

Pré-version. Article final publié comme :
Paquette G., Tchounikine P. (2002), *Contribution à l'ingénierie des systèmes conseillers : une approche méthodologique fondée sur l'analyse du modèle de la tâche*, In: Revue Sciences et Techniques Educatives, Vol. 9 3-4

Contribution à l'ingénierie des systèmes conseillers : une approche méthodologique fondée sur l'analyse du modèle de la tâche

Gilbert Paquette * — Pierre Tchounikine**

* *CIRTA/LICEF – Télé-Université*
4750 avenue Henri Julien, bureau 100
Montréal (Québec) H2T-3E4
CANADA

gpaquett@licef.teluq.quebec.ca

** *LIUM – Université du Mans*
Avenue Laennec
72085 Le Mans cedex 9
FRANCE
Pierre.Tchounikine@lium.univ-lemans.fr

RÉSUMÉ. Un système conseiller est un système qui propose une aide active à l'utilisateur d'un logiciel particulier, aide fondée sur une analyse des actions et des productions de l'utilisateur. Dans le cadre des travaux du Licéf, deux types de systèmes conseillers sont développés : des systèmes conseillers destinés aux apprenants utilisant des environnements d'apprentissage diffusés sur le Web avec Explor@ et des agents conseillers destinés aux concepteurs utilisant la méthode d'ingénierie de systèmes d'apprentissage MISA et son système de support sur Internet, ADISA. Les travaux présentés dans cet article visent à élaborer un cadre méthodologique commun facilitant la construction de ces systèmes conseillers. Les travaux présentés portent sur la phase d'identification des conseils, pour laquelle nous proposons une approche fondée sur l'analyse d'une modélisation de la tâche considérée.

ABSTRACT. An advisor system is a system that proposes a just-in-time first-line help to the users of a given software, on the basis of an analysis of the actions and productions of the user. Within the Licéf laboratory, two different types of advisor systems are developed: advisor systems dedicated to students using the Web-based interactive learning environments developed with Explor@ and advisor agents dedicated to courseware designers using the course design methodology MISA and its Internet-based workbench ADISA. The objective of the work presented in this paper is to elaborate a common methodological framework facilitating the design of these systems. The work presented here focuses on the identification of the pieces of advice; we propose an approach based on the analysis of a model of the considered task.

MOTS-CLÉS : Système conseiller, Identification des conseils, Ingénierie des connaissances.

KEY WORDS: Advisory System, Identification of the pieces of advice, Knowledge Engineering.

1. Introduction

Un système conseiller est un système informatique qui propose une aide active « intelligente » [WIN 92] aux utilisateurs d'un logiciel particulier. Typiquement, un conseil est un texte qui suggère à l'utilisateur de réaliser une action, lui indique que son action est incohérente ou maladroite ou qui critique l'objet qu'il a produit. Un conseil est une information : le système conseiller ne force en rien l'utilisateur à suivre les conseils qu'il prodigue. Les conseils sont fondés sur une analyse des actions et des productions de l'utilisateur. Le terme « aide intelligente » fait référence à l'objectif recherché, c'est-à-dire proposer un conseil qui prend en compte le contexte de l'action et les objectifs poursuivis par l'utilisateur.

Le Centre de recherche Licef¹ mène depuis 1992 différents travaux de recherche liés à la construction de systèmes conseillers. Les systèmes conseillers construits visent deux types d'utilisation :

– Construction de systèmes d'apprentissage. Dans un système d'apprentissage, le rôle du système conseiller est d'analyser les interactions entre l'apprenant et le système et d'apporter une aide à l'apprenant. L'aide peut viser directement un apprentissage ou une remédiation, le système intégrant pour cela des fonctionnalités de diagnostic et des connaissances pédagogiques et didactiques [WEN 87], ou viser un support de type « organisationnel » au niveau de l'organisation des activités que l'apprenant doit effectuer. C'est sur ce type d'aide que les travaux présentés ici se focalisent (conseils liés à la tâche que doit réaliser l'apprenant, par opposition à des conseils liés à un « diagnostic cognitif » de l'apprenant). Ce type de système conseiller peut être ajouté à un logiciel existant par ailleurs (par exemple, à un traitement de texte ou un tableur), intégré comme l'un des composants d'un tuteur intelligent classique (i.e. un système d'enseignement installé sur une machine particulière), mais également dans des dispositifs d'enseignement à distance via le Web tels que pratiqués à la Télé-Université.

– Construction d'ateliers d'ingénierie pédagogique. Au sein du Licef ont été élaborées plusieurs méthodes d'ingénierie de systèmes d'apprentissage, dont la méthode MISA. Ces méthodes permettent une approche rationnelle de la construction de systèmes d'apprentissage ou de formation de grande échelle ; elles ont été commercialisées dans des sociétés comme la Banque de Montréal, Hydro-Québec ou des services de formation de l'armée canadienne, et sont utilisées au sein de la Télé-Université. Associé au guide méthodologique papier présentant les différentes phases de la méthode, un atelier logiciel supporte les différentes étapes du processus de construction du système d'apprentissage. Il contient des éditeurs facilitant la modélisation des connaissances enseignées, des éditeurs permettant la

1. Le Licef (Laboratoire d'Informatique Cognitive et d'Environnement de Formation) est le Centre de recherche de la Télé-Université du Québec. Cette Université pratique exclusivement l'enseignement à distance, en utilisant de façon croissante les Nouvelles Technologies de l'Information et de la Communication (NTIC), en particulier le Web.

définition des événements d'apprentissage et leur structuration en différents scénarios, ainsi que la définition des publics cibles, de leur niveau d'habileté de départ et du niveau visé, etc. Au sein de ce type d'atelier d'ingénierie pédagogique, le rôle du système conseiller est d'analyser les interactions entre l'utilisateur (i.e. le concepteur du système d'apprentissage) et l'atelier logiciel pour aider cet utilisateur à mener à bien sa tâche, en l'occurrence à concevoir son système d'apprentissage.

Dans son activité de conception de systèmes d'apprentissage, le Licef (la Télé-Université) est actuellement amené à élaborer des systèmes conseillers pour les environnements d'apprentissage sur Internet diffusés par la plate-forme Explor@ [PAQ 00]. Par ailleurs, des travaux sont entrepris pour développer un système conseiller pour l'atelier d'ingénierie pédagogique ADISA [PAQ 00], qui est destiné à supporter l'utilisation de la méthodologie d'ingénierie des systèmes d'apprentissage MISA [PAQ 97]. Les travaux présentés dans cet article visent à élaborer un cadre méthodologique commun facilitant la construction de ces systèmes conseillers, selon une approche fondée sur l'analyse de la tâche que doit réaliser l'utilisateur (respectivement, l'apprenant qui suit une formation particulière via Explor@, l'enseignant qui construit un cours avec ADISA).

La construction d'un système conseiller comporte des difficultés liées à la mise en œuvre technique du système et des difficultés liées à l'identification des conseils. L'aspect technique concerne la construction et l'implantation de l'architecture informatique du système conseiller, et, en particulier, le problème de l'espionnage des actions de l'utilisateur. La maîtrise de ces aspects techniques (qui sont loin d'être triviaux) est bien évidemment une condition *sine qua non*. Sur ce point, les travaux du Licef ont conduit à la construction de l'architecture de développement de systèmes conseillers EpiTalk [GIR 95, PAQ-c 96]. Cette architecture a été utilisée avec succès pour plusieurs applications, en particulier pour construire le système conseiller de l'atelier de génie didactique AGD (prédécesseur de ADISA) [PAQ-b 96] et divers conseillers de cours à distance diffusés sur Internet à l'aide du système Explor@ [GIR 99].

L'identification des conseils qui seront embarqués dans le système (et que celui-ci pourra donc délivrer) est le second aspect du problème, et la clé de la valeur ajoutée par le système conseiller. Si les conseils effectivement intégrés au système doivent prendre en compte l'usage « en situation » du système et non simplement son usage théorique, il se pose cependant un problème initial, l'identification de ce que l'on peut appeler les « conseils candidats », c'est-à-dire les conseils directement liés à la tâche². A ce niveau, les deux contextes d'application que nous visons (construction de systèmes d'apprentissage et construction d'ateliers d'ingénierie

2. De façon cohérente avec les différents travaux d'ingénierie menés au Licef, notre approche est résolument fondée sur la tâche à réaliser par l'acteur et non sur une étude ergonomique de l'acteur en situation. Cette hypothèse ne signifie pas que nous construisons un système conseiller sans tenir compte de son usager, mais que l'identification des conseils (qui constitue en quelque sorte la « matière première ») est fondée sur l'analyse de la tâche que doit réaliser l'utilisateur.

pédagogique) se rejoignent sur le fait que les conseils sont liés à une tâche que doit réaliser l'utilisateur. Pour les environnements d'apprentissage sur Internet diffusés par Explor@, on dispose d'une modélisation de la structure pédagogique (modules, unités d'apprentissage et activités) et des différentes interrelations et contraintes régissant l'activité de l'apprenant (niveaux de progression et transitions entre niveaux). Pour l'atelier d'ingénierie pédagogique ADISA, on dispose d'une modélisation de la tâche de conception de cours, c'est la méthode MISA. Dans les deux cas, une difficulté intrinsèque est que les conseils qui doivent être intégrés dans le système conseiller ne sont généralement pas directement disponibles auprès d'experts de la tâche considérée ou de livres, ils doivent être construits, en prenant en compte la légalité et la pertinence des actions de l'utilisateur par rapport à la tâche à réaliser. Identifier quelques conseils est alors relativement simple et rapide, mais élaborer un ensemble de conseils complet, non redondant et non contradictoire est beaucoup plus délicat. Dès que l'on s'intéresse à un système conseiller non trivial, une démarche méthodologique est absolument nécessaire.

