

# Une théorie de la trace informatique pour faciliter l'adaptation dans la confrontation logique d'utilisation/logique de conception

Alain Mille – Yannick Prié  
LIRIS UMR 5205 – Université Claude Bernard Lyon 1  
prenom.nom@liris.cnrs.fr

## 1 Introduction

Un environnement informatique *perçu* offre des possibilités nombreuses et variées d'inter-actions<sup>1</sup> pour accompagner une activité humaine pour des tâches plus ou moins formatées. Nous ne faisons pas d'hypothèse *a priori* sur la notion d'environnement informatique, qui peut être considéré à des niveaux de granularité très variables. Le contour (ou le périmètre) ressenti d'un environnement dépend en effet fortement du contexte d'utilisation, lequel n'a de sens que pour l'utilisateur, acteur dans son environnement. Un environnement pourra alors aller d'une simple fenêtre de configuration d'une application jusqu'à l'ensemble des outils disponibles à l'utilisateur à un instant donné.

Les composants d'un environnement informatique sont conçus sur la base d'une analyse de l'activité des futurs utilisateurs et des tâches impliquées. Ils offrent différentes possibilités d'inter-action conformément à l'analyse menée. D'autre part, beaucoup de composants sont « ouverts » aux autres et peuvent être utilisés les uns avec les autres, les uns à partir des autres<sup>2</sup>, *etc.* Le périmètre de l'environnement informatique *conçu* se révèle donc difficile à stabiliser à l'avance.

Un hiatus est donc inévitable entre environnements informatiques perçus et environnements informatiques conçus. Il nous semble donc que la question de l'accompagnement de l'activité de l'utilisateur par des « facilitateurs » d'appropriation des possibilités offertes se pose donc dans la conception même des environnements informatiques.

Après avoir précisé les notions de *logiques d'utilisation* et de *logiques de conception* qui se confrontent au moment de l'utilisation, nous proposons d'utiliser les traces d'inter-action comme support de facilitation de la gestion du hiatus par l'utilisateur. Enfin, nous présentons le processus de modélisation de l'utilisation nécessaire pour décrire les inter-actions et détaillons le modèle MUSETTE que nous proposons pour y parvenir. Nous illustrons ensuite brièvement la démarche de modélisation de l'utilisation avant de nous intéresser à la notion de « facilitateur » en fournissant un exemple dans le cadre d'un Environnement Informatique d'Apprentissage Humain.

Il ne s'agit pas ici de développer les questions épistémologiques importantes que l'approche soulève et de développer les enjeux d'une telle recherche, mais de

---

<sup>1</sup> Inter-action : dénote le fait que les actions de l'utilisateur et les réactions de l'environnement sont dépendantes les unes des autres dans le cadre d'une activité particulière. L'utilisateur agit avec l'environnement comme intermédiaire avec lui-même d'abord. Il n'est pas possible de dissocier l'action de l'attente du résultat de l'action qui précisément est le lieu de la confrontation logiques d'utilisation / logiques de conception.

<sup>2</sup> Il est possible par exemple d'autoriser de réaliser un document à partir d'un tableur mais d'exploiter intensivement les possibilités d'un autre environnement comme un traitement de texte dans ce tableur (et vice-versa).

contribuer à la réflexion sur la notion de traces et de leurs usages quand il s'agit d'environnements informatiques. Ces questions seront toutefois évoquées dans la discussion qui conclut l'article.

## **2 Logiques d'utilisation et logiques de conception**

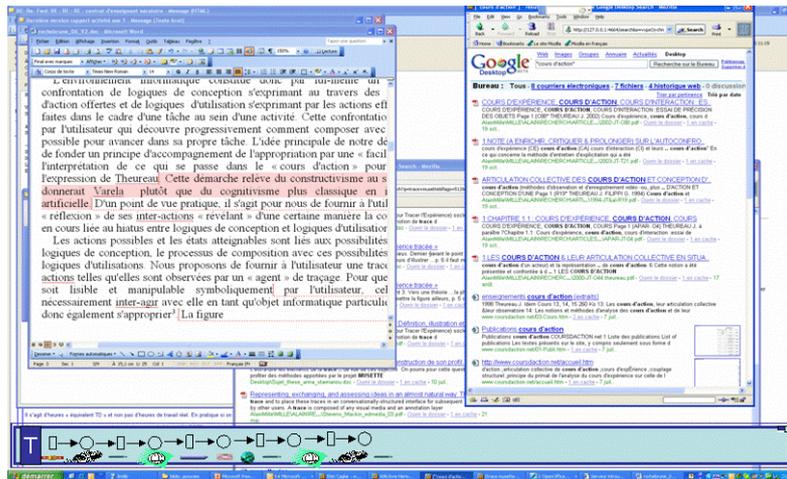
Les environnements mis à disposition sont conçus selon des *logiques de conception* et sont exploités selon des *logiques d'utilisation*. Nous choisissons ce terme *logique* pour référer à une rationalité qui pourrait être revendiquée par les concepteurs ou les utilisateurs dans leurs choix. Nous mettons ce terme au pluriel pour indiquer qu'il y a peu de chance qu'une conception ou une utilisation réponde à une seule « logique ».

Les logiques de conception s'expriment au travers de l'environnement développé par les possibilités d'inter-actions offertes avec des objets informatiques manipulables par l'utilisateur, ces possibilités variant selon les différents états prévus de l'environnement.

