupe et dans un même jeu, elles sont susceptibles de concerner un des items manipulés. La définition des occurrences minimales diffère de celle de (Mannila *et al.*, 1997) où une occurrence minimale est définie comme l'intervalle de temps contenant une occurrence minimale. Dans la définition 5, c'est l'ensemble de toutes les occurrences minimales dans l'intervalle de temps considéré, comme dans l'exemple 2, où les deux occurrences o_1 et o_3 sont minimales.

4.2 Expérimentations

Dans DISKIT, le contexte est pris en compte en répartissant les actions composant la trace dans autant de sous-ensembles qu'il y a de contextes différents. Cela revient à séparer les contextes avant la fouille de façon à assurer que les occurrences d'épisodes sélectionnées par

DMT4SP ont un sens du point de vue de leur contexte. La trace est donc divisée en autant de séquences à explorer qu'il y a de combinaisons de valeurs différentes pour les attributs constituant le contexte. Nous avons expérimenté l'introduction du contexte avec DISKIT afin d'évaluer l'efficacité résultante du point de vue du rappel et de la précision pour des motifs bien identifiés, et leur évolution en fonction de la prise en compte du contexte. Nous avons lancé DISKIT en faisant varier les paramètres de contexte, et nous avons comparé les résultats à ce que nous aurions *idéalement* souhaité trouver dans les résultats et qui a été calculé par FINEPIO. Pour rendre compte de l'efficacité de la stratégie par les indicateurs de rappel et de précision à partir du nombre d'épisodes trouvé par FINEPIO, nous avons focalisés la recherche sur deux épisodes : `show,add,Good` et `show,add,Bad`. Le comptage des occurrences minimales au cours du post-traitement a été réalisé de deux façons différentes par DISKIT. La première utilise la sémantique d'occurrence minimale de DMT4SP et la seconde utilise l'algorithme FINEPIO qui recherche *toutes* les occurrences minimales d'un épisode séquentiel à partir des intervalles fournis par DMT4SP en prenant en compte le contexte. Nous avons exploré les traces de deux sessions de jeu ayant eu lieu en 2015 et en 2016 pour rechercher les deux épisodes `show,add,Good` et `show,add,Bad` et en introduisant progressivement le contexte de ces épisodes : le numéro de groupe, le numéro de jeu, le numéro d'utilisateur et le numéro d'item.

La table 3 contient les principaux paramètres des expériences réalisées. Ils ont été choisis afin de limiter le temps de traitement et le volume de résultats tout en assurant le rappel de toutes les occurrences des épisodes `show,add,Bad` et `show,add,Good`. La longueur minimum et maximum choisie est de 3, la fréquence minimum est de 800, inférieure à celle des deux épisodes `show,add,feed` et rend inutile l'introduction d'une contrainte temporelle pour accélérer le traitement. Le paramètre estampille temporelle permet de calculer les estampilles à l'aide des dates enregistrées dans les actions de la trace (valeur oui) ou à l'aide d'un simple numéro séquentiel (non). Dans ces différentes expériences, nous avons fait varier l'importance de la prise en compte du contexte et noté dans le tableau le nombre de séquences ainsi que le nombre d'événements issus du découpage de la trace. La première expérience a été réalisée sans prise en compte du contexte, la deuxième en introduisant le groupe d'utilisateurs, la troisième expérience en introduisant le groupe et le jeu et la quatrième en introduisant le groupe, le jeu et l'utilisateur. Enfin la cinquième et la sixième ont été réalisées en introduisant le groupe, le jeu et l'item.

Exp.	1	2	3	4	5	6
estampilles temporelles	oui					non
minsup	800					
inclusion de patterns	show,add,feedGood show,add,feedBad					
longueur min/max	3 / 3					
découpage		group	group game	user	group game item	
nb de séquences	2	152	1 177	3 359	17 557	17 557
nb d'événements	46 696	46 696	46 696	47 909	170 884	170 884

TABLE *3 – Les paramètres utilisés pour l'expérimentation.*

Une caractéristique importante qui rend plus délicate la prise en compte du contexte est que tous les types actions du jeu Tamagocours ne possèdent pas les mêmes attributs. Les actions `chat`, `tuto` et `help` en particulier ne possèdent pas l'attribut `item_id`). Il existe cependant des attributs qui sont partagés par tous les types d'actions (au moins obligatoirement l'identifiant, le type d'action et la date). La définition 5 de FINEPIO prend en compte cette