Ainsi, le travail présenté ici a notamment été réalisé pour entreprendre la construction du système conseiller de l'atelier distribué d'ingénierie de systèmes d'apprentissage ADISA, qui est destiné à supporter l'utilisation de la méthodologie MISA. Pour donner un ordre de grandeur, le guide méthodologique de MISA [PAQ 97] fait plus de 300 pages, les principales phases de la méthode sont décrites par trente-cinq tâches qui se décomposent elles-mêmes en sous-tâches, soit plus de 150 tâches au total. Pour reprendre les métaphores couramment utilisées pour les Systèmes à Base de Connaissances [AUS 96], la construction du système conseiller associé ne relève donc pas d'un « artisanat » : une approche méthodologique est absolument nécessaire.

Dans cet article, nous présentons une approche qui vise à guider et à faciliter l'identification des conseils³. L'approche proposée est fondée sur l'analyse d'une modélisation de la tâche considérée et une typologie de conseils. Notre hypothèse (fonder l'identification des conseils sur la tâche) est liée à la qualité des modèles de tâches que nous utilisons dans ces contextes, qui font l'objet d'analyses soigneuses [PAQ 00] et sont considérés comme une base suffisante pour construire des conseils pertinents (rappelons qu'il s'agit de « conseils » et non de contraintes). L'objectif est d'identifier (et d'opérationnaliser à terme) une démarche méthodologique permettant de systématiser l'identification de « conseils candidats ». Nous pensons en effet que la généralisation des services en accès direct sur le Web (enseignement par distance, bases de ressources, etc.) va rendre de plus en plus pressante la construction de systèmes conseillers de première ligne, et justifier l'élaboration d'une méthodologie de construction de ces systèmes. A un niveau général, l'approche proposée peut être

3. Il s'agit donc d'une approche « constructive » (les conseils vont être élaborés à partir de la tâche à réaliser), par opposition à une approche « interprétative » où les conseils seraient élaborés à partir d'un modèle d'intervention prédéfini (cf., en ingénierie des connaissances, l'opposition entre les approches constructives, par exemple Mapcar [TCH 97], et les approches interprétatives, par exemple Kads [SCH 93]).

appliquée à la conception de systèmes conseillers destinés à des environnements informatiques servant de support à la mise en œuvre d'une tâche complexe. De façon plus précise, les travaux présentés ici sont destinés à faciliter, d'une part, la construction de systèmes conseillers destinés à des environnements de support à la tâche diffusés sur le Web avec Explor@, et, d'autre part, les agents conseillers destinés aux concepteurs utilisant la méthode d'ingénierie de systèmes d'apprentissage (MISA) et son système de support sur Internet, ADISA. On profite alors du fait que, dans les deux contextes, les tâches mises en jeu sont modélisées à l'aide du langage graphique de modélisation MOT. De fait, les travaux présentés ici sont fondés sur et poursuivent les travaux liés à MOT et Explor@ préalablement publiés dans la revue STE ([PAQ 96], [PAQ 00]).

Afin de donner une compréhension intuitive du contexte d'application de nos travaux, nous présentons rapidement en section 2 la place et le rôle d'un système conseiller dans un environnement de support à la tâche diffusé sur le Web avec Explor@. Nous détaillons ensuite la démarche que nous proposons pour construire un système conseiller (dans le contexte de ADISA) : tout d'abord, en section 3, les principes de l'approche proposée, puis, en section 4 et 5, les différentes phases que nous proposons de suivre, en indiquant comment des outils d'analyse du modèle de la tâche considérée peuvent contribuer à l'identification des conseils. Ces différentes phases peuvent être vues comme un « guide méthodologique » à adapter en fonction des éventuelles spécificités du domaine considéré. La présentation de ces phases est faite à un niveau abstrait afin de préserver son caractère de généralité à la démarche proposée.

Dans la suite de cet article, l'expression « tâche considérée » désigne la tâche complexe que l'utilisateur doit réaliser (tâche qui se décompose, récursivement, en sous-tâches), et l'expression « système hôte » désigne le logiciel ou système d'aide à la tâche sur lequel le système conseiller va se greffer. Ainsi, dans le cadre de MISA, la tâche considérée est « construire un système d'apprentissage en suivant les préceptes de la méthodologie MISA » et le système hôte est l'atelier logiciel ADISA.

2. Conseil dans les environnements de support à la tâche

Le système Explor@ [PAQ 00] permet de construire un centre virtuel d'apprentissage chargé de diffuser une banque d'événements d'apprentissage en faisant en sorte que les usagers partagent un ensemble de ressources pédagogiques (outils, fichiers, moyens de communication, documents, etc.) maintenues par l'institution de formation ou par l'entreprise. Ces ressources sont sélectionnées par les concepteurs d'un système d'apprentissage, en fonction des particularités de chaque cours et des différentes tâches assumées par les apprenants, les formateurs, les experts de contenu, les gestionnaires et les concepteurs du cours. Explor@ permet donc de construire un ou plusieurs environnements de support à la tâche pour chaque catégorie d'acteur d'une formation à distance.

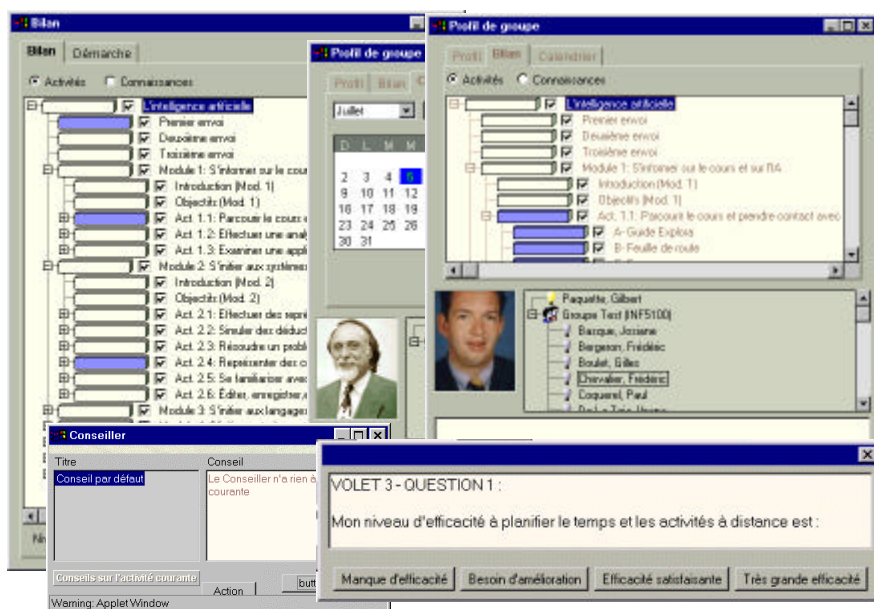


Figure 1. Des outils d'assistance/conseil en regard d'une arborescence des tâches ou des connaissances

Explor@ vise à faciliter l'intégration à un événement d'apprentissage sur le Web d'outils de suivi et de conseil facilitant l'apprentissage individuel et collaboratif, le support pédagogique et la gestion de la formation. C'est le cas notamment des outils bilan, calendrier, profil de groupe ou agent conseiller présentés en Figure 1. Cette intégration d'outils d'assistance « intelligente » nécessite trois opérations principales soutenues par des outils Explor@ destinés spécifiquement au concepteur de l'environnement.

i). Description de la structure des activités et des connaissances. À partir d'un modèle des connaissances et d'un modèle pédagogique du cours, on décrit deux structures hiérarchiques qui seront intégrées à la base de données d'Explor@. La structure pédagogique présente la décomposition des tâches du cours (modules, unités d'apprentissage, activités), dont les entités sont appelées unités pédagogiques (UP). La structure cognitive représente les connaissances traitées dans le cours ; ses entités sont appelées unités cognitives (UC). Des pages d'un ou plusieurs sites Web sont associées à chaque UP et à chaque UC.

Ces deux opérations rendent opérationnel un outil comme le calendrier ou la feuille de route interactive. Dans ce type d'outil, l'utilisateur choisit une date s'affichant dans une couleur spéciale lorsqu'il a utilisé l'environnement. Il obtient alors la liste

des pages visitées ce jour-là, identifiées par le nom de l'activité associée ou par le nom de la connaissance associée à la page. Une fonction permet d'afficher la page correspondante dans le site Web.

ii). Définition des niveaux de progression et des transitions entre niveaux. Pour chaque UP et chaque UC, on définit des niveaux de progression, à l'aide de gabarits prédéfinis ou par l'énoncé de conditions. Le nombre de niveaux d'une UC ou d'une UP dépend de la complexité de l'activité ou de la connaissance. Pour une activité simple, comme l'envoi d'un travail noté à un correcteur, on ne prévoira que deux niveaux : envoyé ou non envoyé. Dans d'autres cas, on définira un plus grand nombre de niveaux. Par exemple, si un apprenant doit lire deux textes, visionner une vidéo et passer un test pour compléter une activité, on aura besoin d'au moins quatre niveaux pour l'activité correspondante, sans doute plus si l'on veut tenir compte de différents niveaux de résultats au test.