Quel que soit le soin apporté au processus de conception, l'utilisateur tentera d'appliquer ses propres « logiques d'utilisation ». Les manifestations de ces logiques d'utilisation se constatent par l'observation. C'est au travers de l'exploitation de « retours d'expérience » qu'elles sont habituellement utilisées pour tenter de réduire le hiatus constaté avec les logiques de conception. Il s'agit alors de mettre en place de nouvelles conceptions intégrant les logiques d'utilisation au mieux en offrant de nouvelles possibilités d'inter-actions avec l'environnement.

L'environnement informatique constitue par lui-même un terrain de confrontation de logiques de conception s'exprimant au travers des possibilités d'inter-action offertes et de logiques d'utilisation s'exprimant par les inter-actions effectivement attendues dans le cadre d'une tâche au sein d'une activité. Cette confrontation est gérée par l'utilisateur pour avancer dans sa propre tâche. Les actions possibles et les états atteignables sont liés aux possibilités issues des logiques de conception ; composer avec ces possibilités relève des logiques d'utilisations. L'idée principale de notre démarche est de fonder un principe d'accompagnement de l'appropriation par une facilitation de l'interprétation de ce qui se passe dans le cours d'action de l'utilisateur pour reprendre l'expression de Theureau [1]. D'un point de vue pratique, il s'agit pour l'utilisateur d'exploiter un dispositif de « réflexion » de ses inter-actions « révélant » d'une certaine manière la confrontation en cours liée au hiatus entre logiques de conception et logiques d'utilisation.

En guise de dispositif de « réflexion », nous proposons de fournir à l'utilisateur une trace des inter-actions telles qu'elles sont observées par un « agent » de traçage. La trace produite doit permettre la représentation des inter-actions et être le support de facilitation de la gestion du hiatus entre logiques de conception et logiques d'utilisation. En conséquence, une telle trace doit être lisible et manipulable symboliquement par l'utilisateur ; celui-ci inter-agira nécessairement avec elle en tant qu'objet informatique particulier (voir **Figure 1** par exemple) qu'il devra donc également s'approprier.



**Figure 1.** Une fenêtre « trace » s'ajoute dans l'environnement. L'environnement informatique perçu comprend les objets (avec leurs propriétés) permettant les inter-actions. Un objet trace (bas de l'écran) des transitions et des états se succédant sur une ligne temporelle, illustrés par les icônes des objets concernés dans l'inter-action. Les composants de la trace sont explorables et manipulables par l'utilisateur.

L'intégration d'objets traces transforme *de facto* un environnement informatique S en un nouvel environnement informatique S'. L'utilisation du système S', intégrant la gestion de traces explicites, peut donner lieu à une modélisation de nouvelles utilisations et leur traçage/présentation, pouvant donner lieu à la mise en place d'un système S'' intégrant l'utilisation des traces de S dans S'. Il s'agit donc bien d'une dynamique de l'utilisation liée à la disponibilité de traces gérées par l'utilisateur, et la connaissance de l'utilisateur sur l'utilisation de l'environnement se construit inter-activement en étant symboliquement représentable par l'utilisateur lui-même à des fins de facilitations de ses propres tâches.

### 3 Modéliser l'utilisation pour pouvoir la tracer

Nous présentons d'abord le principe de modélisation de l'utilisation en introduisant les notions de *modèle d'utilisation (MU)* et de *signatures de tâches expliquées (SiTEx)* avant de détailler une manière de les mettre en œuvre avec l'approche MUNETTE.

#### 3.1 Modèle d'utilisation et signatures de tâches expliquées

Nous appelons *modèle d'utilisation* l'ensemble des éléments qui permettent de produire une trace d'utilisation sur la base des inter-actions avec un environnement informatique. La construction d'un modèle d'utilisation nécessite qu'un modélisateur humain observe des utilisateurs dans leurs pratiques de l'environnement. L'observation « initiale » est réalisée dans le cadre d'un environnement informatique S ne comportant pas d'objets traces. C'est une observation « ouverte »<sup>3</sup>, destinée à identifier ce que l'utilisateur prend en considération comme entités et événements lorsqu'il inter-agit avec l'environnement.

Une façon concrète de construire un modèle d'utilisation initial est de mettre en évidence ce que nous avons convenu d'appeler des « signatures de tâches

<sup>3</sup> Par opposition à une observation « fermée » dans laquelle on ne ferait que vérifier des choses. Il est nécessaire d'observer les situations d'utilisations en tant que telles, sans vouloir les découper *a priori* selon une vision qui ne pourrait venir que de pré-conçus.

expliquées » (SiTEx). Une SiTEx est un motif inter-actionnel que les utilisateurs reconnaissent dans une trace comme typique d'une tâche qu'ils peuvent alors expliquer (décrire). Les signatures peuvent avoir des « grains » variés, depuis une séquence d'opérations élémentaires jusqu'à une combinaison de possibilités d'inter-actions offertes par différents environnements, considérés comme un seul environnement. La manière d'exprimer les signatures fournit matière à symboliser les processus inter-actionnels pour mieux les manipuler ensuite. Le modèle d'utilisation initial doit permettre de produire ce que nous avons convenu d'appeler une « trace primitive ». Celle-ci constitue le premier niveau considéré comme interprétable par l'utilisateur d'une trace et exploitable par la machine via les SiTEx permettant de repérer des séquences inter-actionnelles potentiellement significatives. Nous verrons dans la section 3 comment ce principe est exploité pour construire des facilitateurs pilotés, voire co-conçus par les utilisateurs.

