



HAL
open science

MOCA, une approche expérimentale de l'ingénierie des EIAH

Vincent Barré, Christophe Choquet, Alain Corbière, Philippe Cottier, Xavier
Dubourg, Patricia Gounon

► **To cite this version:**

Vincent Barré, Christophe Choquet, Alain Corbière, Philippe Cottier, Xavier Dubourg, et al..
MOCA, une approche expérimentale de l'ingénierie des EIAH. Environnements Informatiques pour
l'Apprentissage Humain 2003, Apr 2003, Strasbourg, France. pp.55-66. edutice-00000127

HAL Id: edutice-00000127

<https://edutice.hal.science/edutice-00000127>

Submitted on 31 Oct 2003

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

MOCA, une approche expérimentale de l'ingénierie des EIAH.

Vincent Barré, Christophe Choquet, Alain Corbière, Philippe Cottier, Xavier Dubourg, Patricia Gounon*

* *LIUM*

IUT de Laval

52, rue des docteurs Calmette et Guérin

53020 Laval Cedex 9

Prenom.Nom@univ-lemans.fr

RÉSUMÉ. Ce texte présente une approche expérimentale de l'ingénierie des EIAH. Nous avons modélisé un scénario pédagogique à l'aide d'un outil de description normalisée. Sur la base d'une expérimentation réalisée au sein de l'IUT de Laval auprès de 38 étudiants, nous présentons les écarts constatés entre les activités et les scénarios préconçus ainsi que la nature des interactions entre les acteurs humains (tuteur et apprenants). La méthodologie utilisée repose sur trois techniques (traces informatiques, observations in situ et entretiens). Les résultats obtenus nous permettent d'envisager la ré-ingénierie d'un EIAH.

MOTS-CLÉS : ingénierie des EIAH, expérimentation, conception, tutorat, observables, normalisation.

1. Introduction

Les efforts de normalisation internationale en matière de description de ressources pédagogiques tels que la réflexion menée sur les L.O.M. (Learning Object Metadata) ainsi que sur les architectures de plate-formes de Formation Ouverte et A Distance comme la norme IEEE LTSA (Learning and Training System Architecture), sont maintenant incontournables mais portent essentiellement sur la problématique de la diffusion des contenus. Peu de travaux abordent, dans le contexte de FOAD, le problème de la création de ressources et de leur ré-ingénierie. Nous pouvons citer par exemple ici les travaux du LICEF de la Télé Université du Québec (TELUQ) qui intègrent explicitement l'enseignant auteur de contenus comme acteur du cycle de formation et l'instrumente du point de vue méthodologique [PAQUETTE 96] comme du point de vue outils [PAQUETTE 00]. Ces travaux, du fait de leur antériorité, ne relèvent cependant pas pour l'instant de la problématique d'interopérabilité imprégnant fortement les projets de normalisation actuels.

L'utilisation des technologies de l'information et de la communication dans les dispositifs de formation introduit une désynchronisation des deux rôles principaux de l'enseignant – auteur/concepteur de cours et tuteur/formateur – qui se traduit généralement par une spécialisation des acteurs du processus de formation. Le cycle de vie d'un cours traditionnel (i.e. présentiel) dans un cadre institutionnel (i.e. une formation diplômante supportée par une institution de formation) peut-être résumé en quatre temps : (1) l'enseignant définit et crée des contenus en les articulant de manière à former un scénario pédagogique qu'il pense fonctionnel ; (2) l'enseignant utilise ces matériaux en session. Il adapte son scénario et son comportement en fonction des réactions et de la réussite des apprenants, ce qui lui permet (3) de relever des imperfections et des possibilités d'amélioration de son cours et l'amène généralement à (4) revoir ses contenus et/ou son scénario pédagogique en prévision d'une nouvelle session. Seul le deuxième temps de ce cycle de vie est véritablement pris en compte par la norme IEEE LTSA qui, en identifiant quatre processus (diffusion, apprenant, tutorant et évaluant) dans un dispositif de FOAD, exclut de son spectre d'application la définition, l'évaluation et la ré-ingénierie des ressources et des scénarii pédagogiques.

