

Système d'aide adaptatif à base de traces

Karim **Sehaba**
Université Lumière Lyon 2
karim.sehaba@gmail.com

Compte rendu d'expérience

Résumé

L'objectif de notre travail est de développer un système d'aide adaptatif à base de traces d'interaction. Il s'agit de considérer les traces laissées par les utilisateurs comme sources de connaissances que le système peut exploiter pour générer des aides adaptées à leur utilisateur cible. Dans ce cadre, nous avons proposé des modèles permettant de représenter les traces, le profil utilisateur et les connaissances d'adaptation. Nous avons également proposé une méthode d'extraction de connaissances d'adaptation à partir d'une base de traces représentant les activités de plusieurs utilisateurs. Le domaine d'application de notre travail concerne l'aide à l'utilisation d'une plateforme d'apprentissage en ligne dans le cadre de la formation VCIel (Visualisation et conception infographiques en ligne).

Mots clés

Aide basée sur le partage d'expériences, personnalisation des aides, trace d'interaction, système adaptatif, apprentissage en ligne

Abstract

The objective of this work is to develop an adaptive help system based on interaction traces. It consists in considering the traces left by the users as knowledge sources that the system can use in order to generate adapted help to the target user. In this context, we have proposed models for representing trace, user profile and adaptation knowledge. We have also proposed a method of knowledge extraction from a trace-based management system (TBMS) representing the activity traces of several users. The application domain of this work is the help in using an e-learning platform in the context of the VCIel training.

Keywords

Help based on sharing experiences, help personalization, interaction traces, adaptive system, e-learning



©Auteur(s). Cette œuvre, disponible à http://ritpu.ca/IMG/pdf/RITPU_v09_n03_55.pdf, est mise à disposition selon les termes de la licence Creative Commons Attribution - Pas de Modification 2.5 Canada : <http://creativecommons.org/licences/by-nd/2.5/ca/deed.fr>

Introduction

Dans la plupart des environnements informatiques, l'aide aux utilisateurs est prédéfinie, par le concepteur, en fonction d'un certain nombre de situations présumées et ne tient pas compte des spécificités et de l'évolution des besoins des utilisateurs. Les concepteurs de l'aide sont ici censés avoir formalisé l'ensemble des demandes et l'ensemble des trajectoires d'usage.

Ces environnements souffrent souvent de nombreux inconvénients, car ils ne permettent pas l'évolution et l'adaptation des aides aux différentes situations rencontrées dans la pratique. De plus, la conception d'un système d'aide disposant d'une représentation complète des besoins des utilisateurs avec lesquels il interagit est très difficile, et parfois même impossible, pour le concepteur. En effet, comme le souligne Gapenne (2006), si l'exhaustivité des systèmes d'aide est idéalement souhaitable, il apparaît en même temps que la complexité et le nombre des trajectoires d'usage que certains environnements informatiques offrent rendent impossible la satisfaction de cette contrainte.

Pour remédier à ces problèmes, et avec l'évolution des nouvelles technologies de l'information et de la communication, la plupart des éditeurs de logiciels mettent à disposition des utilisateurs des outils de communication et d'échanges tels que le forum, le clavardage, etc. Ces outils permettent aux utilisateurs de s'entraider par le biais du partage d'expériences. Néanmoins, même si ces outils sont ouverts, dans la mesure où chaque utilisateur peut solliciter une aide et/ou répondre aux sollicitations des autres utilisateurs, leurs usages dépendent des motivations des utilisateurs. Également, les échanges que permettent ces outils sont généralement non contextualisés et non structurés. Ils ne permettent donc pas la construction de connaissances facilement réutilisables. En effet, les échanges entre l'utilisateur « aidé » et l'utilisateur « aidant » sont exprimés en langage naturel; il est par conséquent difficile de les traiter d'une manière automatique afin d'aider d'autres utilisateurs dans une situation

similaire à celle de l'utilisateur aidé. En outre, le plus souvent, l'utilisateur aidé ne décrit pas avec précision le contexte de sa situation de blocage, de même que l'utilisateur aidant peut ne pas préciser le contexte de sa solution. C'est pourquoi plusieurs échanges sont parfois nécessaires pour que les utilisateurs, aidant et aidé, se comprennent.

Nous proposons une approche pour une aide adaptative basée sur les traces d'interaction. Par « trace », nous entendons un objet informatique représentant l'historique des actions de l'utilisateur collectées en temps réel à partir de son interaction avec l'environnement informatique. Le principe est de considérer ces traces, représentant l'activité des utilisateurs, comme des sources de connaissances que le système peut exploiter pour faire évoluer ses stratégies d'aide et adapter ses réponses aux différentes situations rencontrées. Ainsi, il s'agit d'utiliser les expériences d'un utilisateur (u1), stockées dans des traces modélisées, pour aider un autre utilisateur (u2) en cas de blocage¹. Cela suppose que l'environnement informatique est capable d'enregistrer les actions des utilisateurs en utilisant des capteurs plus ou moins évolués allant d'un clic de souris à l'enregistrement vidéo.

Dans ce cadre, nous nous intéressons à la problématique d'adaptation des traces aux caractéristiques de leur utilisateur cible. Il s'agit de transformer la trace de l'utilisateur u1 pour qu'elle soit adaptée au profil de l'utilisateur u2. Cette adaptation est particulièrement nécessaire si les utilisateurs u1 et u2 n'ont pas les mêmes :

- Capacités physiques et cognitives. Par exemple, u1 est voyant et u2 est mal ou non-voyant;
- Compétences du domaine. Par exemple, u1 est un expert Linux et u2, un débutant ou un novice;

¹ Par blocage, nous entendons une situation où une personne a un projet d'action (une tâche à réaliser, par exemple) dont les modalités de réalisation sont ignorées (ou oubliées) et où une aide censée rendre explicite ces modalités est nécessaire.

- Préférences. Par exemple, u1 préfère manipuler des systèmes à interfaces graphiques (GUI) et u2 préfère utiliser les lignes de commande.

Dans ces conditions, une simple présentation de la trace de u1 à u2 peut ne pas être efficace pour ce dernier.

L'adaptation des traces peut porter sur le contenu de la trace, sa présentation et ses modalités d'interaction. Cet article se focalise sur l'adaptation des modalités d'interaction. Nous considérons ces dernières comme la façon de faire pour réaliser une action (définition de Martin, 1995) et nous adoptons la formalisation de Nigay (2001) qui considère la modalité comme un couple :

- Dispositif physique – Un ou plusieurs médias qui permettent l'acquisition ou la diffusion des informations;
- Langage d'interaction – Un ensemble d'expressions bien formées et significatives pour l'environnement informatique.

En effet, pour que l'utilisateur réalise une tâche donnée, il agit par des actions sur un ou plusieurs dispositifs physiques (ou médias) à l'aide d'un langage d'interaction. L'objectif est de proposer à l'utilisateur cible la modalité la plus appropriée en fonction de son profil afin de favoriser la convivialité du système. Cet article traite précisément de deux problématiques, à savoir :

- Comment adapter les traces à leur utilisateur cible. Pour cela, nous proposons des modèles et des outils permettant de transformer les traces pour qu'elles soient directement accessibles à leurs utilisateurs cibles;
- Comment extraire les connaissances d'adaptation à partir de traces. Pour cela, nous proposons des outils permettant d'analyser plusieurs traces afin d'extraire des règles d'adaptation.