	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5	Exp. 6
Occurrences d'épisodes retournées par DISKIT	9276	8194	7686	5381	4960	4978
retrouvées	917	1821	1821	4189	4958	4978
non retrouvées	4061	3157	3157	789	20	0
Rappel	18,42%	36,58%	36,58%	84,15%	99,60%	100,00%
Précision	9,89%	22,22%	23,69%	77,85%	99,96%	100,00%

TABLE *4 – Synthèse des résultats obtenus par* DISKIT *pour les occurrences des motifs* `show,add,Good` *et* `show,add,Bad` *en utilisant la sélection des occurrences minimales selon* FINEPIO*. Le nombre d'épisodes trouvés par* FINEPIO *est de 4978.*

	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5	Exp. 6
Occurrences d'épisodes retournées par DMT4SP	6071	5852	5696	5060	4925	4942
retrouvées	650	1375	1375	4171	4923	4942
non retrouvées	4328	3603	3603	807	55	36
Rappel	13,04%	27,62%	27,62%	83,79%	98,90%	99,28%
Précision	10,71%	23,50%	24,14%	82,43%	99,96%	100,00%

TABLE *5 – Synthèse des résultats obtenus par* DISKIT *pour les occurrences des motifs* `show,add,Good` *et* `show,add,Bad`*, lors de l'application de la sémantique d'occurence minimale de* DMT4SP *(le nombre d'épisodes trouvés par* FINEPIO *est de 4978).*

particularité dans la définition d'une occurrence, il ne s'agit pas d'obtenir une identité stricte des contextes dans une occurrence mais une inclusion des contextes, à condition que l'un au moins des événements de l'occurrence possède tous les attributs de contexte. Ceci présente l'avantage l'inclure un contexte de façon plus souple, et d'obtenir des résultats potentiellement plus riches. Mais la contrepartie est que certaines actions doivent être dupliquées dans plusieurs séquences ce qui augmente considérablement le nombre d'événements dans les séquences que DMT4SP doit traiter et a une influence sur le temps de traitement. Dans le cas de Tamagocours, les actions `chat`, `tuto` et `help` possèdent tout de même les deux attributs `group_id` et `game_id`, ce qui limite l'augmentation du nombre d'événements. On peut voir dans la table 3 qu'avec 46 696 actions dans la trace le nombre d'événements générés par le découpage augmente en particulier pour les exp. 5 et 6 pour atteindre 170 884 événements. Dans l'expérimentation réalisée ici, les types d'actions des épisodes `show,add,Good` et `show,add,Bad` possèdent tous les mêmes attributs.

Nous avons expérimenté en post-traitement deux versions de DISKIT conformément aux deux définitions précédentes : la définition 2 de DMT4SP (tableau 5), et la définition 5 de FINEPIO (tableau 4). Chacun d'eux regroupe les résultats obtenus pour les deux épisodes `show,add,Good` et `show,add,Bad` et pour les deux traces. Dans chaque tableau a été reporté le nombre total d'occurrences d'épisodes produits par DISKIT, le nombre d'occurrences d'épisodes pertinents retrouvés parmi ceux qui ont été produits et le nombre d'occurrences qui n'ont pas été retrouvées dans les résultats. Sur ces deux tableaux on peut remarquer que, en l'absence de prise en compte des éléments de contexte, le rappel est très faible et la méthode d'exploration est sans intérêt dans ces conditions car elle s'appuie sur le seul critère de proximité temporelle des événements (Exp. 1) et a très peu de chances de capturer les épisodes signifiants. L'introduction du groupe et du jeu (Exp. 2 et 3) améliore peu le rappel qui reste très faible, du fait des actions mêlées de plusieurs utilisateurs simultanément. Lorsque le numéro d'utilisateur est introduit, le rappel s'améliore de façon significative car le