Lorsque le concepteur a défini les différents niveaux pour les UC et les UP qu'il veut voir soutenus par le système Explor@, d'autres outils de l'utilisateur deviennent opérationnels et peuvent être intégrés dans les environnements. C'est le cas notamment des outils « Bilan » et « Profil de groupe » présentés en Figure 1. Le bilan présente à l'utilisateur la structure des activités du cours, ainsi que la structure des connaissances, accompagnées de bâtonnets indiquant le niveau de progression de l'utilisateur. Cet outil lui permet d'évaluer régulièrement sa progression dans le cours et d'apporter les correctifs qui s'imposent. Les membres d'un groupe peuvent ainsi comparer leur progression à celle d'autres usagers en utilisant l'outil « Profil de groupe ». Ce dispositif permet d'identifier d'autres apprenants avec lesquels il serait intéressant de travailler en équipe. Il facilite également aux formateurs la tâche d'établir un diagnostic sur la progression du groupe et d'apporter des correctifs si nécessaire.

iii). Définition des conseils. Il est possible de créer un environnement encore plus interactif ou interventionniste par la définition d'agents conseillers qui gèrent des conseils relatifs à certains niveaux de progression atteints par l'utilisateur dans une ou plusieurs unités pédagogiques ou cognitives. Les conseils peuvent prendre la forme de courts textes affichés à l'écran. Ils peuvent aussi déclencher un fichier externe (messages sonores ou visuels) ou démarrer un dialogue avec l'utilisateur menant à un conseil (tel qu'indiqué sur la Figure 1). Les conseils peuvent s'afficher à l'initiative de l'utilisateur en choisissant l'option appropriée du menu assistance de l'environnement Explor@ ou à l'initiative du système, lorsque certaines conditions sont réunies.

Cet exemple illustre comment s'intègre un système conseiller dans un environnement d'apprentissage sur Internet diffusé par Explor@. La structure pédagogique (modules, unités d'apprentissage et activités) et les différentes interrelations et contraintes entre les composants de l'environnement que doit prendre en compte l'apprenant (niveaux de progression, transitions entre niveaux, etc.) sont finement modélisées. Les interrelations sont nombreuses et complexes, et

le rôle du système conseiller est d'aider l'apprenant dans la gestion de ses activités. Ainsi, le système conseiller va par exemple indiquer à un apprenant qu'il est en train de démarrer l'activité X alors qu'il n'a pas atteint un niveau suffisant dans des activités préalables ou une évaluation satisfaisante dans certaines UC.

Les outils auteurs d'Explor@ permettent d'intégrer ce type d'agent conseiller aux environnements d'apprentissage sur Internet qu'Explor@ permet de diffuser. Le problème est l'identification des conseils potentiellement pertinents, problème dont la difficulté est liée à la taille et à la complexité des structures hiérarchiques modélisant le cours. Notre objectif est de passer d'une façon de procéder totalement *ad hoc* à chaque système à un processus guidé par une démarche méthodologique.

Par ailleurs, l'équipe du Licef a entrepris d'utiliser le même type de démarche et les mêmes outils pour définir les agents conseillers destinés aux concepteurs de cours utilisant la méthode d'ingénierie de systèmes d'apprentissage (MISA) et son système de support sur Internet, ADISA, atelier logiciel qui supporte les différentes étapes du processus de construction des systèmes d'apprentissage⁴. Comme nous le verrons plus loin, la méthode MISA décrit un graphe des tâches du concepteur à l'aide d'un modèle de connaissances très élaboré, qui n'est pas un simple arbre des tâches. Il faut alors recourir à des analyses de structure extrêmement complexes, du type de celles que nous allons maintenant présenter.

3. Principes de l'approche proposée

3.1. Le système conseiller est un méta-système

Les travaux sur EpiTalk, et plus récemment sur Explor@-conseil, ont montré l'intérêt de définir le système conseiller comme une extension réflexive du système hôte plutôt que d'intégrer les conseils « en dur » dans ce dernier. Cette approche permet en particulier de greffer un système conseiller à un système déjà existant et contribue à la portabilité, l'évolution et la maintenance du système conseiller. EpiTalk est défini comme une architecture épiphyte, qui peut être greffée sur une

4. L'atelier distribué d'ingénierie d'un système d'apprentissage (ADISA), est un système de support à la conception destiné aux utilisateurs de la méthode MISA 4.0, accessible en ligne ou en local via un fureteur Web. ADISA facilite la construction des plans et devis d'une téléformation. Il regroupe un ensemble de 35 outils interreliés. Le format DHTML est utilisé pour créer des formulaires dynamiques qui se modifient en fonction des choix du concepteur. Une fois remplis, ces formulaires, tout comme les modèles graphiques MOT, sont stockés sous la forme de fichiers en format XML, ce qui permet la propagation de données entre les outils. ADISA contient un grand nombre de connaissances pédagogiques provenant de la méthode MISA, notamment 17 typologies d'objets pédagogiques ayant chacune fait l'objet d'une recherche approfondie et recouvrant les principales composantes d'un système de télé-apprentissage.

application quelconque pour espionner et raisonner sur les actions de l'utilisateur [GIR 95].

3.2. L'identification des conseils est fondée sur une modélisation de la tâche

D'un point de vue technique, le déclenchement d'un conseil est lié aux actions de l'utilisateur, et donc aux fonctionnalités du système hôte. Nous pensons cependant que l'identification des conseils ne doit pas être fondée sur l'analyse des fonctionnalités du système hôte, mais sur une modélisation de la tâche que l'utilisateur doit réaliser avec le système hôte : c'est sur MISA (qui constitue de modèle de la tâche) et non sur ADISA (qui est l'atelier logiciel de mise en œuvre de la tâche) qu'il faut se baser.

Tout d'abord, l'analyse des fonctionnalités du système hôte n'est pas un bon support à l'identification de conseils prenant en compte le contexte et les objectifs des actions de l'utilisateur. D'une part, les considérations méthodologiques du type « telle action devrait être effectuée avant telle autre » ont un impact sur la conception du système hôte, mais ces considérations (et leurs justifications) ne sont généralement pas explicitement représentées dans celui-ci. Les retrouver sur la base des fonctionnalités du système hôte s'apparente alors à un périlleux exercice de « reverse-engineering », et ceci d'autant plus que le système hôte n'est qu'une implantation particulière, qui prend en compte un ensemble de spécifications liées à la tâche considérée mais également des considérations techniques, d'ergonomie (etc.). Par ailleurs, fonder la construction du système conseiller sur une modélisation de la tâche à réaliser plutôt que sur le système hôte permet de prendre en compte, au sein de ce système conseiller, la diversité des usages des fonctionnalités du système hôte. Dans le contexte d'une méthodologie comme MISA, ceci permet de prendre en compte le fait que MISA est un guide général, qui propose un certain nombre de principes d'adaptation [PAQ 97] permettant d'adapter la méthode. Ainsi, si le type de support médiatique utilisé par le système d'apprentissage à construire est imposé et est une contrainte identifiée dès le départ, la façon dont certaines tâches de MISA doivent être réalisées est modifiée, ceci étant décrit dans le guide méthodologique ; bien que l'atelier logiciel soit inchangé, les conseils délivrés doivent bien évidemment être relatifs aux tâches « adaptées ».

Dans nos travaux, nous utilisons le langage graphique de modélisation MOT⁵ pour modéliser la tâche considérée (les systèmes d'apprentissage diffusés avec Explor@, la méthode MISA sont modélisés en MOT). Ce langage a déjà été décrit en détail dans la revue STE [PAQ 96]. Rappelons que MOT propose trois types de connaissances (concepts, procédures et principes), trois types de faits (exemples, traces, énoncés) résultant de leur instanciation et sept types de relations

5. MOT est un logiciel commercialisé, accessible aux organismes d'enseignement et de recherche à partir du site du Licef.

(Composition, Spécialisation, Instanciation, Précédence, Intransit/Extransit, Régulation et Application). Ces types de connaissances et ces relations ont été retenus sur la base des résultats de différents travaux en psychologie cognitive, en ingénierie des connaissances et en sciences de l'éducation, notamment les travaux de [ALE 87], les travaux autour de Kads [SCH 93] et, en sciences de l'éducation, les travaux de [MER 94], [ROM 81], [TEN 90] ou [WES 91]⁶. En MOT, une tâche complexe se modélise sous la forme d'un graphe où les sous-tâches à réaliser sont décrites sous la forme de procédures, les entrées/sorties des tâches sont décrites sous la forme de concepts reliés aux procédures par des liens de type Intransit-Extransit (I/P), et les liens de Précédence (P) sont utilisés pour décrire l'enchaînement des tâches (on utilisera tâche et procédure de façon synonymique). Un concept est lié par un lien de composition (C) aux attributs qui le décrivent, chaque attribut étant associé à une spécification des valeurs qu'il peut prendre. Cette spécification peut être réalisée en extension, sous la forme d'une liste de valeurs reliées à l'attribut par un lien d'Instanciation (I), ou en compréhension, sous la forme d'un principe. Les tâches et les produits complexes sont décrits par des sous-modèles (récursivement).

La Figure 2 présente la modélisation MOT de l'une des 35 tâches de base MISA, la construction d'un tableau des compétences. Dans ces modèles, les composantes du devis du système d'apprentissage sont des objets représentés par des concepts (rectangles) et ils sont produits à l'aide de tâches représentées par des procédures (ovales). Le sous-modèle représenté ici est centré sur la tâche « 2.3 Identifier les écarts de compétence à combler » qui produit le devis « 214 – Tableau des compétences ». Cette tâche a comme intrants d'autres devis : 102, 104, 108, 210, 212 et 542. Le produit de cette tâche sera ensuite réutilisé comme intrant d'autres tâches, qui permettront de produire les devis 220, 222, 224, 310, 320, 610, 630 et possiblement, amèneront des modifications au « 212 – Modèle des connaissances ».

