

Trèfle : un modèle de traces d'utilisation

Elöd Egyed-Zsigmond*, Alain Mille*, Yannick Prié*, Jean-Marie Pinon*

Laboratoire d'Ingénierie des Systèmes d'Information, Bâtiment Blaise Pascal (501), 20 avenue Albert Einstein
69621 VILLEURBANNE CEDEX, <http://lisi.insa-lyon.fr> (LIRIS, from 01.01.2003)
elod.egyed-zsigmond@insa-lyon.fr, amille@lisi.univ-lyon1.fr,
yprie@lisi.univ-lyon1.fr, Jean-Marie.Pinon@lisi.insa-lyon.fr

Résumé

Dans cet article nous présentons un modèle de représentation de traces d'utilisation. Nous parlons de l'idée que lors de l'utilisation d'une application informatique des utilisateurs manipulent des objets à l'aide de procédés. Nous proposons donc de représenter l'évolution d'un système à l'aide d'un graphe qui contient des représentations des utilisateurs, procédés et objets manipulés, en les interconnectant. Ce graphe évolue au fur et à mesure de l'utilisation de l'application fournissant ainsi une base toujours plus riche de capital d'expérience que l'on peut réutiliser.

Mots clés : modèle de traces d'utilisation, capitalisation de l'expérience, graphe.

1 Introduction

L'ordinateur et les applications informatiques sont des outils de plus en plus utilisés dans la vie quotidienne. Ils sont exploités lors de la réalisation de tâches toujours plus complexes. Ils permettent à des personnes non initiées d'effectuer des opérations et résoudre des problèmes qui auparavant relevaient de la compétence des spécialistes. Néanmoins, ces applications sont utilisées par des utilisateurs ayant un degré d'expertise varié. Il est important donc que les interfaces et les applications sachent s'adapter, pour pouvoir être utilisées à leur meilleur rendement. Comme de plus, les ordinateurs « n'oublient » pas, il serait intéressant de garder les traces convenablement modélisées des sessions d'utilisation afin de pouvoir les réutiliser pour aider la réalisation de nouvelles tâches.

Dans ce papier nous présentons un modèle qui permet la mémorisation des traces d'utilisation pour capitaliser l'expérience des utilisateurs et la réutiliser pour fournir de l'aide dans des situations non triviales, nécessitant de la créativité. Notre modèle a pour vocation d'utiliser un paradigme tel que celui du raisonnement à partir de cas (RàpC) afin d'effectuer la réutilisation de l'expérience.

Nous considérons que dans une application informatique des utilisateurs manipulent des objets à l'aide des procédés. A l'aide de notre modèle nous mémorisons les traces de manipulation d'objets en représentant les

utilisateurs, les procédés et les objets, connectés dans un graphe.

Nous présentons d'abord le contexte de notre étude, suivi d'un passage en revue de quelques systèmes d'aide existants. Nous décrivons ensuite notre modèle de traces d'utilisation de manière théorique d'abord, puis à travers une application concrète. Pour conclure nous discutons les directions possibles à prendre pour développer le modèle.

2 Contexte et Problématique

Nos travaux de recherche se déroulent dans le cadre du projet RECIS soutenu par le Réseau National de Recherche en Télécommunications. Le projet se fait en commun avec France Télécom Recherche et Développement, le Laboratoire d'Ingénierie des Systèmes d'Information (LISI) et le projet IMEDIA de l'INRIA.

L'objectif de ce projet est d'étudier et de développer des outils permettant d'enrichir les services de recherche et d'accès au contenu multimédia.

Il s'agit d'élaborer un système qui permette à l'utilisateur d'exprimer facilement et précisément une requête, un système qui trouve rapidement des résultats pertinents, puis propose des moyens pour raffiner la requête et qui mette à disposition de l'utilisateur plusieurs méthodes de visualisation des résultats.

Les travaux de notre équipe portent sur l'annotation de documents et sur l'étude de l'aide à l'utilisateur dans la mise en place et l'exploitation de ces annotations.

Nous avons en effet identifié trois tâches principales qui se retrouvent dans la plupart des tâches d'un utilisateur : l'annotation d'un fragment de document, la recherche de fragments de document et la navigation entre fragments de documents. Nous chercherons donc à mettre en place une modélisation des connaissances, un modèle de l'expérience. Modèle qui permettra d'assister ces trois tâches génériques dans le contexte des tâches réelles de l'utilisateur.

Dans la suite de cette section, nous présentons le modèle théorique (Modèle Étendu des Strates Interconnectées par les Annotations (E-SIA) [25; 12; 1]) qui nous permet de représenter et d'utiliser les annotations.

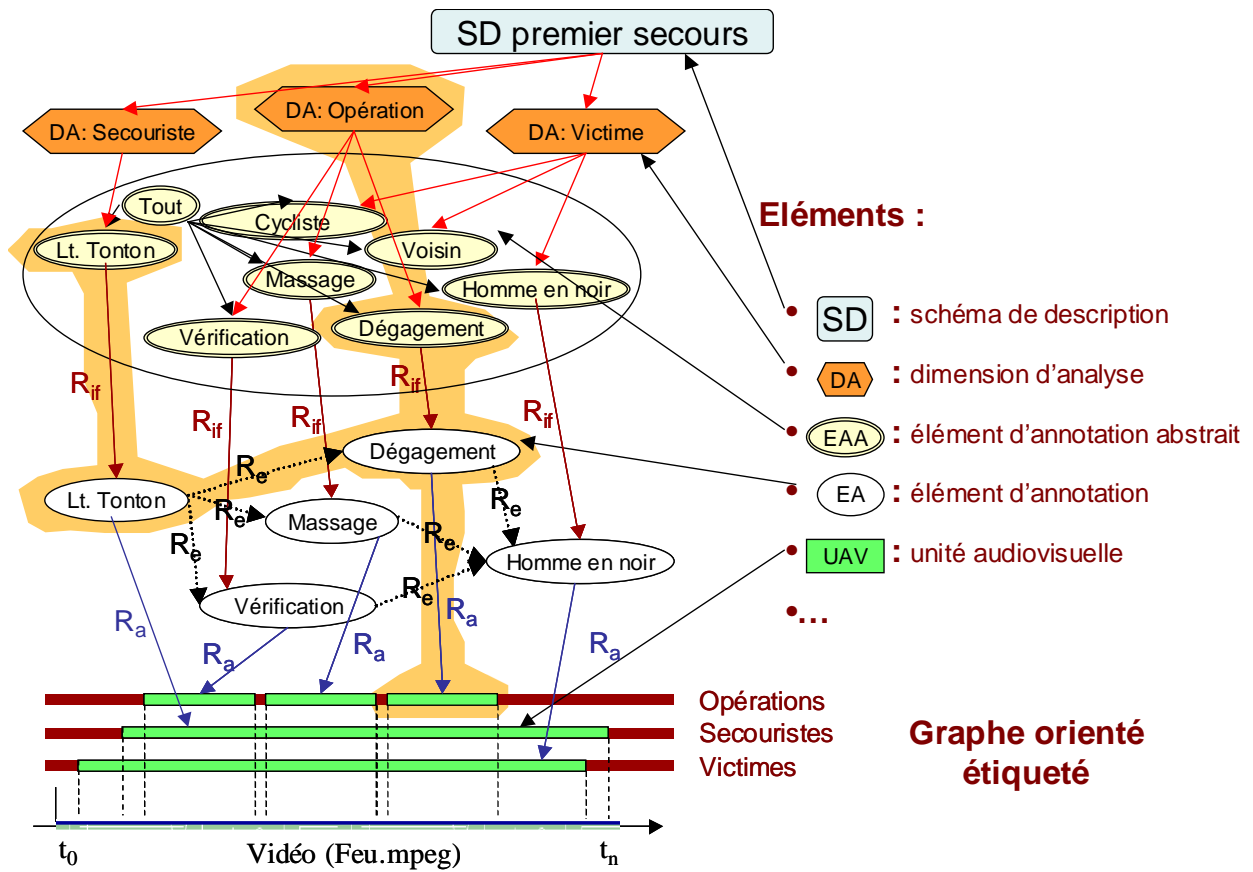


FIG. 1: Exemple de graphe des différentes « couches » de l'annotation dans E-SIA (document vidéo sur le secourisme)

Dans E-SIA (FIG. 1), les documents sont décrits par des *éléments d'annotation* (EA). Ces éléments d'annotations (EA) sont tirés d'un vocabulaire utilisable pour l'annotation. Les termes, *éléments d'annotation abstraits* (EAA) constituant ce vocabulaire peuvent être regroupés dans des *dimensions d'analyse* (DA) formant ainsi des ensembles utiles dans le cadre d'une tâche d'annotation donnée. Les éléments d'annotation (EA) annotent des fragments de documents matérialisés dans le graphe par des *unités audiovisuelles* (UAV). Les éléments d'annotation peuvent être contextualisés en les mettant en relation (au sein même de l'annotation) à l'aide d'une *relation élémentaire* (Re). L'ensemble des éléments forme un graphe connexe orienté étiqueté.

