Concevoir des assistants intelligents pour des applications fortement orientées connaissances : problématiques, enjeux et étude de cas

Amélie Cordier, Marie Lefevre, Stéphanie Jean-Daubias, Nathalie Guin

Concevoir des assistants intelligents pour des applications fortement orientées connaissances : problématiques, enjeux et étude de cas

Amélie Cordier, Marie Lefevre, Stéphanie Jean-Daubias et Nathalie Guin

Université de Lyon, CNRS
Université Lyon 1, LIRIS, UMR5205, F-69622, France
{prenom.nom}@liris.cnrs.fr

Résumé : Les applications permettant de manipuler des connaissances complexes sont souvent difficiles à prendre en main, et donc rebutantes pour des « novices ». En conséquence, les utilisateurs ont tendance à les abandonner trop vite, se privant ainsi de leurs nombreux avantages. Doter ces applications d'assistants adaptés aux utilisateurs peut permettre d'éviter cet écueil. Dans cet article, nous nous intéressons aux problématiques d'ingénierie des connaissances soulevées par le développement d'un « assistant intelligent » pour de tels systèmes. Nous traitons cette question à travers l'étude de cas du logiciel Adapte, un outil pour la proposition d'activités pédagogiques personnalisées. Dans cette étude, nous identifions les besoins d'assistance dans Adapte et nous mettons en évidence les verrous posés par le développement d'un assistant pour cette application. Nous nous intéressons en particulier aux connaissances nécessaires au fonctionnement d'un tel assistant, ainsi qu'aux processus qui permettent l'acquisition et l'évolution de ces connaissances.

Mots-clés : Assistance à l'utilisateur, modèles de connaissances, étude de cas et recommandations, Raisonnement à Partir de l'Expérience Tracée (RàPET), Environnements Informatiques pour l'Apprentissage Humain (EIAH).

1 Introduction

Les récents progrès en ingénierie des connaissances, tant d'un point de vue conceptuel que technologique, contribuent à l'accélération des développements d'applications « fortement orientées connaissances ». Ces applications sont conçues pour collecter, capitaliser et surtout réutiliser des connaissances issues des expériences humaines afin d'aider des utilisateurs dans une tâche donnée. Cependant, ces applications sont souvent très difficiles à prendre en main, et bon nombre d'utilisateurs novices les abandonnent trop vite, malgré leur utilité potentielle. Les causes de ce phénomène sont nombreuses : interfaces chargées, modèles de connaissances complexes et inaccessibles pour l'utilisateur, investissement personnel requis trop important avant de pouvoir tirer profit de l'outil, etc.

Différentes solutions visent à faciliter l'appropriation de ces applications par leurs utilisateurs. Outre les formations en présentiel, il existe des solutions logicielles :

logiciels de formation, tutoriels ou encore assistants s'exécutant en parallèle de l'application principale. En permettant une prise en main progressive, les assistants apportent un début de réponse au problème de l'appropriation des applications complexes par des utilisateurs novices. De plus, puisqu'ils fournissent de l'aide en contexte, au moment où elle est requise, les assistants sont souvent plus efficaces que les tutoriels et les outils de formation. Les assistants représentent un facteur essentiel dans le processus d'appropriation d'une application par un utilisateur. Il est donc regrettable que leur développement ne bénéficie ni du même engouement ni de la même célérité que celui des applications elles-mêmes. Il est néanmoins important de reconnaître que le développement d'un assistant « intelligent » est une entreprise particulièrement compliquée. Autant les applications sont développées pour accomplir un ensemble de tâches particulières, souvent prédéfinies, autant un assistant doit pouvoir s'adapter au contexte d'utilisation et doit être efficace dans des situations qui n'auront pas été anticipées par les concepteurs. Ainsi, dans la suite, lorsque nous parlons d'assistant intelligent, nous désignons à la fois un assistant simple, flexible, capable de s'adapter aux besoins et aux usages des utilisateurs, mais aussi compétent sur l'ensemble des fonctionnalités de l'application pour laquelle il fournit une assistance (notée application cible).

