Approche de modélisation de l'expérience d'utilisation de systèmes complexes pour l'assistance aux tâches de veille informatiquement médiées

Laflaquière Julien - Champin Pierre Antoine - Prié Yannick - Mille Alain

Laboratoire d'InfoRmatique en Images et Systèmes d'information FRE 2672
CNRS/INSA de Lyon/Université Claude Bernard Lyon 1/Université Lumière Lyon 2/Ecole Centrale de Lyon

LIRIS -Bât Nautibus - UFR Informatique
Université Claude Bernard Lyon 1
F-69622 Villeurbanne Cedex

prénom.nom@liris.cnrs.fr
http://liris.cnrs.fr/prenom.nom

Cadre de la conférence

L'une des évolutions les plus marquantes de ces dernières années dans les domaines de recherche en système d'information porte sur l'orientation de l'utilisation de ces systèmes. Il ne s'agit plus simplement de modéliser les informations en fonction de leurs contenus mais en fonction des utilisations qui en seront faites. Il faut donc prendre en compte à la fois les caractéristiques liées aux problèmes à résoudre par les informations ainsi que les particularités des utilisateurs de ces systèmes. Les contextes d'utilisation s'élargissent, des différents types de veille (technologique, environnemental, concurrentiel, juridique, etc.) à la veille stratégique et à l'intelligence économique.

Thème de la conférence abordé :

La représentation orientée utilisation des informations (le représentation de l'information selon les usages potentiels).

Résumé

Dans le cadre de tâches informatiquement médiées de haut niveau, comme la veille, la modélisation des informations selon leur *utilisation* potentielle pose nécessairement la question de la modélisation de l'utilisation des *outils* qui en permettent l'exploitation, appelant le renouvellement d'approches de modélisation classiques, pour prendre en compte spécifiquement les *tâches* et le *contexte* de l'utilisateur. Cet article présente une approche générale de modélisation de l'expérience d'utilisation d'un système informatique, avec pour objectif de réutiliser cette expérience en contexte pour assister l'utilisateur à effectuer sa tâche. L'approche de modélisation présentée, dénommée MUSETTE (Modéliser les USages Et les Tâches pour Tracer l'Expérience) se base sur la capture d'une trace d'utilisation conforme à un modèle d'utilisation général, lequel décrit les objets et les relations manipulés par l'utilisateur du système informatique visé. Cette trace primitive peut être considérée comme un conteneur de connaissances par rapport à la tâche, qui peut être analysée *a posteriori* à l'aide de signatures de tâche expliquées permettant d'y localiser des épisodes signifiants. Ces épisodes pourront alors être réutilisés par des assistants logiciels comme connaissances contextualisées, afin de soutenir l'utilisateur dans sa tâche. Les avantages de notre approche seront présentés au travers de quatre scenarii d'assistance à l'utilisateur qui nous permettront également de présenter des exemples tournés vers la tâche de veille informatiquement médiée, en s'intéressant en particulier à l'utilisation de Human-Links, un outil cartographique dédié à la veille sur le Web.

Mots-clefs: approche de modélisation, processus de veille, outils informatiques de veille, assistance à l'utilisateur, assistance à la tâche, expérience d'utilisation, réutilisation d'expérience, trace d'utilisation.

1 Introduction

1.1 Veille sur le Web : tâche complexe

L'immense ressource documentaire, que représente depuis quelques années le Web, le place au cœur de la veille, quelque soit son type, technologique, environnementale, stratégique, juridique, etc. Si le Web ne modifie pas le principe fondamental de la veille, qu'on peut considérer, de façon classique, comme un *processus* divisé en plusieurs *étapes*, allant de l'*identification du problème décisionnel* aux *décisions et applications au problème identifié*, il en actualise, en revanche, radicalement la mise en œuvre[1]. Il suffit pour s'en convaincre de se pencher par exemple, sur l'étape de *transformation du problème décisionnel en problèmes de recherche d'information*, complexe par nature, qui exige aujourd'hui du veilleur la prise en compte, l'anticipation, de nouvelles possibilités de recherche d'information, liées au Web et à ses outils. Cette étape appelle ainsi la mobilisation de connaissances et de savoir-faire dans un domaine dont les pratiques, complexes, instables et en évolution permanente, sont encore l'objet de nombreux travaux de recherche [Mey04, Pir02].

Cette remarque nous permet de rebondir pour souligner que finalement, le succès d'un processus de veille, repose encore et toujours sur le veilleur dont le travail ne semble pas avoir été nécessairement facilité par ces changements technologiques (Internet, logiciels dédiés, automatisation, etc.). La difficulté n'étant plus d'accéder à de l'information, elle est transposée sur des problèmes de plus haut niveau, aux contours plus abstraits, comme c'est le cas dans la question de la *validation* et de l'*authentification* des sources d'information[2].

C'est dans ce contexte qu'est aujourd'hui proposée une kyrielle de logiciels dédiés, « solutions et outils de veille » en tout genre, représentant un panel très large d'outils [Laf03]. Dans leur grande majorité, quoi qu'en laisse penser leur label, ces outils ne couvrent en réalité qu'une partie du processus et ne vise (même pour les plus complets d'entre eux) généralement que deux étapes : la recherche et la *collecte d'informations* d'une part, et le *traitement, l'analyse et la présentation des informations collectées* d'autre part. Ces outils, s'ils ne soutiennent pas le processus complet, sont déjà d'une grande complexité. Pour généraliser, nous considérons ces logiciels avant tout comme des SI documentaires dédiés dont l'utilisation s'intègre à la gestion d'un « espace documentaire numérique » (EDN[3]). Leur prise en main est un premier challenge, et l'exploitation optimale de leurs capacités reste souvent un objectif plus qu'une réalité.

La complexité de l'exploitation de ces outils, et à travers eux celle des informations qu'ils permettent de manipuler, explique la tendance à la modélisation de l'*utilisation* de ces informations, pour en enrichir et faciliter l'exploitation. Cette même complexité justifie notre volonté d'aborder le problème en terme d'*assistance* à l'utilisateur, s'appuyant sur une modélisation de l'*utilisation* de ces outils en contexte, tournée vers l'appropriation de l'outil, qui reste incontournable, mais également vers la tâche elle-même. Nous proposons dans la section suivante, de faire un point rapide sur la question de l'assistance à l'utilisateur (dans le cadre général de tâches complexes non forcément liées directement à la veille), et d'introduire par des exemples notre propre point de vue.

---

[1] « *Avant Internet nous passions 80% du temps à chercher et 20% à analyser l'information. Désormais, l'objectif est quasiment d'arriver au ratio inverse* » [Cha03].

[2] Où aux connaissances du domaine viennent s'ajouter de nécessaires connaissances des outils manipulés.

[3] Que l'on peut définir comme étant composé d'une part, de l'ensemble des documents directement accessibles à un utilisateur, et d'autre part de l'ensemble des outils et moyens d'aide à l'exploitation de ces documents.

1.2 Assister l'utilisateur dans sa tâche

Les outils informatiques en général et les SI en particulier, possèdent des systèmes d'aide à l'utilisateur, aujourd'hui connus sous diverses formes, regroupés dans des classifications qui distinguent (entre autres) les « conseillers » des « assistants », les systèmes « conversant » des systèmes « autonomes », ou encore les systèmes « *ad hoc* » des systèmes « évolutifs » [Egy03]. Quelle que soit leur forme, on peut distinguer deux types d'objectifs dont le premier, vise l'*appropriation* de l'outil. Nous entendons par ce terme, l'acquisition par l'utilisateur des connaissances relatives aux fonctionnalités du logiciel, indépendamment de la tâche effectuée[4]. Pour le second, l'enjeu est l'*assistance à la tâche* qui elle est au contraire intimement liée à la tâche effectuée et à son contexte. Si l'appropriation de l'outil reste (et restera) un objectif important, il nous semble que le véritable enjeu aujourd'hui est bien, dans le contexte de l'utilisation de logiciels complexes (logiciels de veille entre autres), l'assistance à la *tâche*, i.e. la création de systèmes capables d'être un support à la réalisation de la tâche plus qu'un guide prescriptif à l'utilisation de l'outil.