L'application de notre travail concerne la formation VCIel² (Visualisation et conception infographiques en ligne). Il s'agit d'un Master 2 professionnel entièrement en ligne et à distance. Notre objectif est de développer un système d'aide à base de traces permettant aux tuteurs et aux apprenants de s'entraîner par le biais du partage de traces. L'aide porte ici sur l'utilisation de la plateforme de cours SPIRAL³.

Cet article est organisé comme suit : la section 2 présente et discute le fondement de notre approche d'aide à base de traces; la section 3 décrit la théorie de la trace sur laquelle est basée notre proposition; la section 4 présente l'architecture générale du système d'aide que nous proposons; la section 5 détaille les formalisations que nous avons proposées pour la représentation des traces, le profil de l'utilisateur et les connaissances d'adaptation; la section 6 présente le principe des méthodes d'adaptation de trace et d'extraction de connaissances d'adaptation que nous proposons; la section 7 décrit le cadre applicatif de notre travail qui concerne le développement d'un système d'aide à base de traces dans le cadre de la formation VCIel; la section 8 présente une conclusion et des perspectives.

Nécessité d'un système d'aide adaptatif

Avec l'évolution des environnements informatiques, la conception des systèmes d'aide est plus que jamais nécessaire afin de permettre aux utilisateurs grand public d'en faire usage de manière efficace, sans avoir à fournir un effort d'apprentissage spécifique. En effet, comme le montrent les statistiques de Netcraft⁴, le nombre de sites Web est en constante augmentation, et pour répondre au besoin d'un accès ubiquitaire à l'information, les utilisateurs utilisent divers dispositifs pour accéder au contenu de ces sites : assistant numérique personnel, téléphone mobile, tablette tactile, etc. Comme l'expliquent Capobianco et Carbonell (2006),

² <http://vciel.univ-lyon2.fr>

³ <http://spiralconnect.univ-lyon1.fr>

⁴ <http://news.netcraft.com>

cette richesse du contenu et des modalités d'accès entraîne une complexité d'utilisation. En outre, les logiciels grand public se multiplient et les versions successives d'un même logiciel se succèdent à un rythme soutenu. Ainsi, d'une version à la suivante, aux fonctionnalités de plus en plus nombreuses et difficiles à mettre en œuvre s'ajoute parfois un changement profond de l'interface, comme c'est le cas dans les dernières versions des logiciels de bureautique commercialisés.

Pour faire face à cette complexité, l'aide a suscité de nombreuses recherches dans différents domaines : l'ergonomie, les sciences du langage, les sciences cognitives et l'informatique. En ce sens, la notion d'aide a fait l'objet de plusieurs publications, notamment dans le numéro spécial « Systèmes d'aide : enjeux pour les technologies cognitives » de la revue *Intellectica* en 2006, ou au colloque Tildem (TICE et didactique des langues étrangères et maternelles : la problématique des aides à l'apprentissage) en 2006 (Foucher et Pothier, 2007). Malgré cet intérêt, il est apparu qu'une confusion régnait à propos de cette notion. En effet, l'usage du terme « aide » s'est généralisé, comme le souligne Gapenne (2006), à un ensemble de situations et d'outils parfois très hétérogènes : l'aide aux personnes en situation de handicap, les systèmes d'aide à la décision, la fonction d'aide dans les logiciels grand public, les aides au travail, etc. Cette confusion s'explique en partie par les diverses acceptions de cette notion selon le domaine d'étude et aussi par la proximité de la notion d'aide avec l'assistance, la substitution, la suppléance (Gapenne, 2006) ou la facilitation (voir Mille, Caplat et Philippon, 2006).

Dans Gapenne (2006), l'auteur propose de définir l'aide comme une relation spécifique entre deux agents, l'aidé et l'aidant, dont la mise en œuvre et la dynamique ont pour effets l'appropriation et l'usage d'un schème nouveau pour l'aidé et une mise à l'épreuve d'un parcours didactique pour l'aidant. Cette aide peut prendre plusieurs formes telles que les procédures (De Brito, 2006) (un ensemble d'opérations et/ou d'actions à exécuter dans le but d'atteindre un but donné), les modes d'em-

ploi (Boullier et Legrand, 1993) (un texte, sur support papier ou numérique, décrivant les éléments d'une situation et proposant des actions à réaliser) ou l'aide des logiciels (Capobianco et Carbonell, 2006).

Dans le cadre des environnements informatiques, un système d'aide est considéré comme une famille d'outils logiciels destinés à accompagner l'aidé dans l'appropriation et l'usage des objets informatiques (Gapenne, 2006). Ainsi, un tel système d'aide doit rendre explicites les modalités d'appropriation et d'usage des différents services de l'environnement informatique. Dans ce cadre, plusieurs études (voir Capobianco et Carbonell, 2006) ont démontré que les systèmes d'aide sont généralement ignorés ou rarement consultés. Selon ces études, les novices préfèrent recourir à la formation pour acquérir les connaissances nécessaires à l'utilisation des logiciels, et les utilisateurs occasionnels font appel à leurs collègues pour sortir d'une situation de blocage.

Plusieurs raisons expliquent ce constat, notamment l'écart entre le point de vue de l'utilisateur et celui du concepteur de l'aide. En effet, une des conditions déterminantes de l'efficacité des aides est la compatibilité entre deux représentations : la vision du concepteur qui les automatise sous forme de fonctions logicielles et celles des utilisateurs qui auront à utiliser ces aides. Comme le souligne Leplat (1998), l'origine de cet écart vient du fait qu'il est difficile d'anticiper de manière très précise les compétences des utilisateurs cibles, en particulier si l'aide est envisagée dans un système grand public, et par conséquent, ces compétences supposées peuvent être superflues ou sous-estimées.

En ce sens, certaines études, comme celle de Sellen et Nicol (1990), ont montré que les utilisateurs novices éprouvent des difficultés à comprendre les messages d'aide, généralement formulés dans les termes du modèle conceptuel du logiciel. Ces utilisateurs avaient des difficultés à se construire une représentation suffisamment fidèle des fonctionnalités des logiciels (Capobianco et Carbonell, 2006).

Pour combler ces lacunes, nous proposons un système d'aide à base de traces. Il s'agit de considérer les traces d'interaction laissées par les utilisateurs comme une source de connaissances que le système peut exploiter pour générer des aides adaptées aux utilisateurs cibles. Dans ce contexte, une trace est une séquence de toutes les actions de l'utilisateur pour réaliser une tâche donnée. Le principe consiste à utiliser les expériences des uns pour aider d'autres utilisateurs en situation de blocage. Cette approche a l'avantage de réduire l'écart entre la vision du concepteur, d'une part, et celle des utilisateurs et la réalité du terrain, d'autre part.

Dans ce cadre, cet article s'intéresse précisément à la problématique de l'adaptation des traces au profil de son utilisateur cible. En effet, même si on peut supposer que les traces sont à la fois disponibles et accessibles, il semble peu probable qu'elles puissent aider l'utilisateur cible par une simple visualisation. Une exploitation directe ne peut intervenir que dans une situation identique ou très proche de la situation dans laquelle la trace a été créée. Ainsi, notre objectif est de transformer les traces partagées par l'utilisateur aidant (ou source) pour qu'elles soient adaptées aux compétences, aux préférences et aux capacités de l'utilisateur aidé (ou cible).