Les assistants intelligents s'appuient sur un certain nombre de connaissances : connaissances sur le domaine de l'application (si elles sont disponibles), connaissances sur la tâche à accomplir avec l'application cible, connaissances relatives aux compétences et préférences des utilisateurs, etc. Ils doivent également être capable d'enrichir leurs connaissances au cours du temps afin de faire évoluer l'aide qu'ils fournissent aux utilisateurs et de s'adapter à leurs besoins changeants. Dans la section 2, nous dressons un rapide panorama de ce type d'assistants en mettant l'accent sur la façon dont les connaissances sont manipulées dans ces outils. Dans la section 3, nous discutons de la façon dont le raisonnement à partir de l'expérience tracée peut contribuer à fournir à un utilisateur, au fil de l'eau, une aide adaptée et contextualisée. Nous discutons en particulier de la circulation de la connaissance entre les différents acteurs de la situation : utilisateur, application et assistant. D'un point de vue plus concret, dans la section 4, nous présentons les résultats d'une réflexion que nous avons menée afin de proposer des recommandations pour le développement d'un assistant intelligent pour l'application Adapte. Cet outil permet à des enseignants, de tous niveaux et disciplines, de construire des séquences d'activités pédagogiques. Notre étude a consisté à identifier les besoins en assistance (du point de vue de l'enseignant utilisateur de l'application), puis à mettre en évidence les verrous au développement de l'assistant : modèles de connaissances manquants, mécanismes de raisonnement nécessaires et anticipation des modalités d'interactions avec l'enseignant. La section 5 discute des principes proposés pour l'assistant d'Adapte et de leurs liens avec le raisonnement à partir de l'expérience tracée. À cette occasion, nous soulevons un certain nombre d'interrogations : comment extraire des expériences des traces d'interactions ? Comment permettre à un agent assistant de réutiliser ces expériences pour proposer une assistance personnalisée et contextualisée ? La section 6 conclut cet article en présentant brièvement nos différentes perspectives de recherche dans ce domaine.

2 Un aperçu de quelques assistants intelligents

Les travaux de recherche visant à construire des assistants dits « intelligents » pour aider un utilisateur dans une tâche donnée ne manquent pas. Pour ne donner que quelques exemples, on peut mentionner les travaux visant à aider à la classification de messages électroniques (Freed et al. 2008), à l'utilisation d'outils de conception (Hawes 2009) ou encore à l'aide à la navigation sur des sites complexes (Richard *et al.* 2004). Dans ces assistants, la composante « intelligente » réside dans l'implémentation de mécanismes de raisonnement permettant de faciliter ou d'automatiser certaines tâches : utiliser des techniques de classification pour organiser des messages, appliquer un raisonnement à base de règles pour compléter un cas de conception, utiliser des modèles de parcours prédéfinis et indépendants des utilisateurs, etc. Ces assistants reposent donc sur l'utilisation de connaissances et de stratégies prédéfinies. En général, les modules d'aide sont activés lorsqu'un certain nombre de conditions prédéfinies sont vérifiées. Ils ne peuvent pas évoluer pour s'adapter à de nouvelles situations et apporter de l'aide sur des aspects non anticipés par leurs concepteurs. Il est vrai que ces assistants apportent souvent une aide efficace aux utilisateurs « standards », mais ils ne sont pas capables de s'adapter aux utilisateurs ayant des besoins singuliers.

Les outils adaptatifs, en revanche, sont conçus pour s'adapter aux utilisateurs. Selon (Oppermann 1994), les outils adaptatifs sont capables de modifier automatiquement leurs propres caractéristiques en fonction des besoins des utilisateurs. Ces outils sont particulièrement pertinents dans les contextes où les utilisateurs doivent s'approprier rapidement des environnements alors qu'ils n'ont parfois même pas conscience de leurs propres besoins. Les outils adaptatifs exploitent les connaissances dont ils disposent sur les utilisateurs pour adapter leur comportement. Ils font également usage de connaissances relatives au domaine et à l'application elle-même afin de pouvoir faire des inférences et d'identifier les éléments de l'application qui peuvent être adaptés à l'utilisateur. Ainsi, les outils adaptatifs sont centrés sur la façon dont l'utilisateur interagit avec le système et sur la façon dont les interfaces peuvent s'adapter pour faciliter ces interactions. Une revue de plusieurs systèmes adaptatifs peut être trouvée dans (Weibelzahl 2002).

Dans ce travail, nous nous intéressons au développement d'assistants qui soient à la fois capables de s'adapter aux utilisateurs (en prenant en compte leurs compétences et leurs préférences) et capables d'apporter de l'aide pertinente sur l'application cible. Pour cela, nous nous intéressons aux outils s'appuyant sur le partage et la réutilisation de l'expérience pour la construction de l'assistance, tel que celui décrit dans (Champin 2003). Dans (Philippon *et al.* 2005), les auteurs discutent de la façon d'apporter une aide qu'ils nomment *spontanée* (c'est-à-dire pertinente et au bon moment) aux utilisateurs d'une application informatique. Pour cela, ils proposent d'utiliser le raisonnement à partir de l'expérience tracée pour alimenter les propositions d'aide et établir un dialogue entre le système et l'utilisateur. Cette première ébauche de l'utilisation du raisonnement à partir de l'expérience tracée a montré l'utilité d'une telle approche. Nous approfondissons cette question dans la section suivante.