Une tâche d'annotation est partiellement décrite à l'aide d'une structure appelée *schéma de description* (SD). Cette structure permet d'imposer un cadre plus ou moins contraignant à l'annotation en précisant quelles sont les dimensions d'analyse à utiliser, ainsi que les relations à mettre en place entre les descripteurs posés en instanciant des termes (EAA) de ces dimensions d'analyse. Nous parlons d'annotation « selon un schéma de description ». Les schémas de description permettent de documenter et de guider la façon de décrire un document.

En pratique lors d'une annotation, l'utilisateur choisit un schéma de description, repère dans le document à annoter les fragments intéressants et les annote en ins-

tanciant les termes (EAA-s) contenus dans les dimensions d'analyse composant le schéma de description. Dans notre exemple, le SD « premier secours » a été utilisé pour annoter.

Pour exploiter les annotations nous utilisons des Graphes Potentiels (GP), constituant l'expression des requêtes dans E-SIA. Le graphe potentiel est décrit par des nœuds typés (SD, DA, EAA, EA, ...) qui correspondent à des nœuds du graphe global. Une partie de ces sommets peuvent être rendus génériques en laissant certains attributs non renseignés; ils peuvent alors correspondre à plusieurs nœuds du graphe global. Pour traduire la requête « trouver les séquences vidéo où Lt. Tonton effectue une opération » nous construisons le graphe potentiel de la FIG. 2. Les nœuds de départ sont le terme EAA « Lt.Tonton » et la dimension d'analyse DA « Opération ». Le nœud représentant l'unité audiovisuelle cherchée est N1. Ce nœud est un *nœud caractérisé*, représentant l'objet de notre requête.

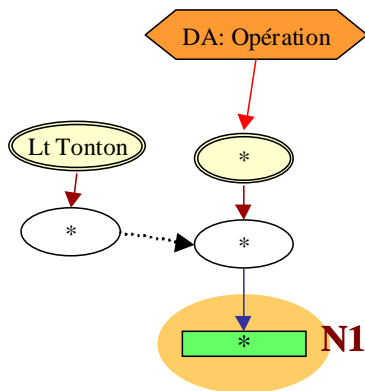


FIG. 2 Graphe potentiel représentant la requête : « Les séquences où Lt.Tonton manipule la victime »

Une fois la requête traduite en graphe potentiel, nous recherchons les sous-graphes dans le graphe global qui lui sont isomorphes. L'instanciation du GP dans le graphe global utilise un algorithme de multipropagation [26]. Un résultat est illustré sur la FIG. 1 (le sous graphe marqué par la forme foncée). Ici l'UAV cherchée, correspondant au nœud N1 du graphe potentiel est celle annotée par l'EA « Dégagement ». Il y a deux autres fragments correspondant à la requête : l'UAV annotée par l'élément d'annotation EA « Vérification » et celle annotée par l'EA « Massage ».

3 Systèmes d'aide

Dans cette partie nous passons en revue quelques catégories et systèmes d'assistance à l'utilisateur dans sa tâche d'utilisation d'une application. Si l'aide à la manipulation de documents multimédias, telle que nous l'avons présentée, n'a pas encore vraiment été abordée, nous nous appuyons sur des travaux existants pour proposer à la fin de cette section quelques directions à suivre pour la mise en place des systèmes d'aide.

Aujourd'hui nous sommes loin des aides de quelques lignes, expliquant les paramètres à mettre après une instruction dans une ligne de commande. Ces systèmes d'aide, si nous pouvons les appeler ainsi, étaient assez bien adaptés et efficaces pour ces tâches très simples. Les possibilités des systèmes, aussi bien que celles des utilisateurs ayant évolué et pouvant être parfois extrêmement complexes, l'aide à l'utilisateur doit donc devenir de son côté plus performante.

Les systèmes d'aide à l'utilisateur peuvent se caractériser selon de nombreux critères.

Selker [30] propose de séparer les systèmes d'aide en systèmes de type *conseiller* et systèmes de type *assistant*.

Un système d'aide de type conseiller, donne des informations, propose des solutions mais n'intervient pas directement dans le travail de l'utilisateur. La plupart des systèmes d'aide des logiciels sous Windows sont de ce type. Nous pouvons les consulter, ils nous conseillent parfois des solutions, comme le compagnon MS Office, sans modifier les documents sur lesquels nous travaillons.

Les systèmes de type assistant font à notre place les tâches répétitives. C'est le cas typique des macros ou des aides fondées sur la prédiction des actions (compléter automatiquement des mots clés, adresses, listes de paramètres, etc.).

Selker [30] et Lieberman [19] distinguent également des systèmes d'aide *conversants* et *autonomes*. Les systèmes d'aide conversants attendent nos questions, c'est le cas des aides traditionnelles. Nous les parcourons chapitre par chapitre, lançons des recherches par mots clés ou bien posons des questions. Les systèmes autonomes fonctionnent en parallèle avec l'utilisateur et lui proposent de l'aide sans qu'il ait été fait appel à eux.

Un autre critère de classification des systèmes d'aide est leur capacité à *s'enrichir* [20; 31] Certains systèmes d'aide *évoluent* en effet au cours de leur utilisation. D'autres [14] s'adaptent certes aux différentes situations par exemple en présentant l'aide selon différentes vues suivant la situation, mais leur savoir ne s'enrichit pas.

L'enrichissement peut s'effectuer de manière *manuelle* ou *automatique*. En effet certains environnements suivent les actions de l'utilisateur et les réponses du système pour apprendre les habitudes de l'utilisateur et adapter l'aide proposée [30; 18; 15; 13]. D'autres systèmes permettent à l'utilisateur d'intervenir dans son processus d'apprentissage et de le guider [17] en lui précisant ses préférences. Une autre méthode d'enrichissement qui peut être manuelle ou semi-automatique est la possibilité de créer des macros [20; 21; 28]. Des macros peuvent être créées en programmant, en réutilisant l'historique [21; 8] ou à l'aide d'enregistreurs de macros [17; 24; 28].

L'« apprentissage » de l'assistance peut exploiter des profils d'utilisateur [31], des profils de situations (métaphore des salles) [24], combiner les deux, accumuler l'expérience d'un seul [18] ou plusieurs [15] utilisateurs. Alors l'assistance peut être adaptée selon le type d'utilisateur et de situation. Les systèmes peuvent employer différentes techniques de fouille de données pour calculer leurs recommandations [31] : statistiques, raisonnement à partir de cas ou catégorisation des données.

La tâche d'annotation et de manipulation de documents multimédias étant très complexe et subjective, les critères de description et de recherches étant souvent flous, très variables et difficiles à formuler, l'aide doit pouvoir s'adapter le plus possible à la situation et à l'utilisateur.

En étudiant les systèmes d'aide existants nous avons synthétisé quelques directives émergentes :

- Pour pouvoir s'adapter, il faut modéliser les tâches, les utilisateurs, les contextes (situations) et les applications de telle façon qu'ils puissent être manipulés aussi bien par le système que par les utilisateurs.
- Concevoir des modèles de tâche qui permettent de faire ressortir de nouvelles façons d'utiliser le système.
- Si le système garde l'historique des opérations, structurer celle-ci de façon arborescente voire en graphe et prévoir d'utiliser cet historique pour générer des macros, en rendant générique (quand cela est possible) les suites d'opérations.

- Dans la plupart des cas combiner des systèmes d'aide de type assistant et de type conseiller, qui agissent néanmoins sans interrompre l'utilisateur dans son travail.
- Quand c'est possible profiter des pauses de l'utilisateur pour faire travailler des assistants, donc créer des systèmes autonomes.
- Concevoir des systèmes d'aide qui apprennent vite et donnent des réponses rapides et intelligibles.
- S'il s'agit d'un système qui apprend, envisager l'apprentissage automatique combiné avec les instructions et les interventions de l'utilisateur.
- Pour permettre à l'utilisateur d'intervenir dans le processus d'apprentissage, représenter celui-ci sous une forme lisible par l'homme.
- Adapter le média, le type (exemple, explication, etc.) et la quantité de l'information mis à la disposition de l'utilisateur.

Dans notre modèle nous avons essayé de prendre en compte ces directives.

4 Le modèle Trèfle

4.1 Présentation générale

L'ingénierie des connaissances étudie depuis longtemps le problème de la capitalisation et le réutilisation de l'expérience d'utilisation d'un système informatique. Plusieurs modèles de représentation de connaissances ont été mis au point [3; 16; 5; 2; 4; 7; 10; 13; 23], traitant de la formalisation des tâches, profils de l'utilisateur, historiques de manipulations d'objets mais pas des trois en même temps.