Théorie de la trace

Le contexte scientifique de notre travail est basé sur la théorie de la trace développée par l'équipe SILEX⁵. Par définition, une trace est un ensemble d'observés temporellement situés. On appelle « observé » toute information structurée issue d'une observation. Dans notre cadre de recherche, les observés sont générés à partir de l'observation des interactions entre l'utilisateur et l'environnement informatique. Chaque observé possède un type, défini par le modèle de trace, et peut être en relation avec d'autres observés de la même trace. Formellement, un observé possède un sujet, l'utilisateur observé durant la phase de collecte, et un ensemble d'attributs/valeurs qui caractérisent l'événement observé. Le modèle de trace définit les types d'observés et les types de relations qui composent la trace. On appelle « trace modélisée », notée M-Trace, toute trace issue d'un processus de collecte, composée d'observés situés et conforme à un modèle de trace.

5 <http://liris.cnrs.fr>

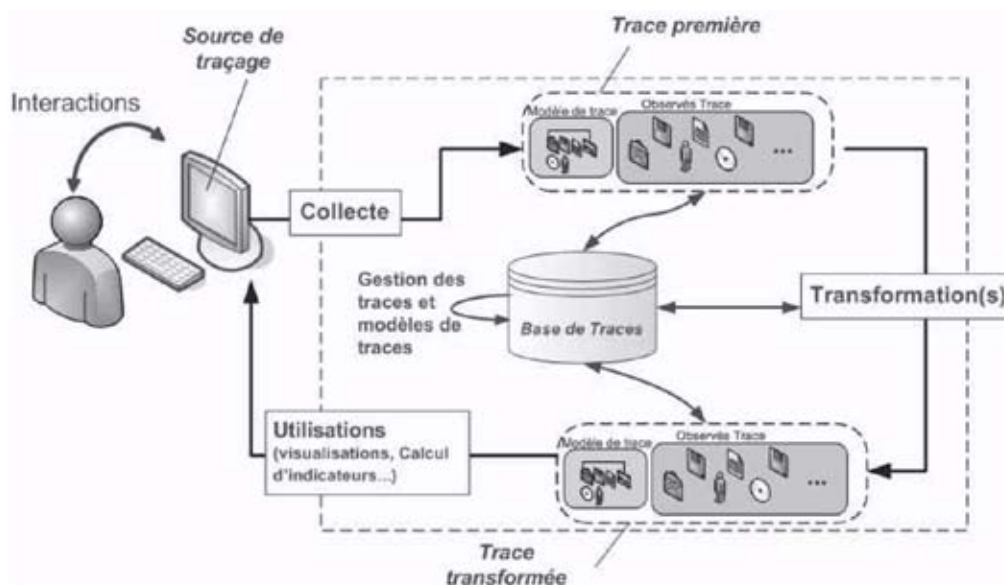


Figure 1. Système de gestion de base de traces

Afin de faciliter la gestion et l'exploitation des traces, SILEX a développé un système de gestion de base de traces (SGBT) (Lafraquière, 2009; Settouti, 2011)). Ce système permet le stockage, la transformation et la visualisation des traces modélisées. Comme le montre la figure 1, durant la phase de collecte, le SGBT récupère les événements fournis par des sources de traçage (par exemple, capteur logiciel, enregistreur de frappe, etc.) et les représente dans une trace première. Cette dernière est issue directement du système de collecte. Généralement, les traces premières sont difficilement exploitables du point de vue de leur quantité et de leur qualité. En effet, les traces premières contiennent un grand nombre d'informations parfois inutiles, et des filtrages sont nécessaires pour pouvoir les exploiter. Également, les informations de ces traces sont de très bas niveau et donc souvent difficiles à comprendre. C'est pourquoi le SGBT propose des mécanismes de transformation permettant de générer des traces de haut niveau à partir des traces premières.

Le SGBT permet également la visualisation des traces pour que l'utilisateur puisse interagir avec elles. Cette visualisation peut être en temps réel (pour l'apprentissage réflexif, tel que dans Clauzel,

Sehaba et Prié (2011) et Cram, Jouvin et Mille (2007), ou *a posteriori* (pour un bilan ou un retour d'expériences, tel que dans Champalle, Sehaba, Mille et Prié (2011)).

Architecture générale

Comme le montre la figure 2, l'architecture du système d'aide que nous proposons est composée de quatre modules : une base de traces, des connaissances d'adaptation, des profils utilisateurs et un module d'adaptation. Ce système permet de collecter les traces des utilisateurs et de les stocker dans une base de traces. Une trace est une séquence de toutes les actions de l'utilisateur pour réaliser une tâche donnée où chaque action, dans l'environnement informatique, est représentée par un observé typé.

Les traces stockées dans la base vont permettre à l'environnement informatique de disposer des connaissances qu'il peut mobiliser pour aider ses utilisateurs. Pour cela, le système d'aide doit sélectionner la ou les bonnes traces qui permettent à l'utilisateur de réaliser sa tâche en cas de blocage. La ou les traces sélectionnées doivent ensuite être adaptées à leur utilisateur cible en fonction de son profil. L'aide fait suite à une demande explicite de l'utilisateur qui perçoit une difficulté particulière.

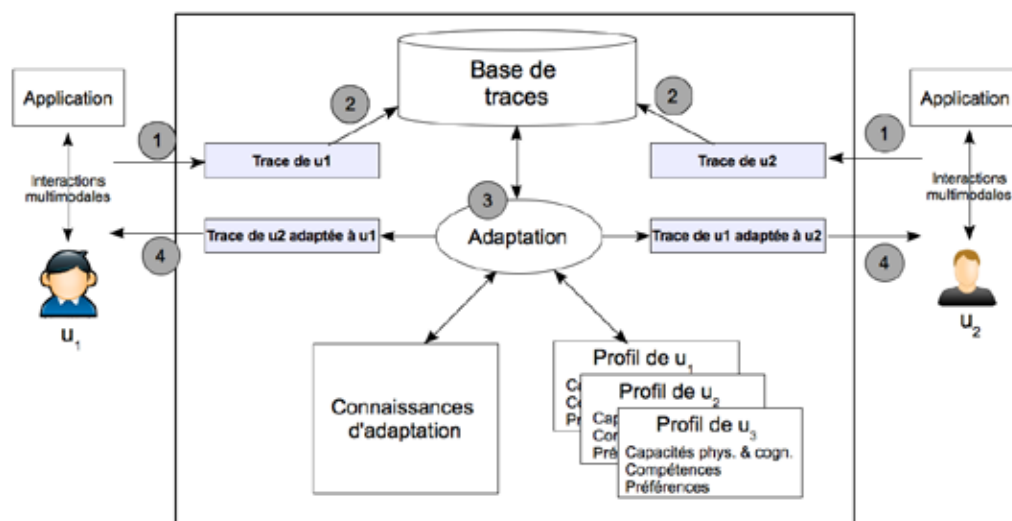


Figure 2. Architecture générale du système d'aide

Dans les prochaines sous-sections, nous présentons le principe des principales fonctions du système d'aide, à savoir la collecte, la sélection et l'adaptation des traces ainsi que l'extraction de connaissances à partir d'une base de traces. Tout au long de ces sous-sections, nous ferons référence aux différentes parties de la figure 2.