3 Un outil d'assistance basé sur le RàPET

Comme nous l'avons mentionné précédemment, pour être utile et efficace, un assistant doit être capable de s'adapter aux besoins des utilisateurs ainsi qu'aux changements de contexte. Cette exigence a un impact fort sur l'évolution attendue des connaissances dans le système. Or, puisqu'il n'est pas possible d'anticiper tous les comportements de l'utilisateur au moment de la conception de l'assistant, il est nécessaire de doter celui-ci de mécanismes lui permettant de faire évoluer au fil de l'eau les connaissances sur lesquelles il s'appuie. Ces connaissances ne concernent pas uniquement les utilisateurs (compétences, préférences), mais d'une façon plus générale, l'ensemble du domaine d'application concerné.

Le raisonnement à partir de l'expérience tracée (RàPET) propose de réutiliser les principes désormais bien connus du raisonnement à partir de cas pour exploiter des expériences non structurées (Mille 2006). Dans le paradigme du RàPET, les traces d'interactions sont collectées et stockées par un système de gestion de traces. Ces traces sont ensuite exploitées comme des conteneurs de connaissances qui permettent de conserver les informations sur des expériences « en contexte ». C'est le rôle des mécanismes de raisonnement que d'extraire des traces les connaissances nécessaires aux différentes tâches, et de les utiliser à bon escient. Le RàPET, qui s'appuie naturellement sur les interactions entre l'utilisateur et l'application (Cordier et al. 2009), apporte donc un élément de réponse à la problématique de l'évolution des connaissances exposée plus haut.

La figure 1 s'intéresse au couplage utilisateur / application et met en évidence les processus qui assurent la coévolution du système en interaction : l'assistant s'adapte à l'utilisateur et l'utilisateur s'approprie progressivement l'application, ses usages s'en trouvant donc modifiés.

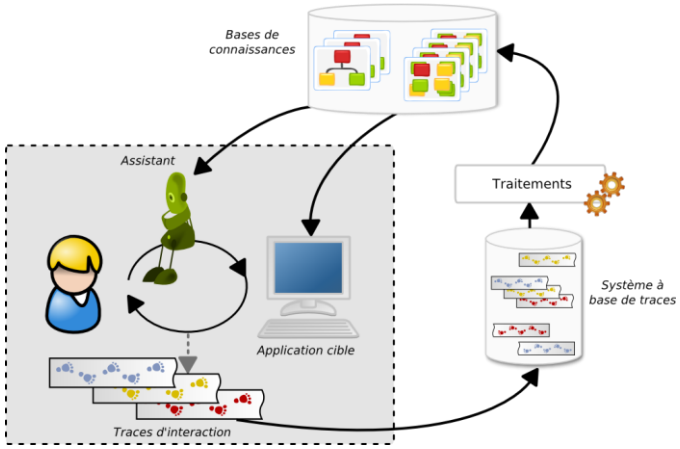

Fig. 1 – Partage des connaissances entre l'utilisateur, l'application et l'assistant.

Comme l'indique la figure 1, l'application cible s'appuie sur des bases de connaissances qui peuvent être initialisées par des processus variés (construction manuelle, résultat d'une fouille, etc.). Ces bases de connaissances vont évoluer tout au long du cycle de vie du système. Pour cela, des mécanismes d'acquisition interactive de connaissances doivent être mise en place (Cordier 2008). L'utilisateur manipule une partie des connaissances via l'application. Il en a donc une vue spécifique à ses besoins. Les interactions entre l'utilisateur et l'application peuvent être médiées par l'assistant qui utilise les bases de connaissances et les expériences passées pour établir ses recommandations. Le système à base de traces assure que l'ensemble des interactions est tracé et collecté. Des mécanismes de raisonnement complémentaires, qui peuvent être *online* ou *offline*, supervisés ou non, sont nécessaires pour exploiter les traces. Ces mécanismes permettent d'enrichir les bases de connaissances existantes, mais également de retrouver des expériences passées, utilisées comme support pour l'assistance ou bien partagées entre différents utilisateurs.

Dans une telle architecture, les bases de connaissances sont partagées et enrichies par tous les acteurs du processus (utilisateur, assistant, application). Chaque acteur dispose de sa propre *vue*, adaptée, sur les connaissances dont il a besoin. En pratique, la mise en place d'une telle architecture n'est pas simple : il est nécessaire d'instrumenter les interactions, de définir les mécanismes de raisonnement et de prévoir les procédures nécessaires à la validation des connaissances acquises au fil de l'eau. Ces aspects sont discutés sur un cas concret dans la section suivante.

4 Étude de cas : l'application cible Adapte

Cette section rapporte les conclusions d'une étude que nous avons menée afin d'établir des recommandations pour le développement d'un assistant intelligent pour l'application Adapte. Nous présentons tout d'abord l'application, puis nous décrivons les besoins identifiés pour assister ses utilisateurs. Nous discutons ensuite des questions soulevées par le développement d'un tel assistant.