Le spectre des tâches informatiquement médiées s'élargissant, tant en diversité qu'en complexité, il devient crucial que l'assistance prenne en compte les particularités des tâches de l'utilisateur, et le *contexte* d'utilisation des outils exploités pour les mener à bien [Cap02]. Ainsi voit-on se développer le besoin d'agents assistants qui seraient capables de tirer avantage de connaissances décrivant la tâche elle-même, pour interpréter, dans leur(s) contexte(s), les traces laissées par l'utilisation des outils informatiques. Notons que la notion de « contexte » n'a ici rien à voir avec ce qui est communément appelé « *contextual help* », qui ne considère que les caractéristiques de l'état d'un système à un instant donné, en se focalisant particulièrement sur l'interface (i.e. les éléments sélectionnés, les menus courants, etc.). Nous parlons ici de porter notre attention sur le contexte *de l'utilisateur*, dans la réalisation de la tâche qu'il est en train de mener.

Dans cette problématique de l'assistance à la tâche, nous distinguons plus précisément deux types de tâches. Premièrement, les tâches bien définies, pour lesquelles l'assistance devrait s'appuyer sur des connaissances décrites minutieusement dans une « ontologie de développements de systèmes ». Une grande part de l'effort du Web Sémantique [Han03] est dédiée à la modélisation de ce genre de connaissances, à leur représentation dans des documents Web avec des langages basés sur XML, à leur extraction plus ou moins automatique depuis des textes, ou à leur obtention grâce à des annotations utilisateur. Un triplet RDF peut indiquer, par exemple, qu'un fragment particulier d'une page Web est une adresse postale, comme définit dans une « ontologie du travail de bureau » (au sens large), permettant sa gestion automatique par un assistant capable de la rajouter au carnet d'adresse[5]. L'assistant pourrait très bien ajouter cette nouvelle connaissance au carnet d'adresse sans tenir compte de ce que l'utilisateur est en train de faire à ce moment là.

Deuxièmement, il s'agit des tâches difficiles à anticiper, devant être reconnues sur la base de leur manifestation et définies à la volée, en considérant l'expérience courante d'utilisation du système. Dans notre exemple, l'utilisateur pourrait exploiter dans son travail le fragment contenant l'adresse pour chercher un numéro de téléphone dans un annuaire électronique, afin d'ajouter ce dernier à son répertoire. Bien sûr, ce genre de comportement ne peut être assisté la première fois, mais nous pensons que s'il devient plus ou moins courant, des assistants devraient être capables d'en tirer profit.

---

[4] L'illustration typique de ce type d'aide est le « trombone » de Microsoft Office, à l'efficacité discutable.

[5] Dans ce genre de cas, il faut considérer bien entendu, qu'il est pertinent de « tagger » les adresses postales, afin de les utiliser pour envoyer du courrier postal dans une tâche de travail de bureau

Développer ce genre d'assistant requiert des modèles de description de tâches hautement versatiles, mais capables également de prendre en compte l'expérience courante de l'utilisateur exploitant son système. En fait, nous pensons que le défi réside dans le fait d'être capable de *tracer l'expérience* concrète (i.e. l'expérience acquise à travers les usages réels) de sorte qu'il soit possible de la ré-exploiter pour fournir des réponses utiles à des questions non complètement définies dès le départ. Notre principale question devient : comment est-il possible de modéliser et capturer l'expérience d'utilisation d'un système, de façon à ce qu'elle soit réutilisable et constitue une connaissance pour un agent assistant ? De plus, comment serait-il possible de faire évoluer, avec l'expérience concrète, cette assistance elle-même ?

L'équipe CEXAS du LIRIS, provenant du domaine du Raisonnement à Partir de Cas (RàPC), voit dans ce type de démarche, un moyen de tracer et réutiliser l'expérience et offre quelques propositions pour relever ce défi [Pri00, Cor97, Cha03b]. Cette idée a abouti à l'élaboration de MUSETTE (Modéliser le USages et les Tâches pour Tracer l'Expérience), une approche générale pour représenter l'expérience concrète, en relation avec le contexte d'utilisation d'un système informatique.

Nous souhaitons souligner, dans cet article, que la modélisation des informations selon leur(s) utilisation(s) potentielle(s), pose indirectement le problème de la modélisation de l'utilisation des outils qui en permettent l'exploitation. Cette dernière, pour dépasser la simple juxtaposition de fonctionnalités normalisées, se doit de prendre en compte les *usages réels* des outils, et tenir compte des *tâches* et du *contexte* de l'utilisateur. Cette démarche nécessite une nouvelle approche globale de modélisation, que nous présentons dans cet article, qui s'inscrit dans le cadre d'une problématique d'assistance à l'utilisateur confronté à des tâches et des outils complexes, dont le veilleur est la parfaite illustration.

Dans la section suivante, nous commencerons donc par décrire brièvement l'approche dans sa globalité. Se succèderont ensuite deux sections décrivant les notions de *modèle d'utilisation* et de *trace primitive*, puis de *signature de tâche expliquée* comme un moyen de distinguer dans cette trace des *épisodes*, comparables à des cas réutilisables. Une section sera ensuite consacrée à des scenarii illustrant plusieurs types d'assistance que permet d'envisager l'approche MUSETTE (notamment dans le cadre de l'exploitation d'un outil dédié à la veille, Human-links). La septième section exposera des travaux liés et leur discussion, enfin la conclusion s'attachera, après un rapide bilan, à évoquer nos perspectives.

2 Musette, approche globale

Dans le schéma suivant (fig. 1) est présentée la structure générale de l'approche MUSETTE, du niveau de l'observation (en haut à droite) jusqu'à celui de la ré-utilisation de l'expérience (à gauche), en passant par deux niveaux de modélisation d'expérience. Le schéma peut se lire de la façon suivante. Un *utilisateur* interagit avec un *système*, opérant des *modifications* dans son espace de travail. Un *agent observateur*, guidé par un *modèle d'observation*, génère à partir de ces modifications une *trace primitive* en 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 être soit directe (avec des agents clairement distincts du système), soit médiée (grâce à des modifications directes du système). Dans ce cas, l'interaction peut elle-même être captée par l'agent observateur et devenir un épisode réutilisable. Il est très important de souligner que MUSETTE n'est qu'une approche générale de modélisation, elle peut être implémentée grâce à divers langages ou formalismes de représentation, à même de représenter les différents composants de l'approche MUSETTE détaillés dans les sections suivantes.