6. MOT est un langage de modélisation informel élaboré pour répondre aux besoins des concepteurs non-informaticiens qui désirent construire un système d'apprentissage ou un système d'aide à la tâche. Ce type de système est généralement constitué d'un certain nombre d'outils dont les relations doivent être précisées relativement à un ensemble de tâches à accomplir. Le concepteur a besoin d'un formalisme intégré, facile d'accès et capable de l'aider à construire une vue d'ensemble cohérente des principaux processus, concepts et stratégies qui décrivent une classe de problèmes à résoudre. Dans de nombreux formalismes de représentation des connaissances développés en intelligence artificielle ou encore en UML, qui tend à s'imposer comme la *lingua franca* du génie logiciel, un système d'information est spécifié au moyen de plusieurs modèles destinés à représenter différents points de vue sur le système. Le formalisme MOT, qui se destine à des applications éducatives, vise au contraire un traitement intégré de deux niveaux d'abstraction, les faits et les connaissances abstraites, celles-ci étant à leur tour subdivisées en trois types (concepts, procédures et principes). L'intégration de ces divers types de connaissances dans un même modèle est apparue nécessaire puisque, selon le type de connaissance, le traitement pédagogique sera différent.

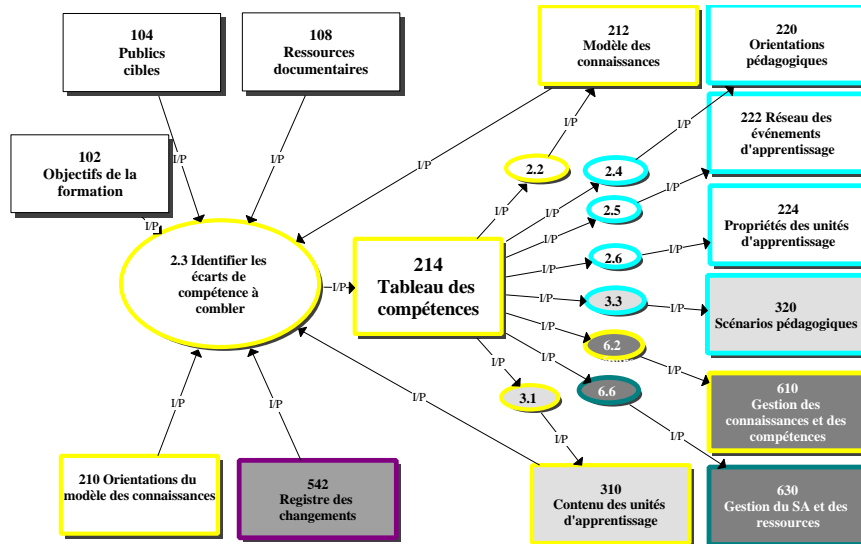


Figure 2. Modélisation MOT d'une tâche de MISA

Ce modèle représente un niveau intermédiaire du modèle global de la méthode MISA. Au niveau supérieur se trouvent les six phases de la méthode dont la deuxième, « 2 – Établir une solution préliminaire », laquelle se compose notamment des tâches 2.2, 2.3, 2.4, 2.5 et 2.6 représentées sur la Figure 2. À un niveau inférieur, chaque tâche est subdivisée en sous-tâches visant à construire les composantes de chaque devis. Par exemple, la tâche 2.3 sera subdivisée en sous-tâches permettant d'établir, pour chaque connaissance principale du modèle des connaissances (212), un énoncé de la compétence actuelle et de la compétence visée pour chaque public cible établi dans le devis 104.

3.3. L'apport d'outils d'analyse du modèle de la tâche

Un modèle MOT est une structure de donnée qui peut être analysée par des outils informatiques. Il est donc possible d'utiliser des outils qui prennent en entrée un graphe MOT et produisent en sortie des informations relatives à un point de vue particulier sur le graphe. Ces outils ne créent pas d'informations supplémentaires puisque, par définition, les informations qu'ils calculent sont représentées dans le graphe. Ils permettent simplement de mettre en évidence et de structurer des informations qui, parce qu'elles correspondent à un point de vue particulier, ne sont pas facilement accessibles avec les outils standards de visualisation.

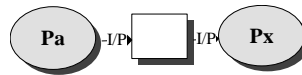
Dans le contexte de la construction d'un système conseiller, l'analyse du graphe modélisant la tâche considérée permet d'identifier systématiquement les composants

à prendre en compte dans le système conseiller (les différents concepts, attributs et procédures), et, surtout, de mettre en évidence leurs différentes interactions. Ainsi, étant donnée une procédure Px, l'analyse du graphe permet de mettre en évidence les procédures qui précèdent Px, les sous-procédures que la réalisation de Px va nécessiter, les procédures qui succèdent à Px, les procédures qui ne sont pas liées par un lien de précedence à Px, l'influence de Px sur d'autres procédures (...), ou encore les concepts pré-requis à Px. Étant donné un concept Cx, l'analyse du graphe permet de mettre en évidence les procédures où Cx est un extrant et les procédures où Cx est un intrant, en indiquant les autres concepts associés à Cx comme intrants ou extrants et, dans le cas où la procédure concernée est une sous-procédure, la procédure « mère » (l'analyse permet de situer et de structurer l'information). Ces différentes informations sont les données sur la base desquelles on va identifier les conseils visant à aider l'utilisateur dans le choix des procédures à réaliser, à l'empêcher de commencer une procédure pour laquelle il n'a pas les ressources (intrants) nécessaires, etc.

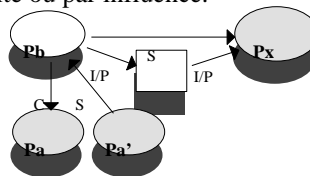
L'utilisation d'outils d'analyse n'est pas simplement utile, elle est indispensable. Ainsi, dans un modèle complexe, identifier l'ensemble des procédures qui précèdent une procédure Px n'est en rien trivial. Les liens explicites (lien P entre Pa et Px) indiquent les principales relations de précedence, mais il existe également le plus souvent des relations indirectes beaucoup plus délicates à déceler⁷.

A titre d'exemples, on peut mentionner [TCH 98] les cas suivants :

– Les liens par influence : Pa précède Px parce que Pa a des extrants qui sont des intrants de Px ; c'est ainsi que sont dénotés les liens de précedence dans la Figure 2.

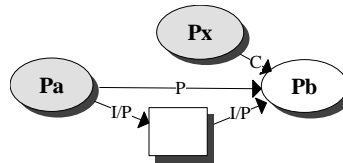


– Les liens hérités : Pa et Pa' précèdent Px parce que Pa est une sous-procédure (lien C) et Pa' une spécialisation (lien S) de Pb et Pb précède Px, que le lien de précedence Pb-Px soit explicite ou par influence.

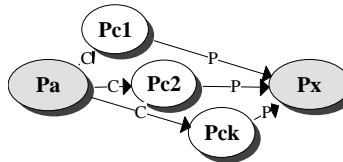


7. Le fait qu'il y ait dans un graphe MOT des relations d'influence indirectes n'est pas une limitation du langage de modélisation : pour un modèle complexe, expliciter toutes les relations ne serait ni possible ni souhaitable, il faut pouvoir se focaliser sur certains concepts et/ou certaines relations.

– Les liens propagés : Pa précède Px parce que Px a une sous-procédure Pb, Pa précède Pb (lien explicite ou par influence) et Pa n'est pas uniquement liée à des sous-procédures de Px.



– Les liens *par factorisation* : Pa précède Px parce que toutes les sous-procédures Pci de Pa précèdent Px ; il s'agit là d'un cas particulier de lien propagé.



L'une des difficultés de la construction d'un système conseiller est d'identifier les relations indirectes entre composants et leurs implications (liens de précédence à respecter, incohérences, etc.). Analyser le graphe MOT de la tâche considérée n'assure en rien d'identifier tous les conseils pertinents, mais c'est cependant la meilleure référence disponible. Utiliser des outils d'analyse automatique permet de tirer le meilleur parti possible de cette ressource, et, en particulier, d'identifier les relations indirectes.

3.4. Une démarche en trois phases

Nous proposons de considérer trois phases dans la construction d'un système conseiller. La Phase A, présentée en section 4, vise à construire ce que nous appelons l'ontologie du système conseiller, c'est-à-dire à identifier et définir les composants que l'on va considérer dans le système conseiller et comment on va les considérer. La Phase B, présentée à la section 5, vise à identifier individuellement les conseils. La Phase C vise à modéliser la tâche « conseiller » (structuration des conseils et modélisation des principes de sélection des conseils). Nous nous focalisons ici sur les deux premières phases, la dernière, qui fait l'objet de travaux en cours, étant évoquée en section 6.

Dans la présentation ci-dessous, nous indiquons les principes et notions de base de notre approche. Dans un souci de généralité, nous ne proposons pas un vocabulaire de description des conseils ni une typologie « prêts à l'emploi », mais des indications et exemples qui peuvent être repris et/ou adaptés lors de la construction d'un système conseiller particulier.

4. Phase A : construction de l'ontologie⁸

L'objectif de cette étape est d'identifier et de décrire les composants de la tâche considérée et les notions qui vont être pris en compte dans le système conseiller.