Le modèle d'utilisation d'un environnement<sup>4</sup>  $S$  est utilisé pour mettre en place l'objet informatique *Trace* qui permet donc de « parler » de ce que l'on fait en interagissant avec  $S$ . Le contour de l'environnement  $S$  a donc évolué en  $S'$  ( $S$  + la trace) mais il n'est pas possible de « parler » de l'objet *Trace* dans la trace elle-même puisqu'elle est construite sur la modélisation d'utilisation de  $S$ .

La dynamique d'utilisation possède deux dimensions :

- Dynamique d'utilisation du système  $S$ , ce qui amène à passer de  $MU_0(S)$  (MU initial de  $S$ ) à  $MU_1(S)$  (évolution de  $MU_0(S)$ ) et d'une manière générale de raffiner chaque nouveau  $MU_j(S)$  en un autre  $MU_k(S)$  pour des raisons très diverses (appropriation d'éléments liés aux logiques de conception, construction de nouveaux éléments généralisant, introduction de nouvelles possibilités, etc.).
- Dynamique liée à l'intégration de l'objet trace dans le modèle d'utilisation. Cette fois, l'objet *trace* et ses usages étant intégrés dans le modèle d'utilisation, le  $MU$  résultant permet de décrire le système  $S'$ . Un nouvel objet trace peut alors être créé, correspondant à l'objet trace pour  $S$  qui sera décrit dans l'élaboration du  $MU_0(S')$ . Cette dynamique est d'une nature différente, introduisant une mise en abîme du dispositif. La représentation des traces peut alors être combinée, offrant à l'utilisateur la capacité de donner du sens à l'usage des traces elles-mêmes.

### 3.2 MUSETTE : Modéliser les Utilisations et les Tâches pour Tracer l'Expérience

L'approche Musette [2] a été mise en place dans l'objectif de fournir un cadre pour l'assistance à base de trace au sein de l'équipe CEXAS<sup>5</sup>. La *Figure 2* illustre l'approche MUSETTE, articulant le niveau du traçage (en haut à droite) et celui de la ré-utilisation d'épisodes d'utilisation (à gauche). Le schéma se lit de la façon suivante : un utilisateur inter-agit avec un environnement (système). Un agent observateur, guidé par un modèle d'observation, génère à partir de ces inter-actions une trace primitive respectant un modèle d'utilisation. Un analyseur générique de trace en extrait alors des épisodes, en accord avec des signatures de tâches expliquées. Ces épisodes sont ré-utilisés par des agents assistants, dont l'action peut

<sup>4</sup> Il faut remarquer que le modèle d'utilisation d'un environnement est une façon de donner le contour *perçu* de l'environnement informatique. Le contour de l'environnement *conçu* n'est en général pas accessible.

<sup>5</sup> *Cognition, Expérience et Agents Situés*, équipe de l'axe *Données, document, et connaissances* du LIRIS.

être soit spécifique avec des agents clairement distincts du système, soit intégrée grâce au système lui-même. Musette est une approche générale de modélisation, et peut être implémentée grâce à divers langages ou formalismes de représentation, à même d'en représenter les différents éléments, que nous précisons dans ce qui suit<sup>6</sup>.

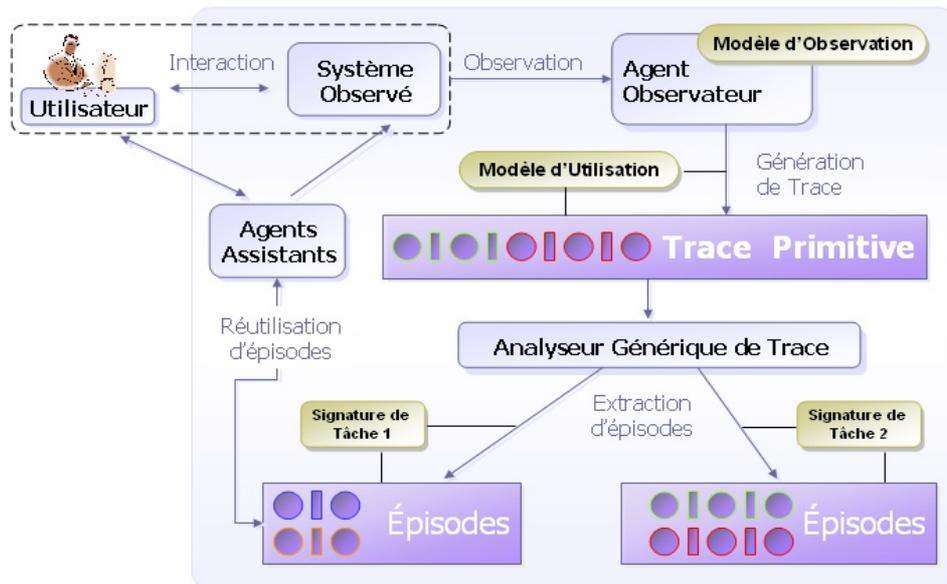


Figure 2. Approche Musette

Il convient de déterminer de quoi la trace sera constituée (modèle d'utilisation), et comment celle-ci sera générée (modèle d'observation). Les éléments d'interaction constituant la trace sont des *objets d'intérêt* (OI) de trois types : *entités*, *événements* et *relations*. Les entités sont des objets « présents » pour l'utilisateur dans son interaction ; les événements sont des objets qui « ont lieu », qui « se passent », durant cette même interaction ; les relations peuvent lier indifféremment des entités et/ou des événements. Un modèle d'utilisation décrit les entités, événements et relations à observer pour construire une trace primitive, à la manière d'une « ontologie de l'observation ». Associé au MU, un modèle d'observation décrit les règles nécessaires à la détermination des données pertinentes issues de l'environnement et à la construction effective de la trace primitive. Le modèle d'observation n'est pas spécifié dans l'approche Musette et doit être adapté à chaque environnement observé.