Collecte des traces

Initialement, les utilisateurs interagissent avec l'environnement informatique pour réaliser leurs tâches (1). L'environnement enregistre leurs actions, ainsi que les modalités d'interaction utilisées pour réaliser chacune de ces actions, dans des traces adoptant la formalisation présentée dans la section « Formalisations ». Ces traces sont ensuite stockées dans la base de traces (2).

Cette base sera interrogée par le système chaque fois qu'un utilisateur fera appel à l'aide pour sortir d'une impasse.

Sélection des traces

L'aide consiste à présenter à l'utilisateur la succession d'actions qu'il doit réaliser pour atteindre ses objectifs en cas de blocage. Pour cela, le système d'aide doit sélectionner la ou les bonnes traces qui lui permettent de combler ses besoins. Cette sélection peut être manuelle ou automatique.

La sélection manuelle est basée sur une recherche par mots clés. Elle consiste à chercher dans la base de traces la ou les traces dont la description contient un ou plusieurs mots saisis par l'utilisateur. La sélection automatique est basée sur une méthode prédictive dont le principe consiste à inférer, à partir des premières actions de l'utilisateur, l'ensemble des traces qui couvrent ces actions.

Adaptation des traces

Une fois que la trace pertinente pour l'utilisateur cible est sélectionnée de la base de traces, elle sera transformée par le module d'adaptation (3), en tenant compte de son profil et en se basant sur les

connaissances d'adaptation. Le processus d'adaptation est détaillé dans la section « Adaptation des traces et extraction de connaissances ».

Extraction de connaissances

Le module d'adaptation est également chargé d'extraire des connaissances d'adaptation à partir des traces. Pour cela, ce module analyse l'ensemble des traces stockées dans la base de traces. Il s'agit de déterminer, pour chaque type d'action, l'ensemble des modalités qui peuvent être utilisées pour sa réalisation, ainsi que l'ensemble des conditions (relatives au profil) qui doivent être respectées pour chacune de ces modalités. Le processus d'extraction de connaissances est présenté dans la section « Adaptation des traces et extraction de connaissances ».

Formalisations

Dans cette section, nous présentons les formalisations que nous proposons pour la représentation des traces, le profil utilisateur et les connaissances d'adaptation.

Trace d'interaction

Une trace T est une séquence de toutes les actions de l'utilisateur, ainsi que des modalités utilisées, pour réaliser une tâche donnée. Formellement,

$$T = \langle u, \text{task}, (o_1, o_2, \dots, o_n) \rangle$$

- u : l'utilisateur tracé,
- task : une description de la tâche de l'utilisateur tracé,
- o_i : un observé de la trace. Chaque o_i est un couple $o_i = (A_i, M_i)$, où :

A_i est une action de l'utilisateur, sachant que chaque action A_i possède un type, noté \bar{A}_i , qui définit sa structure par un ensemble d'attributs (a_1, a_2, \dots, a_m) . Par exemple, l'action « écrire une date » peut avoir comme type écrire_date(dd, mm, yyyy), et l'action « déplacer un fichier » peut avoir comme

type `déplacer_fichier` (fichier, dossier_cible).

M_i est une modalité d'interaction. La notion de modalité ayant diverses acceptions selon le domaine d'étude (voir par exemple Bellik, 1995; Bernsen, 1996; Martin, 1995), il est par conséquent difficile de trouver une définition qui fasse l'unanimité. En se basant sur Sehaba (2011), nous considérons une modalité comme un couple (Nigay, 2001) : dispositif physique et langage. Par exemple, l'action

`déplacer_fichier` peut être réalisée selon les modalités suivantes :

- `ls` fichier dossier_cible sous Linux,
- `cp` fichier dossier_cible sous MS-DOS,
- `glisser-déplacer` sous GUI Windows.

Nous avons supposé, dans notre formalisation, que chaque trace représente l'activité de l'utilisateur pour réaliser une tâche donnée. Néanmoins, en pratique, le système collecte toutes les actions de l'utilisateur sans faire la distinction entre celles qui font partie de la même tâche et celles qui n'en font pas partie. Pour cela, c'est le SGBT (présenté dans la section « Théorie de la trace ») qui assure le passage, par des transformations manuelles ou automatiques, entre la trace première (contenant toutes les activités de l'utilisateur sur l'application) et les traces associées aux tâches.

Profil utilisateur

C'est en fonction des propriétés du profil que le module d'adaptation va transformer la trace sélectionnée pour qu'elle soit conforme à son utilisateur cible. Le profil intervient également dans le processus d'extraction de connaissances. Il permet de déterminer les conditions dans lesquelles une modalité donnée peut être appliquée.

Formellement, le profil P d'un utilisateur donné est un ensemble de propriétés p_i qui caractérisent l'utilisateur :

$$P = \{p_1 = v_1, p_2 = v_2, \dots, p_n = v_n\}$$

où chaque propriété p_i appartient à un des composants suivants :

- Capacités physiques et cognitives telles que les capacités sensorielles, l'attention, la mémorisation, etc.,
- Compétences du domaine telles que la maîtrise de l'utilisateur des commandes Linux, MS-DOS, etc.,
- Préférences telles que les couleurs préférées, le style d'apprentissage, etc.

v_i est la valeur de la propriété, avec $v_i \in [0, 1]$. Par exemple, la capacité de vue d'un utilisateur non-voyant est $p_{vue} = 0$, et la capacité de vue d'un utilisateur sans déficience visuelle est $p_{vue} = 1$.

Le profil utilisateur peut être défini par l'utilisateur lui-même en remplissant les valeurs des différentes propriétés, ou généré automatiquement à partir de l'analyse de ses traces d'interaction comme dans Hussaan (2010).

Connaissances d'adaptation

Une connaissance d'adaptation permet au système de choisir, pour chaque action de la trace, la bonne modalité en fonction des propriétés du profil de l'utilisateur cible.

Le principe est d'associer à chaque type d'action \bar{A}_i une ou plusieurs facettes, chacune étant définie par un mode et une modalité. Le mode définit les conditions dans lesquelles la modalité peut être réalisée. Ces conditions portent sur le profil de l'utilisateur. La modalité définit le dispositif physique et le langage d'interaction permettant de réaliser l'action. Formellement, la connaissance d'adaptation de \bar{A}_i est :

$\bar{A}_i \{$

Facette₁ :

Mode₁ : $\langle (m^1_{i1} \in I^1_{i1}, w^1_{i1}), \dots, (m^1_{ik1} \in I^1_{ik1}, w^1_{ik1}) \rangle$

Modalité₁ : $\langle media_{i1}, langage\ d'interaction_{i1} \rangle$

Facette₂ :

Mode₂ : $\langle (m^2_1 \ I^2_1, w^2_1), \dots, (m^2_{k2} \ I^2_{k2}, w^2_{k2}) \rangle$

Modalité₂ : $\langle \text{media}_2, \text{langage d'interaction}_2 \rangle$

...