4.1. Identification et définition des composants de la tâche considérée

La première étape consiste à identifier l'ensemble des procédures et des concepts de la tâche considérée. Cette liste peut être identifiée par l'analyse du graphe MOT. Il est possible de prendre en compte l'ensemble des procédures et des concepts ou de se restreindre à un sous-ensemble.

La seconde étape consiste à définir précisément les différents concepts, c'est-à-dire les objets produits et utilisés par les procédures (par exemple, dans MISA, les concepts « réseau des événements d'apprentissage » ou « scénario pédagogique des unités d'apprentissage » de la Figure 2). Pour chacun des concepts (et sous-concepts associés), il s'agit d'identifier les attributs le caractérisant puis, pour chacun de ces attributs, identifier les valeurs qu'il peut prendre. Si la modélisation MOT qui est disponible est suffisamment précise, ces informations sont directement accessibles dans le modèle (liens de composition, instanciation et régulation), sinon la modélisation MOT doit être complétée.

Les relations MOT définissent implicitement des groupes de procédures et de concepts sémantiquement liés, par exemple l'ensemble des sous-tâches d'une tâche, l'ensemble des produits d'une tâche, etc. Identifier ces regroupements est important pour l'étude de la cohérence entre concepts (cf. section 5.2). Les regroupements liés à des relations MOT peuvent être retrouvés automatiquement. Cependant, certaines procédures et/ou concepts, sans être explicitement liés entre eux par des relations MOT, peuvent avoir des liens sémantiques (même type d'objet, même objet mais considéré selon différents points de vue, etc.). La dernière étape consiste à identifier les regroupements pertinents de ce type. Dans la suite de cet article, le terme « composant » désignera indifféremment les concepts, les procédures et les regroupements.

4.2. Définition des notions prises en compte dans le système conseiller

L'objectif de cette étape est d'identifier comment, dans le système conseiller, on veut pouvoir qualifier les composants identifiés précédemment. Il s'agit donc de définir un vocabulaire abstrait adapté à la tâche considérée, à ses composants et aux types de conseils que l'on veut délivrer. Il est important de noter que le vocabulaire

8. Le terme « ontologie » ne désigne ici rien d'autre que la définition (structurée mais informelle) des concepts de la tâche considérée et d'un vocabulaire (ensemble de termes) permettant de dénoter certaines caractéristiques de ces concepts.

choisi induit un point de vue particulier, et que, implicitement, il sert de guide à la définition des conseils. Cette étape, si elle ne consiste qu'à définir le vocabulaire que l'on va utiliser par la suite, est donc cependant extrêmement importante.

Nous proposons de dissocier les notions d'importance, de statut et de qualité.

4.2.1. *Notion d'importance*

La notion d'importance dénote le degré de nécessité de la réalisation d'une tâche ou de la production d'un produit. Cette notion va par exemple permettre d'indiquer que certains composants, plus importants que d'autres, doivent absolument être produits pour qu'une tâche soit considérée comme réalisée, ou encore que, dans un certain contexte d'adaptation de la tâche, certains composants ne sont pas pertinents et n'ont pas à être considérés. Le vocabulaire utilisé pour décrire l'importance d'un composant est générique (il n'est pas spécifique de la tâche considérée, il est possible d'utiliser des termes comme « peu important », « essentiel », etc.). Selon la tâche considérée, il peut s'avérer utile d'utiliser plus ou moins de termes, dénotant différentes graduations.

On définira successivement un vocabulaire permettant de décrire l'importance d'un attribut en tant que caractéristique d'un concept (ensemble de termes comme « doit absolument être valué », « peut ne pas être valué », etc.), l'importance d'un concept en tant que ressource ou produit de la tâche (ensemble de termes comme « non pertinent », « doit absolument être défini », « peut ne pas être défini », etc.), l'importance d'une (sous-)tâche (ensemble de termes comme « non pertinent », « doit absolument être réalisé », « peut ne pas être réalisé », etc.) et, éventuellement, l'importance d'un regroupement.

4.2.2. *Notion de statut*

La notion de statut dénote le fait qu'un composant a été considéré (au moins une des actions effectuées par l'utilisateur a trait à ce composant). Cette notion va par exemple permettre d'indiquer qu'une tâche n'a pas été commencée ou qu'un produit est défini. La notion de statut est donc liée à la réalisation de la tâche considérée (le statut d'un composant va être modifié au fur et à mesure de la réalisation de la tâche). La notion de statut permet de qualifier l'état d'avancement d'une tâche ou de définition d'un produit à un niveau abstrait indépendant de la sémantique des composants considérés. Ici encore, le vocabulaire utilisé est donc générique, mais il peut s'avérer utile d'utiliser différentes graduations et/ou sémantiques. Le cas échéant, il peut être intéressant de dissocier différents types de produits ou de tâches, et d'associer des vocabulaires différents à chaque type.

On définira successivement un vocabulaire permettant de décrire les différents statuts possibles d'un attribut (ensemble de termes comme « valué », « pas encore valué », etc.), les différents statuts possibles d'un concept (ensemble de termes comme « pas encore défini », « partiellement défini », « complètement défini », etc.), les différents statuts possibles d'une tâche (ensemble de termes comme « non encore

considéré », « commencé », « presque terminé », « terminé », etc.) et, éventuellement, les différents statuts possibles d'un regroupement.

4.2.3. Notion d'état

La notion d'état dénote un point de vue abstrait sur le niveau de réalisation d'une tâche en fonction de ses objectifs, et donc de façon spécifique à la sémantique de la tâche. De façon similaire à la notion de statut, la notion d'état est liée au niveau de réalisation de la tâche. En revanche, le vocabulaire n'est pas générique, mais spécifique de la tâche. Ainsi, dans MISA, à la tâche « Etablir la solution préliminaire (phase 2) » (du système d'apprentissage à réaliser) sont associés les états « besoins établis », qui dénote le fait que le modèle de connaissance (212) du domaine enseigné est terminé et que le tableau des compétences a été défini, et « solution préliminaire complétée », qui dénote le fait que, de plus, le réseau des événements d'apprentissage et les autres devis de la phase 2 ont été construits. Le vocabulaire utilisé doit donc être défini tâche par tâche, lors de la Phase B.

La notion de statut, qui permet un niveau de description assez général, est utilisée pour (par exemple) décrire des conseils incitant l'utilisateur à poursuivre une tâche presque terminée ou l'autorisant à commencer une tâche dont les intrants sont définis. Seule, la notion de statut n'est pas suffisante, car une tâche peut être à des niveaux de réalisation qui lui sont propres et qui ne sont pas dénotables par des termes génériques sans qu'il y ait perte d'information trop importante. Par exemple, quand une tâche comme « Etablir la solution préliminaire (phase 2) » produit plusieurs extrants, dire que son statut est presque terminé est une information trop générale pour être suffisante, alors qu'évaluer son état à « besoins établis » permet de délivrer un conseil précis tel que « examiner si l'on peut regrouper des publics cibles ». Seule, la notion d'état n'est cependant pas suffisante, car pour décrire (par exemple) un conseil fondé sur le simple fait qu'une tâche est commencée, on serait obligé de faire intervenir des niveaux de détails inutiles. Les notions de statut et d'état sont donc complémentaires.

4.3. Principes de valuation et de propagation

4.3.1. Importance

En général, l'importance d'un composant n'est pas intrinsèque, elle est contextuelle à une réalisation particulière de la tâche considérée. L'importance d'un composant doit donc être définie par rapport à un contexte décrivant l'importance attribuée aux autres composants (concepts, tâches, regroupements) et leur statut. Par ailleurs, il est généralement possible d'identifier des règles de composition définissant les valeurs de certains composants par rapport à d'autres, ce qui permet de ne se focaliser que sur un sous-ensemble des composants.

On commencera donc par identifier les composants clés (tâches / concepts / regroupements) qui définissent les principes d'adaptation principaux de la tâche⁹ et qui déterminent l'importance des autres composants. On définira ensuite les règles générales de composition d'importance (par exemple, « une tâche dont certains extrants doivent absolument être définis doit absolument être réalisée » ou « les concepts extrants d'une tâche non pertinente sont non pertinents »). On établira enfin l'importance relative des composants, selon un format de type :

*Dans un contexte où
liste-de {importance(concept/tâche/regroupement) = x}
et liste-de {statut(concept/tâche/regroupement) = y},
Alors importance(concept/tâche/regroupement) = z*

Les différentes ressources nécessaires à l'identification des relations d'importance (attributs d'un concept, extrants d'une tâche, sous-tâches d'une tâche, liens entre composants, etc.) sont accessibles par analyse du graphe. Dans la mesure où l'importance d'un composant est contextuelle, si cela est techniquement possible, il est beaucoup plus pertinent de définir dynamiquement (à l'exécution) l'importance des composants que de la définir une fois pour toute, lors de la construction du système conseiller. Ainsi, si le système conseiller détecte que, étant donné les actions de l'utilisateur, un composant particulier n'est plus pertinent, cette information peut être propagée pour éviter de conseiller une action importante *a priori*, mais qui ne l'est plus du tout ou plus autant dans le contexte courant.

4.3.2. Statut

La notion de statut est essentiellement « syntaxique », elle n'est ni contextuelle ni relative. Cependant, les statuts de certains composants sont liés.