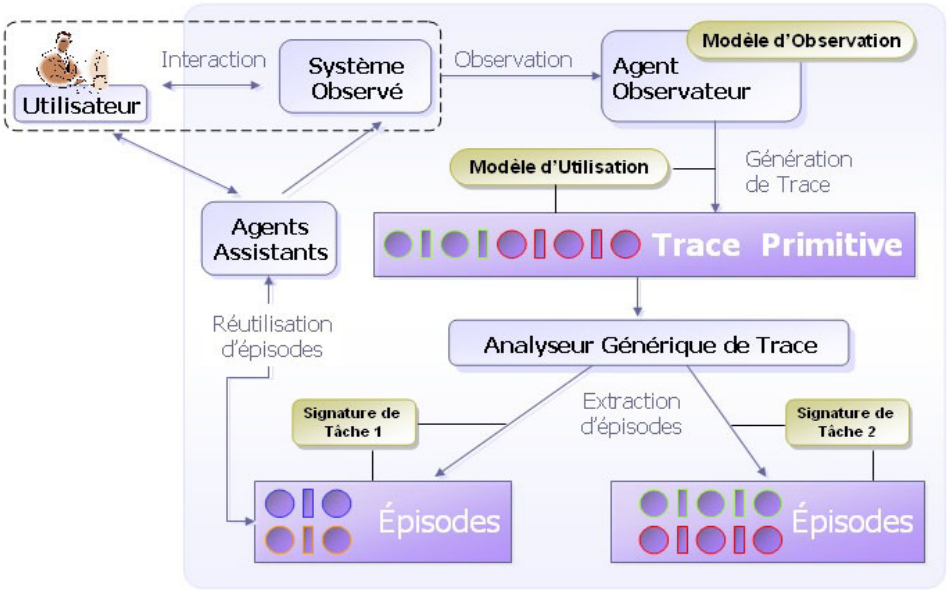

Figure 1 : Schéma introductif de l'approche et de son vocabulaire.

3 Modèle d'utilisation et trace primitive

3.1 Que devons nous observer ?

La trace primitive visée étant une représentation des interactions entre l'utilisateur et le système observé, la première étape de mise en œuvre de l'approche consiste à déterminer *de quoi* la trace sera constituée, et *comment* celle-ci sera générée. Ces deux questions trouvent leurs réponses, respectivement, dans le *modèle d'utilisation* et le *modèle d'observation*, dont la combinaison permet le fonctionnement d'un *agent observateur* en charge de la production d'une *trace primitive* (fig. 1).

Les éléments d'interaction choisis pour être constitutifs de la trace sont des *objets d'intérêt* (OI), chacun d'eux pouvant appartenir à une des trois catégories suivantes : *entités*, *évènements* et *relations*. Les entités peuvent se caractériser comme des objets « présents » pour l'utilisateur dans son interaction avec le système[6]. Les évènements, quant à eux, se caractérisent plutôt comme des objets qui « *ont lieu* », qui « se passent », durant cette même interaction. Enfin les *relations*, sont binaires et peuvent lier indifféremment des entités et/ou des évènements. Tous ces objets, qui peuvent posséder des attributs, doivent impérativement *faire sens*, aussi bien pour l'utilisateur que pour le système.

Le modèle d'utilisation d'un système particulier décrit quels sont concrètement les entités, les évènements et les relations qui seront observés pour construire la trace primitive. Le modèle d'utilisation intègre aussi d'éventuelles contraintes supplémentaires, dont celles concernant la structure interne des objets d'intérêt, qui restent dépendantes de l'expressivité et des possibilités offertes par le langage utilisé pour les exprimer (non détaillées ici). Pour simplifier, il est possible de voir un modèle d'utilisation comme une « ontologie de l'observation », décrivant les entités, les évènements, et leurs relations, à laquelle on adjoint des contraintes supplémentaires[7].

---

[6] Et *ne* correspondant *pas nécessairement* avec des « objets » du système (de son interface en particulier).
[7] Décrites directement, ou pas, dans l'ontologie et qui restent dépendantes de l'expressivité du langage employé.

Il faut souligner le fait que la détermination de ces objets d'intérêt, comme le terme « intérêt » le laisse sous-entendre, dépend complètement du *point de vue*, aussi général soit-il, adopté relativement au modèle d'utilisation. En d'autres termes, la création du modèle d'utilisation qui détermine ce qui, dans l'interaction utilisateur-système sera observé, est inévitablement biaisée par les buts que se fixe le modélisateur (l'assistance à la tâche de l'utilisateur par exemple).

3.2 Comment devons nous observer ?

Savoir de quels objets d'intérêt la trace se compose n'est bien sûr pas suffisant pour construire un agent observateur, encore faut-il déterminer les moyens de les observer. Le *modèle d'observation* décrit ainsi les règles nécessaires à la détermination des données pertinentes issues du système, ainsi que celles permettant la construction effective de la trace primitive. Contrairement au modèle d'utilisation, le modèle d'observation n'est pas, pour le moment, spécifié dans l'approche MUSETTE. En effet, produire une trace conformément à un modèle d'utilisation donné, par l'observation des interactions utilisateur-système, dépend grandement du système lui-même, sur lequel (dans le cadre général) nous ne faisons aucune supposition. On peut espérer néanmoins qu'à terme, les applications seront suffisamment auto-explicatives pour qu'on puisse mettre en place un observateur générique contrôlé par un ensemble de règles formelles d'observation. Pour le moment la seule solution est de coder un observateur *ad hoc* pour chaque système étudié, lequel génère une première trace – essentiellement événementielle – appelée *trace brute* à partir de laquelle sera générée la trace primitive. Le modèle d'observation est alors finalement codé « en dur » pour chaque observateur.

3.3 Structure de la trace

Un agent observateur étant défini par un modèle d'utilisation et un modèle d'observation (éventuellement codé en dur), une *trace* peut être produite à partir de l'observation. Cette trace n'est pas un flot ininterrompu d'entités, d'évènements et de relations, elle se subdivise en deux types de structures distinctes : les *états,* regroupant des entités, et les *transitions* regroupant des évènements. Les premiers représentent l'état du système observé (différent d'un « état-machine ») à un instant donné (ou période déterminée), par rapport au modèle d'utilisation. Une transition regroupe quant à elle un ensemble d'évènements, qui se produisent entre deux états. La trace est donc *in fine* une séquence alternée d'états et de transitions. Le modèle d'observation définit certains évènements « machine » (ou combinaison de ces évènements, qui peuvent correspondre (ou pas) à un évènement du modèle d'utilisation), comme des déclencheurs de changements d'état[8].

On remarquera que cette structuration peut faire perdre leur rôle temporel aux entités et aux évènements, qui sont regroupés sans ordre temporel. En fait cela importe peu, car aucune relation causale n'est définie dans le modèle ; en particulier les entités d'un état associé aux évènements de la transition à suivre, *n'impliquent pas* l'état suivant[9].

3.4 Exemple

La figure 2, ci-dessous, donne un exemple simplifié d'un modèle d'utilisation de navigateur Web, associé à un fragment de trace primitive, conforme au modèle. Les entités du modèle d'utilisation sont les pages Web (Page), les Hyperliens (Lien), les images (Img) et les

[8] C'est-à-dire la création d'une transition puis d'un nouvel état dans la trace.

[9] Bien entendu, un modèle d'utilisation particulier peut très bien définir cette sémantique pour certains types d'entités, d'évènements et relations que lui-même définit (cf. ci-dessous remarque sur la transition n°6, fig. 2).

préférences (Pref). Les évènements du modèle d'utilisation sont les clics de l'utilisateur (Clic), ses pages sauvegardées localement (Sauv), ses *bookmarks* sur une page (Bm) et le changement de langue (Lang). Dans la suite de cet exemple, nous considèrerons que les pages possèdent également un attribut URL contenant leur adresse.