}

Telle que pour chaque facette j :

- $m^j_i \in P$: propriété du profil,
- $I^j_i \subseteq [0, 1]$: intervalle de valeurs,
- w^j_i : poids de la propriété m^j_i dans le mode de la facette j.

Le mode exprime des contraintes sur les propriétés du profil, où chaque contrainte indique l'intervalle auquel doit appartenir la valeur de la propriété ainsi que le poids de la propriété dans le mode. Par exemple, la connaissance d'adaptation de l'action « afficher le contenu d'un dossier » (\overline{ACD}_i) peut être :

\overline{ACD}_i {

Facette₁ :

Mode₁ : $\langle \text{Compétences_Linux} \in [0.5, 1], 1 \rangle$

Modalité₁ : $\langle \text{Clavier, ls -l Dossier} \rangle$

Facette₂ :

Mode₂ : $\langle \text{Compétences_Msdos} \in [0.5, 1], 1 \rangle$

Modalité₂ : $\langle \text{Clavier, dir Dossier} \rangle$

Facette₃ :

Mode₃ : $\langle \text{Compétences_GUI} \in [0.3, 1], 1 \rangle$

Modalité₃ : $\langle \text{Souris, Double_clique_sur Dossier} \rangle$

}

Cette connaissance peut être définie par le concepteur ou générée automatiquement à partir de l'analyse des traces d'interaction. Ce dernier cas est présenté dans la section suivante.

Adaptation des traces et extraction de connaissances

Dans cette section, nous présentons le principe de nos méthodes d'adaptation de traces et d'extraction de connaissances d'adaptation.

Adaptation des traces

L'adaptation des traces est un processus de transformation qui permet de générer à partir d'une trace T, de la base de traces, une autre trace avec des modalités adaptées à son utilisateur cible.

Le principe du processus d'adaptation est d'associer à chaque action de la trace T la modalité qui maximise la similarité entre son mode et le profil de l'utilisateur cible. Le choix de la mesure de similarité est tout à fait crucial pour une bonne exécution du processus d'adaptation. Il s'agit en effet de trouver la meilleure adéquation entre le profil de l'utilisateur et la modalité à associer à chaque action. Bisson (2000) présente un état de l'art sur les fonctions de similarité.

Le but du processus d'adaptation n'est pas d'exécuter automatiquement les traces transformées, mais de permettre à l'utilisateur cible de savoir comment il peut procéder pour réaliser ses tâches. Pour l'exécution des traces, en plus du profil, d'autres critères doivent être pris en compte tels que les dispositifs physiques disponibles, la cohérence entre le langage d'interaction sélectionné et les logiciels accessibles dans l'environnement informatique de l'utilisateur cible, etc.

Extraction de connaissances d'adaptation

L'objectif est d'extraire des connaissances d'adaptation concernant les modalités d'un type d'action \overline{A}_i à partir des traces stockées dans la base de traces. Le principe de la méthode d'extraction de connaissances consiste à :

Chercher dans la base de traces l'ensemble des utilisateurs qui ont réalisé une action de type \bar{A}_i en utilisant une modalité M_i ,

- Déterminer, dans les profils de ces utilisateurs, les propriétés qui sont très proches ou identiques,
- Affecter ces propriétés au mode $mode_i$ associé à la modalité M_i de \bar{A}_i ,
- Ajouter la facette $(mode_i, M_i)$ à la connaissance de \bar{A}_i , puis
- Refaire le même raisonnement pour toutes les modalités associées aux actions de type \bar{A}_i .

Formellement, pour tous les observés $o_k = (A_k, M_k)$ de la base de traces, on définit la matrice $M = (v_{ij})$ où $1 \leq i \leq m$ et $1 \leq j \leq n$, avec :

- m : le nombre d'utilisateurs qui ont réalisé l'action A_k avec la modalité M_k ,
- n : le nombre de propriétés du profil utilisateur,
- v_{ij} : la valeur de la propriété p_j de l'utilisateur u_i .

Par la suite, on applique l'algorithme suivant :

Algorithme d'extraction de connaissances

Entrée : base de traces, seuil proche ϵ de 0.

Sortie : connaissances d'adaptation

Pour (chaque action A_k de type \bar{A}_k dans la base de trace) **faire**

$E = L$ l'ensemble des modalités associées à l'action A_k

Pour (chaque modalité $\in E$) **faire**

Construire la matrice M

$Mode_j = \{\}$

Pour (chaque colonne p_j de la matrice) M **faire**

si $((\max(v_{1j}, \dots, v_{mj}) - \min(v_{1j}, \dots, v_{mj})) < \epsilon)$ **alors**

$X = \text{moyenne}(v_{1j}, \dots, v_{mj})$

$mode_j = mode_j \cup \{p_j \in [\xi - \epsilon, \xi + \epsilon]\}$

fin

fin

Ajouter la facette $\langle Mode_j, M_j \rangle$

au type \bar{A}_k

fin

fin

Cadre applicatif : formation VCIel

L'application que nous visons dans ce travail concerne la formation VCIel (Sehaba et Metz, 2011). VCIel⁶ (Visualisation et conception infographiques en ligne) est un Master 2 entièrement en ligne et à distance. Il vise à former des professionnels dans les domaines de la production multimédia et les technologies de l'Internet, l'infographie 2D-3D et la conduite de projet.

Les apprenants de cette formation accèdent aux différents modules de la formation à travers la plateforme SPIRAL⁷ (Renaut, Batier, Flory et Heyde, 2006). Cette dernière met à disposition des utilisateurs des outils collaboratifs permettant des échanges synchrones et asynchrones entre les différents étudiants et entre les étudiants et leurs tuteurs. Afin de faciliter l'utilisation de cette plateforme par les tuteurs et les apprenants, SPIRAL dispose d'un système d'aide offrant une description détaillée de ses principales fonctionnalités telles que l'inscription d'un étudiant à un module, l'activation d'une ressource (clavardage, forum, etc.), l'affichage des statistiques de connexion, etc.

Le système d'aide de cette plateforme est organisé sous forme d'un ensemble de briques. Chaque brique possède un identifiant, un nom, une description et un ensemble de fonctions. Chaque fonction permet aux utilisateurs (tuteurs ou apprenants) ne maîtrisant pas la plateforme d'être guidés dans leurs actions pour réaliser leurs tâches.

Les statistiques montrent que les fonctions d'aide sont très peu consultées bien que les utilisateurs aient des difficultés à utiliser certaines fonctionnalités de la plateforme. Ils préfèrent souvent faire appel au responsable de la formation ou à leurs collègues pour sortir d'une situation de blocage plutôt que de consulter l'aide. Une des raisons qui explique ce faible engouement réside dans le fait que le contenu de l'aide ne couvre pas toutes les fonctionnalités de la plateforme. En effet, l'aide SPIRAL est gérée par l'administrateur qui peut créer, modifier ou supprimer des briques et/ou des fonctions,

6 <http://vciel.univ-lyon2.fr>

7 <http://spiralconnect.univ-lyon1.fr>

ce qui est très contraignant et coûteux pour l'administrateur, souvent préoccupé par d'autres tâches.