4.1 Présentation de l'application

Le logiciel Adapte a pour but d'aider les enseignants, de toutes disciplines et de tous niveaux scolaires ou universitaires, à proposer des séquences de travail adaptées à la fois aux besoins de chaque élève et aux habitudes pédagogiques de chaque enseignant (Lefevre *et al.* 2009). Il s'appuie pour cela sur des modèles de connaissances génériques et utilise des processus qui instancient et articulent ces modèles (Lefevre 2009). Les instanciations des modèles sont propres à chaque enseignant et à chaque type d'activités à personnaliser. Les enseignants ne manipulent cependant pas directement les modèles, leur instanciation passe par l'utilisation d'une interface appropriée.

Actuellement, l'utilisation d'Adapte par un enseignant se déroule selon trois étapes : les deux premières permettent à un enseignant de définir son modèle de

personnalisation et la troisième lui permet de vérifier les séquences de travail proposées par le système en fonction du modèle de personnalisation (cf. figure 2).

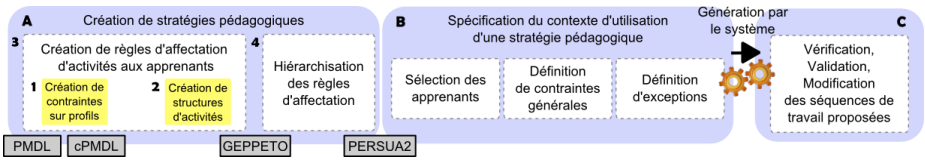

Fig. 2 – Étapes réalisées par l'enseignant dans Adapte et lien avec les modèles de connaissances sous-jacents.

Dans la première étape, l'enseignant décrit sa stratégie pédagogique (cf. A) sous forme d'un ensemble de règles hiérarchisées. Chaque règle indique quelles activités un apprenant possédant une caractéristique donnée devra effectuer. Les caractéristiques peuvent être liées à ses connaissances, ses compétences, son comportement, etc. L'enseignant définit pour cela des contraintes sur le profil des apprenants permettant de sélectionner les apprenants répondant à certains critères (cf. A-1). Les profils des apprenants sont décrits selon le langage de modélisation PMDL (Eyssautier-Bavay 2008) et les contraintes sur profils respectent le modèle cPMDL. L'enseignant définit ensuite des contraintes sur activités permettant d'adapter les activités pédagogiques à ses habitudes de travail (cf. A-2). Ces contraintes respectent les modèles de l'approche GEPPETO qui permettent d'adapter autant les activités à faire sur papier que celles se déroulant sur support logiciel. L'enseignant relie ensuite les contraintes sur profils aux contraintes sur activités afin de créer des règles permettant d'affecter les activités aux apprenants (cf. A-3). Ces règles sont associées à un niveau d'importance (cf. A-4). Chaque stratégie pédagogique définie par un enseignant est conforme au modèle de personnalisation PERSUA2.

Dans la seconde étape, l'enseignant décrit un contexte d'utilisation pour une stratégie pédagogique donnée (cf. B). Ce contexte est lui aussi conforme au modèle PERSUA2 et contient d'une part la liste des élèves pour lesquels l'enseignant souhaite obtenir des séquences de travail et d'autre part un ensemble de contraintes sur la séquence en général (matériel disponible, durée, nombre d'activités, etc.). L'enseignant peut de plus préciser des exceptions pour certains élèves, limitant par exemple le nombre d'activités à fournir à un apprenant en difficulté, changeant la police d'écriture pour un élève dyslexique ou encore demandant une séquence de travail plus longue pour des élèves ayant des facilités dans certaines matières.

Dans la troisième étape (cf. C), l'enseignant prend connaissance des séquences de travail proposées par Adapte pour chacun des élèves. Il peut alors agir sur le nombre et sur le contenu des activités proposées dans les séquences, ainsi que sur leur ordre de présentation, avant de valider ou non les propositions du système. Une fois les séquences de travail validées par l'enseignant, Adapte crée pour chaque élève une feuille d'exercices à imprimer et/ou les fichiers nécessaires à la configuration de logiciels pédagogiques.

4.2 Comment aider l'enseignant ?

Le logiciel Adapte peut être vu comme une boite à outils pouvant être utilisée par n'importe quel enseignant. La force de l'application est qu'elle peut s'adapter aux besoins de chacun, puisqu'elle repose sur des modèles génériques. En contrepartie, elle ne propose aucune aide lors de son utilisation. Chaque enseignant doit apprendre à manipuler les outils disponibles, *via* l'interface du logiciel, pour obtenir les séquences de travail adaptées à chacun de ses élèves. La multiplicité des modèles cités dans la partie précédente laisse entrevoir les difficultés rencontrées par les enseignants lors des premières utilisations du logiciel : ceux-ci doivent comprendre les diverses options offertes par chacun des modèles et comprendre comment les combiner.