Nous partons de l'idée que dans une application informatique, ce qui fait sens, c'est ce qui est interconnecté du point de vue de l'utilisateur lors de l'utilisation des objets disponibles. Au départ cette interconnexion correspond à (forme signature de) une intention (objectif à atteindre), tandis qu'au fur et mesure des actions lors de l'utilisation de l'application elle se concrétise en un résultat (objectif atteint). L'utilisateur rend indirectement compte de ses différentes tâches en manipulant des objets à l'aide d'outils. Nous considérons que de manière générale l'utilisateur, lors de l'utilisation d'une application informatique, manipule des objets à travers des procédés (FIG. 3).

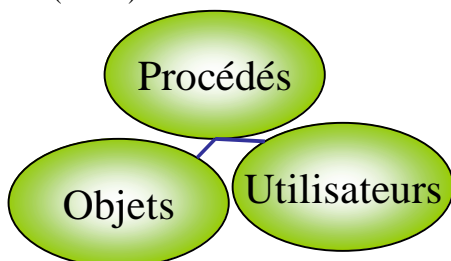


FIG. 3 Schéma général du modèle Trèfle

L'ensemble des objets qui peuvent être manipulés forme le *modèle d'utilisation*. Pour une application de

traitement de texte, le modèle d'utilisation est par exemple composé de caractères, sections, chapitres, figures, tableaux, documents... Pour notre application de manipulation de documents audiovisuels, les documents, les annotations (EA-s), les différents éléments de structurant le vocabulaire (EAA, DA, SD), forment le modèle d'utilisation. Lors de l'utilisation de l'application les objets du modèle d'utilisation s'instancient, les instances reflétant les manipulations subies par les objets à travers les actions de l'utilisateur. Nous considérons qu'un utilisateur peut créer, modifier, et visualiser un objet grâce à des procédés.

L'ensemble de contraintes de manipulation imposé par l'interface d'une application forme le *modèle de tâches*. Ce modèle de tâche peut se représenter à l'aide de procédés. Les procédés sont typiquement les actions accédées par des menus/ sous-menus, boutons ou icônes de l'interface graphique, ou simplement en tapant des instructions. Les procédés d'une application sont rarement entièrement indépendants. Par exemple dans RECIS, l'annotation d'un document suppose le choix d'un schéma de description, le choix d'une dimension d'analyse et le choix d'un terme dans cette dimension d'analyse. La création d'un document dans un traitement de texte passe par le choix d'un modèle et une suite de créations de paragraphes conformes au modèle. Nous considérons : annotation d'un document, choix de schéma de description, création d'un document texte, choix de modèle, etc. des procédés.

Pour adapter le modèle trèfle à une application donnée nous définissons tout d'abord l'ensemble d'objets dont nous désirons tracer la manipulation (tiré du modèle d'utilisation), ensuite l'ensemble de procédés qui permettent d'effectuer ces manipulations (tiré du modèle de tâches) et nous devons décider si l'application est mono ou multi-utilisateur. L'ensemble de procédés dont on a décidé le traçage avec la représentation des utilisateurs constitue le *modèle d'observation* : en effet, il s'agit d'indiquer quel point de vue on souhaite avoir sur l'utilisation de l'application informatique pour en former un modèle explicatif ciblé. Nous pouvons représenter les différents liens entre procédés par des relations d'explication dans le graphe, leur sémantique dépendant de l'application. Les relations d'explication traduisent certaines des relations du modèle de tâches. Ils permettent d'exprimer le fait qu'un procédé soit lié d'une manière ou d'une autre à un autre procédé. Par exemple il est lancé parce qu'un autre a déjà été exécuté. Nous pouvons nous servir des prototypes d'applications existants [9; 6; 29; 27] pour formaliser les modèles de tâches et choisir les parties à tracer.

Le modèle d'observation pour une application de traitement de texte peut être constitué par exemple des procédés de création d'un document, choix d'un modèle, et création de paragraphes. La création du document explique le choix d'un modèle ainsi que la création des paragraphes. A titre d'illustration, nous pouvons considérer que pour créer un document « Word » il faut choisir un modèle et créer les paragraphes en conformité avec ce modèle.

4.2 Le graphe

Nous choisissons d'inclure dans le modèle Trèfle les éléments nécessaires afin de tracer l'usage d'une application : utilisateurs, procédés et objets. Nous avons choisi de représenter ces éléments sous la forme d'un graphe. Nous pouvons représenter ainsi les différents liens qui relient les utilisateurs, procédés et objets, nous avons également les éléments du modèle dans une structure homogène et fortement interconnectée. Dans le cadre de nos travaux précédents [25; 12; 11] nous avons déjà représenté les annotations des documents sous la forme de graphe, nous avons également mis au point des outils d'exploitation que nous pouvons ainsi réutiliser. Nous représentons donc le modèle d'utilisation, le modèle d'observation et les traces observées sous la forme d'un graphe dont les nœuds sont des objets, procédés ou utilisateurs (FIG. 4).

Concrètement : une instance d'un modèle M trèfle est composé

- du graphe G contenant les objets du modèle d'utilisation, les éléments du modèle d'observation ainsi que la trace,
- de l'ensemble des graphes potentiels $\{GP\}$ permettant d'exploiter ce graphe.

Plus particulièrement :

Le **graphe global** $G = \langle N, R \rangle$ est le graphe représentant l'ensemble des éléments N du modèle, mis en relation par l'ensemble de relations R . $N = (U \cup O \cup P)$, un nœud peut donc être soit un utilisateur $u \in U$, soit un objet du modèle d'utilisation $o \in O$, ou bien un procédé $p \in P$. Formellement les arêtes du modèle ne portent pas de sémantique, leur signification est déterminée par les nœuds qu'elles relient.

Un **nœud** a des attributs, et chaque attribut a une ou plusieurs valeurs. Chaque nœud a un identificateur unique dans le graphe global G . Les nœuds sont abstraits ou concrets. Un nœud abstrait est défini lors de la mise en place initiale du modèle, alors que les nœuds concrets forment la trace d'utilisation et sont créés lors de l'utilisation de l'application. Les nœuds concrets sont situés dans le temps. Les nœuds concrets seront notés : $[\text{nom}, t]$ où nom est le nom du nœud abstrait dont il est l'instance et t est le time-code de sa création. Les nœuds abstraits par les relations qui les unissent définissent des ensembles, alors que les nœuds concrets (objets et procédés) peuvent être ordonnés selon le temps, formant ainsi des listes, traces. Chaque nœud concret est l'instance d'un nœud abstrait. Dans le graphe cette relation se matérialise par des arêtes d'instanciation entre les nœuds abstraits et concrets.

Dans la suite nous introduisons les différents types de nœuds : objets, procédés et utilisateurs.

Les **objets** sont les nœuds représentant tout ce que l'utilisateur peut manipuler et manipule de façon consciente. Les objets du modèle Trèfle sont des données du modèle d'utilisation dont on a décidé de tracer la manipulation.

Un **objet abstrait** est un objet dont on a décidé de tracer la manipulation. A chaque objet abstrait correspond

au moins un procédé abstrait qui permette de l'instancier. Mais il peut être en relation avec plusieurs procédés abstraits qui permettent sa manipulation. Les objets abstraits peuvent être reliés à d'autres objets abstraits par des relations de composition. Par exemple un document est composé de d'un modèle et de paragraphes. On note un objet abstrait « type et nom objet ».

Un **objet concret** est une instance d'objet abstrait, il a un type (est relié à un seul objet abstrait), a été créé par un procédé concret, à l'initiative d'un utilisateur concret. Il peut être relié à d'autres procédés concrets qui le modifient, réutilisent ou le visualisent.

Toute utilisation de l'application informatique passe par des procédés. Un **procédé** est inclus dans le graphe parce que l'on choisit de tracer ses utilisations. L'utilisateur se sert des procédés pour manipuler les objets. Il est obligé de passer par des procédés. Un procédé peut être relié à un ou plusieurs objets par des relations d'utilisation, de création, visualisation. Chaque objet est relié à au moins un procédé.

Un **Procédé abstrait** est un type de procédé, et concerne des objets abstraits (on peut le considérer comme un opérateur, tandis que les objets abstraits sont les opérants). Les procédés abstraits seront notés : $\|\text{nom procédé}\|$. Les procédés abstraits peuvent être reliés entre-eux par des arêtes d'explication tirées du modèle de tâches.

Un **procédé concret** est une instance située dans le temps et reliée à un utilisateur d'un procédé abstrait. Il est également relié à ou moins un objet concret.

Un **utilisateur** est une représentation de l'utilisateur humain, qui utilise le système. Les utilisateurs sont en relation avec des procédés par des arêtes « est à l'origine de ». Particulièrement :

Un **utilisateur abstrait** est une personne, identifiée par un identificateur, ou un groupe, ayant accès à un certain nombre de procédés abstraits pour manipuler des objets. Un utilisateur abstrait est noté $\langle \text{nom utilisateur} \rangle$.

Un **utilisateur concret** est une personne identifiée, qui manipule des objets concrets à l'aide des procédés concrets. Un utilisateur peut appartenir à un groupe. Dans ce cas il n'a accès qu'aux procédés de son groupe.