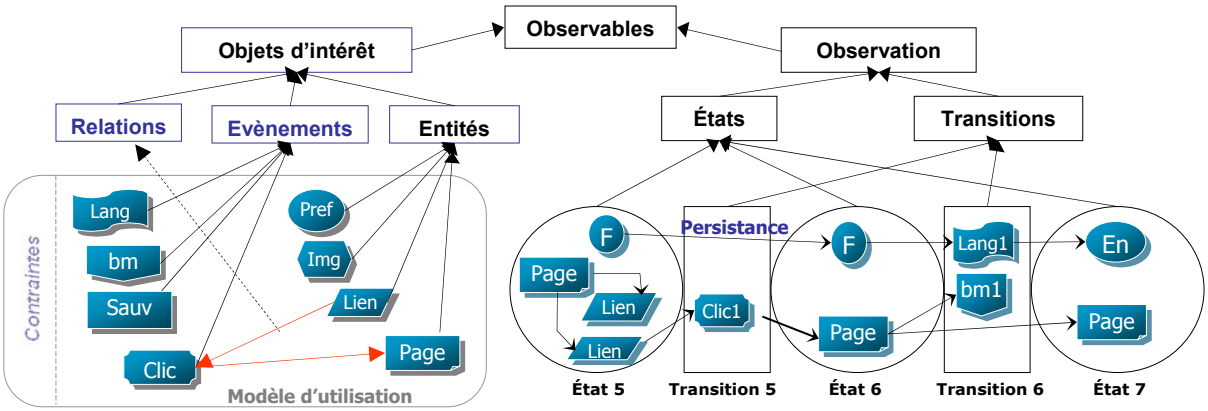

Figure 2 : Un modèle d'utilisation simplifié un fragment de trace de navigation**.**

Pour la lisibilité de la figure, toutes les relations possibles (dans le modèle d'utilisation) 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. Le fait qu'aucun changement n'intervienne entre eux vient du fait que le modèle d'observation correspondant (non décrit dans cet article) spécifie qu'un nouvel état n'est construit que lorsqu'une page est rafraîchie (ou chargée). 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.

4 *Extasi* et épisodes

Si l'on considère que le modèle d'utilisation donne une bonne description des interactions utilisateur-système, l'expérience courante de l'utilisateur doit être récupérable à partir de la trace primitive. Plus précisément, on appelle *épisode* toute partie de la trace correspondant à une expérience de réalisation d'une tâche spécifique, et qui peut être réutilisée en situation similaire. Ces « morceaux » de trace sont repérés et liés à une tâche particulière grâce à des *signatures de tâche expliquées* (*EXplained TAsk Signature* ou *Extasi*). Pour localiser dans la trace des épisodes significatifs, on considère qu'ils doivent impliquer les mêmes entités et/ou évènements, produits à un même moment ou dans un ordre particulier (en fonction des états et transitions). Pour le moment, ces points communs sont exprimés 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, dépendantes du langage, sur la structure interne des objets d'intérêt.

Quand ces points communs ont été identifiés pour une tâche particulière ils sont considérés comme une *signature* de cette tâche. En effet, leur instanciation dans la trace peut être interprétée comme un indice de sa réalisation, dans la partie de la trace correspondante. Les épisodes ne sont pas (nécessairement) limités au segment de trace instanciant la signature : l'identification de la réalisation d'une tâche permet surtout l'amélioration de l'interprétation de la tâche elle-même. Les rôles joués par les objets d'intérêt dans la trace peuvent être compris plus facilement ; des relations supplémentaires, non posées lors de l'observation et non déterminées dans le modèle d'utilisation peuvent être inférées, etc. Ainsi, un épisode peut-être annoté, ou *expliqué*, par un certain nombre d'informations venant du fait qu'on y a reconnu l'occurrence de la réalisation d'une tâche donnée. Ces explications peuvent prendre la forme d'annotations libres (accessibles à l'utilisateur) ou formelles (interprétables par un agent automatique). Un *analyseur générique* peut ainsi extraire de la trace primitive des épisodes instanciant la signature de l'*Extasi*, i.e. des épisodes dans lesquels on aura reconnu une signature de tâche puis poser les explicitations. L'efficacité de cette méthode repose sur un compromis dépendant du modèle d'utilisation : si celui-ci est trop général, des tâches différentes ne seront pas différenciées, et les épisodes, trop vagues ne pourront être correctement ré-utilisés. Si celui-ci est trop spécifique, certaines tâches lui échapperont.

4.1 Exemples d'*Extasi*

Nous présentons ici deux tâches simples de navigation Web pouvant 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 (i.e. charge une page dont l'Url est le préfixe de la première) et y pose un *bookmark* (Bm), (notons que les flèches en pointillés représentent des contraintes dépendant du langage quant aux attributs Url). Des explications sont également fournies dans cette *Extasi* : 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.

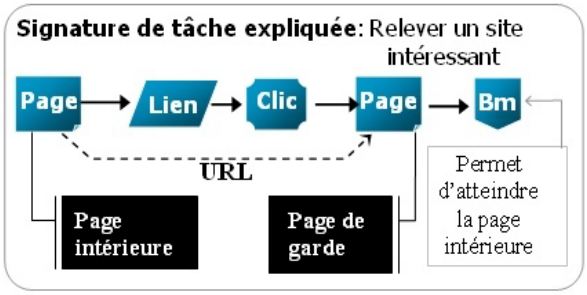

Figure 3 : *Extasi* et épisode (exemple 1)

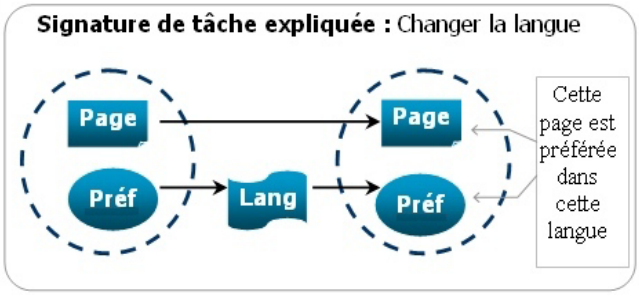

Figure 4 : *Extasi* et épisode (exemple 2).

Une autre tâche consiste à modifier les préférences de langue pour visualiser différentes versions (selon la langue) de pages Web. Notre modèle permet de détecter ce genre de tâche, le changement étant représenté par un évènement Lang (lorsque la page change sans avoir clique sur un lien). L'*Extasi* de la figure 4, possède une signature de ce type, les cercles en pointillés représentant une contrainte de co-occurrence des objets d'intérêt Page et Pref (représentant le langage courant de l'utilisation). Dans cet exemple également des annotations textuelles donnent des explications quant aux objets d'intérêt concernés.

Après avoir détaillé l'approche MUSETTE de façon générique, il nous faut préciser que sa mise en œuvre est éprouvée sur un certain nombre d'applications. L'objet d'une de ces mises en œuvre était la modélisation de l'utilisation d'un logiciel de veille sur le Web (Human-Links), pour l'assistance à la tâche [Laf03]. Afin de développer des exemples relatifs au domaine de la veille, nous présenterons brièvement l'outil en question et ses spécificités, puis nous nous inspirerons de ce travail pour donner quelques exemples de scénarii.

5 Human-Links, outil de veille Web

Human-Links est un logiciel de cartographie dynamique d'information, commercialisé par Social-Computing[10], permettant la gestion d'espaces documentaires textuels (appelés « bases de connaissances »), adapté à la veille sur le Web. Ses objectifs affichés sont la visualisation, l'organisation, la recherche et le partage d'informations. La carte « thématique » de son interface (fig. 4) permet de visualiser et de manipuler un ensemble de documents appartenant à des catégories thématiques dans un espace où l'écartement entre les items correspond à une distance sémantique calculée[11]. Les résultats des requêtes qui peuvent être lancées sur divers moteurs[12] depuis Human-Links, sont représentés sur la carte, permettant de les situer « thématiquement » par rapport aux catégories et autres documents de l'espace de travail.