La trace primitive est structurée selon deux types de structures : les états, regroupant des entités, et les transitions regroupant des événements. Les premiers représentent l'état du système observé à un instant/période donné par rapport au modèle d'utilisation. Une transition regroupe un ensemble d'événements, observés entre deux états. La trace est donc *in fine* une séquence d'états et de transitions<sup>7</sup>.

La figure 3 présente un exemple simplifié d'un modèle d'utilisation de navigateur Web, associé à un fragment de trace primitive. Les entités du modèle d'utilisation sont les pages Web (*Page*), les Hyperliens (*Lien*), les images (*Img*) et les préférences (*Pref*). Les événements du modèle d'utilisation sont les clics de l'utilisateur (*Clic*), ses pages sauvegardées localement (*Sauv*), ses bookmarks sur une page (*Bm*) et le

<sup>6</sup> Les contraintes éditoriales nous contraignant à aller à l'essentiel, nous invitons le lecteur intéressé à consulter les références citées, qui discutent plus précisément de nombreux points ici survolés ou passés sous silence.

<sup>7</sup> Le choix des termes *État* et *Transition* est lié à la volonté de considérer le cours d'action comme un processus.

changement de langue (*Lang*). Dans la suite de cet exemple, nous considérerons que les pages possèdent également un attribut URL contenant leur adresse.

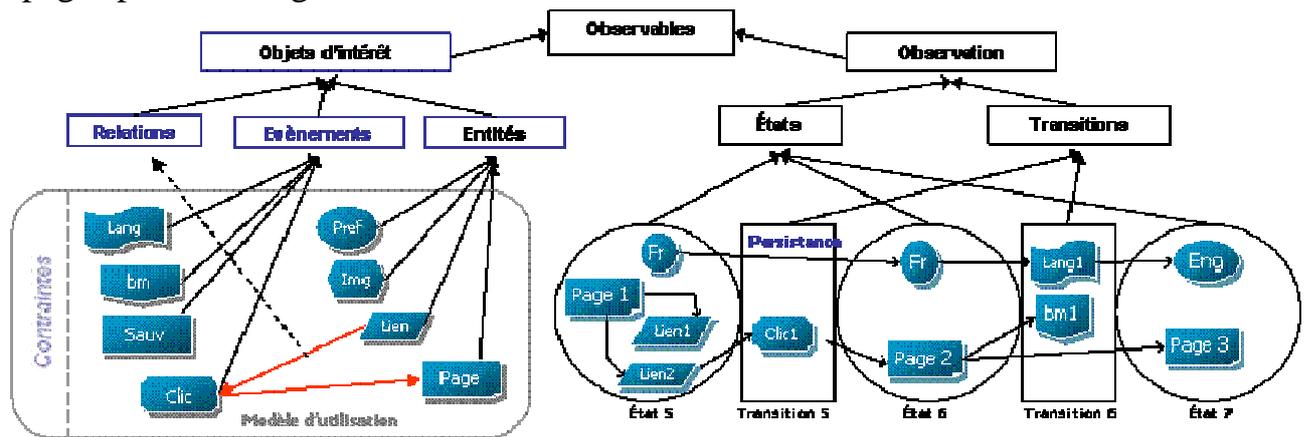


Figure 3. Un modèle d'utilisation simplifié et un fragment de trace de navigation

Pour la lisibilité de la figure, toutes les relations possibles du MU n'y ont pas été reportées. Seules apparaissent les relations d'une entité Lien vers un évènement Clic, et celle d'un évènement Clic vers une entité Page - signifiant simplement que le lien cliqué mène à la nouvelle page affichée. D'autres relations apparaissent uniquement dans le fragment de trace, mais sont en principe définies dans le modèle d'utilisation : d'une page à un lien (« la page contient le lien »), d'une page à un bookmark (« le bookmark s'applique à cette page »), de page à page (« cette page est rafraîchie »), de préférence à changement de langage, puis à une autre préférence (quand les préférences représentent les choix de l'ancien et du nouveau langage). On peut noter que la transition 6 comporte deux évènements, indépendants l'un de l'autre dont la trace ne donne pas l'ordre d'apparition. La relation Persistance permet quant à elle, dans ce modèle d'utilisation particulier, de traduire le fait que c'est un même objet du point de vue du système, qui se retrouve dans deux états successifs.

Nous appelons épisode une sous-partie de la trace primitive, qui peut y être découpée à l'aide d'une signature de tâche expliquée. Une telle signature s'exprime par : (a) des motifs de graphe constitués d'objets d'intérêt et leurs relations, (b) des contraintes relatives aux positions (distances relatives) des OIs dans la trace, (c) des contraintes sur la structure interne des objets d'intérêt. Les « explications » d'une signature de tâche permettent d'apporter des précisions sur les épisodes qu'elle a détectés : un épisode sera au minimum nommé, éventuellement annoté librement – en vue d'une lecture par l'utilisateur – ou formellement – en vue d'une utilisation automatique.