Un **graphe potentiel** $GP = \langle N_{GP}, R \rangle$. est composé de nœuds de graphe potentiel $n \in N_{GP}$ qui peuvent être partiellement renseignés et ainsi s'instancier parmi les nœuds du graphe global G . L'instanciation se fait en comparant les attributs renseignés par des fonctions de comparaison. Les arêtes $r \in R$ sont de même type que les arêtes du graphe G . Les graphes potentiels sont utilisés dans le modèle Trèfle pour calculer des traces, découper et adapter des épisodes d'utilisation. Un graphe potentiel a des nœuds caractérisés. Ce sont des nœuds génériques dont on veut récupérer les correspondances dans le graphe global G , car ils constituent l'objet de la requête exprimé par le graphe potentiel. Ils ont un attribut « caractérisation ».

4.3 Traces

Les actions de l'utilisateur faisant partie du modèle d'observation sont tracées par la création de nouveaux

nœuds et arêtes dans le graphe. Ces nouveaux nœuds sont des nœuds concrets. Ils sont situés dans le temps et ensemble avec les arêtes qui les lient forment la *trace brute*. La *trace brute expliquée* est l'ensemble du graphe global G. En général une *trace* est un sous-graphe du graphe global G. Nous pouvons filtrer les traces, grâce à des graphes potentiels pour obtenir des *épisodes*. Un épisode est une liste ordonnée de nœuds concrets du graphe global G, construite par l'instanciation d'un graphe potentiel et organisé selon un critère d'ordre total (le temps par exemple). L'épisode contenant tous les procédés et objets concrets est la *trace linéaire brute*.

On ne trace pas la création et la modification des objets, procédés et utilisateurs abstraits, comme on ne trace pas la manipulation des procédés concrets non plus.

Le but du modèle est de fournir un cadre riche de traçage des actions de l'utilisateur pour capitaliser et réutiliser son expérience. Pour ce faire nous construisons à partir des traces des épisodes qui pourront devenir des cas réutilisables dans le cadre d'un système basé sur le raisonnement à partir de cas (RàpC)[22].

Sur la FIG. 4 nous présentons un exemple du graphe global G. Supposons que nous choisissons l'observation de la manipulation des objets « O_{abs1} », « O_{abs2} » et « O_{abs3} », qui font partie dans notre cas du modèle d'utilisation. Leur manipulation se fait à travers les procédés ||P_{abs1}||, ||P_{abs2}||, ||P_{abs3}|| et ||P_{abs4}||. Cette utilisation est observée pour un seul utilisateur : <U_{abs1}>.

L'ensemble des procédés observés ainsi que la représentation de l'utilisateur forme le modèle d'observation. Par exemple, dans une application de traitement de texte, les objets observés (*modèle d'utilisation*) peuvent être : paragraphe, section, document ; les procédés (*modèle d'observation*) par exemple : créer un nouveau document ou paragraphe, commencer une nouvelle section, modifier le style d'un paragraphe, déplacer une section.

Les traces de la FIG. 4 nous indiquent que l'utilisateur a effectué deux sessions (chaque session se traduit par la création d'un nœud d'utilisateur concret : les nœuds [U_{concr1},t0] et [U_{concr2},t5'] qui sont situés dans le temps). Lors de la première session l'utilisateur [U_{concr1},t0] a lancé le procédé ||P_{abs1}||, pour manipuler un objet « O_{abs1} ». Cette manipulation se traduit dans la trace par la création des nœuds [P_{concr1},t1] et [O_{concr1},t2]. [O_{concr1},t2] est l'instance de l'objet « O_{abs1} » manipulé (par exemple créé) par l'utilisateur [U_{concr1},t0] à travers le procédé [P_{concr1},t1]. [P_{concr1},t1] est situé également dans le temps et il est relié à l'instance de l'utilisateur qui est à son origine ainsi qu'à l'instance de l'objet [O_{concr1},t2] qu'il manipule.

L'utilisateur [U_{concr1},t0], toujours lors de sa première session a créé ensuite une instance de l'objet « O_{abs2} », par un instance du procédé ||P_{abs2}||, laissant la trace [P_{concr2},t3], [O_{concr2},t4] et les relations qui leur sont adjacentes.

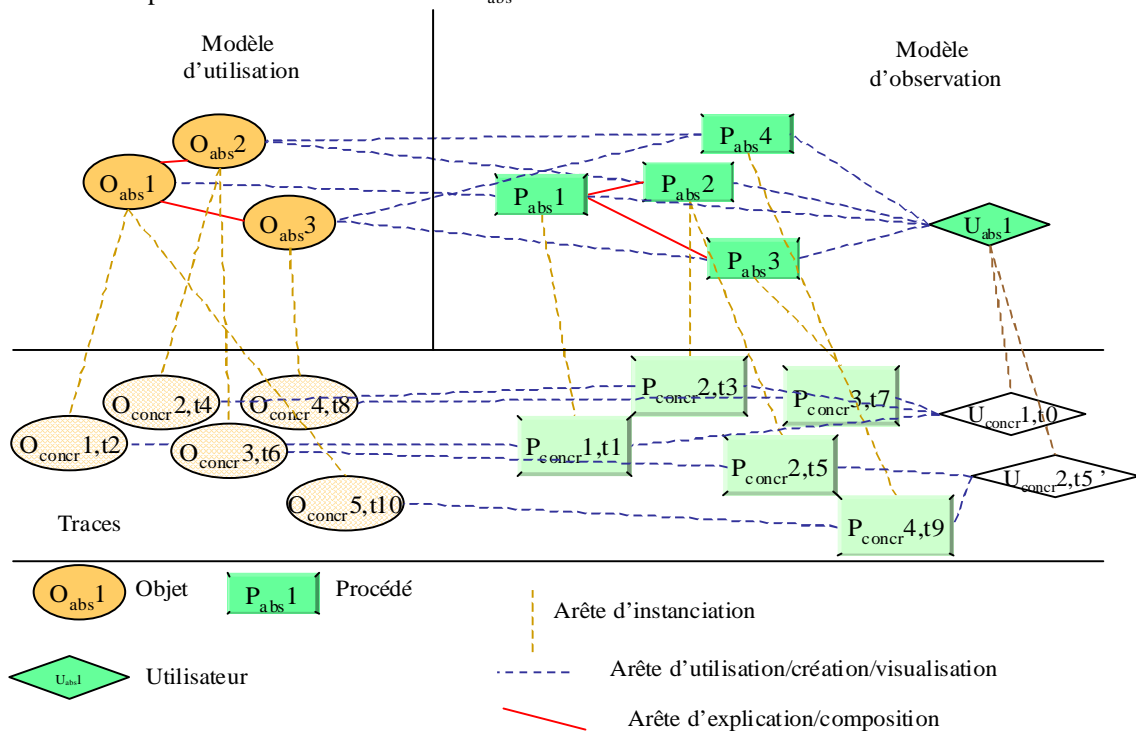


FIG. 4 Découpage du graphe en modèle d'utilisation, modèle d'observation et les traces résultant de l'observation de l'utilisation

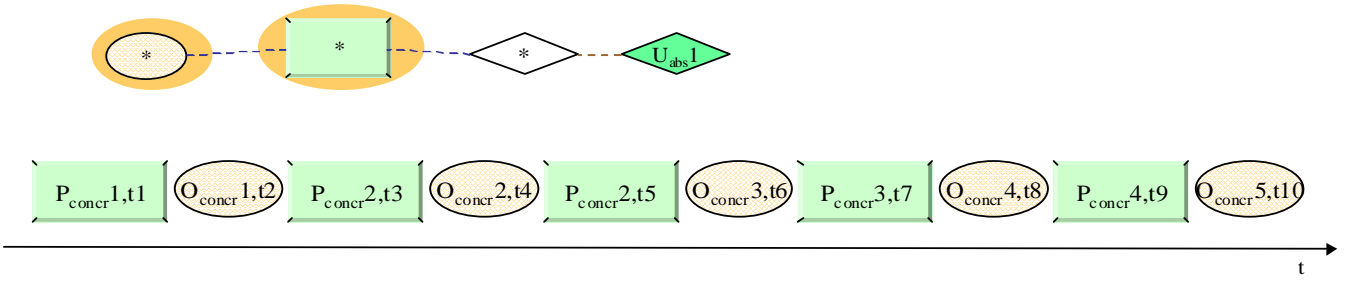


FIG. 5 La trace linéaire brute et le graphe potentiel qui a permis de la construire