Human-links, permettant la gestion quasi-complète d'un espace documentaire de travail est un outil qui correspond aux besoins du processus de veille. Il prend en charge une partie de la *recherche* d'information et de sa *collecte* (méta-moteur et recherche distribuée[13]). Les fonctionnalités de sa cartographie thématique soutiennent le traitement et l'*analyse* visuelle rapide de documents, et d'ensembles de documents, en structurant l'espace de travail. En plus d'une production de rapports de synthèse assistée, la carte elle-même peut jouer un rôle de *présentation* des informations collectées. Dans la mise en œuvre de l'approche MUSETTE sur la modélisation de l'utilisation de ce logiciel, le système observé retenu comporte quatre types d'outils logiciel (éditeur de texte, navigateur, OS-gestion de fichiers et outils de communication) ainsi que Human-Links lui-même.

Figure 5 : *Interface de Human-Links*

[10] Voir www.social-computing.com.
[11] Évolution de la méthode *Term Frequency, Inverse Document Frequency (tf-idf)*.
[12] Ce peut être aussi une recherche sur une BdD en ligne ou sur l'espace de travail d'un autre utilisateur.
[13] Human-Links déploie en outre une architecture *PeerToPeer* sur laquelle s'appuient la recherche et le partage d'informations. Chaque utilisateur de Human-Links représente un « contact » pour les autres.

Un modèle d'utilisation a été créé, définissant 35 entités, 42 évènements et plusieurs dizaines de relations. L'utilisation du système observé, et en particulier de Human-Links et son interface cartographique offre une grande diversité d'interactions observables. L'intérêt n'est pas de détailler ici les modalités techniques exigées par la mise en place des modèles d'utilisation et d'observation. Il est plus intéressant de donner un aperçu des situations potentielles d'assistance aux tâches complexes, impliquant l'utilisation de Human-Links, dans lesquelles notre approche offre des solutions originales. C'est ce que s'attachent à exposer les scenarii de la section suivante.

6 Scenarii

Cette section présente quatre types de situations et d'assistants différents concevables sur l'approche MUSETTE. Le travail de modélisation effectué sur l'utilisation de Human-Links dans le cadre de la veille, constitue une source d'inspiration pour plusieurs exemples présentés ici.

Agent Assistant spécifique - Dans notre premier scénario, un agent assistant spécifique peut prendre en compte une tâche donnée et réutilisant l'*Extasi* correspondante ; prenons comme exemple simple, l'*Extasi* de la figure 3. Nous pouvons construire un assistant qui signalerait à un utilisateur qu'il parcourt une page dans un site intéressant (i.e. dès qu'il est possible pour l'agent de trouver une *Extasi* « *Bookmark* d'un site » dont le site est le site courant).

En utilisant le même principe simple, d'un assistant spécifique tenant compte d'une tâche particulière, nous disposons en fait d'un moyen assez puissant de (re)contextualisation des informations par rapport à leur *utilisation*. Si nous reprenons l'activité du veilleur utilisant Human-Links par exemple, il est possible de construire un assistant capable d'indiquer qu'un document a été (ou non) consulté[14]. Un tel agent spécifique pourrait appliquer des mécanismes de RàPC[15] [Aam94] pour réutiliser des épisodes pertinents, dans un contexte donné. Il faut souligner ici un avantage de l'approche MUSETTE : une seule et même source de connaissance (la trace primitive) peut être partagée par plusieurs assistants, cherchant chacun à en extraire des épisodes différents. Ainsi, il est toujours possible de rajouter un nouvel assistant, sans avoir à modifier la source de connaissances[16].

Le second scénario implique cette fois un assistant générique, capable de prendre en charge plusieurs tâches, via leur *Extasi*, de façon systématique : en explorant la trace courante par rapport à certaines tâches (à la différence de la fonctionnalité d'historique des navigateurs Web qui ne donnent qu'une liste plate d'items), suggérant ou faisant au fur et à mesure des actions qui semblent pertinentes, à la façon d'un correcteur orthographique ou grammatical.

À la suite de l'exemple précédent, considérons cette fois l'utilisateur de Human-Links dans une phase de tri des résultats d'une requête (cf. Section 5). Un assistant serait capable de proposer par exemple tous les résultats non encore consultés, ou bien ceux qui l'ont été, ou encore ceux qui ont été rapidement écartés (consultés un temps très court) pour en proposer la suppression (sélective, ou groupée). L'avantage de MUSETTE est ici que chaque tâche particulière est réifiée et extraite par l'*Extasi* correspondante, et que les explications fournies par l'*Extasi* peuvent être utilisées aussi bien pour guider l'assistant que pour les rendre intelligibles à l'utilisateur.

---

[14] Avec la possibilité de graduer la consultation en fonction du temps d'ouverture au premier plan par exemple.
[15] Raisonnement à Partir de Cas.
[16] Il suffit d'ajouter une nouvelle *Extasi* pour extraire de nouveaux épisodes dans la trace existante.

Assistance basée sur une amorce d'épisode - Dans le troisième scénario, l'utilisateur demande spécifiquement de l'aide à l'assistant lors de son interaction avec le système. Ce dernier doit alors identifier la tâche engagée par l'utilisateur. Cette identification peut être la sélection explicite d'une *Extasi* par l'utilisateur, ou être totalement automatique et basée sur des indices présents dans la trace courante. Une solution intermédiaire pour le système est de proposer plusieurs interprétations de l'activité de l'utilisateur, en se basant sur l'instanciation partielle d'*Extasi*, celui-ci pouvant accepter ou rejeter chaque proposition. L'assistant peut alors retrouver des épisodes similaires, basés sur la même *Extasi*, pour les réutiliser.

Prenons encore une fois comme exemple l'utilisation de Human-Links, cette fois-ci dans le cadre d'une recherche d'information sur le Web et considérons la situation suivante. Une première *Extasi* appelée « Affinage d'une requête », correspond à la succession de deux lancements d'une requête dont les premiers résultats, trop nombreux, conduisent l'utilisateur à relancer la requête en y ajoutant des mots clefs supplémentaires pour filtrer les résultats. Une explication textuelle peut alors être attachée à la première page de résultats et indiquer « Trop de résultats ». Une autre *Extasi* appelée « Demander à quelqu'un » correspond à une succession de lancement d'une requête suivie de l'émission d'un e-mail, contenant tout ou partie des mots clefs de la requête, à un contact (cf. Section 5). Dans ce cas, l'explication attachée peut être « Aucun résultat pertinent ».

Considérons maintenant que l'utilisateur demande de l'aide à l'assistant après avoir lancé une requête. L'assistant reconnaît le début des deux *Extasi* que nous venons de décrire. Il peut demander à l'utilisateur de préciser sa situation en proposant une première description « Trop de résultats », mais l'utilisateur la rejette. Ce dernier accepte l'alternative, i.e. « Aucun résultat pertinent ». L'assistant peut alors chercher les épisodes correspondant à l'*Extasi* « demander à quelqu'un » et y rechercher parmi les destinataires d'e-mails ceux qui sont le plus à même d'aider l'utilisateur sur le thème actuel. Pour cela il peut profiter du fait que les contacts dans Human-Links (destinataires potentiels du mail) sont liés à des profils thématiques (basés sur des mots-clefs). Finalement l'assistant peut proposer à l'utilisateur d'envoyer un mail à un interlocuteur particulier susceptible de le renseigner. L'intérêt de MUSETTE dans ce type de scénario tient au fait que l'interrogation de la base d'expérience est réalisée simplement en *utilisant* le système (sans faire appel à un langage de requête particulier), et réduisant le risque d'une assistance hors de propos.