Afin qu'ils s'approprient plus rapidement et plus efficacement l'application, il est donc nécessaire d'apporter une forme d'aide aux enseignants dans leur utilisation d'Adapte. Un préalable à la proposition de tout type d'aide est une étude ergonomique approfondie de l'interface de l'application afin de la rendre plus simple à comprendre et à utiliser. Ce type d'amélioration est nécessaire mais n'entre pas dans le cadre de cette étude : nous nous intéressons ici à la mise en œuvre d'un assistant intelligent capable de s'adapter à l'utilisateur. Nous proposons de faire reposer une telle assistance sur la capitalisation et le partage d'expériences entre utilisateurs ainsi que sur l'exploitation de connaissances sur les utilisateurs, leurs compétences, leurs préférences et leurs objectifs, connaissances qui ne sont pas présentes actuellement dans le système. Dans la suite de cette section, nous décrivons les différents types d'aides que nous avons identifiés lors de notre analyse de l'application et de notre étude des usages qui en étaient faits par certains utilisateurs.

Dans la première étape d'Adapte, l'enseignant définit sa stratégie pédagogique en créant des règles d'affectation liant des contraintes sur profils à des contraintes sur activités. À ce niveau, il est possible de conseiller l'enseignant sur les parties du profil à utiliser pour personnaliser les activités ou encore sur le choix des activités à fournir en fonction des parties de profils utilisées pour la personnalisation. Une telle aide doit s'appuyer sur la réutilisation d'expériences passées similaires. Il est en effet possible de réutiliser des exploitations des profils possédant une même structure faites précédemment par l'enseignant lui-même ou par d'autres enseignants.

Dans la seconde étape, l'enseignant définit le contexte dans lequel il souhaite utiliser une stratégie pédagogique donnée. Il peut ensuite définir des exceptions pour certains élèves. Ces exceptions permettent à l'enseignant d'intégrer au modèle de personnalisation des connaissances dont il dispose sur ses élèves, mais qui ne sont pas représentées dans les profils. Un second niveau d'aide consiste donc à généraliser ces exceptions et à les intégrer dans les profils ou les modèles de personnalisation afin que l'enseignant n'ait plus à les redéfinir. Cette intégration implique la mise en œuvre d'un mécanisme permettant d'expliquer les causes des exceptions et d'établir le lien avec les éléments manquants dans les modèles de connaissances.

Enfin, dans la dernière étape, Adapte utilise le modèle de personnalisation d'un enseignant pour générer des séquences de travail adaptées aux profils des apprenants. Ces séquences de travail sont soumises à l'enseignant pour validation. Il peut alors en modifier le contenu (ajout ou suppression d'activités, nouvel énoncé pour une activité)

ou la présentation (ordre des activités, mise en page pour les activités papier, etc.). Ces modifications effectuées par l'enseignant peuvent être exploitées comme des sources de connaissances pour améliorer progressivement les connaissances du système d'assistance. En effet, il est possible d'inférer, à partir de ces modifications, un ensemble de connaissances utiles à l'assistance (sous forme de règles d'affectation, de stratégies pédagogiques, de nouveaux éléments à prendre en compte dans les profils, de contraintes sur profils, etc.).

Il est donc possible de fournir trois types d'aides intelligentes dans Adapte. L'assistant qui fournira ces différentes aides devra s'appuyer sur des modèles de connaissances et des mécanismes associés. Certains de ces modèles et mécanismes existent déjà dans l'application et sont formalisés. C'est le cas pour PMDL (structures de profils), cPMDL (contraintes sur profils), GEPPETO (activités et contraintes sur activités) et pour les deux parties de PERSUA2 (stratégies pédagogiques et contexte d'utilisation). D'autres modèles sont implicites : ils sont mis en œuvre dans l'application, mais ne reposent pas sur une formalisation explicite. C'est le cas du modèle des corrections possibles par l'enseignant lors de la dernière phase d'utilisation d'Adapte, ou encore du modèle de navigation et d'interaction avec l'interface. Enfin, des modèles n'existent pas encore et devront être définis. C'est le cas par exemple du modèle de connaissances relatif aux différents éléments qui concernent l'utilisateur. La table 1 récapitule les types d'assistance envisagés pour Adapte et établit, pour chacun, les liens avec les modèles de connaissances concernés.

Table 1. Types d'assistance et modèles de connaissances associés pour Adapte.