plus grand nombre d'occurrences d'épisode `show,add,feed` est réalisé par le même utilisateur. Néanmoins, la nature collaborative du jeu fait que parfois plusieurs utilisateurs d'un même groupe contribuent ensemble à la réalisation d'une séquence `show,add,feed` sur un item donné. Par ailleurs un utilisateur peut travailler en même temps sur plusieurs ressources à la fois. Ce critère n'est par conséquent pas fiable. Enfin lorsque un contexte plus précis est introduit avec le numéro de groupe, de jeu et d'item, les séquences sont très contextualisées et le rappel des épisodes devient beaucoup plus important. Le nombre de séquences générées par le découpage avec le contexte devient très important, la taille des séquences devient beaucoup plus faible, ce qui rend le traitement plus rapide et il est alors inutile d'introduire de contraintes temporelles car les intervalles de temps des séquences sont réduits. La légère amélioration observée entre les expériences 5 et 6 vient de la précision des dates. L'unité de temps utilisée dans le jeu est la seconde, et deux actions réalisées de façon très rapide par un utilisateur peuvent par conséquent posséder la même date, même si l'ordre dans lequel elles apparaissent reflète leur instant d'apparition. Il serait nécessaire de re-séquentialiser les actions de la trace pour en tenir compte. La seule différence entre les expériences 5 et 6 est l'utilisation d'un simple numéro séquentiel à la place des dates, et qui reflète l'ordre d'apparition des actions. On obtient une légère amélioration du rappel, et la précision qui devient 100%. Dans le tableau 5, les occurrences d'épisodes non retrouvés dans l'expérience 6 et que l'on retrouve également dans l'expérience 5 sont des occurrences non minimales au sens de DMT4SP, elles sont toutes dues à l'hésitation des utilisateurs qui ont réalisé les séquences `show,add,remove,add,feed`. En effet, dans le jeu Tamagocours, une action `remove` supprime l'effet de l'action `add` qui la précède, et ce n'est que la deuxième action `add` qui rend possible l'action `Good`, comme dans l'exemple 2. Cependant, d'une part cette information n'est pas disponible explicitement et d'autre part elle est spécifique à Tamagocours. Il n'est donc pas possible *a priori* de décider quelle occurrence est la plus pertinente et FINEPIO sélectionne toutes les occurrences. De ce fait, si cette règle était prise en compte pour le comptage des occurrences d'épisodes, la précision de FINEPIO pour l'expérience 6 devrait diminuer légèrement.

4.3 Discussion

La préparation des données joue un rôle crucial d'une part pour l'efficacité du traitement et la diminution de la redondance des résultats et d'autre part pour assurer un traitement convenable des données prenant en compte l'organisation des données fournies. La sémantique d'occurrence minimale telle que définie dans DMT4SP est intéressante du point de vue de l'efficacité de la fouille, mais empêche le rappel de toutes les occurrences d'épisode si le contexte n'est pas pris en compte pour préparer convenablement les données. Une conséquence est que l'ensemble des occurrences minimales doivent être recherchées au moment du post-traitement et cette opération augmente légèrement le temps de traitement, mais le fractionnement n'augmente pas significativement le temps de traitement global. Seule la première expérience a un temps de traitement beaucoup plus long car il est plus coûteux de traiter une longue séquence plutôt que plusieurs séquences plus petites.

Dans FINEPIO, deux stratégies ont été mises en oeuvre pour rechercher les occurrences minimales. La première effectue une recherche contextuelle sans découper la trace. Cette méthode doit traiter une très longue séquence et reste assez lente malgré une indexation pour accélèrer le traitement. La deuxième s'appuie sur les attributs présents dans tous les événements (groupe et jeu) pour découper la trace puis effectue une recherche contextuelle dans l'ensemble des traces ainsi découpées. Cette stratégie est beaucoup plus rapide car la recherche est réalisée sur des séquences beaucoup plus petites.

L'intérêt de la contextualisation des épisodes séquentiels est double. Il permet tout d'abord

de fractionner une grande trace en de nombreuses séquences plus petites, rendant inutile l'introduction de contraintes temporelles. L'étape de fouille peut traiter ces séquences de façon efficace et produit moins de résultats redondants. De plus, les épisodes ainsi générés sont cohérents au regard du contexte dans lequel elles portent.

La prise en compte du contexte est réalisée *au sens large*, c'est à dire que si une action ne possède pas de valeur pour l'un au moins des attributs du contexte, alors cette action est rattachée à tous les contextes possédant une valeur pour les attributs considérés. Ce rattachement a pour conséquence une augmentation significative du nombre d'événements que DMT4SP doit traiter. La consultation du *modèle* associé à la trace permet de connaître les attributs associés à chaque type d'action (Fuchs, 2017) et ainsi de vérifier qu'une contrainte de contexte est applicable.

La modélisation proposée sous la forme de contrainte contextuelle est générique et peut s'appliquer à tout type de séquence si les informations présentes dans la trace le permettent. La contextualisation s'avère intéressante pour les traces du jeu Tamagocours, mais il reste encore à vérifier qu'elle l'est également pour d'autres traces et dans d'autres domaines. Elle repose sur l'hypothèse selon laquelle il est possible d'exprimer le contexte sous la forme d'un ensemble d'attributs représentant des notions telles que le sujet de l'action, l'objet de l'action, la phase de l'activité. Il n'est pas évident que toutes les applications et les traces en résultant soient organisés de cette manière, mais on peut raisonnablement penser qu'il existe dans les traces, outre les les types d'actions et leur instant temporel, au moins l'identification des utilisateurs et des objets manipulés dans le jeu. Nous avons observé les traces d'un autre jeu, ClassCraft, dans lequel ces informations sont bien présentes. Mais il reste à bien étudier et comprendre le jeu pour formuler des requêtes et introduire en conséquences des contraintes associées de sorte que l'exploration soit efficace.