Nous présentons ci-dessous deux tâches simples de navigation Web qui peuvent être identifiées par notre modèle d'utilisation simplifié, et montrons comment leurs signatures peuvent être expliquées par des annotations sur les objets d'intérêt correspondants. Premier exemple : lorsqu'un utilisateur trouve une page intéressante, il lui arrive souvent de vouloir poser un bookmark sur le site dans lequel il a trouvé la page, considérant qu'il en contient – ou contiendra – d'autres. Ce genre de tâche est aisément repérable dans la trace primitive (fig. 3) : depuis une page (Page) d'un site, l'utilisateur remonte à la page d'accueil et y pose un bookmark (Bm). Des

explications sont également fournies dans cette SiTEx : l'évènement bookmark (Bm) est annoté avec une explication textuelle, potentiellement utile pour l'interprétation de l'épisode et les deux pages Web impliquées peuvent être annotées par « page interne » et « page de garde », selon une ontologie des sites Web, définie en dehors de Musette, et utilisable par un assistant dédié.

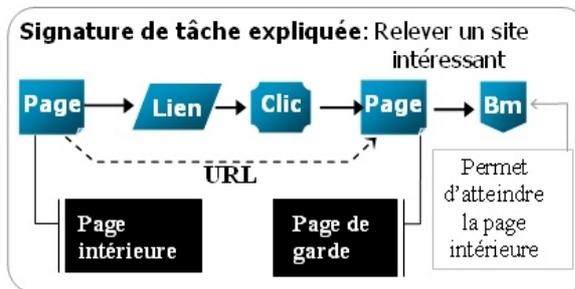


Figure 4. SiTEx et épisode (exemple 1)

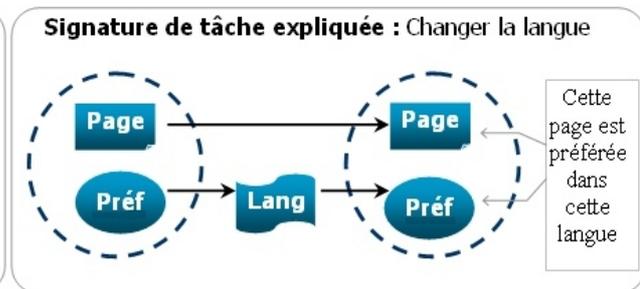


Figure 5. SiTEx et épisode (exemple 2).

Une autre tâche consiste à modifier les préférences de langue pour visualiser différentes versions de pages Web. Le changement étant représenté par un évènement Lang (lorsque la page change sans avoir cliqué sur un lien). La SiTEx de la figure 4, exprime une signature de ce type, les cercles en pointillés représentant une contrainte de co-occurrence des objets d'intérêt Page et Préf (représentant le langage courant de l'utilisation). Dans cet exemple des annotations textuelles donnent des explications quant aux objets d'intérêt concernés.

### 3.3 Illustration : construction d'un modèle d'utilisation pour un environnement de veille stratégique

L'approche Musette a été utilisée [3, 4] pour la modélisation de l'utilisation de l'outil de veille stratégique cartographique Human-Links<sup>8</sup>. Le modèle d'utilisation résultant contient 35 entités, 42 évènements et plusieurs dizaines de relations, et sa définition a permis d'éprouver Musette ainsi que de mettre en place des éléments méthodologiques de conception de MU :

- la nécessité de définir précisément l'activité dans laquelle est utilisé l'environnement étudié (en observant les utilisateurs),
- la validation *a priori* des objets d'intérêt par leur appartenance à une SiTEx,
- la nécessité que tous les objets d'intérêt soient intelligibles par les utilisateurs (qui permettront de trancher le cas échéant),
- les compromis rendus nécessaires par le fait de travailler à plusieurs niveaux d'assistance différents (niveau de l'interface, niveau de la tâche de veille), et les manières de mêler des objets d'intérêts de ces différents niveaux, *etc.*

## 4 Application à la « facilitation » de tâches médiées par les environnements informatiques

### 4.1 Principes d'un facilitateur

Nous avons choisi d'appeler « facilitateur » [5] tout agent informatique exploitant le contenu de la trace d'utilisation pour fournir à l'utilisateur des éléments de compréhension de l'environnement par rapport à ses propres utilisations et la capacité

<sup>8</sup> <http://www.human-links.com/>

de s'approprier les possibilités d'inter-action offertes dans le cadre de ses utilisations propres. Le facilitateur doit donc « suivre » au plus près le contexte d'utilisation, défini par l'utilisateur lui-même en train d'agir. Une requête de facilitation consiste pour l'utilisateur à fournir, directement ou non, un contexte d'utilisation qui fait sens pour lui et pour lequel il espère trouver des éléments complémentaires élargissant sa compréhension de la situation. Il utilise le mécanisme de signature de tâche expliquée comme support de sa requête. La signature peut être fournie de différentes façons selon le degré d'anticipation de la facilitation :