Type d'aide proposée	**(1) Conseil lors de la création du modèle de personnalisation**	**(2) Intégration des exceptions dans le modèle de personnalisation**	**(3) Révision des propositions faites par Adapte**
Modèles formalisés	PMDL, cPMDL, GEPPETO, PERSUA2	PMDL, cPMDL, GEPPETO	PMDL, cPMDL, GEPPETO, PERSUA2
Modèles implicites			Modèle de corrections Modèle de navigation
Verrous identifiés	Comment identifier des parties de profils portant sur un même domaine mais n'ayant pas la même structure ? Comment utiliser ces parties proches pour donner des conseils sur les activités à fournir aux élèves ?	Comment trouver une équivalence entre des exceptions sur le contexte d'utilisation et des règles d'affectation pouvant être ajoutées dans la stratégie pédagogique ? Les informations se trouvent-elles dans les profils ?	Comment traduire des corrections de l'enseignant en contraintes pouvant être ajoutées au modèle de personnalisation ? Quelles informations peuvent être déduites de l'ordre dans lequel les modifications sont faites ?
Pistes de solutions	Demander à l'enseignant de décrire le profil lors de sa création, à l'aide d'annotations. Se servir des métadonnées décrivant les activités pour proposer des liens entre profils d'élèves et activités.	Demander à l'enseignant de décrire le profil lors de sa création, à l'aide d'annotations. Utiliser des mécanismes de découverte de connaissances.	Identifier le lien entre modèle d'activités, contraintes et activités refusées pour préciser les contraintes sur activités. Préciser les contraintes de sélection des profils en analysant les profils pour lesquels des activités ont été ajoutées / supprimées

4.3 Réutiliser l'expérience pour fournir une assistance adaptée

Comme le montre la section précédente, l'assistance que nous proposons pour Adapte repose essentiellement sur la réutilisation des expériences des utilisateurs. À ce stade, il est important de rappeler qu'Adapte est une application générique qui manipule des modèles de connaissances et non les connaissances elles-mêmes. En effet, ce sont les enseignants qui, au travers des manipulations qu'ils effectuent via l'interface, instancient les modèles de connaissances. La généricité de l'application cible introduit une difficulté supplémentaire dans le développement de l'assistant : il n'est pas possible d'instancier les bases de connaissances qui seront utilisées par celui-ci *a priori.* L'assistant doit donc construire ses connaissances au fur et à mesure. Il doit également être capable d'exploiter les connaissances acquises progressivement par l'application cible. Ces aspects ont été mis en évidence dans les dernières lignes de la table 1. Nous en discutons plus précisément ci-dessous.

Le premier type d'assistance identifié pour Adapte repose clairement sur la réutilisation des expériences des utilisateurs, au sens où nous l'avons décrit dans la section 3. Il s'agit d'identifier des situations passées similaires (des enseignants ayant utilisé des profils identiques) et de réutiliser les solutions comme source d'exemple pour la situation courante. La difficulté majeure ici est de définir ce que sont des profils (ou parties de profils) similaires. En effet, Adapte manipule des profils à partir de leur structure sans disposer d'information sur la nature de leur contenu. Par conséquent, il est possible de retrouver des profils (ou parties de profils) dont la structure est identique mais, à moins de disposer de connaissances additionnelles permettant de déterminer que deux profils sont similaires, il n'est pas possible de retrouver des profils « proches » de part leur contenu. Une solution à ce problème serait d'offrir la possibilité aux enseignants d'annoter des parties de profils pour apporter des informations sur le contenu et donc permettre la comparaison de parties de profils. Néanmoins une telle solution demande un effort important de la part de l'enseignant.

Le deuxième type d'assistance est relativement compliqué à mettre en œuvre dans la mesure où il requiert la définition d'un mécanisme capable d'établir une équivalence entre des exceptions définies par les enseignants sur le contexte d'utilisation de certains élèves et des contraintes qui, en étant intégrées à la stratégie pédagogique, auraient permis d'éviter de définir ces exceptions. En revanche, l'ensemble des connaissances nécessaires à l'établissement de cette équivalence est présent dans le système. Pour ce problème, on pourrait envisager d'implémenter des mécanismes de découverte de connaissances afin de faire émerger les règles les plus fréquentes, avant de les faire valider par les enseignants et de les intégrer aux connaissances du domaine.

Le troisième type d'assistance relève également du raisonnement à partir de l'expérience tracée, mais à un niveau différent de celui discuté pour le premier type. Comme nous l'avons expliqué, dans la dernière étape de l'utilisation d'Adapte, l'enseignant peut vérifier, modifier et valider les activités proposées par le système. Cette étape peut être assimilée à l'étape de révision en raisonnement à partir de cas. Actuellement, le modèle de navigation et d'interaction dans cette interface est

implicite. Par exemple, l'enseignant peut ajouter ou supprimer une activité, ce qui correspond implicitement à un changement du niveau de priorité d'une règle d'affectation dans la stratégie pédagogique. De même, il peut décider de régénérer une activité, ce qui revient à modifier ou préciser certaines contraintes de génération de l'activité. Il peut également modifier la présentation des activités (par exemple, les trier dans un certain ordre), ce qui apporte des informations sur ses préférences. Ces différentes actions traduisent des « façons de faire » de l'enseignant. Par conséquent, nous pensons que l'étude des interactions qui ont lieu durant cette phase peut permettre d'acquérir des connaissances précieuses sur les pratiques pédagogiques des enseignants. En pratique, cette découverte de connaissances requiert un processus d'analyse des traces d'interactions (Laflaquière *et al.* 2008).