Finalement, la façon de prendre en compte le contexte fait que l'on se ramène à une situation proche de la fouille d'une base de séquences, mais l'adéquation de cette méthode pour les traces reste à étudier, du fait de la non prise en compte de la dimension événementielle dans la forme des résultats.

La dimension temporelle est importante essentiellement du point de vue de l'ordre des actions, mais la précision des intervalles de temps entre actions ou entre les première et dernière actions d'un épisode est de moindre importance. Ces expérimentations ont mis en évidence que le contexte est bien plus important que la précision temporelle pour le rappel des occurrences d'épisodes.

5 Conclusion

Nous avons proposé une approche pour l'exploration de traces numériques fondée sur les épisodes séquentiels exploitant le contexte des actions pour focaliser l'analyse sur des épisodes signifiants.

La principale perspective à ce travail serait d'intégrer la prise en compte du contexte dans une méthode de fouille de traces à la manière de FINEPIO.

Il reste encore à continuer à développer DISKIT pour être capable de répondre à des requêtes plus complexes que pourraient se poser des utilisateurs d'EIAH. Par exemple, une des options de DISKIT permet notamment de formuler des requêtes sur l'absence de certains types d'événements dans des épisodes, ce qui est utile pour étudier sur l'utilité de certains dispositifs pédagogiques sur les performances des apprenants. D'autres expérimentations restent à mener pour l'éprouver dans d'autres domaines avec des contraintes et des requêtes plus variées. DISKIT ne se limite pas aux traces d'interaction mais à tout type de données symboliques et séquentielles. Nous l'avons utilisé dans des travaux précédents pour des partitions musicales (Fuchs & Cordier, 2016) ayant des caractéristiques différentes des traces.

Références

AGRAWAL R. & SRIKANT R. (1995). Mining sequential patterns. In *Data Engineering, 1995. Proceedings of the Eleventh International Conference on*, p. 3–14 : IEEE.

BÉCHET N., CELLIER P., CHARNOIS T. & CRÉMILLEUX B. (2014). Fouille de motifs séquentiels pour la découverte de relations entre gènes et maladies rares. *Revue d'Intelligence Artificielle*, **28**(2), 245–270.

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.

CRAM D. (2010). Découverte interactive et complète de chroniques.

FRAWLEY W. J., PIATETSKY-SHAPIRO G. & MATHEUS C. J. (1992). Knowledge discovery in databases : An overview. *AI Magazine*, **13**(3), 57–70.

FUCHS B. (2017). Assister l'utilisateur à expliciter un modèle de trace avec l'analyse de concepts formels. In C. ROUSSEY, Ed., *Ingénierie des connaissances 2017*, Actes des 28èmes Journées francophones d'Ingénierie des connaissances - IC 2017, p. 151–162.

FUCHS B. & CORDIER A. (2016). Interprétation interactive de connaissances à partir de traces. In N. PERNELLE, Ed., *Ingénierie des connaissances 2016*, Actes des 27e Journées francophones d'Ingénierie des connaissances - IC 2016, p. 167–178.

GENG L. & HAMILTON H. J. (2007). Choosing the right lens : Finding what is interesting in data mining. In *Quality measures in data mining*, p. 3–24. Springer.

MANNILA H., TOIVONEN H. & INKERI VERKAMO A. (1997). Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, **1**(3), 259–289.

NANNI M. & RIGOTTI C. (2007). Extracting trees of quantitative serial episodes. In S. DŽEROSKI & J. STRUYF, Eds., *Knowledge Discovery in Inductive Databases : 5th International Workshop, KDID 2006 Berlin, Germany, September 18, 2006 Revised Selected and Invited Papers*, p. 170–188 : Springer Berlin Heidelberg.

PERER A. & WANG F. (2014). Frequence : Interactive mining and visualization of temporal frequent event sequences. In *Proceedings of the 19th International Conference on Intelligent User Interfaces*, IUI '14, p. 153–162 : ACM.

SANCHEZ E., EMIN-MARTINEZ V. & MANDRAN N. (2015). Jeu-game, jeu-play, vers une modélisation du jeu. une étude empirique à partir des traces numériques d'interaction du jeu tamagocours. *Revue STICEF*, **22**.

VAN LEEUWEN M. (2014). Interactive data exploration using pattern mining. In *Interactive Knowledge Discovery and Data Mining in Biomedical Informatics*, p. 169–182. Springer.