Assistance à une tâche non définie - Pour finir, évoquons un quatrième scénario, en reprenant le précédent et en admettant qu'aucune proposition d'*Extasi* ne satisfasse l'utilisateur. L'assistant pourrait alors lui fournir le moyen de préciser quelle tâche exacte celui-ci entreprend. Notre utilisateur peut ne vouloir par exemple ni demander à quelqu'un, ni affiner sa requête mais plutôt interroger un autre moteur de recherche, plus spécifique (dédié à un domaine, ou à langage particulier). L'utilisateur est alors en train de *décrire une nouvelle tâche*, avec l'*Extasi* correspondante - sans la spécifier complètement bien sûr, puisque l'utilisateur demande de l'aide pour l'accomplir.

Cependant si cette *Extasi* était jusqu'alors inconnue, la tâche en elle-même a pu être déjà menée à bien par le passé, et la trace peut contenir des épisodes correspondant à l'*Extasi* en question. L'utilisateur pourrait ainsi décrire une nouvelle tâche à la volée, et exiger une assistance à son sujet. Nous pouvons constater ici dans quelle mesure la séparation dans MUSETTE, entre le modèle d'utilisation général et les signatures de tâches spécifiques, influence l'assistance. Quelle que soit sa forme, l'assistance aux utilisateurs de systèmes informatiques a depuis longtemps motivé de nombreux travaux de recherche, comme s'attachera à le montrer la section suivante.

7 Travaux liés et discussion

Bien avant l'existence des ordinateurs personnels, V. Bush (1945) [Bus45] rêvait avec MEMEX, d'un appareil capable de capturer toutes les informations qu'un scientifique consulterait ou annoterait, rassemblant des informations restant disponibles pour d'autres scientifiques dans le futur. Dans les années 90, Hill et al. [Hil92] voyaient dans les interactions cumulées entre les utilisateurs et les objets numériques, une forme de dégradation, comme les pages abîmées d'un manuel. Plus récemment, Wexelblat et Maes [Wex97] proposèrent un outil permettant aux utilisateurs de visualiser les chemins empruntés par d'autres sur un site Web. Tous ces travaux se sont en fait focalisés sur ce qui pourrait être montré directement à l'utilisateur comme des « empruntes » laissées par l'utilisation, capables de répondre à des questions qu'ils n'auraient pas encore formulées. Ces travaux visaient une assistance générique.

Certains travaux se sont focalisés sur la tâche de l'utilisateur afin de contextualiser ce genre d'assistance [Far00, Rev00], tandis que d'autres ont essayé de capter des épisodes généraux de navigation sur le Web, sur la base de signatures statiques [Cor97] comme c'est le cas avec *Broadway*[17] [Jac98], qui constitue des bases de cas de chemins parcourus sur le Web par différents utilisateurs pour leur en proposer certains, qui semblent pertinents (sans que ce mécanisme soit transparent pour l'utilisateur).

L'assistance que permet d'envisager l'approche MUSETTE s'éloigne quant à elle de l'observation *passive* des actions d'un utilisateur, dont la plus simple illustration est sans doute la fonction « historique » des navigateurs Web - dont la logique fut poussée par des assistants comme *Letizia* [Lie95], un agent autonome dont l'assistance consiste à proposer à l'utilisateur des sites « pertinents » selon sa navigation. Dans le domaine de la navigation Web, l'exploitation de *logs* bruts montre d'ailleurs rapidement ses limites et ne permet au mieux que la représentation graphique, sous forme d'arbres ou de graphes, du parcours Web d'un utilisateur [Bea04]. L'approche MUSETTE se tourne explicitement vers la *tâche* de l'utilisateur, son *contexte* et l'*utilisation* de l'outil informatique qu'elle induit.

Si la tâche, et son contexte, sont bien des préoccupations centrales de l'approche MUSETTE, il ne faut pour autant pas la confondre avec une approche, plus classique, de *modélisation de tâche* [Bal94]. Là où la modélisation de tâche se donne les moyens de représenter (de façon abstraite) une *tâche*, l'approche MUSETTE modélise des objets d'intérêt, dont l'*agencement* particulier des instanciations dans la trace primitive va représenter une partie de l'activité (interaction utilisateur-machine) de l'utilisateur. Ce n'est donc pas la tâche elle-même qui est modélisée, pas même indirectement, mais bien l'*utilisation* (induite par la tâche) dudit système. Il est peut-être plus facile d'exprimer ces différences en distinguant les *signatures de tâche* de l'approche MUSETTE, des fameux *modèles de tâches* [Lac02, Bal94, Pat99].

Un modèle de tâche décrit une tâche selon les *contraintes* de l'approche de modélisation adoptée, pour CTT (ConcurTaskTrees) par exemple, toute action doit avoir un auteur spécifié appartenant à un des quatre types définis par l'approche [Pat03][18]. Une instanciation de ce modèle (un chemin dans l'arbre obtenu) représente une des façons de réaliser « correctement » une tâche. Par ailleurs, la prise en compte du contexte de la tâche se fait indirectement par l'intégration éventuelle d'autres modèles (domaine, situation, etc.). Les signatures de tâches ne représentent rien de plus qu'un motif (souple), une combinaison d'objets d'intérêt, dont l'instanciation dans la trace indique qu'elle s'y déroule. Elles ne

[17] BROwsing ADvisor reusing path WAYs.
[18] Machine, utilisateur, interaction, ou « aucun » dans le cas d'une « tâche abstraite ».

prescrivent donc pas la tâche et leur objectif n'est pas de servir de « référent » [Bal94], comme c'est souvent le cas pour les modèles de tâche, aussi bien dans le cadre de la conception de systèmes interactifs [Smi96, Pat99], en particulier de leur interface [Bar02, Win02], que dans l'évaluation des performances (utilisateur ou système) [Bal94]. Comme nous l'avons dit, les signatures de tâches, ou *Extasi* si elles sont expliquées, sont indépendantes de la trace elle-même et peuvent être ajoutées, modifiées, etc. Elles ne dépendent que du point de vue adopté par le modélisateur (en terme d'assistance souhaitée dans le cas de l'assistance).

La phase clef de l'approche MUSETTE est évidemment la création du modèle d'utilisation. Nous avons commencé à mettre en place pour cette phase, une démarche méthodologique dédiée, se rapprochant plus d'une démarche d'*Analyse de Tâche* [Kie95], mais s'inspirant également des travaux d'analyse de situation, de domaine, des interactions, etc. Nous partageons avec ces différents travaux un problème central : celui du niveau de *granularité* de la description[19] [Bla00]. Rappelons que la modélisation des objets d'intérêt, doit être effectuée à un niveau de granularité qui doit leur permettre de faire sens pour l'utilisateur (la participation de ce dernier à ce processus et sa validation est indispensable pour espérer rendre transparent les traces d'utilisation)[20]. Les méthodes de conception participatives [Mul01], constituent dans ce cas une source d'inspiration pertinente.

8 Conclusion

Cet article a mis en avant les difficultés auxquels se confrontent des utilisateurs de systèmes informatiques de plus en plus complexes (comme les veilleurs), pour justifier une remise en question des démarches d'assistance à l'utilisateur dans sa tâche. Soutenant l'idée que l'*utilisation* des systèmes, dans le cadre de la réalisation d'une tâche, doit être la problématique centrale d'une démarche d'assistance à l'utilisateur, cet article a détaillé l'approche MUSETTE. Cette approche consiste à modéliser des objets d'intérêts pour en permettre l'observation durant l'utilisation d'un système. Un agent observateur peut ainsi construire une trace primitive de laquelle sont extraits des épisodes significatifs, qui constitueront à leur tour une ressource de connaissances réutilisables en contexte pour supporter l'utilisateur dans sa tâche. Après avoir détaillé les moyens d'obtenir cette trace, nous avons illustré, grâce à quelques scenarii, différents exemples d'assistances envisageables et qui montrent comment l'utilisation d'un système, une fois modélisée, peut elle-même être source d'assistance à l'utilisateur.