En outre, le contenu de certaines fonctions n'est pas adapté à tous les apprenants. En effet, chaque année, VCIel accueille à distance une vingtaine d'étudiants, en formation initiale ou en formation continue, provenant de différents pays. Cette diversité de provenance, tant géographique qu'institutionnelle, nécessite des mécanismes adaptatifs afin que le système d'aide puisse adapter ses réponses selon les spécificités de chaque apprenant : ses compétences, son rythme d'apprentissage, sa façon d'apprendre, etc.

L'objectif de notre travail dans ce cadre applicatif était de développer un module permettant aux tuteurs et aux étudiants de s'entraider par le biais de partage de traces. Ce module devait être capable de transformer la trace d'un utilisateur donné pour qu'elle soit adaptée à l'utilisateur cible en tenant compte de son profil.

Dans le cadre de cette formation et sur la base des modèles présentés dans cet article, nous avons développé une application qui permet aux étudiants de s'entraider par le biais du partage d'expériences. Cette application a été implémentée en JAVA J2EE. Les profils des utilisateurs et les connaissances d'adaptation sont stockés dans des fichiers XML. Le kTBS⁸ (*kernel for Trace Base System*) a été utilisé pour stocker les traces d'interaction. Les données du kTBS sont codées en RDF⁹ (Resource Description Framework). L'interface de cette application est basée sur les technologies JavaScript et CSS

8 Le kTBS est un prototype du SGBT (système de gestion de base de traces) : <http://liris.cnrs.fr/sbt-dev/ktbs>

9 <http://www.w3.org/RDF>

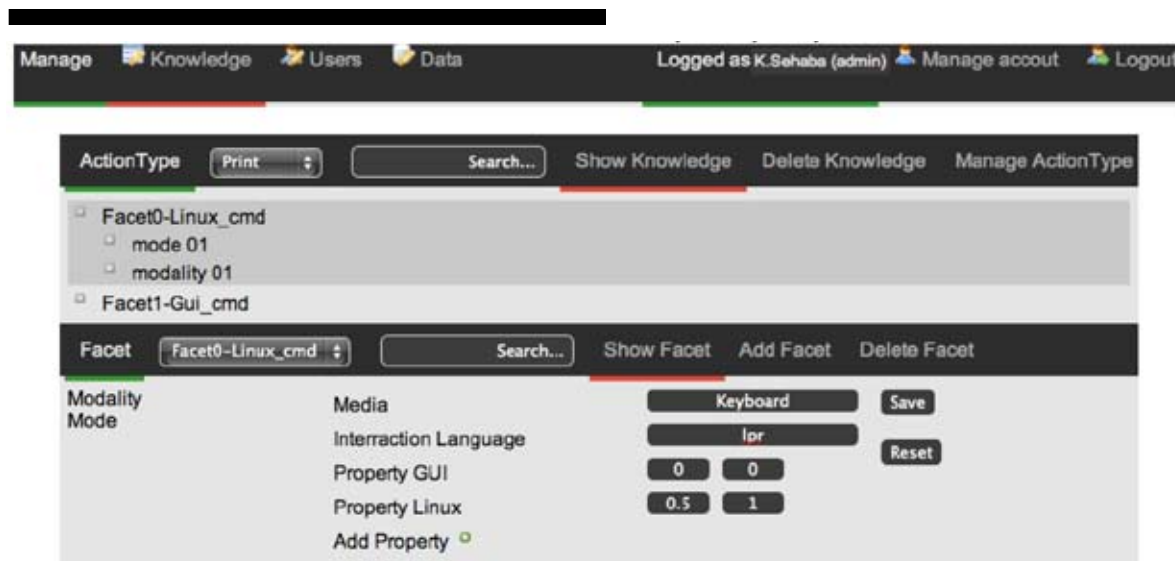


Figure 3. Interface de création de connaissances d'adaptation

Cette interface permet à chaque utilisateur d'introduire son profil, de partager ses traces avec d'autres utilisateurs, de recevoir les traces des autres utilisateurs et, si nécessaire, de transformer ces traces afin qu'elles soient adaptées à son profil. Quant à l'administrateur, il peut créer, modifier ou supprimer manuellement des connaissances d'adaptation et extraire ces connaissances automatiquement à partir des traces stockées dans le kTBS.

La figure 3 montre l'interface qui permet de configurer les connaissances d'adaptation. Cette interface correspond au compte de l'utilisateur « K.Sehaba ». En tant qu'administrateur, cet utilisateur peut gérer les connaissances d'adaptation et les profils des utilisateurs. Les modèles de ces derniers ont été présentés dans les sections « Formalisations ». Comme le montre la figure, cette interface est composée de deux parties : la première partie permet de sélectionner un type d'action, puis de visualiser ses facettes. La deuxième partie permet de manipuler la modalité et le mode de la facette sélectionnée dans la première partie. Ainsi, l'administrateur peut créer ou modifier la modalité de la facette (média et langage d'interaction) et/ou son mode (propriétés et leurs intervalles).

La figure 4 présente un exemple d'une trace partagée et le résultat de sa transformation. Ces deux traces sont stockées dans le kTBS. Comme le montre cette figure, les traces peuvent être visualisées dans différents formats : HTML, RDF, TTL, etc. (1). La trace de la figure 4.A a été créée par l'utilisateur Bob et stockée dans sa base (nommée user_1) avec l'identifiant t_4 (2). Cette trace représente l'activité de l'utilisateur Bob. Il s'agit de déplacer, compiler et exécuter un fichier LaTeX. Faute de place, la figure ne montre que deux observés : déplacer un fichier (3) et compiler un fichier LaTeX (4). Les actions de cette trace sont effectuées avec des commandes Linux. Cette trace a été partagée avec l'utilisateur Alice, puis transformée en tenant compte de son profil (Alice possède peu de connaissances concernant les commandes Linux et MS-DOS, et préfère utiliser l'interface graphique).