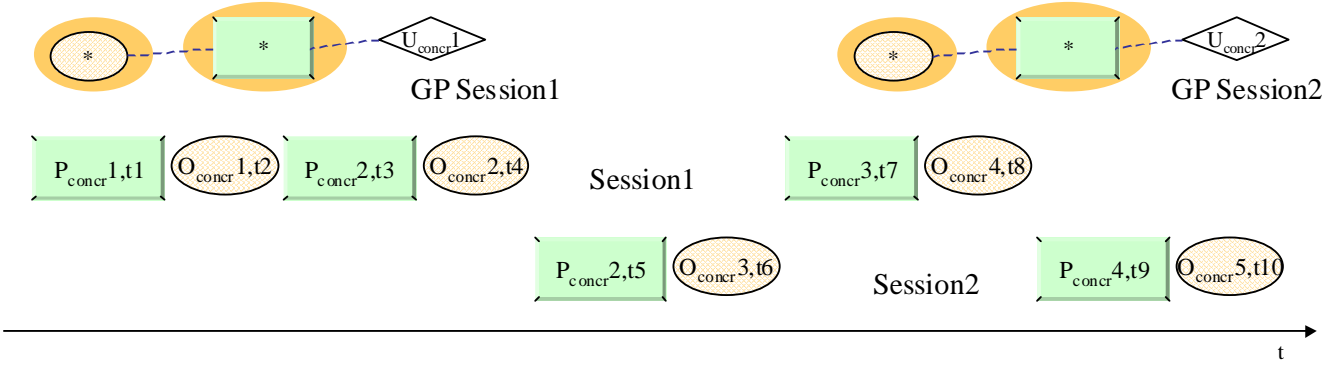


FIG. 6 Graphes potentiels qui permettent la construction d'épisodes représentant des sessions

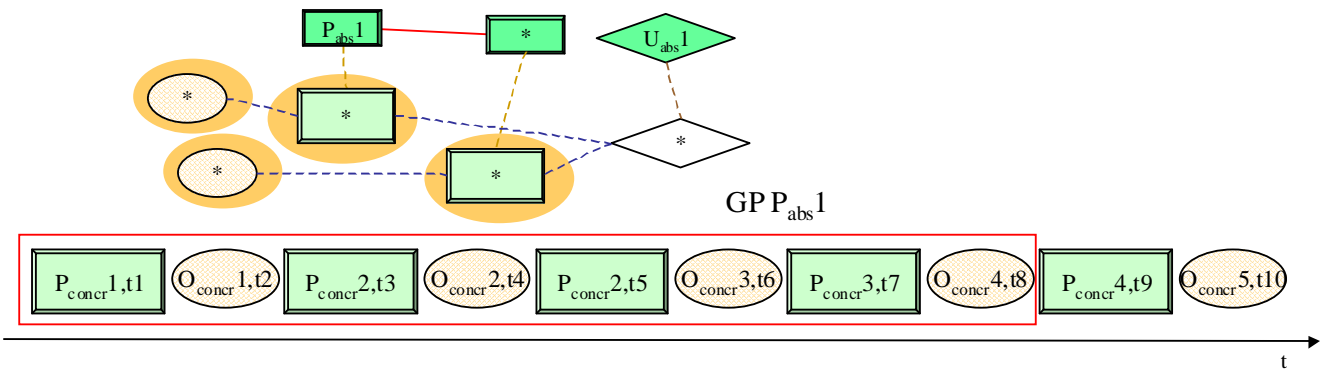


FIG. 7 Graphe potentiel qui permet la construction des épisodes selon le modèle de tâches : procédés et objets dans le cadre du procédé $P_{abs,1}$

Les épisodes se calculent à l'aide des graphes potentiels : il s'agit en effet de requêtes sur le graphe global G. Les épisodes sont le résultat du tri chronologique de listes de correspondances des graphes potentiels, ou des projections sur la ligne de temps de sous-graphes du graphe global G.

La trace linéaire brute est illustrée sur l'exemple de la FIG. 5. Cet exemple nous indique l'évolution du système dans le temps. Le premier objet manipulé est $[O_{concr,1,t2}]$, il est manipulé par le procédé $[P_{concr,1,t1}]$, ils forment les premiers éléments de la trace, qui s'arrête avec la manipulation de l'objet $[O_{concr,5,t10}]$.

Pour construire des épisodes pertinents nous faisons des graphes potentiels qui exploitent les relations qui lient les éléments du graphe. La trace linéaire brute doit être découpée en épisodes qui donneront des cas réutilisables au sens RàpC du terme [11]. Le découpage de la trace linéaire brute peut se baser sur le modèle

d'observation représenté par les relations d'explication entre les procédés abstraits, elle peut également se faire en se basant sur les relations qui lient entre-eux les objets abstraits, se servir des instances d'utilisateurs ou bien encore exploiter n'importe quelle autre relation du graphe global G. Pour construire les épisodes nous considérons non plus seulement la trace linéaire brute mais aussi différents contextes de chaque nœud qui la compose.

Les objets ainsi que les procédés du graphe peuvent avoir des relations entre-eux. Ces relations expriment des liens fonctionnels ou procéduraux spécifiques à l'application. Par exemple si nous avons les objets : document, paragraphe et titre, nous pouvons dire qu'un document est composé d'un titre et de paragraphes. Dans le cadre de notre application nous savons que l'annotation selon un schéma de description passe par le choix d'un terme dans une dimension d'analyse, nous pouvons

donc mettre en relation les représentants de ces objets dans le modèle d'utilisation.

Ces liens peuvent être matérialisés dans le graphe sous la forme de relations de composition et d'explication (FIG. 4). Ces relations peuvent servir à découper les traces en épisodes sémantiquement disjoints.

La trace linéaire brute de la FIG. 5 peut être découpée par exemple selon les sessions ou selon les procédés. La FIG. 6 présente un découpage selon les sessions. Le graphe potentiel permet de récupérer les procédés et objets manipulés lors d'une session. Comme il y avait deux sessions la trace se découpe en deux épisodes: l'un regroupant les procédés et les objets manipulés lors de la première session, l'autre ceux manipulés lors de la deuxième session.

La FIG. 7 présente un découpage de cette trace selon le modèle d'observation. Le modèle de tâche indique que le procédé $\|P_{abs,1}\|$ explique les procédés $\|P_{abs,2}\|$ et $\|P_{abs,3}\|$. Cela peut par exemple représenter le fait que le procédé $\|P_{abs,1}\|$ impose l'exécution des procédés $\|P_{abs,2}\|$ et $\|P_{abs,3}\|$. Pour la construction des épisodes basés sur le modèle de tâche nous avons donc mis en place un graphe potentiel qui exploite les relations d'explication adjacentes au procédé abstrait $\|P_{abs,1}\|$. Dans ce cas l'épisode qui commence par l'instanciation du procédé $\|P_{abs,1}\|$, se termine avec la dernière instance d'objet qui a été manipulé par un procédé expliqué par $\|P_{abs,1}\|$, dans notre cas $[O_{concr,4,t8}]$, manipulé par $[P_{concr,3,t7}]$, instance de $\|P_{abs,3}\|$.

Avec les graphes potentiels nous pouvons calculer plusieurs types de traces et en se basant sur des relations de composition et d'explication nous pouvons calculer, exploiter une même trace avec des granularités différentes. Nous pouvons choisir d'inclure dans l'épisode seulement la création d'objets complexes, sans nous occuper des détails, ou bien au contraire nous pouvons inclure les procédés et objets les plus fins de notre modèle d'utilisation et modèle d'observation.

Nous avons des épisodes, qui peuvent alimenter une base de cas. Ces cas sont ensuite adaptables en utilisant par exemple à nouveau les relations de composition ou d'explication, pour être réutilisés selon le cycle du raisonnement à partir des cas. Nous pouvons choisir de remplacer des objets avec d'autres objets faisant partie du même objet composé, ou bien des procédés avec d'autres qui ont une explication similaire.

Pour l'adaptation on peut remonter à des niveaux de granularité plus élevés pour proposer des alternatives à une solution source.

5 Application à RECIS.

Dans cette section nous présentons un cas concret simplifié de mise en place du modèle trèfle, depuis l'adaptation du modèle à cette application particulière jusqu'aux étapes de réutilisation de l'expérience. L'exemple s'applique à notre étude d'annotation de vidéos. L'aide du système consiste à amener les utilisateurs à annoter les vidéos d'une manière cohérente, et leur faciliter le choix des annotations à poser.

5.1 Modèle d'utilisation

Pour notre exemple nous choisissons le traçage de l'annotation. Le modèle d'utilisation est composé des objets suivants (FIG. 8): un schéma de description SD« premier Secours », ainsi qu'un SD« général » (SDG) selon lesquels les utilisateurs peuvent annoter la vidéo, des dimensions d'analyse : DA« Victime », DA« Opération » et DA« Secouriste » dans lesquels les termes doivent être choisis, complétés par une DA« générale » (DAG) contenant tous les termes, ainsi que les termes eux-mêmes et la relation élémentaire qui permet de relier des instances de termes (éléments d'annotation). En effet il existe toujours un schéma de description général qui ne contraint pas l'annotation (mise en place libre de terme et de relations), de la même façon il existe toujours une dimension d'analyse générale regroupant tous les termes. Les schémas de description sont reliés aux dimensions d'analyse par des relations de composition tout comme les dimensions d'analyse aux termes. Dans la pratique ce modèle d'utilisation décrit une application qui permet d'annoter soit selon un schéma prédéfini SD(« premier secours ») qui impose la mise en place d'un secouriste, d'une victime et une opération, soit une annotation sans contraintes qui suit le SD« général ».