Nous avons débuté dans cet article en soulignant, notamment à travers l'exemple de la veille, les difficultés de mener des tâches informatiquement médiées complexes. L'intérêt nouveau du domaine des SI, pour la modélisation des informations par rapport à leur utilisation est une réponse aux besoins de repères des utilisateurs. L'approche MUSETTE, dans le cadre d'une démarche d'assistance à l'utilisateur, partage une préoccupation similaire. En effet, la modélisation de l'*utilisation* d'un système qu'elle propose, orientée vers une tâche, permet d'observer, tracer puis réutiliser une expérience concrète d'utilisation. La mise en relation, dans la trace, d'éléments d'interaction *par* l'utilisation effective, permet de contextualiser ces éléments d'interaction dans le cours des tâches menées par l'utilisateur. De plus, ces éléments d'interaction, dont les informations de l'espace de travail font partie, se contextualisent réciproquement *par l'utilisation* dont ils ont fait ensemble l'objet.

---

[19] Tant dans le cadre d'analyse que celui du modèle d'utilisation. Pour nous le niveau d'abstraction des OIs.
[20] Condition pour espérer faire émerger de nouvelles signatures de tâches.

8.1 Perspectives

La mise en œuvre de l'approche MUSETTE est développée et diversifiée dans les travaux de l'équipe CEXAS du LIRIS. Certains travaux s'attachent par exemple à étudier les possibilités de MUSETTE appliquée à de multiples utilisateurs et au partage et à la réutilisation d'ontologies (approche multi-agents de MUSETTE, modèle MAZETTE) [Ara04]. D'autres, s'appuyant également sur l'approche MUSETTE, visent l'étude des processus métacognitifs dans le cadre de l'apprentissage (projet Clever@ [Oll04]), ou l'assistance à la réutilisation de l'expérience dans un contexte coopératif de conception [Stu03]. Enfin, une approche dénommée « MUSETTE-Analyse » vise non plus à assister directement l'utilisateur, mais à fournir des outils à des analystes d'activités telles que la conduite automobile (projet INRETS), ou bien la manipulation d'outils informatiques pour les personnes âgées (projet MNESIS). Pour faciliter la phase de prototypage de l'approche, un outil graphique a été conçu sous la forme d'un *plug in* de PROTÉGÉ[21] (fig. 6).

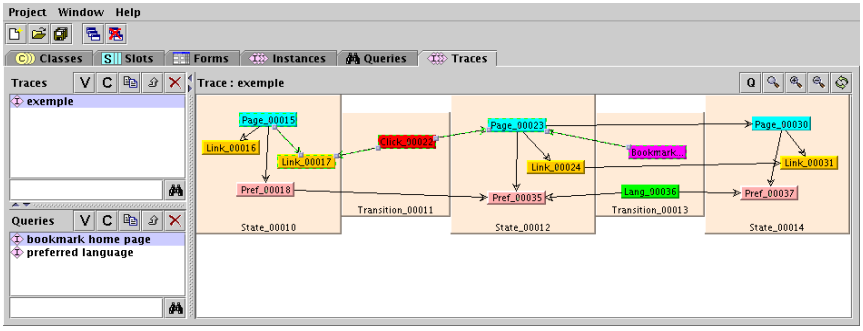

Figure 6 : Interface du *plug-in* Musette dans PROTÉGÉ.

En ce qui concerne l'assistance aux tâches de haut niveau informatiquement assistées, en particulier celles du processus de veille, il n'existe pas, à notre connaissance de démarche facilement comparable à MUSETTE. Pour notre part, nous menons des travaux d'approfondissement et de mise en œuvre de la démarche, dont nous pouvons évoquer rapidement trois dimensions : la réflexivité de l'activité basée sur des traces, l'étude des stratégies d'exploitations des outils (en particulier de veille) grâce aux traces, et enfin une dimension concernant la modélisation utilisateur.

En terme d'assistance à l'utilisateur, dans le cadre de tâche de haut niveau, la question de l'impact de la présentation à l'utilisateur de sa propre trace d'utilisation, se pose. On sait que la *réflexivité* de l'activité a des effets positifs en terme d'apprentissage [Tho02]. Il sera intéressant de voir comment la représentation de l'activité, basée sur la trace MUSETTE pourra influencer l'activité d'un utilisateur d'un système complexe comme Human-Links, tant au niveau de la manipulation de l'outil que, de façon plus globale, dans la réalisation de sa tâche.

Les *stratégies* employées par l'utilisateur, quel que soit leur niveau d'abstraction, sont justement un autre objet d'étude intéressant, dans le cadre de l'application de MUSETTE. Plusieurs travaux montrent combien il est difficile d'appréhender en détail la question des stratégies d'un utilisateur à travers l'utilisation d'un système, c'est le cas en particulier pour les stratégies de recherche d'information sur le Web [Cia04]. Il s'agira donc pour nous de voir en quoi la modélisation de l'utilisation peut constituer un outil efficace à la mise en évidence de stratégies, et à leur éventuelle réutilisation en terme d'assistance. L'étude des stratégies de recherche d'information par l'utilisation de systèmes informatiques dédiés et complexes, a

[21] Voir http://protege.stanford.edu/

une saveur toute particulière dans le cadre de la veille. En effet, l'enjeu ici est de dépasser le cadre de la recherche d'information au sens strict, et de comprendre quels effets l'utilisation de ces outils à sur l'étape de transformation du problème décisionnel en problèmes de recherche d'information. Il s'agit de comprendre comment se co-influencent ce travail de transformation (qui anticipe l'utilisation d'un outil) et les stratégies mises en œuvre pour répondre aux problèmes de recherche d'information (à travers l'utilisation réelle de l'outil). Des résultats sur ce sujet pourront apporter une contribution pertinente d'une part aux travaux liés à la recherche d'information (dans le cadre de la veille en particulier) et d'autre part pourraient amener à reconsidérer la conception des outils dédiés, ainsi que la formation de leurs utilisateurs.

Une des conséquences d'un travail sur les stratégies de l'utilisateur, concerne directement un autre domaine, celui de la *modélisation utilisateur* [Vas96]. En effet, s'il est possible grâce à l'approche MUSETTE, de mettre en évidences des stratégies à travers l'utilisation d'un système informatique, leur détection peut permettre de qualifier le niveau d'expertise sur des critères, ou des combinaisons de critères de haut niveau (stratégies) tant pour la maîtrise de la manipulation de l'outil, que pour la réalisation de la tâche elle-même. Nous terminerons enfin sur l'idée que toutes les informations relatives à l'utilisation de l'outil, de la trace elle-même, des stratégies employés tant au niveau de la manipulation de l'outil que de la tâche elle-même, ainsi que les informations liées aux assistants pourraient représenter une source précieuse dans le cadre de la reconception des outils, en particulier les outils complexes dédiés à la veille. C'est une de perspectives que nous développerons dans notre travail futur.

9 Références

[Aam94] A. Aamodt, E. Plaza. *Case-based reasoning: Foundational issues, methodological variations, and system approaches*. AI Communications, Vol.7, No.1, pages Introduction.

[Ara04] J. Arana, S. Hassas, and Y. Prié (2004). *MAZETTE: Multi Agent MUSETTE for Sharing and Reusing Ontologies*. In Workshop on Ontologies, Semantics, and E-learning, LNCS 3292, oct 04.