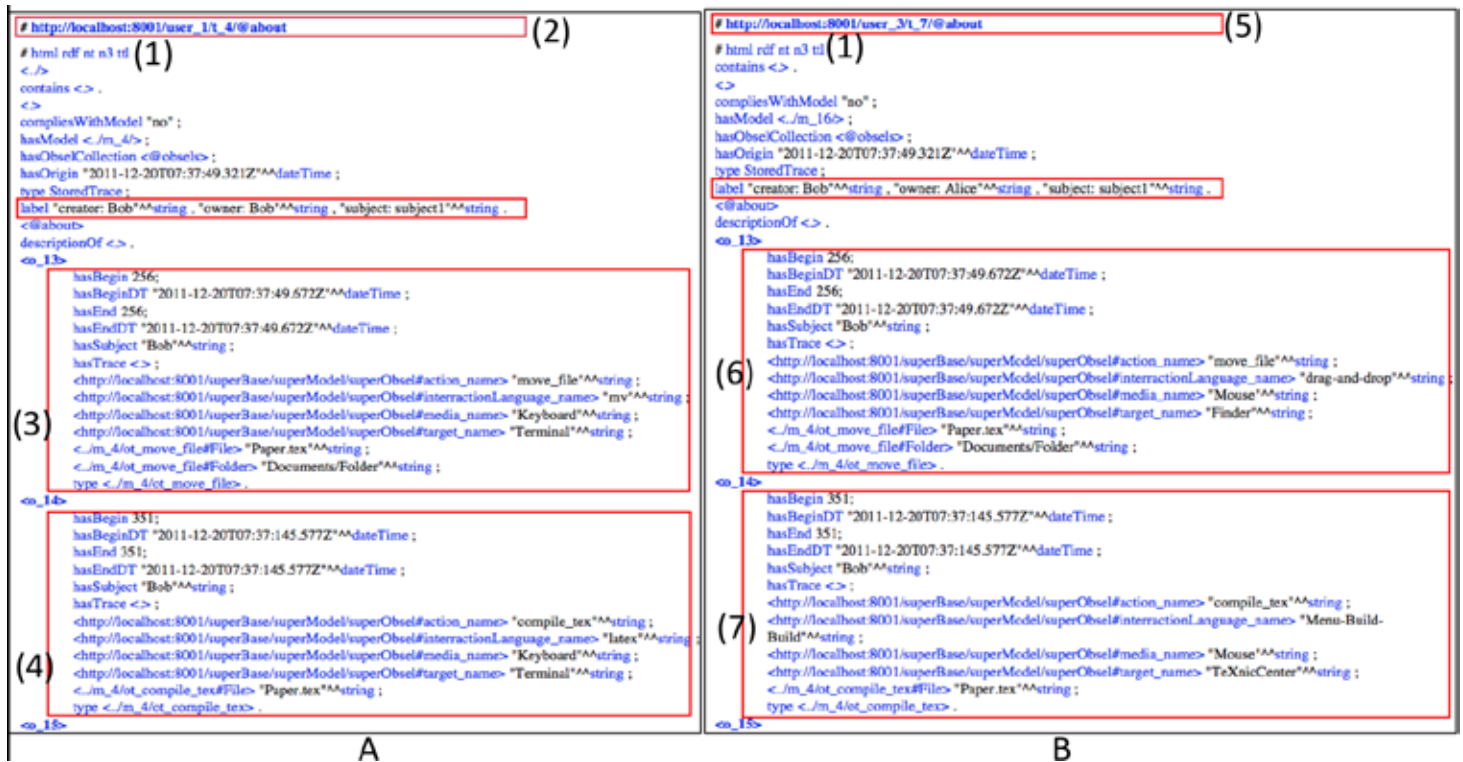


Figure 4. Exemple d'une trace et résultat de sa transformation

Pour cela, pour tout observé de la trace partagée, le système a associé la modalité qui maximise la similarité entre son mode et le profil d'Alice. Comme le montre la figure 5, pour l'observé *move_file* (déplacer fichier), le système a calculé les similarités entre le profil d'Alice et les différents modes de la connaissance d'adaptation du type d'action « *move_file* ». Ainsi, le système a sélectionné la modalité GUI (interface graphique) parce qu'elle est la plus proche du profil d'Alice.

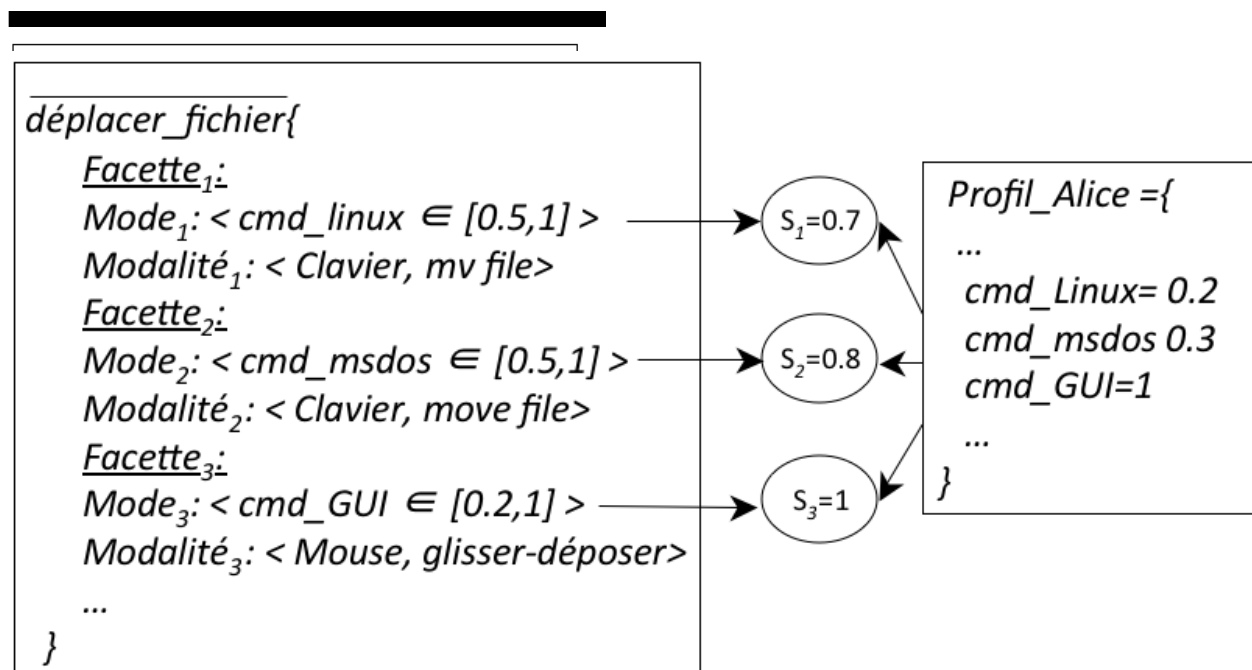


Figure 5. Exemple « calcul de similarité »

Le résultat de cette transformation a été stocké dans la base d’Alice (nommée user_7) avec l’identifiant t_7, comme le montre la figure 5.B. Tous les observés de cette trace ont été transformés avec des modalités adaptées [voir (6) et (7)] au profil d’Alice (commandes GUI).

Conclusion

Cet article tente d’apporter une réponse à la problématique de compatibilité entre la vision du concepteur de l’aide et les compétences effectivement requises des utilisateurs. En effet, plusieurs études (Capobianco et Carbonell, 2006; Leplat, 1998) posent le problème de l’écart entre ces deux représentations, ce qui explique en partie les difficultés rencontrées par les utilisateurs des systèmes d’aide.

Dans ce travail, nous avons proposé une approche d’un système d’aide à base de traces. Il s’agit de considérer les traces d’interaction comme des sources de connaissances que le système peut utiliser pour aider ses utilisateurs en cas de situation de blocage. L’avantage de cette approche réside dans sa

capacité à faire évoluer le contenu de l’aide d’une manière dynamique ainsi qu’à générer des aides adaptées au profil de leur utilisateur cible. Ainsi, nous avons proposé des modèles de représentation de traces, de profil utilisateur et de connaissances d’adaptation. Nous avons également proposé des méthodes d’adaptation de traces et d’extraction de connaissances d’adaptation à partir d’une base de traces.

L’application de notre travail concerne la formation VCIel. Il s’agit de proposer une aide adaptative à base de traces permettant de faciliter l’utilisation de la plateforme SPIRAL.

Nous travaillons actuellement sur la validation de notre approche par des expérimentations sur des traces issues de cette plateforme. Également, nous sommes intéressés par le partage d’expériences entre utilisateurs avec et sans handicaps.