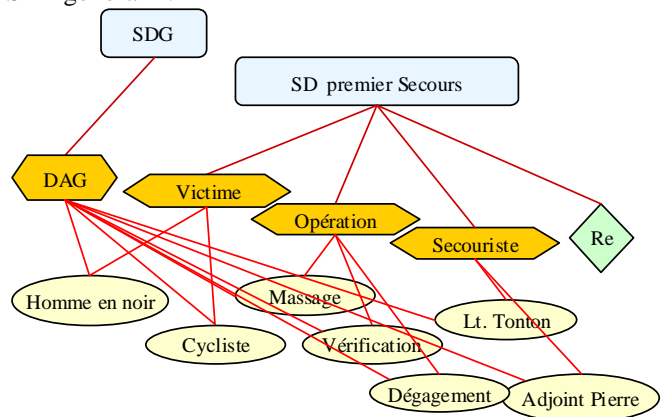


FIG. 8 Modèle d'utilisation

5.2 Modèle d'observation

Le modèle d'observation contient les utilisateurs, ainsi que les procédés qui permettent de créer les objets. (FIG. 9). Nous traçons ainsi l'annotation depuis le choix du schéma de description (dans notre cas, le SD« premier secours », ou le SD« général »), en passant par le choix de la dimension d'analyse, la mise en place ou réutilisation des éléments d'annotation et leur mise en relation. Le modèle d'observation indique que le procédé de choix de SD explique les procédés de choix de dimension d'analyse et le procédé de mise en relation. Dans la pratique dans l'interface de l'application le choix d'un SD pour annoter implique le choix des termes (EAA) dans des dimensions d'analyse, et leur mise en relation. Dans l'application elle-même un utilisateur a le choix d'annoter une vidéo, soit selon le SD« général », soit selon le SD« premier secours ».

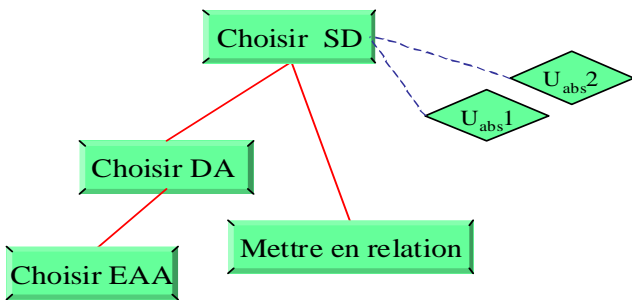


FIG. 9 Modèle d'observation

Ces procédés du modèle d'observation sont reliés par des relations de création/utilisation aux objets du modèle d'utilisation (FIG. 10).

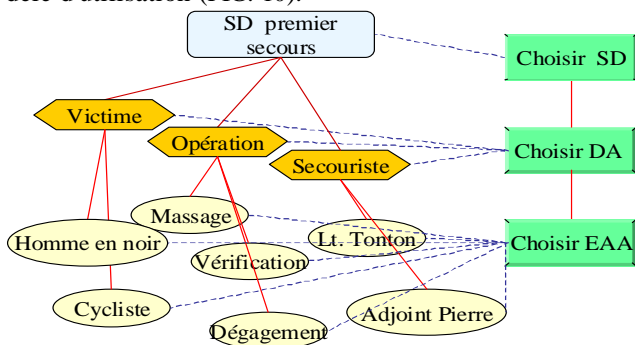


FIG. 10 Fragment du graphe global G représentant des relations entre le modèle d'utilisation et le modèle d'observation

Le procédé $\| \text{Choisir SD} \|$ est en relation de création avec les SD « général » et SD « premier secours », le procédé $\| \text{Choisir DA} \|$ est en relation de création avec les dimensions d'analyse, les procédés $\| \text{Choisir EAA} \|$ et $\| \text{Mettre en relation} \|$ sont reliés à tous les termes. Le procédé $\| \text{Mettre en relation} \|$ est relié de plus à l'objet Re

et lors de son instanciation il sera relié à trois objets concrets : l'instance de Re et les instances des deux termes que l'on met en relation avec Re. Une partie de ces relations est représentée sur la FIG. 10.

Le modèle d'utilisation et le modèle d'observation ainsi définis, nous pouvons étudier le fonctionnement du système. Dans les sections suivantes nous allons dérouler une première annotation selon le SD « premier secours », suivi d'une deuxième session, d'un autre utilisateur qui annote selon le SD « général », et qui sera dirigé par le système vers le SD « premier secours » après avoir posé ses premières annotations.

5.3 Scénario et construction des épisodes

Nous allons suivre l'évolution du système du point de vue du modèle trèfle pendant la mise en place des annotations en indiquant les méthodes de construction de traces ainsi que leur découpage et leur réutilisation pour proposer de l'aide à l'utilisateur.

La FIG. 11 présente les trois premières étapes de la première session. L'utilisateur 1 se connecte, créant ainsi le nœud $[U_{\text{concr}}1, t_0]$. Il choisit le SD « premier secours » laissant la trace formée par les nœuds procédé concret $[\text{ChoisirSD}, t_1]$ et l'objet concret $[\text{SD premier secours}, t_2]$, reliés entre-eux. Ensuite l'utilisateur choisit la dimension d'analyse DA « Victime » et l'EAA « Homme en noir », laissant les traces $[\text{ChoisirDA}, t_3][\text{DA : Victime}, t_4]$ et $[\text{ChoisirEAA}, t_5][\text{Homme en noir}, t_6]$. La FIG. 12 présente la trace linéaire brute des trois premières étapes de la première section.

Après cela il annote en choisissant un terme de la DA « opération » : $[\text{Vérification}, t_7]$ et le met en relation avec le terme $[\text{Homme en noir}, t_6]$ (FIG. 13).

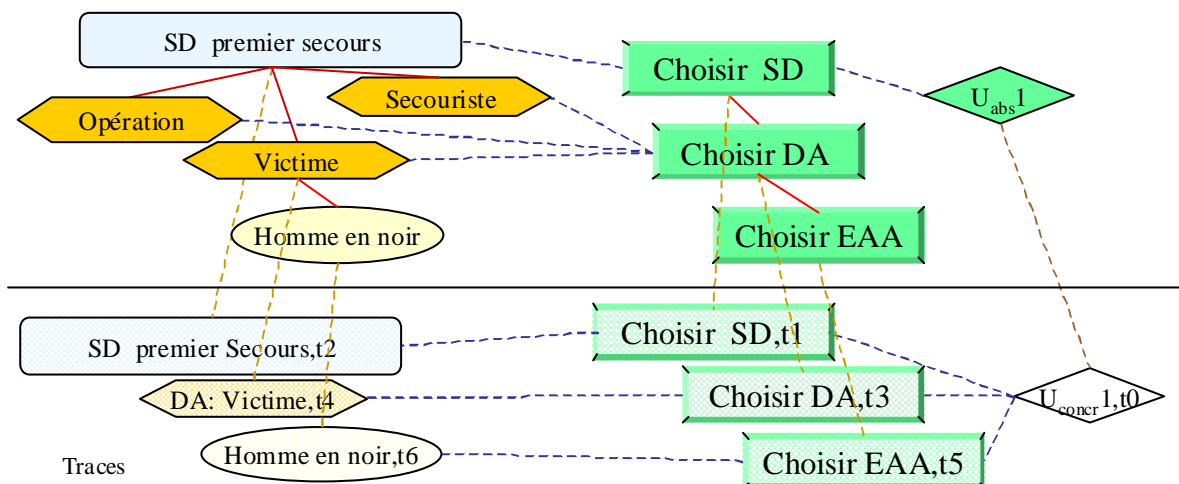


FIG. 11 Fragment du graphe global G représentant les traces laissées par premières pas de l'annotation



FIG. 12 Trace linéaire brute des premières pas de l'annotation

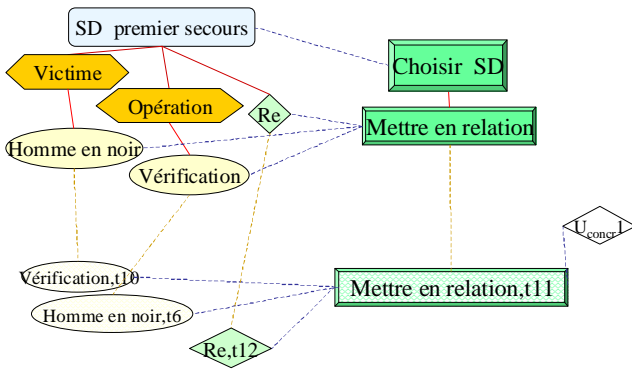


FIG. 13 Traces laissées par la mise en relation

Finalement il précise un secouriste [Lt Tonton,t12] et le met en relation avec l'opération [Vérification,t7]. Chaque fois laissant des traces comme pour la mise en place de l'EA [Homme en noir,t6](FIG. 13).