On commencera par définir les règles établissant les statuts par défaut des composants (par exemple, « tous les attributs sont non valués » et « toutes les tâches sont non encore considérées »). On définira ensuite les règles de composition de statut entre un concept et ses attributs, par exemple « un concept dont tous les attributs sont définis est défini » ou « un concept dont tous les attributs importants sont définis est presque entièrement défini ». On définira enfin les règles de composition de statut entre une tâche et ses produits et entre une tâche et ses sous-tâches. Sur ce dernier point, on peut différencier deux façons de procéder. Une première approche consiste à établir des règles de composition entre les tâches terminales (les tâches qui ne sont pas décomposées en sous-tâches) et leurs extrants, puis d'établir des règles de composition entre les tâches non terminales et les tâches terminales. Une seconde approche consiste à établir directement des règles de

9. A titre d'exemple, dans MISA/ADISA, il est prévu d'identifier l'importance de certains de ces composants clés à partir d'un questionnaire permettant de caractériser les objectifs et contraintes de l'utilisateur (propriétés du système d'apprentissage visé, support médiatique imposé ou non, etc.), certains autres pouvant être déduits de ses actions.

composition entre les tâches complexes et leurs extrants (extrants directs et extrants des sous-tâches). Le choix entre ces deux approches peut être fondé sur l'étude de l'homogénéité des tâches (la seconde approche, plus globale, est peu pertinente si elle conduit à considérer indistinctement des composants qui sont sémantiquement peu liés).

D'un point de vue technique, pour établir le statut d'un composant, il faut mettre en place au niveau de l'architecture du système conseiller les mécanismes d'espionnage des actions et d'évaluation des produits de l'utilisateur, et des mécanismes de valuation du statut des composants correspondants. Il faut ensuite appliquer les règles de composition pour valuer les statuts des composants qui leur sont liés.

4.3.3. État

Ainsi qu'indiqué précédemment, la notion d'état, qui dénote le niveau de définition d'un produit ou le niveau de réalisation d'une tâche, est essentiellement « sémantique ». Il s'agit, pour chaque tâche, d'identifier un vocabulaire, propre à la tâche, qui permet de décrire ses différents niveaux de réalisation, puis les conditions d'application de ce vocabulaire.

L'état d'une tâche est généralement relatif aux concepts produits et à l'état d'autres tâches. Dans certains cas, il peut également être contextuel à une utilisation particulière de la tâche considérée. L'état d'une tâche doit donc être défini par rapport à, d'une part, un contexte décrivant l'importance et le statut attribués aux autres composants (concepts, tâches, regroupements) et, d'autre part, à une condition sur les valeurs des attributs des concepts extrants et/ou sur l'état d'autres tâches. On pourra par exemple utiliser un format de type :

Dans un contexte où
liste-de {importance(concept/tâche/regroupement) = x}
et liste-de {statut(concept/tâche/regroupement) = y},
Si liste-de {état(autre tâche) = z}
et liste-de {valeur(attribut) = t},
Alors état (tache) = r

Ainsi, pour reprendre l'exemple de la tâche « Etablir la solution préliminaire » présenté précédemment, l'évaluation des états « besoins établis » et « solution préliminaire complétée » est liée à l'état des sous-tâches « Construire le modèle des connaissances », « Définir les besoins d'apprentissage » et « Construire le réseau des événements d'apprentissage » ; pour évaluer l'état de la tâche « Définir les besoins d'apprentissage », on utilise des descriptions où interviennent la notion de statut (l'outil de définition du tableau des compétences a été utilisé) et la notion d'état (par exemple pour évaluer si les compétences relatives attribuées aux publics cibles sont adéquates), la notion d'importance n'intervenant pas dans ce cas.

Les différentes ressources nécessaires à l'identification des relations (liste des tâches, liste des sous-tâches d'une tâche, liste des extrants d'une tâche, etc.) sont calculables par analyse du graphe. L'analyse des liens entre composants facilite également l'identification de leurs interrelations. Des outils spécifiques mettant en évidence les importances relatives des composants peuvent également être utiles. Au niveau des tâches considérées, on peut travailler de façon isomorphe à l'arbre des tâches (considérer en premier lieu les sous-tâches terminales, définir leur état une à une, par rapport aux valeurs possibles de leurs extrants, puis définir l'état des tâches composées par rapport à l'état de leurs sous-tâches) ou travailler de façon plus générale (considérer, pour une tâche, l'ensemble de ses extrants directs et des extrants des sous-tâches). Ici encore, le choix de l'une ou l'autre approche doit être fondé sur le degré d'homogénéité des tâches.

On peut noter qu'il est possible, dans certains cas, d'établir une liaison entre le statut et l'état d'un composant. Cette liaison peut s'exprimer sous la forme de règles du type :

Dans le contexte x,
Si situation = y
et z1 < état (composant) < z2,
Alors statut (composant) = v

Ce type de règle peut venir en remplacement ou en complément des règles de composition de statut, auquel cas elles doivent être considérées comme prééminentes.

D'un point de vue technique, pour établir l'état d'un composant, il faut, de façon similaire à la notion de statut, mettre en place au niveau de l'architecture du système conseiller des mécanismes d'espionnage des actions de l'utilisateur et d'analyse de ses produits. La difficulté tient ici au fait que les outils d'analyse doivent juger de la qualité du produit, ce qui est bien évidemment beaucoup plus délicat. Il s'agit d'identifier des formules (essentiellement heuristiques) qui, sur la base de l'analyse des objets produits par l'utilisateur, permettent d'attribuer une valeur d'état. Ainsi, dans MISA, un certain nombre de « normes » de qualité du modèle (MOT) des connaissances du domaine enseigné ont été identifiées, sur la base du nombre et du type des concepts et liens mis en jeu, le nombre de connaissances pour lesquelles un besoin d'apprentissage a été précisé, etc. Il ne s'agit pas de juger que le modèle est bon ou mauvais, mais qu'il est (par exemple) plutôt factuel ou plutôt procédural, information qui est ensuite comparée avec la nature de modèle des connaissances qui est souhaitable pour le type de public cible visé ou pour la stratégie pédagogique adoptée.

5. Phase B : identification des conseils

Nous proposons de différencier les conseils essentiellement liés à la démarche préconisée et les conseils essentiellement liés aux produits élaborés. Pour ces derniers, on dissociera les notions de cohérence et de qualité. La notion de cohérence permet de dénoter le fait qu'une valeur associée à un attribut est légale ou illégale, ou qu'une configuration (ensemble de composants liés entre eux) a un sens ou n'a pas sens. La notion de qualité permet de dénoter des informations plus subjectives ou plus contextuelles. Le fait de dissocier ces deux aspects permet deux niveaux de jugement, absolu (« ceci n'a pas de sens ») et plus relatif (« il serait préférable de ... »). Cette distinction facilite également la maintenance et l'évolution du système conseiller, en permettant de dissocier les aspects intangibles des aspects qu'il peut être nécessaire de reconsidérer en cas d'adaptation de la tâche considérée, d'introduction de nouveaux concepts, etc.

La dissociation de différents types de conseils vise à faciliter l'identification des conseils. A chaque type de conseil correspond un ensemble de questions que les concepteurs du système conseiller doivent se poser. La dissociation proposée ne doit cependant pas conduire à une partition des conseils. Ainsi, certains conseils liés à la démarche peuvent également intégrer des aspects liés à la qualité des produits, et inversement.

Actuellement, la démarche proposée se focalise essentiellement sur les conseils de type démarche, qui sont les conseils pour lesquels le processus d'identification est le plus systématisable.

5.1. Conseils de type « démarche »

Les conseils de type « démarche » sont liés à des relations de précédence, par exemple des conseils signalant qu'il serait préférable de réaliser une action A (ou de définir un produit P) avant d'entreprendre une action B. Un conseil de ce type peut être déclenché lorsque le système conseiller détecte une action qui correspond à un début de tâche (lancement d'une fonction, valuation d'un concept qui est un extrait de la tâche, etc.) ou à une fin de tâche (lancement d'une fonction, début d'une autre tâche, etc.). Ainsi, pour la tâche présentée en Figure 2, il s'agit (par exemple) de signaler à un utilisateur qui vient de développer le modèle des connaissances (2.2) et qui tente d'élaborer le devis du réseau des événements d'apprentissage (2.5) qu'il faudrait auparavant définir le tableau des compétences et les orientations pédagogiques (2.4), en justifiant ce conseil par « pour pouvoir élaborer le RÉA, il faut avoir défini le tableau des compétences (214), ainsi que les orientations pédagogiques (220) ».

Les conseils de type démarche sont liés à des relations de précédence entre les tâches, relations qui sont représentées dans la modélisation MOT, soit explicitement (lien P), soit implicitement (liens Intrants/Extrants, etc. ; cf. section 3). Les situations

susceptibles de nécessiter des conseils de type démarche peuvent donc être recherchées systématiquement. Pour un type de conseil, on identifie les configurations MOT (c'est-à-dire le type d'objet et le type de lien) qui dénotent les situations pour lesquelles ce type de conseil peut être pertinent, et on utilise ensuite la configuration comme un « patron » : chaque configuration (sous-graphe) du graphe MOT qui correspond au patron est un conseil potentiel, dont la justification correspond à la justification des relations mises en jeu.