Références

- Bellik, Y. (1995). *Interfaces multimodales : concepts, modèles et architectures* (thèse de doctorat, Université Paris XI, France). [Récupéré](http://www.limsi.fr/~bellik) du site de l'auteur : <http://www.limsi.fr/~bellik>
- Bernsen, N. O. (1996). A reference model for output information in intelligent multimedia presentation systems. Dans G. P. Faconti et T. Rist. (dir.), *Proceedings of ECAI 96 Workshop: Towards a standard reference model for intelligent multimedia systems*. [Récupéré](http://www.nislab.dk) du site de l'auteur : <http://www.nislab.dk>
- Bisson, G. (2000). La similarité : une notion symbolique numérique. Dans E. Diday, Y. Kodratoff, P. Brito, M. Moulet (dir.), *Induction symbolique numérique à partir de données*. Toulouse, France : Cépaduès. Manuscrit [récupéré](http://www-leibniz.imag.fr/Apprentissage) du site de l'équipe Apprentissage Machine du laboratoire Leibniz (IMAG-Grenoble) : <http://www-leibniz.imag.fr/Apprentissage>
- Boullier, D. et Legrand, M. (dir.). (1993). *Les mots pour le faire. Conception des modes d'emploi*. Paris, France : Descartes.
- Capobianco, A. et Carbonell, N. (2006). Aides en ligne à l'utilisation de logiciels grand public : problèmes spécifiques de conception et solutions potentielles. *Intellectica*, 44, 87-120. [Récupéré](http://hal.archives-ouvertes.fr) de l'archive HAL : <http://hal.archives-ouvertes.fr>
- Champalle, O., Sehaba, K., Mille, A. et Prié, Y. (2011). A framework for observation and analysis of learners' behavior in a full-scope simulator of a nuclear power plant: Approach based on modelled traces. Dans I. Aedo, N.-S. Chen, D. G. Sampson, J. M. Spector et Kinshuk (dir.), *Proceedings of the 11th IEEE International Conference on Advanced Learning Technologies* (p. 30-31). Piscataway, NJ : IEEE. [Récupéré](http://liris.cnrs.fr/publis) du site du laboratoire LIRIS, section Publications : <http://liris.cnrs.fr/publis>
- Clauzel, D., Sehaba K. et Prié, Y. (2011). Enhancing synchronous collaboration by using interactive visualisation of modelled traces. *Simulation Modelling Practice and Theory*, 19(1), 84-97. [Récupéré](http://liris.cnrs.fr/publis) du site du laboratoire LIRIS, section Publications : <http://liris.cnrs.fr/publis>
- Cram, D., Jouvin, D. et Mille, A. (2007). Visualisation interactive de traces et réflexivité : application à l'EIAH collaboratif synchrone eMédiathèque. *STICEF*, 14. [Récupéré](http://sticef.univ-lemans.fr) du site de la revue : <http://sticef.univ-lemans.fr>
- De Brito, G. (2006). Statut et non-respect des procédures écrites. *Intellectica*, 44, 63-85.
- Foucher, A.-L. et Pothier, M. (dir.). (2007). Sélection d'articles du colloque international Tidilem « TICE et didactique des langues étrangères et maternelles : la problématique des aides à l'apprentissage » (numéro thématique). *Alsic*, 10(1). [Récupéré](http://alsic.revues.org) du site de la revue : <http://alsic.revues.org>
- Gapenne, O. (2006). Relation d'aide et transformation cognitive. *Intellectica*, 44, 7-16.
- Gapenne, O. et D. Boullier (dir.). (2006). Systèmes d'aide : enjeux pour les technologies cognitives (numéro thématique). *Intellectica*, 44.
- Hussaan, A.-M. (2010). Utilisation des traces pour la mise à jour des connaissances du domaine et du profil de l'apprenant dans les EIAH adaptatifs. Dans P. Cottier (dir.), *Actes des 3^e rencontres Jeunes chercheurs en EIAH* (p. 159-160). [Récupéré](http://liris.cnrs.fr/publis) du site du laboratoire LIRIS, section Publications : <http://liris.cnrs.fr/publis>
- Laflaquière, J. (2009). *Conception de système à base de traces numériques pour les environnements informatiques documentaires* (thèse de doctorat, Université de technologie de Troyes, France). [Récupéré](http://hal.archives-ouvertes.fr) de l'archive HAL : <http://hal.archives-ouvertes.fr>
- Leplat, J. (1998). À propos des procédures. *Performances humaines et techniques*, 94, 6-15.

- Martin, J.-C. (1995). *Coopérations entre modalités et liage par synchronie dans les interfaces multimodales* (thèse de doctorat non publiée). École Nationale Supérieure des Télécommunications, Paris, France.
- Mille, A., Caplat, M. et Philippon, M. (2006). Faciliter les activités des utilisateurs d'environnements informatiques : quoi, quand, comment? *Intellectica*, 44, 121-143.
- Nigay, L. (2001). *Modalité d'interaction et multimodalité* (mémoire présenté en vue de l'habilitation à diriger des recherches, Université Joseph Fourier – Grenoble I, France). [Récupéré](http://tel.archives-ouvertes.fr) de l'archive Tel (Thèses en ligne) : <http://tel.archives-ouvertes.fr>
- Renaut, C., Batier, C., Flory, L. et Heyde, M. (2006). Improving Web site usability for a better e-learning experience. Dans A. Méndez-Vilas, A. Solano Martin, J. A. Mesa Gonzalez et J. Mesa Gonzalez (dir.), *Current developments in technology-assisted education* (p. 891-895). Badajoz, Espagne : FORMATEX. [Récupéré](http://citeseerx.ist.psu.edu) du site CiteSeerX : <http://citeseerx.ist.psu.edu>
- Sehaba, K. (2011). Partage d'expériences entre utilisateurs différents : adaptation des modalités d'interaction. *Actes des 22^e journées francophones d'Ingénierie des connaissances (IC 2011)* p. 639-655. Chambéry, France. [Récupéré](http://liris.cnrs.fr/publis) du site du laboratoire LIRIS, section *Publications* : <http://liris.cnrs.fr/publis>
- Sehaba, K. et Metz, S. (2011). Using interaction traces for evolutionary design support: Application on the Virtual Campus VCIel. Dans A. Verbraeck, M. Helfert, J. Cordeiro et B. Shishkov (dir.), *Proceedings of the 3rd International Conference on Computer Supported Education* (vol 2, p. 237-242). Noordwijkerhout, Pays-Bas : SciTePress. [Récupéré](http://liris.cnrs.fr/publis) du site du laboratoire LIRIS, section *Publications* : <http://liris.cnrs.fr/publis>
- Sellen, A. et Nicol, A. (1990). Building user-centered on-line help. Dans B. Laurel (dir.), *The art of human-computer interface design* (p. 143-153). New York, NY : Addison Wesley.
- Settouti, L. S. (2011). *Systèmes à base de traces modélisées : modèles et langages pour l'exploitation des traces d'interactions* (thèse de doctorat, Université Claude Bernard – Lyon 1, France). [Récupéré](http://liris.cnrs.fr/publis) du site du laboratoire LIRIS, section *Publications* : <http://liris.cnrs.fr/publis>