Pour récupérer des épisodes représentant seuls les procédés employés ou seuls les objets manipulés, nous construisons les graphes potentiels de la FIG. 14. Nous

les instancions, nous trions les résultats en ordre chronologique et nous obtenons les épisodes de la FIG. 15

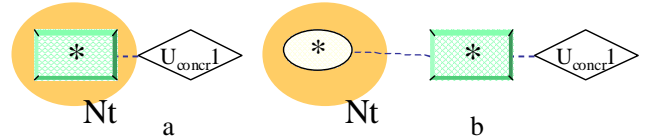


FIG. 14 Graphes potentiels pour le calcul des traces linéaires

En effet le graphe potentiel FIG. 14 (a), permet de récupérer les procédés concrets effectués par un utilisateur donné. Le nœud caractérisé (Nt :Nœud de trace) du graphe potentiel désigne ces procédés. En triant ces procédés concrets pour notre utilisateur en ordre chronologique nous obtenons l'épisode de la FIG. 15 (a). En instanciant le graphe potentiel FIG. 14 (b) pour avoir les objets manipulés par l'utilisateur [U_concr,1], nous aurons, après un tri l'épisode de la FIG. 15 (b).

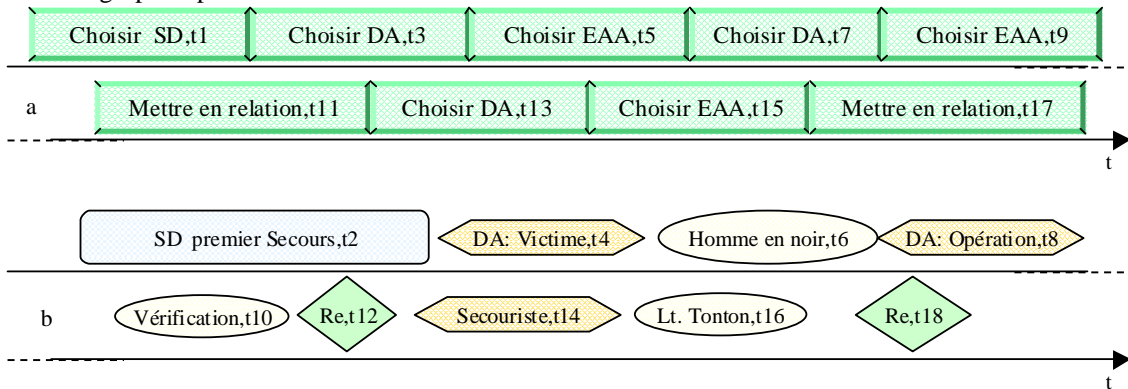


FIG. 15 Traces linéaires calculées

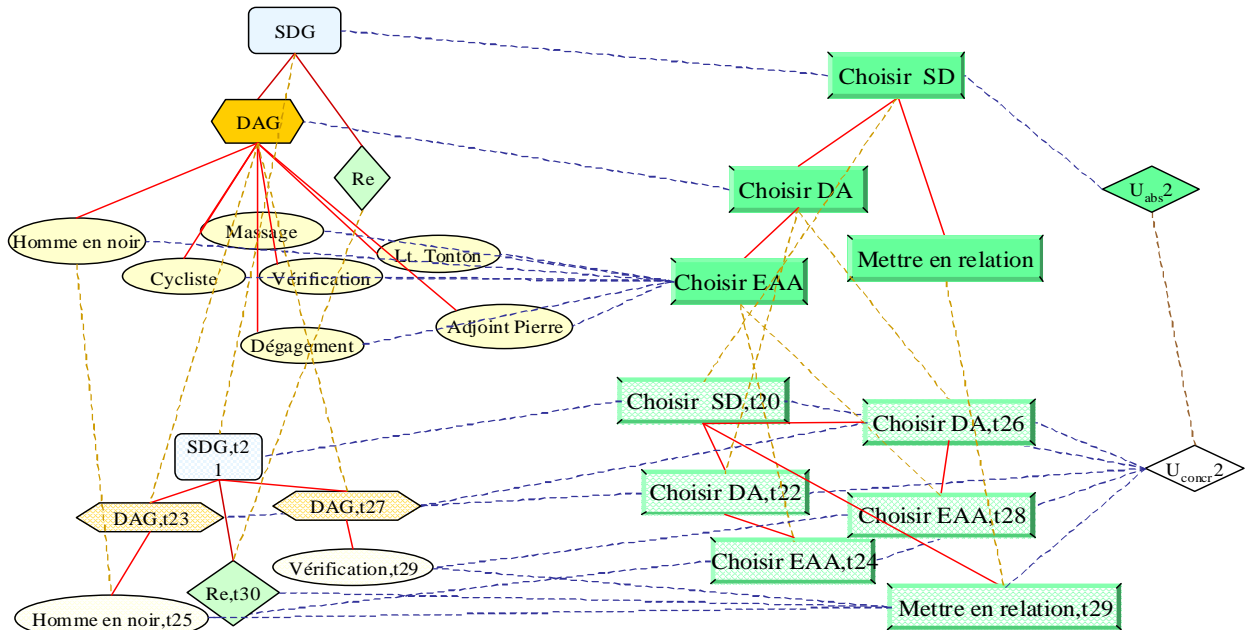


FIG. 16 Fragment du graphe global G, représentant de traces laissées par une annotation selon le SD« générique »

En fonction de type d'aide que l'on désire fournir, nous pouvons mettre en place différentes méthodes de calcul d'épisodes.

Supposons maintenant qu'un deuxième utilisateur commence l'annotation d'une autre vidéo. Il ne connaît pas le SD « premier secours », il choisit donc le SD « général ». En regardant la vidéo il l'annote avec le terme « Lt Tonton », choisi dans la DA « générale », avec le terme EAA « dégagement » choisi toujours dans la DA « générale », les met en relation et demande le système s'il y a d'autres termes, avec lesquels il pourra décrire sa vidéo. En faisant ses annotations il laisse les traces de la FIG. 16.

Pour construire nos épisodes nous mettons en place le graphe potentiel de la FIG. 17. Le graphe potentiel de la FIG. 17 permet de récupérer la trace formée par les annotations posées par un utilisateur selon un schéma de description. Dans notre cas, l'instanciation du graphe potentiel « Annotations posées par un utilisateur » appliqué au premier utilisateur dans le graphe global G et le tri chronologique du résultat donne la trace de la FIG. 18.

Si l'on applique le même graphe potentiel aux traces laissées par le deuxième utilisateur (FIG. 16) nous trouvons la trace calculée de la FIG. 19

Comme nous pouvons le constater les épisodes calculés dans les figures FIG. 18 et FIG. 19 commencent avec des instances des mêmes objets. Après comparaison des épisodes ainsi calculés le système propose au deuxième utilisateur d'annoter sa vidéo selon le SD « premiers secours » et donc de préciser encore une victime et la mettre en relation avec [Vérification,t29]. De cette façon nous obtenons des annotations cohérentes.

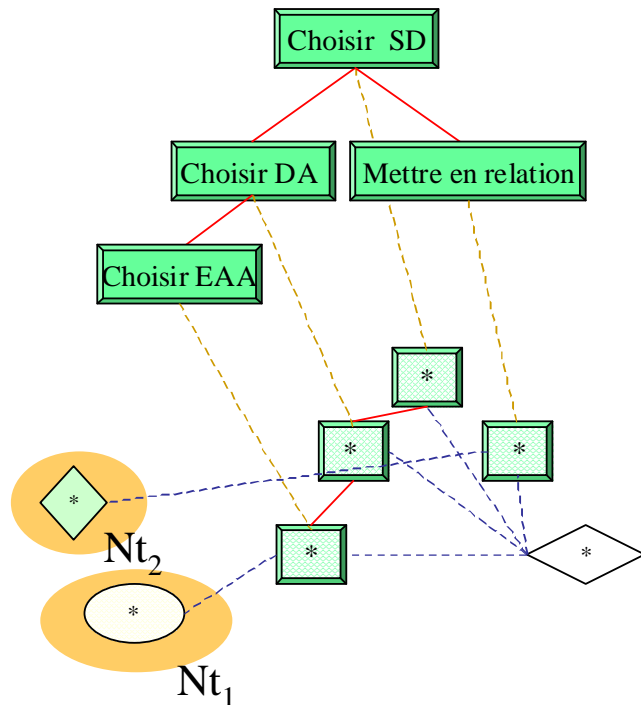


FIG. 17 Graphe potentiel pour calculer des traces : « Annotations posées par un utilisateur »

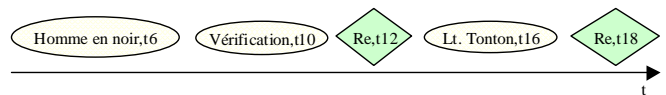


FIG. 18 Trace calculée à l'aide du GP « Annotations posées par un utilisateur » pour l'utilisateur U_{concr1}

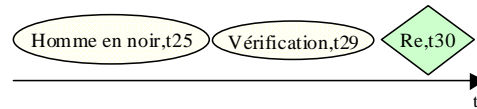


FIG. 19 Trace calculée à l'aide du GP « Annotations posées par un utilisateur » pour l'utilisateur U_{concr2}

6 Discussion et perspectives

Nous avons présenté un modèle de mise en place de traces d'utilisation qui permet de suivre l'utilisation d'une application en mettant en contexte aussi bien les objets manipulés que les procédés grâce auxquels cette manipulation a eu lieu ainsi que les utilisateurs qui ont œuvré. Nous avons défini les outils pour pouvoir tracer l'utilisation d'applications complexes pour fournir de l'aide pour la réalisation des tâches où l'expérience est importante. Grâce à ce modèle nous pouvons facilement récupérer les actions d'un utilisateur, ses sessions, mais également le contexte observé de son travail. Ainsi nous pouvons calculer, identifier, comparer, adapter et réutiliser des épisodes selon plusieurs critères.