Dans cette étude, nous n'avons pas détaillé les modalités de présentation de l'assistance qui pourraient être proposées aux enseignants. L'assistance peut prendre plusieurs formes : conseils dispensés de façon externe à l'application, suggestions de modifications directement sur l'interface principale, exécution automatique de certaines procédures, ou encore combinaison des différentes approches. Quelles que soient les modalités retenues, elles exigent que l'assistant dispose de connaissances relatives à l'interface de l'application cible et aux possibilités qu'elle offre. Ces modèles de connaissances ne sont actuellement pas formalisés dans le cas d'Adapte et devront nécessairement l'être afin de permettre le fonctionnement de l'assistant.

5 Discussion

En s'appuyant sur l'analyse d'une application particulière (Adapte), cet article traite des difficultés liées au développement d'assistants intelligents pour des applications fortement orientées connaissances. La conception d'assistants intelligents est une problématique complexe et les scénarios d'assistance sont nombreux. Dans certains cas, il s'agit de guider l'utilisateur pour lui « faciliter la tâche ». Dans d'autres, il faut l'aider à surmonter une difficulté bloquante. Dans (Richard *et al.* 2004), les auteurs proposent un système conseiller (pour la navigation sur certains sites Web) qui s'appuie sur des modèles de parcours prédéfinis. Dans une optique différente, nous cherchons à nous appuyer sur des connaissances qui évoluent au fil du temps. Par exemple, dans le cas d'Adapte, l'application manipule exclusivement des connaissances fournies par les utilisateurs. L'enjeu est donc de doter l'assistant de la capacité à intégrer progressivement ces connaissances et à les utiliser judicieusement dans l'assistance qu'il fournit. Nous proposons donc d'exploiter les traces d'interactions qui permettent de conserver les expériences passées « en contexte » et donc de faciliter leur réutilisation dynamique dans d'autres tâches.

Cette approche n'est pas spécifique au cas d'Adapte décrit dans cet article. Dans un travail précédent (Cordier *et al*. 2009), nous avons proposé d'utiliser le paradigme du raisonnement à partir de l'expérience tracée pour fournir une assistance à des utilisateurs présentant des déficiences visuelles. Dans ce travail, l'accent était mis sur le partage d'expériences entre utilisateurs « très différents » (en l'occurrence, entre utilisateurs voyants et non voyants). Il s'agissait alors non pas de proposer une

assistance sur le contenu de l'application, mais sur les modalités d'accès à ce contenu. Nous avons montré comment l'expérience d'un utilisateur voyant (par exemple, réserver un billet de train) pouvait être réutilisée par un non-voyant par simple adaptation des modalités d'interactions avec l'environnement informatique.

L'architecture d'assistant intelligent que nous avons proposée dans la section 3 de cet article vise non seulement à apporter une aide pointue pour l'utilisation d'applications complexes, mais aussi à prendre en compte les spécificités propres à des utilisateurs très différents. L'assistance que nous souhaitons fournir cherche donc à faciliter l'accès au contenu, en fonction des connaissances et des compétences des utilisateurs, mais aussi à la présentation de ce contenu, en prenant en compte les capacités et les préférences de ces utilisateurs. Une telle architecture permet donc de concevoir des assistants utiles dans les différentes situations évoquées plus haut.

Comme nous l'avons montré, garantir l'adaptabilité de l'assistant requiert de doter ce dernier de la faculté d'évoluer progressivement et implique donc la mise en place de mécanismes lui permettant de mettre à jour les connaissances sur lesquelles il s'appuie. Des mécanismes d'acquisition interactive de connaissances intégrés à l'environnement sont donc nécessaires. Ils doivent supporter les interactions avec l'utilisateur pour la négociation et la validation des connaissances acquises (Stuber 2007). Nous pensons que nombre d'applications informatiques pourraient tirer profit d'assistants capables de s'adapter rapidement aux utilisateurs et de leur fournir une assistance appropriée. C'est en particulier le cas pour les applications pour lesquelles l'utilisateur doit s'adapter très rapidement à un environnement complexe et souvent changeant.

6 Conclusion

Les expérimentations menées avec Adapte ont donné des résultats très positifs et ont montré que cet outil permettait aux enseignants de réaliser la tâche pour laquelle il est conçu : la personnalisation de séquences d'activités pédagogiques. Cependant, ces expérimentations ont également mis en évidence « la complexité de la tâche demandée aux enseignants lors de la formalisation de leurs règles de personnalisation » (Lefevre 2009). Cette complexité perçue par les enseignants tient au fait que l'application est difficile à prendre en main, notamment parce qu'elle propose un très grand nombre de possibilités dès la première phase d'utilisation.