On peut dissocier (cf. [TCH 98] pour une typologie plus précise) :

- Les conseils positifs, qui visent à aider l'utilisateur dans le choix de la prochaine action à réaliser. Typiquement, ce type de conseil va être déclenché lorsque l'utilisateur demande de l'aide. La forme générale est : « Si P1 est terminée Alors commencer P2 ». Différents sous-cas (et donc différents patrons) correspondent aux situations où « réaliser P2 permettra de continuer/compléter la procédure P » (P1 et P2 sont des sous-procédures de P, P2 est la dernière sous-procédure de P non encore terminée), « réaliser P2 permettra de compléter le regroupement R », « réaliser P2 permettra de produire C, et ainsi de passer à P3 » (P2 produit un intrant de P3 et P3 doit être réalisée), etc.

- Les conseils de type avertissement, qui visent à signaler à l'utilisateur les implications de son choix concernant la prochaine action à réaliser. Typiquement, ce type de conseil va être déclenché lorsque l'utilisateur entreprend une action licite mais qui diffère de celle qui est préconisée. La forme générale est : « Il est licite de faire P1, mais il serait préférable de faire P2 ». Les sous-cas sont les mêmes que pour les conseils « positifs », mais formulés différemment : « si l'on ne fait pas P2 maintenant, P restera inachevée » (l'utilisateur a commencé P et terminé certaines sous-procédures de P, P2 est une sous-procédure de P non encore terminée et l'utilisateur passe à P1, qui n'est pas une sous-procédure de P), « si l'on ne fait pas P2 maintenant, le regroupement R restera incomplet/inachevé » (l'utilisateur passe à P1 qui n'est pas dans R), « si l'on ne fait pas P2 maintenant, on ne produira pas C, et on ne pourra pas faire P3 » (P3 est une procédure qui succède normalement à la procédure en cours, P3 a en intrant C, C est un extrant de P2, et l'utilisateur passe à P1 qui ne produit pas C), etc.

- Les conseils négatifs, qui visent à signaler à l'utilisateur que son choix de la prochaine action à réaliser contrevient à une règle de précedence implicite ou explicite. Pour une situation où les pré-requis d'une procédure n'ont pas été définis, la forme générale est « avant de commencer P il faut avoir défini C », et les sous-cas sont « C est un intrant de P », « C est un intrant de P1, qui est une sous-procédure de P », etc. Pour une situation où un lien de précedence entre procédures n'est pas respecté, la forme générale est « avant de commencer P2 il faut avoir réalisé P1 », et les sous-cas sont « P1 précède P2 » (lien de précedence explicite), « P2 succède bien à P0 qui est la tâche en cours, mais P1 est une sous-procédure de P0 et P1 n'a pas été réalisée », etc.

La recherche systématique des instances de ces patrons dans le graphe modélisant la tâche considérée (processus qui peut être automatisé par composition d'outils d'analyse) permet l'identification d'un ensemble de conseils potentiels. Une fois les conseils potentiels identifiés, il s'agit de les analyser pour décider s'ils sont retenus et, éventuellement, de les contextualiser. Contextualiser un conseil consiste à le préciser en faisant intervenir les notions d'importance, de statut et d'état, ou encore à remplacer le texte générique par un message spécifique. Un conseil contextualisé pourra avoir la forme :

Dans un contexte où
liste-de {importance(concept/tâche/regroupement) = x et justification de l'importance},
liste-de {statut(concept/tâche/regroupement) = y}
et liste-de {état(concept/tâche/regroupement) = z},
Si (condition générique instanciée et éventuellement affinée),
Alors (message spécifique de la situation)

D'un point de vue technique, s'ils ne sont pas contextualisés, les conseils de type démarche peuvent être implantés avec un formalisme de type « règle de production d'ordre 1 » (par exemple, « si l'utilisateur entreprend une procédure P1, un des intrants C de P1 n'est pas défini et la procédure P2 produit C, alors conseiller de réaliser P2 avant de commencer P1 »). Outre le fait que cela évite de définir plusieurs conseils ne différant que par les composants mis en jeux, si l'on maintient un modèle MOT représentant la démarche effectivement suivie par l'utilisateur, ces conseils « génériques » pourront être calculés par rapport au processus effectivement réalisé, et non simplement par rapport au processus « idéal ».

5.2. Conseils de type « cohérence »

Les conseils de type « cohérence » sont liés à des valeurs de description des concepts qui sont incohérentes, indépendamment des notions d'importance, de statut ou d'état. On peut distinguer les règles de cohérence endogène (cohérence interne du produit) et les règles de cohérence exogène (cohérence entre produits). A titre d'exemple, on utilise dans MISA un concept « besoin d'apprentissage » d'un public cible, qui est défini par la différence entre le niveau d'habileté de départ et le niveau visé. Le fait que le niveau de départ d'un public cible n'ait pas été défini est un problème de cohérence endogène (le concept complexe « besoin d'apprentissage » est mal défini car l'un de ses sous-concepts n'est pas défini) ; le fait que la stratégie pédagogique retenue soit incohérente avec le niveau de départ du public cible ou avec le besoin d'apprentissage identifié est un problème de cohérence exogène (relation entre les concepts « besoin d'apprentissage » et « stratégie pédagogique »).

– Cohérence endogène : il s'agit d'identifier les conseils délivrés lorsqu'un concept est défini de façon incohérente. Pour cela, il faut analyser, pour chaque concept, les cas de violations de contraintes d'intégrité d'un attribut (attribut qui doit

être valué et ne l'est pas, valeur incohérente, etc.) et les cas de violations de contraintes d'intégrité entre attributs. Un premier ensemble de conseils, liés aux valeurs individuelles des attributs, se déduit directement de la définition précise de ces attributs. Un second ensemble de conseils, liés aux interrelations entre attributs, peut être identifié en analysant systématiquement les croisements des valeurs possibles des différents attributs des concepts (recherche de configurations incohérentes).

– Cohérence exogène : il s'agit d'identifier les conseils délivrés lors de problèmes de cohérence entre concepts. Pour cela, il faut identifier les points de vue qui font qu'un ensemble de concepts sont liés entre eux (notion de regroupement), puis analyser la cohérence de ces concepts selon ces différents points de vue. Par exemple, l'analyse des interrelations entre les concepts intrants d'une même procédure est susceptible de mettre en évidence des configurations incohérentes, et donc des conseils. Un même concept peut bien évidemment faire partie de plusieurs regroupements correspondants à différents points de vue. Le processus de construction de ces regroupements peut être automatisé en combinant des outils d'analyse du graphe MOT. Par exemple, pour analyser les problèmes de cohérence entre concepts intrants d'une même procédure, il faut, pour chaque concept, identifier les procédures où le concept est un intrant, puis, pour chacune de ces procédures, identifier les autres intrants de la procédure, les extrants de la procédure et les principes liés à la procédure.

Techniquement, les problèmes de cohérence (en particulier de cohérence endogène) qui sont identifiés peuvent être intégrés sous forme de conseils dans le système conseiller, mais ils peuvent également être utilisés comme spécifications de l'interface du logiciel hôte, afin de prévenir les saisies incohérentes.

5.3. Conseils de type « qualité »

Les critères de qualité des produits sur lesquels reposent l'évaluation des composants ne sont pas représentés dans le graphe MOT modélisant la tâche considérée. Il est donc à la charge des concepteurs du système conseiller de définir ces critères, puis de les utiliser pour définir les différents conseils. Ainsi, dans MISA, sur la base des « normes » de qualité du modèle des connaissances du domaine enseigné, un exemple de conseil de qualité est « il faut augmenter le nombre de faits et/ou réduire celui des procédures et principes », avec pour justification « le public cible est au niveau sensibilisation, pour ce niveau il faut généralement présenter essentiellement des connaissances factuelles et peu de procédures et principes ».

Les conseils de qualité font essentiellement intervenir la notion d'état et la notion d'importance, éventuellement la notion de statut. Leur identification sera facilitée si l'on définit au préalable une typologie des conseils que l'on veut pouvoir donner. Contrairement aux conseils de démarche, cette typologie est nécessairement

spécifique de la tâche considérée, et, les connaissances mises en jeu n'étant pas présentes dans le graphe MOT, il n'est pas possible d'établir des patrons permettant de systématiser la recherche. Afin de guider le processus, il nous paraît nécessaire de dissocier la qualité intrinsèque des produits, la qualité d'un produit en tant que ressource pour une autre tâche et la qualité d'un produit en tant que produit final. Pour chaque type de conseil, il faut ensuite identifier les informations à prendre en compte pour établir les conseils et, le cas échéant, les configurations qui expriment les informations de cette nature dans un graphe MOT. Les outils d'analyse du graphe MOT peuvent ensuite être utilisés pour rechercher les informations sur lesquelles seront fondés les conseils.

6. Discussion

Nous avons présenté dans cet article une approche qui vise à rationaliser et systématiser l'identification de conseils potentiels. Cette approche peut-être résumée comme suit. Tout d'abord, la tâche considérée est modélisée à un niveau abstrait. Ce modèle doit décrire la tâche et ses différentes sous-tâches, leurs intrants et extrants et les principes de régulation. Dans nos travaux nous utilisons le langage de modélisation MOT, mais les principes de l'approche proposée sont transposables pour d'autres langages de modélisation s'ils permettent le même type de description. Ce modèle est utilisé, d'une part, comme spécification du système hôte (si celui-ci ne préexiste pas) et, d'autre part, comme support à l'identification de conseils potentiels par les experts de la tâche considérée. Ces conseils sont ensuite adaptés au système hôte (en particulier leurs conditions de déclenchement, qui sont liées aux fonctionnalités du système hôte). Le système conseiller est enfin implanté comme un méta-système, par exemple à l'aide de l'architecture EpiTalk.