- *Nouvelle signature de tâche expliquée* : l'utilisateur réalise des inter-actions, exploite la trace pour choisir les éléments qui font sens dans sa tâche courante et utilise les éléments choisis comme constitutifs de la signature de tâche expliquée qui sera sa requête. Le facilitateur recherche dans la trace les séquences « similaires » et les propose comme épisodes candidats en les situant temporellement et par les objets qui étaient alors en relation, ce qui permet à l'utilisateur d'élargir son interprétation et d'agir en conséquence.
- *Signature de tâche expliquée cataloguée* : l'utilisateur choisit un squelette de requête dans une bibliothèque de signatures de tâches expliquées. Celle qui correspond à son utilisation courante (qui sert d'index) et les épisodes correspondants sont retrouvés par l'agent facilitateur comme dans le premier cas. Les écarts constatés entre les épisodes retrouvés et l'épisode en cours de construction sont mis en évidence pour faciliter l'adaptation par l'utilisateur lui-même.
- *Signature de tâche expliquée canonique* : un modélisateur de la trace a conçu (en liaison avec des utilisateurs) des signatures canoniques à respecter, ainsi qu'un agent de type assistant permettant la mise en œuvre d'une utilisation conforme aux signatures en exhibant des épisodes réels qui pourront être adaptés. Cet usage des signatures de tâche expliquée revient à faire du Raisonnement à Partir de Cas à des fins d'assistance. La partie suivante illustre l'utilisation de cette dernière approche pour la réalisation d'un agent d'assistance dans le domaine de l'apprentissage humain, en se plaçant dans le dernier cas.

#### **4.2 Illustration d'un facilitateur : Pixed (Projet d'intégration de l'expérience pour l'enseignement à distance)**

##### **4.2.1 Présentation générale**

Dans le contexte de l'apprentissage à distance, l'apprenti construit sa connaissance en inter-agissant avec les environnements d'apprentissage, tentant d'exploiter au mieux les activités éducatives disponibles. Apprendre est vu comme mobilisant (entre autres) une capacité à résoudre un problème : « Quel est le meilleur chemin à suivre pour maîtriser cette notion ? ». C'est donc une signature de tâche expliquée correspondant aux opérations d'orientation dans l'hyperespace du cours qui est à la base de l'agent d'assistance réalisé.

Le modèle de conception correspond en partie aux spécifications PAPI<sup>9</sup> et intègre ce que nous avons appelé [6] un réseau notionnel annoté (RNA). Les noeuds du réseau sont des notions définies comme des buts d'apprentissage et les arcs sont

---

<sup>9</sup> Public and Private Information (PAPI) for learners (IEEE). <http://edutool.com/papi/>

constitués de relations de précédence avec un certain seuil de franchissement et de relations de suffisance possédant un attribut de seuil de maîtrise et un ratio de contribution à la notion cible. PIXED [7-10] permet aux enseignants comme aux apprentis d'annoter tout ou partie des activités éducatives (document hypertextes simples : HTML). Les annotations servent de support de communication entre enseignants, entre enseignants et apprentis et entre apprentis ; elles sont à la base des procédures d'adaptation de présentation de contenus.

#### **4.2.2 *Modèle d'utilisation***

Dans cette partie, nous mettons en **gras** les éléments du modèle d'utilisation. Les objets d'intérêt sont constitués par les objets avec lesquels l'apprenant inter-agit durant l'apprentissage. Les **activités pédagogiques, documents numériques** et **Quiz** sont associés aux noeuds du **réseau notionnel annoté** (RNA). Le **cours** est construit initialement autour d'un RNA conçu par les enseignants sur des bases didactiques. Chaque apprenti navigue dans le cours par l'intermédiaire du RNA et son avancement est représenté par un sous-graphe de ce RNA qu'il peut donc enrichir en l'annotant avec des notions qui lui sont propres et des **activités pédagogiques** qu'il aura exploitées de sa propre initiative. Les activités pédagogiques sont intégrées en tant que documents numériques avec leur description. Les activités pédagogiques sont annotées par les **notions** qu'elles expliquent, illustrent, référencent, etc., en précisant quelles sont les notions utiles, nécessaires, suffisantes, etc., pour les exploiter. Les notions étant reliées entre elles par des **relations de précédence** et **de suffisance** pour former un **cours** sous la forme d'un RNA particulier, ces informations sont utilisées pour construire automatiquement une première version du réseau initial.

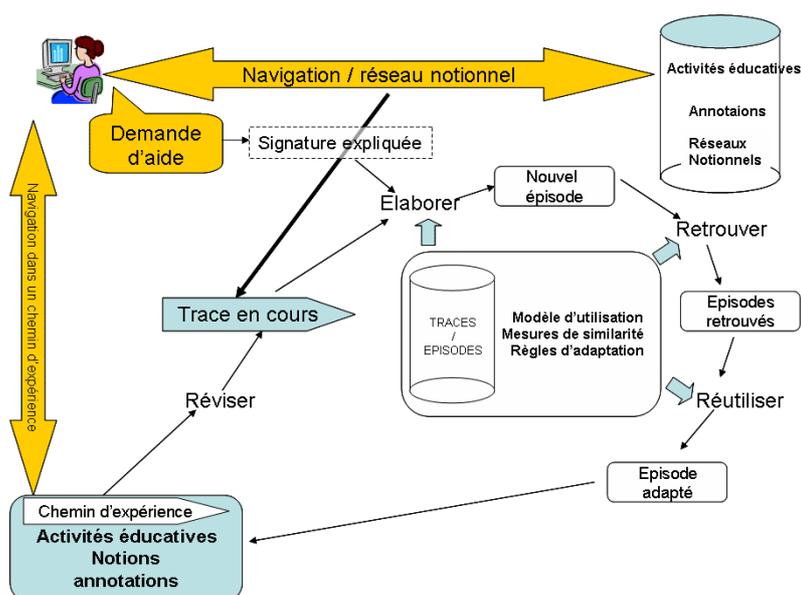
La construction du modèle d'utilisation a été faite empiriquement, et la sémantique des relations (précédence, suffisance) est clairement orientée par des objectifs didactiques, relevant de logiques de conception. Naviguer dans un cours à l'aide d'un réseau notionnel annoté peut se révéler difficile pour l'apprenti qui par définition ne maîtrise pas les notions qui lui sont proposées à l'apprentissage. Il lui est pourtant nécessaire de se construire un chemin parmi les différentes navigations possibles pour atteindre la notion cible de l'apprentissage (au départ, cette notion est la notion cible du cours lui-même). PIXED intègre une façon originale de construire un chemin pertinent sous la forme d'un arbre présentant analogiquement (longueur et position des branches d'accès aux notions) l'effort à fournir pour atteindre la notion cible courante. Les sémantiques des relations de suffisance et de précédence permettent en effet de construire un **arbre de recommandation** pour suivre un **chemin** selon une progression notionnelle didactiquement adaptée.