Nous avons donc mené une étude afin de proposer les principes d'un assistant intelligent pour Adapte. Par intelligent, nous entendons à la fois compétent sur l'ensemble des aspects traités par l'application et capable d'évoluer pour s'adapter aux besoins de l'utilisateur. Cet article rend compte des conclusions de cette étude, en mettant l'accent sur les aspects relevant de l'ingénierie des connaissances. Nous avons notamment montré comment les principes du raisonnement à partir de l'expérience tracée peuvent servir le développement d'un tel assistant et apporter des réponses aux problématiques d'enrichissement des connaissances dans de tels systèmes.

Le travail présenté dans cet article reste une étude préliminaire et doit être poursuivi par la construction des modèles de connaissances manquants ainsi que par le

développement et l'expérimentation de l'assistant. Cette perspective soulève de nombreuses questions relatives à la représentation des connaissances, aux capacités de raisonnement de l'assistant et aux problèmes d'interfaces homme-machine. Cependant, nous sommes convaincues que l'enjeu principal de ce travail est l'étude de la co-construction de sens au sein du système couplé utilisateur / application. Nous pensons que l'instrumentation des interactions, par l'intermédiaire de l'assistant par exemple, permettra le renforcement de ce couplage et sera donc un support solide à la coévolution du système en interaction.

Références

CHAMPIN P.A. (2003). ARDECO: an assistant for experience reuse in Computer Aided Design. In *From structured cases to unstructured problem solving episodes - WS 5 of ICCBR'03*. Trondheim, Norway.

CORDIER A., MASCRET B. & MILLE A. (2009). Extending Case-Based Reasoning with Traces. In *Grand Challenges for reasoning from experiences*, Workshop at IJCAI'09, Pasadena, CA.

CORDIER A. (2008). *Interactive and Opportunistic Knowledge Acquisition in Case-Based Reasoning*. Thèse de doctorat. Université Claude Bernard Lyon 1, France.

EYSSAUTIER-BAVAY C. (2008). *Modèles, langage et outils pour la réutilisation de profils d'apprenants*. Thèse de doctorat. Université Joseph Fourier Grenoble 1, France.

FREED M., CARBONELL J.G., GORDON G.J., HAYES J., MYERS B.A., SIEWIOREK D.P., SMITH S., STEINFELD A. & TOMASIC A. (2008). RADAR: A Personal Assistant that Learns to Reduce Email Overload. In AAAI 2008 proceedings. pp.1287-1293.

HAWES N. (2009). Architectures by Design: The Iterative Development of an Integrated Intelligent Agent. In *Proceedings of AI-2009,* Cambridge, UK. pp. 349-362.

LAFLAQUIERE J., PRIE Y., MILLE A. (2008). Ingénierie des traces numériques d'interaction comme inscriptions de connaissances. In *19èmes Journées Francophones d'Ingénierie des Connaissances*, Nancy. pp. 183-195.

LEFEVRE M. (2009). *Processus unifié pour la personnalisation des activités pédagogiques : méta-modèle, modèles et outils*. Thèse de doctorat. Université Claude Bernard Lyon 1, France.

LEFEVRE M., CORDIER A., JEAN-DAUBIAS S. & GUIN N. (2009). A Teacher-dedicated Tool Supporting Personalization of Activities. In *World Conference on Educational Multimedia, Hypermedia & Telecommunications (ED-MEDIA 2009)*, Honolulu, Hawaii, Chesapeake, VA:AACE. pp. 1136-1141.

MILLE A. (2006). From case-based reasoning to trace-based reasoning. In *Annual Reviews in Control*, 30(2):223-232, Elsevier, ISSN 1367-5788.

OPPERMANN R. (1994). Adaptively supported adaptability. In *International Journal of Human Computer Studies*, 40(3), 455-472.

PHILIPPON M., MILLE A., CAPLAT G. (1994). Aide à l'utilisateur : savoir quand intervenir. In *17th conférence Francophone sur l'Interaction Homme-Machine*, pp. 155-162.

RICHARD B., TCHOUNIKINE P. (2004). Une approche centrée modèle pour la construction d'un système conseiller pour un site Web. In *15èmes Journées Francophones d'Ingénierie des Connaissances*, Lyon. pp. 151-162.

STUBER A. (2007). *Co-construction de sens par négociation pour la réutilisation en situation de l'expérience trace*. Thèse de doctorat. Université Claude Bernard Lyon 1, France.

WEIBELZAHL S. (2002). *Evaluation of Adaptive Systems*. Ph.D. Thesis, University of Trier, Germany, October 2002.