Nous pouvons calculer des traces contenant les utilisateurs ayant manipulé certains objets, ayant exécuté des procédés dans un certain ordre, pour pouvoir par exemple repérer les utilisateurs qui travaillent de la même manière, les mettre en contact ou se servir des méthodes de l'un pour aider l'autre.

Lors de l'élaboration du modèle nous avons pris en compte les directives énumérées dans la section 3. Ainsi le modèle Trèfle permet de modéliser les tâches, les utilisateurs, les contextes d'utilisation et les applications d'une façon transparente aussi bien pour le système que pour les utilisateurs. Il permet également de faire émerger de nouvelles façons d'utiliser un système. Il peut être implanté aussi bien sous la forme d'un système d'aide de type assistant que de type conseiller. Finalement il s'agit d'un modèle qui s'enrichit avec l'utilisation et qui laisse la possibilité d'intervention de l'utilisateur dans la manière dont ces actions sont tracées et réutilisées.

Nous pensons que le modèle trèfle peut s'adapter à une large gamme d'applications permettant ainsi de représenter leurs traces d'utilisation sous la forme d'un graphe et d'utiliser les outils fondés sur la théorie de graphes pour exploiter ces traces.

Nous travaillons actuellement sur l'étude de l'adaptabilité du modèle sur une application type CAO, tout en continuant de formaliser et « polir » le modèle théorique général. Nous prévoyons également de proposer des représentations de nos traces dans les formalismes standards tels que RDFS [32].

Références

- [1] Bénel, A., E. Egyed-Zs., et al., 2001. *Truth in the Digital Library: From Ontological to Hermeneutical Systems*. ECDL 2001 European Conference on Research and Advanced Technology for Digital Libraries, Darmstadt (D), Springer-Verlag, p.366-377.
- [2] Birnbaum, L., R. Bareiss, et al., 1998. *Interface Design Based on Standardized Task Models*. Intelligent User Interfaces, San Francisco CA, USA, ACM, p.65-72.
- [3] Bush, V., 1945: *As we may think*. Atlantic Monthly 176, p.101-108.
- [4] Chandrasekaran, B., J. R. Josephson, et al., 1998. *Ontology of tasks and methods*. Eleventh Workshop on Knowledge Acquisition, Modeling and Management (KAW '98), Banff, Canada, p.25.
- [5] Charlet, J., B. Bachimont, et al., 1996: Ontologie et réutilisabilité: expérience et discussion. Acquisition et Ingénierie des Connaissances. C. Editions. Paris. ISBN 2-85428-417-8, p.69-87.
- [6] Clarke, T., 2001, *Epistemics PC-Pack*, <http://www.epistemics.co.uk/products/pcpack/tools/>
- [7] Delgado, J. and N. Ishii, 1999. *Formal Models for Learning User Preferences, A Preliminary Report*. International Conference on Artificial Intelligence (IJCAI-99), Workshop on Learning about Users, Stockholm, Sweden, p.8p.
- [8] Derthick, M. and S. F. Roth, 2000. *Data exploration across temporal contexts*. ACM International Conference on Intelligent User Interfaces, New Orleans, LA USA, p.60 - 67.
- [9] Dukas, 1998, *Dukas Graphical Task Modeling Tool*, http://www.cc.gatech.edu/gvu/user_interfaces/Mastermind/Dukas/
- [10] Dussaux, G. and J.-P. Pécuchet, 2000. *Création collective de bases de connaissances sur le Web. Indexation par l'usage des documents*. CIDE 2000, Lyon, FR, Europa Production, p.185-203.
- [11] Egyed-Zs., E., 2001. *Manipulation des documents multimédia, idées d'application du RàPC pour une aide à l'utilisateur*. Séminaire RàPC, Grenoble, F, p.73-78.
- [12] Egyed-Zs., E., Y. Prié, et al., 2000. *A graph based audio-visual document annotation and browsing system*. RIAO 2000, Paris, France, p.1381-1389.
- [13] Farrell, R., P. Fairweather, et al., 2000. *A task-based architecture for application-aware adjuncts*. ACM International Conference on Intelligent User Interfaces, New Orleans, LA USA, p.82-85.
- [14] Francisco-Revilla, L. and F. M. Shipman, 2000. *Adaptive medical information delivery combining user, task and situation models*. International Conference on Intelligent User Interfaces, New Orleans, Louisiana, United States, ACM Press New York, NY, USA, p.94 - 97.
- [15] Jaczynski, M. and B. Trousse, 1998. *WWW Assisted Browsing by Reusing Past Navigations of a Group of Users*. In Advanced in Case-based Reasoning, 4th European Workshop on Case-Based Reasoning, p.160-171.
- [16] Lehman, J. F., J. Laird, et al., 1995, *A Gentle Introduction to Soar, an Architecture for Human Cognition*, <http://ai.eecs.umich.edu/soar/docs/Gentle.pdf>
- [17] Lieberman, H., 1993: *Mondrian: A Teachable Graphical Editor. Watch What I Do: Programming by Demonstration*. A. Cypher, MIT Press.
- [18] Lieberman, H., 1995. *Letizia: An Agent That Assists Web Browsing*. Proceedings of the International Joint Conference on Artificial Intelligence, Montreal, Canada,
- [19] Lieberman, H., 1997. *Autonomous Interface Agents*. ACM Conference on Human Factors and Computing Systems, Atlanta, GA, USA, ACM, p.67-74.
- [20] Lieberman, H. and D. Maulsby, 1996: *Instructible Agents: Software That Just Keeps Getting Better*. IBM Systems Journal 35, p.20.
- [21] Macromedia, 1999: *Dreamweaver 3*, <http://www.macromedia.com/>
- [22] Mille, A., 1998: *Associer expertise et expérience pour assister les tâches de l'utilisateur*. Université Claude Bernard, Lyon1, CPE
- [23] Paterno, F., 2000: *Model-based design of interactive applications*. intelligence, ACM Press New York, NY, USA
- [24] Pfister, H.-R., M. Wessner, et al., 1999. *Soziale und kognitive Orientierung in einer computergestützten kooperativen Lernumgebung*. Software-Ergonomie '99. Design von Informationswelten, Stuttgart, DE, p.265-274.
- [25] Prié, Y., 1999: *Modélisation de documents audiovisuels en Strates Interconnectées par les annotations pour l'exploitation contextuelle*. Informatique, INSA-Lyon
- [26] Prié, Y., T. Limane, et al., 2000. *Isomorphisme de sous-graphe pour la recherche d'information audiovisuelle contextuelle*. 12ème congrès Reconnaissance de Formes et Intelligence Artificielle, RFIA2000, Paris, FR, p.277-286.
- [27] Protégé, 2002, *Protégé 2000*, Stanford Medical Informatics
- [28] Ruvini, J.-D. and C. Dony, 2000. *APE: learning user's habits to automate repetitive tasks*. ACM International Conference on Intelligent User Interfaces, New Orleans, LA USA, p.229 - 232.
- [29] Schreiber, G. and J. Wielemaker, 2001, *ModelDraw*, <http://www.commonkads.uva.nl/INFO/tools/modeldraw.html>
- [30] Selker, T., 1994: *Coach: A teaching Agent that Learns*. Communications of the ACM 37, p.92-99.
- [31] Trousse, B., M. Jaczynski, et al., 1999. *Une approche fondée sur le raisonnement à partir de cas pour l'aide à la navigation dans un hypermédia*. Hypertexte & Hypermedia : Products, Tools and Methods (H2PTM'99), Paris,
- [32] W3C, 2000: *Resource Description Framework (RDF)*, <http://www.w3.org/RDF/>