#### **4.2.3 *La trace et son exploitation dans PIXED***

Tout choix dans l'orientation du cours (choix d'une activité pédagogique) est tracé (y compris les inter-actions liées à l'évaluation par l'intermédiaire des Quiz). L'exploitation interne d'une activité pédagogique elle-même n'est pas tracée dans cette version de PIXED. Nous nous intéressons à renvoyer à l'apprenti une possibilité d'exploiter des traces d'apprentissages comme sources de connaissances pour assister son orientation dans la progression d'apprentissage dans un cours.

La SiTEx considérée ici est « ce que j'ai fait (objets inter-agis), ce que j'ai appris (notions "réussies") et ce que je souhaite apprendre (notion à acquérir) » et est utilisée par l'assistant pour suggérer l'adaptation « d'un chemin » notionnel avec des activités pédagogiques alternatives qui, selon l'expérience similaire, pourraient faire sens dans la situation courante.

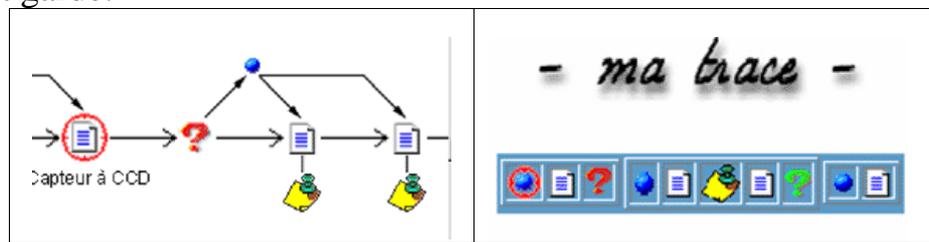
Quand l'apprenti navigue dans son cours en parcourant le RNA correspondant, le système trace les inter-actions d'apprentissage. Un épisode courant est constitué de la partie de trace démarrante avec la notion ciblée pour le cours (la situation finale souhaitée). Sur la base des mesures de similarités<sup>10</sup> et des différents potentiels de réussite, le système sélectionne des épisodes similaires à la situation courante. A partir de ces épisodes, le système crée un épisode adapté à la situation en cours, tentant de maximiser le potentiel de l'épisode courant de l'apprenti. Cet épisode permet de fournir à l'apprenti un « chemin d'expérience » (sorte de façon de continuer son épisode) qu'il pourra personnaliser et exploiter. L'apprenti peut en effet naviguer dans ce chemin d'expérience, choisir des notions relais, exploiter les activités éducatives qui ont aidé d'autres apprentis dans la même situation d'apprentissage, consulter des annotations sur ces activités éducatives (posées par d'autres apprentis, ou des enseignants). L'épisode courant est tracé pour être ajouté aux épisodes de la base dès que la notion ciblée est atteinte, tel qu'il aura été vraiment réalisé par l'apprenti.



**Figure 6.** Le cycle Pixed : au cours de la navigation, l'apprenant demande une assistance pour le guider ; la situation d'apprentissage est représentée par une signature de tâche correspondant à « comment continuer l'apprentissage » ; l'épisode en cours correspondant à cette signature est élaboré, et les traces disponibles sont exploitées pour rechercher des épisodes passés similaires candidats ; une proposition de poursuite d'apprentissage est construite sur la base d'un épisode passé sélectionné, adapté et réapproprié par l'apprenant ; la trace en cours inscrit « naturellement » ce nouvel épisode dans la base de traçage

<sup>10</sup> Mesures de similarité : elles sont construites à partir « d'explications » particulières liées à la signature de tâche expliquée permettant de donner une importance plus ou moins grande aux éléments. Ces explications doivent pouvoir être construites par l'utilisateur.

L'adaptation se fonde sur l'ensemble des épisodes ayant le potentiel le plus important pour proposer une fin d'épisode alternative à l'épisode courant. Une liste ordonnée des essais potentiellement les plus intéressants de ces épisodes est construite. Si plusieurs épisodes ont la même notion ciblée, l'essai au plus fort potentiel est gardé.



**Figure 7.** Chemin d'expérience proposé à l'apprenti (figure de gauche) et trace telle qu'elle s'affiche à l'apprenti pendant son travail (figure de droite). Les deux graphiques sont navigables.