Rappelons que nous considérons que cette démarche permet d'identifier des « conseils candidats » : c'est un travail d'identification préliminaire qui ne dispense pas d'une étude de l'acteur en situation.

Dans cet article, nous avons essentiellement abordé l'identification des conseils. Pour obtenir un système conseiller pertinent, il faut cependant également gérer les conseils pour ne les délivrer qu'à bon escient (Phase C). Si le système conseiller comprend peu de conseils, cette phase peut être limitée à l'étude des aspects ergonomiques (étudier comment l'utilisateur peut visualiser les conseils positifs, comment le système va signaler à l'utilisateur les conseils négatifs, etc.). En revanche, si le système conseiller comprend beaucoup de conseils, il faut également définir les principes de gestion de ces conseils. En effet, plusieurs conseils de même type ou de types différents peuvent être déclenchables simultanément¹⁰. Certains

10. Il est déconseillé de chercher à prendre en compte les problèmes d'interactions entre conseils lors de la définition des conseils, c'est-à-dire au niveau de la phase B. En effet, cela conduit le plus souvent à intégrer dans les conditions des conseils des critères qui ne sont pas relatifs au thème traité par le conseil, mais à des considérations externes, d'un niveau

peuvent être concurrents, c'est-à-dire conseiller des actions différentes. Qu'ils soient concurrents ou non, présenter plusieurs conseils en même temps risque en fait de noyer l'utilisateur sous trop d'informations et de rendre l'activité de conseil inefficace, voire nuisible.

Pour gérer les conseillers, une première approche (simpliste) consiste à identifier un certain nombre de principes généraux permettant de donner des priorités aux conseils. Par exemple, les conseils de cohérence auront une priorité supérieure aux conseils d'avertissement, les conseils portant sur des concepts importants auront une priorité supérieure à ceux qui sont moins importants, etc. Ceci peut être réalisé en associant des valeurs numériques de priorité aux différents termes utilisés pour décrire les concepts (termes définis lors de la Phase A), puis des règles de composition de ces valeurs de priorité. L'avantage de cette approche est sa simplicité. Lorsque plusieurs conseils sont déclenchables, le choix peut être fondé sur le calcul de la priorité associée à chaque conseil. Cependant, en ramenant différentes notions à des valeurs numériques, cette approche ne permet pas de gérer de façon satisfaisante les problèmes de concurrence, ni d'adapter l'activité de conseil en fonction du contexte et/ou de l'utilisateur. Une seconde approche consiste à considérer l'activité de conseil comme une tâche complexe qui nécessite une activité de résolution de problème. Il s'agit alors de modéliser un raisonnement qui vise à déterminer quel(s) conseil(s) doi(ven)t être délivré(s), raisonnement dont les données sont l'ensemble des conseils déclenchables et déjà déclenchés, une modélisation du processus effectivement suivi par l'utilisateur (ce qui a été fait et ce qui n'a pas été fait) et des connaissances spécifiques à la tâche « conseiller ». Il est alors possible de gérer la complexité de la tâche de production des conseils (identification des conseils déclenchables, analyse des problèmes de concurrence, analyse de la pertinence des conseils par rapport à l'utilisateur et par rapport au contexte courant, etc.) et d'obtenir un système conseiller plus pertinent, plus fiable et plus facilement maintenable. Nous envisageons d'aborder ces problèmes sur la base de travaux similaires de G.Kassel sur la modélisation d'explication pour l'usager d'un SBC [GRE 94] et nos propres travaux sur la modélisation d'interactions entre un élève et un système d'enseignement [CHO 98].

L'approche proposée dans cet article constitue un processus méthodologique assez lourd, mais dont les différentes phases peuvent être vues comme un guide à adapter en fonction des éventuelles spécificités du système considéré. Pour un système d'apprentissage de faible amplitude, une application des idées de base de la démarche proposée permet d'accélérer la construction du système conseiller. Pour un système de large amplitude ou pour un contexte comme la construction du système conseiller d'ADISA, où les interrelations entre composants peuvent être de

différent. Ceci conduit à des conseils peu pertinents, inutilement complexes, et qui ne contiennent pas leur propre justification car ils sont contraints par d'autres considérations. Par ailleurs, les conseils étant implicitement interreliés, la maintenance et l'évolution du système conseiller sont rendues beaucoup plus difficiles.

plusieurs centaines, la démarche proposée permet de rationaliser le processus, ce qui est la seule façon de le mener à bien.

Le travail que nous avons présenté est pour l'instant une « méthodologie papier ». Son application systématique à la construction de systèmes conseillers destinés à des environnements de support à la tâche diffusés sur le Web avec Explor@ et la construction du système conseiller de l'ADISA devraient nous apporter des retours d'expérience qui nous permettraient de l'affiner. Nous pensons engager alors des travaux visant à développer une architecture informatique support de la tâche « identifier les conseils » (intégrant notamment les outils informatiques d'analyse d'un graphe MOT), architecture qui sera fondée sur une modélisation MOT de l'approche présentée dans cet article (et qui sera donc, bien entendu, pourvue elle-même d'un système conseiller).

7. Bibliographie

- [AUS 96] AUSSENAC-GILLES N., LAUBLET P., REYNAUD C., *Acquisition et Ingénierie des Connaissances : tendances actuelles*, Éditions Cépaduès, 390 Pages, 1996.
- [ALE 87] ALEXANDER J.R., FREYLING M., SHULMAN S., REHFUSS S., MESSICK S., « Ontological analysis: An ongoing experiment », *International Journal of Man-Machine Studies*, Vol. 26, No 4, 1987, p. 473-485.
- [CHO 98] CHOQUET C., DANNA F., TCHOUNIKINE P., TRICHET F., « Modeling Knowledge-Based Components of a Learning Environment within the Task/Method Paradigm », *International Conference on Intelligent Tutoring Systems (ITS'98, San Antonio, USA)*, *Lecture Notes in Computer Science*, Vol. 1452, Springer-Verlag, 1998, p. 56-65.
- [GIR 99] GIRARD J., PAQUETTE G., MIARA A., LUNDGREN K., « Intelligent Assistance for Web-based TeleLearning », *International conference on AI in Education (AI-Ed'99)*, IOS Press, 1999.
- [GIR 95] GIROUX S., PACHET F., PAQUETTE G., GIRARD J., « Des systèmes conseillers épiphytes », *Revue d'Intelligence Artificielle*, Vol. 9, No 2, 1995, p. 165-190.
- [GRE 94] GREBOVAL C., KASSEL G., « The production of explanations seen as a design task: a case study », *European Conference on Artificial Intelligence (ECAI'94)*, 1994, p. 351-355.
- [MER 94] MERRILL D., *Principles of Instructional Design*, Educational Technology Publications, Englewood Cliffs, New Jersey, 465 pages, 1994.
- [PAQ-a 96] PAQUETTE G., « La modélisation par objets typés : une méthode de représentation pour les systèmes d'apprentissage et d'aide à la tâche », *Science et Techniques Educatives*, Vol 3, No 1, 1996, p. 9-42.
- [PAQ-b 96] PAQUETTE G., GIRARD J., « AGD: a course engineering support system », *International Conference on Intelligent Tutoring Systems (ITS-96, Montreal, Canada)*, *Lecture Notes in Computer Science*, Vol. 1086, Springer-Verlag, 1996, p. 382-391.

- [PAQ-c 96] PAQUETTE G., PACHET F., GIROUX S., GIRARD J., « EpiTalk: Generating Advisor Agents for Existing Information Systems », *Journal of Artificial Intelligence in Education*, Vol. 7, No 3&4, 1996, p. 349-379.
- [PAQ 97] PAQUETTE G., AUBIN C., CREVIER F., « Design and Implementation of Interactive TeleLearning Scenarios », *International Council for Distance Education (ICDE'97)*, PennState University, USA, 1997.
- [PAQ 00] PAQUETTE G., « Construction de portails de télé-apprentissage Explor@ - Une diversité de modèles pédagogiques », *Science et Techniques Educatives*, Vol. 7, No 1, 2000.
- [ROM 81] ROMISZOWSKI A.J., *Designing Instructional Systems*, Kogan Page London/Nichols Publishing, New York, 415 pages, 1981.
- [SCH 93] SCHREIBER G., WIELINGA B., BREUKER J., *KADS: A Principled Approach to Knowledge-Based System Development*, Academic Press, London, 1993.
- [TCH 98] TCHOUNIKINE P., « Des outils d'analyse d'un modèle MOT et une méthode d'ingénierie de systèmes conseillers », Rapport technique du LICEF (LICEF98R150), 1998.
- [TCH 97] TCHOUNIKINE P., « Mapcar : a framework to support the elaboration of the conceptual model of a Knowledge Based System », *International Journal of Intelligent Systems*, Vol. 12, No 6, p. 441-468, 1997.
- [TEN 90] TENNYSON R.D., « Cognitive Learning theory Linked to Instructional Theory », *Journal of Structured Learning*, Vol. 10, No 3, 1990, p. 249-258.
- [WEN 87] WENGER E., *Artificial Intelligence and Tutoring Systems, Computational and Cognitive Approaches to the Communication of Knowledge*, Morgan Kaufmann, Los Altos, USA, 1987, 486 pages, 1987.
- [WES 91] WEST C.K., FARMER J.A., WOLFF P.M., *Instructional Design, Implications from Cognitive Science*, Allyn and Bacon, Boston, 271 pages, 1991.
- [WIN 92] WINKELS R., *Explorations in Intelligent Tutoring and Help*, IOS Press, Amsterdam 1992, 227 pages, 1992.