[Bal94] Balbo S., *Evaluation ergonomique des interfaces utilisateur : un pas vers l'automatisation*. Thèse Informatique, Grenoble, 5 septembre 1994, IMAG, université J.Fourier Grenoble 1, 295 p.

[Bar02] Mickaël Baron, *Intégration d'un modèle de tâche dans une démarche sûre de construction d'interface*. IHM 2002, conférence, Poitiers (France), 26-29 Novembre 2002.

[Bea04] Beauvisage T., *Sémantique des parcours des utilisateurs sur le Web*. Thèse Sciences du langage. Paris : Université Paris X – Nanterre, 2004, 361p.

[Bla00] Blandford A., Bryan-Kinns N., & Thimbleby H., *Interaction Modelling for Digital Libraries*. dans Workshop on Evaluation of Information Management Systems, QMW, Sept. 2000. pp. 1-10.

[Bus45] V. Bush, *As we may think*, Atlantic Monthly, 176(1), 1945, pp. 101-108.

[Cap02] A. Capobianco. *Stratégies d'aide en ligne contextuelles: acquisition d'expertises, modélisation et évaluation expérimentale*. Thèse, Université Henri Poincaré Nancy 1, 31 octobre 2002.

[Cha03] Chartrand M., *La veille stratégique, un outil de la décision et du changement*. Coup d'œil, février 2003, vol.9, n°1, pp 2-5.

[Cha03b] P-A. Champin. *Ardeco: an assistant for experience reuse in Computer Aided Design*. In WS5: From Structured Cases to Unstructured Problem Solving Episodes for Experience-Based Assistance, Workshop at ICCBR'03, Trondheim (NO). 2003.

[Cor97] F. Corvaisier, A. Mille, et J.-M. Pinon. *Information retrieval on the world wide web using a decision making system*. In RIAO 1997, pages 284–295, Jun 1997.

[Cia04] A. Ciaccia, D. Martins. *Recherche d'informations sur le Web. Étude de l'influence de facteurs lies à l'interface, à l'utilisateur et à la tâche*, RSTI-RIA, actes ARCo'04, 8-10 Nov. 2004, Compiègne, pp.159-179.

[Egy03] E. Egyed-Zsigmond, *Gestion des connaissances dans une base de documents multimédias*, Thèse en Informatique, INSA-Lyon, 2003, 217p.

[Far00] R. Farrell, P. Fairweather and E. Brumer. *A task based architecture for application aware adjuncts*, ACM International Conference on Intelligent User Interfaces, New Orleans, LA USA, pages 82-85, 2000.

[Han03] S. Handschuh, S. Staab (eds.). *Annotation in the Semantic Web,* IOS Press, Amsterdam (NL), 2003.

[Hil92] W.C. Hill, J.D. Hollan, D. Wroblewski and T. McCandless. *Edit Wear and Read Wear*. In Proc. of ACM Conference on Human Factors in Computing Systems, CHI'92, pages 3-9, New York ACM Press, 1992.

[Jac98] Jaczinski, M. and TrousseB. *WWW Assisted Browsing by Reusing Past Navigations of a Group of Users*. Advanced in Case-based Reaosning, 4th European Workshop on Case-Based Reasoning, 1998, pp 160-171.

[Kie95] D. Kieras, *Task analysis and the design of functionality*. CRC Handbook of Computer Science and Engineering, CRC Press Inc.

[Lac02] Lacaze X., P. Palanque, et al. (2002). *Analyse de performance et modèles de tâches comme support à la conception rationnelle des systèmes interactifs*. In 14eme Conférence Francophone sur l'IHM, Poitiers.

[Laf03] Laflaquière Julien, *L'expérience d'utilisation d'un espace documentaire pour améliorer l'assistance à l'exploitation de l'information dans le cadre de la veille*. Informatique, mémoire de DEA, Lyon : Université Claude Bernard Lyon 1, 2002, 31 p.

[Lie95] Lieberman Henry. *Letizia : an agent that assists Web Browsing*. Proceedings of the International Joint Conference on Artificial Intelligence, Montreal, Canada, 1995.

[Mey04] Meyer, T. and C. Rodon, (2004). *Trouver sur Internet une réponse à une question*. In Critique de la raison numérique, HERMÈS - cognition, communication, politique. N° 39, pp. 27-34.

[Mul01] Muller J. M, Tudor L.S, Dayton T., *A c.a.r.d game for participatory task analysis and redesign: macroscopic complement to pictive*, Proceedings of InterCHI 93, ACM Amsterdam 1993.

[Oll04] Ollagnier-Beldame Magali, *Projet Clever@: Reutilisation de traces et procesus metacognitifs,* communication Poster, Colloque Arco 2004, Compiègne, 8-10 Novembre 2004.

[Pat99] Paternò, F., *Model-Based Design and Evaluation of Interactive Applications*. Springer Verlag, Berlin, Novembrer 1999, ISBN 1-85233-155-0.

[Pat03] Paternò, F., *ConcurTaskTrees: an engineering notation for task models*, dans D.Diaper & A.Neville, Stanton (Eds), The hanbook of Task Analysis for Human Computer Interaction. London : lawrence Erlbaum Assosiates, pp. 483-503.

[Pir02] Pirolli, P. et al. (2002). *A user-tracing architecture for modeling interaction with the World Wide Web*, dans Advanced Visual Interfaces, AVI 2002. 2002. Trento, Italy: ACM Press.

[Pri00] E. Egyed-Zsigmond, Y. Prié, A. Mille, and J.-M. Pinon. *A graph-based audiovisual document annotation and browsing system*. In RIAO 2000, volume 2, pages 1381–1389, Apr 2000.

[Rev00] L. Francisco-Revilla and E. Breimer. *Adaptive medical information delivery combining user, task and situation models*, International Conference on Intelligent Interfaces, New Orleans, LA USA, p. 94-97, 2000.

[Smi96] Smith, M. J. & O'Neill E. J. *Beyond Task Analysis: Exploiting Task Models in Application Implementation*. CHI 96, 13-18 avril 1996, Vancouver, Canada Pages, (1996), p. 263 - 264.

[Stu03] Stuber A., Hassas S., Mille A., *Combining MAS and Experience Reuse for Assisting Collective Task Achievement.* Mc Guinty L., Ed., *Workshop Proc. ICCBR 2003*, Trondheim, Norway, June 2003, p. 314–323.

[Tho02] Mermoud-Thomassi M. (2002), *Gestion des connaissances et dynamique d'apprentissage par une reconsidération du rôle de la mémoire organisationnelle*, XIème conférence internationale ESCP-EAP, Paris, 5-7 juin 2002.

[Vas96] Vassileva, J. (1996). *A Task-Centered Approach for User Modeling in a Hypermedia Office Documentation System*. User Modeling ans User Adapted Interaction. N° 2-3.

[Wex97] A. Wexelblat and P. Maes. *Footprints: History-rich web browsing*, In Proc. Conf. Computer - Assisted Information Retrieval (RIAO), pages 75-84, 1997.

[Wink02] Winkler M., Palanque P. & al. (2002). *Une démarche structurée pour la conception et l'évaluation d'applications Web par l'exploitation synergique des modèles de tâche et de navigation*. 14eme conférence Francophone sur l'IHM, Poitiers.

[Zaj99] Zajmovic I. and Tourigny N. (1999). *Intégration du modèle conceptuel du domaine au système exécutable : vers des systèmes à base de connaissances extensibles*. IC'99, Palaiseau, Ecole polytechnique.