## 5 Discussion

L'usage de traces d'inter-actions pour faciliter l'appropriation d'un environnement informatique s'applique à de nombreuses activités ou situations comme la conception [11, 12] les tâches collaboratives [13], l'apprentissage humain associé à un Environnement Informatique pour l'Apprentissage Humain (EIAH), le partage et la réutilisation d'ontologies [14], *etc.* Dans la mesure où il est possible de générer les traces à partir d'une démarche explicite de modélisation de l'utilisation, l'approche MUNETTE facilite également le développement d'ateliers pour l'analyse de situations d'utilisations d'environnements techniques instrumentés [15] et différents projets nationaux ou régionaux exploitent cette approche actuellement (acteurs des TICE, pratiques de personnes âgées, traces et EIAH).

Deux orientations complémentaires caractérisent notre recherche actuelle : expérimentation sur des terrains variés en développant des ateliers d'analyse et des facilitateurs ; formalisation de l'approche en fondant plus précisément la notion de modèle d'observation (nous faisons intervenir les notions de modèle de collecte, de source de traçage, *etc.*) et le statut des modèles d'utilisations qui se déclinent à partir d'un modèle d'utilisation initial.

De plus, nous considérons la possibilité de concevoir un « système d'exploitation » informatique incluant l'objet trace comme objet natif, disponible pour les concepteurs et les utilisateurs au même titre que l'objet fichier par exemple.

Enfin, nous nous intéressons aux résonances de cette approche réflexive de l'interaction avec un environnement informatique avec les principes d'ontogenèse étudiés en philosophie et en psychologie.

1. Theureau, J. and G. Filippi, *Cours d'action et conception d'une situation d'aide à la coordination: le cas de la régulation du trafic du RER*. Sociologie du Travail, 1994. XXXVI(4/94): p. 547--562.
2. Champin, P.-A., A. Mille, and Y. Prié, *MUNETTE: Modelling USEs and Tasks for Tracing Experience*, in *Workshop From structured cases to unstructured*

- problem solving episodes - WS 5 of ICCBR'03, Trondheim (NO)*, B. Fuchs and A. Mille, Editors. 2003, NTNU.
3. Laflaquière, J., et al. *Approche de modélisation de l'expérience: utilisation de systèmes complexes pour l'assistance aux tâches de veille informatiquement médiées*. in *ISKO-France 2005*. 2005. Vandoeuvre les Nancy, France: Presses Universitaires de Nancy.
  4. Laflaquière, J. and Y. Prié, *Modélisation d'utilisation de système d'assistance à base de trace: une application de Musette à la tâche de veille documentaire*, in *Actes de l'atelier Traces, Interactions, Co-constructions collectives et relations à la Cognition, AS CoMETE, CNRS STIC RTP38, Paris, 19 décembre 2003*. 2003.
  5. Laflaquière, J. and A. Ciaccia. *Facilitation de tâches informatiquement médiées: une approche centrée sur la réflexivité de l'utilisation*. in *6ème colloque des jeunes chercheurs en sciences cognitives*. 2005. Bordeaux.
  6. Héraud, J.-M., A. Mille, and J.-M. Jolion, *Les réseaux notionnels: un outil pour guider la navigation dans un cours hypermédia*, in *Proceedings 3ème Colloque Jeunes Chercheurs en Sciences Cognitives*. 1999: Soulac, France. p. 116-121.
  7. Héraud, J.-M., *PIXED: une approche collaborative de l'expérience et l'expertise pour guider l'adaptation hypermédia en enseignement à distance*, in *Informatique*. 2002, Université Lyon1: Villeurbanne.
  8. Héraud, J.-M., *Pixed: improving adaptable navigation with Case-Based Reasoning*, in *EDMEDIA'2004 16th World Conference on Educational Multimedia Hypermedia and Telecommunications*. 2004: Lugano. p. 656--661.
  9. Héraud, J.-M., L. France, and A. Mille. *Pixed: An ITS that guides students with the help of learners' interaction log"*. in *Workshop Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes*. 2004. ITS 2004, Maceio.
  10. Héraud, J.-M. and A. Mille, *Pixed: assister l'apprentissage à distance par la réutilisation de l'expérience*, in *Proceedings de l'Atelier Raisonnement à Partir de Cas, Plateforme AFIA'03 Laval*, L. Jean, Editor. 2003.
  11. Champin, P.-A., *Ardeco: an assistant for experience reuse in Computer Aided Design*, in *From Structured Cases to Unstructured Problem Solving Episodes for Experience-Based Assistance*, A. Mille and B. Fuchs, Editors. 2003: Trondheim, Norway.
  12. Philippon, M., G. Caplat, and A. Mille. *Aide à l'utilisateur: savoir quand intervenir*. in *IHM'05: 17ème conférence francophone sur l'interaction homme-machine*. 2005. Toulouse, France.
  13. Stuber, A., S. Hassas, and A. Mille. *Languages games for meaning negotiating between human and computer agents*. in *ESAW 2005: Engineering Societies in the Agents' World*. 2005. Kusadasi, Aydin TURKEY: Springer Verlag.
  14. Arana-Lozano, J., Y. Prié, and S. Hassas, *MAZETTE: Multi Agent MUSERTE for Sharing and Reusing Ontologies*, in *WOSE: Workshop on Ontologies, Semantics and E-Learning, OTM Workshops*. 2004, Springer. p. 741--752.
  15. Georgeon, O., et al., *Driver behaviour modelling and cognitive tools development in order to assess driver situation awareness*. 2005.