Le projet MOCA, pour Modélisation et Optimisation des Cycles d'Apprentissage, a pour objectif principal d'opérationnaliser les deux cycles d'adaptation d'une part, entre le processus apprenant et le processus tutorant et d'autre part, entre le processus d'apprentissage et les processus de conception. Nous pensons qu'il est fondamental, dans un souci d'amélioration de la qualité de l'apprentissage dans un dispositif FOAD normalisé et, plus généralement, dans un contexte d'ingénierie d'EIAH [TCHOUNIKINE 02], de supporter les quatre temps du cycle de vie d'une formation (cf. supra) et donc d'associer étroitement le concepteur à ce cycle.

Afin d'alimenter notre réflexion, et partant de l'idée qu'il n'y a pas isomorphie entre activités prescrites et activités émergentes, il nous a semblé nécessaire de

penser une expérimentation mettant à jour, d'une part, les distorsions entre activités et scénarios anticipés par le concepteur et ceux produits par l'action des apprenants et, d'autre part, les modalités d'interactions entre les apprenants et le tuteur. Cette expérimentation a été réalisée auprès d'étudiants de seconde année d'IUT. Elle est supportée par un dispositif informatique dédié à la formation, FreeStyle Learning [BROCKE et al. 00].

Dans notre approche, nous intégrons la phase de conception dans cet environnement normalisé en modélisant les activités pédagogiques à partir d'un langage de type XML [DE LA PASSARDIERE et al. 01], l'EML (Educational Modelling Language) [KOPER 01]. Ce choix nous apparaît pertinent au regard des récentes réflexions des groupes de normalisation, notamment européens [RAWLINGS et al. 02].

Nous développons dans ce texte la méthodologie que nous avons utilisée et décrivons les caractéristiques de l'EIAH mis en œuvre en nous attardant plus particulièrement sur l'une des activités proposées. Nous appuyant sur un résultat de notre expérimentation, nous montrons comment les données recueillies sur le déroulement d'une session de formation peuvent contribuer, à différents niveaux, dans le cadre des normes actuellement en construction, à une ré-ingénierie de la conception des activités pédagogiques.

2. Méthodologie

Notre expérimentation a pour objectifs : (1) de déterminer les écarts entre les activités des apprenants et les scénarios préconçus, (2) et de recueillir des informations sur les interactions entre les acteurs humains (stratégies et contenus des interventions). Les informations que nous cherchons ainsi à obtenir sont essentiellement qualitatives.

L'expérimentation a été effectuée auprès de 38 étudiants, repartis en trois groupes, en seconde année de DUT Services et Réseaux de Communication de l'IUT de Laval. L'unité d'enseignement proposée, d'une durée de deux heures, vise à former des étudiants aux architectures logicielles des services sur réseau. Chacune de ces sessions a eu lieu de façon synchrone et dans une même salle de cours. Cette option a été choisie afin de favoriser la mise en évidence des rôles tenus et des actes posés par les apprenants et le tuteur. Ce dernier demeurerait à disposition en cas de difficultés, pouvant intervenir, soit en réaction à la sollicitation d'un étudiant (modalité réactive), soit de sa propre initiative lorsqu'il le jugeait nécessaire (modalité proactive). Cette expérimentation a exigé le recours à trois méthodes : des entretiens guidés, de face à face, avec les apprenants, une observation *in situ* des sessions de travail et la récolte de traces informatiques de l'activité des apprenants. Chacune d'elles correspond à trois instances au sein desquelles il est possible d'intervenir en vue d'une optimisation : l'interaction apprenant - objet technique, les interactions entre acteurs humains, l'objet technique lui-même.

Les entretiens effectués auprès des étudiants ont été réalisés immédiatement après la session. Le but était de déterminer la manière dont chacun avait réalisé l'ensemble des activités mises à sa disposition. Nous voulions connaître leur sentiment sur leur activité et, particulièrement, sur les difficultés rencontrées pendant la session d'apprentissage. Les questions portaient essentiellement sur l'utilisation des activités et de l'environnement.

Deux observateurs étaient présents dans la salle afin de recueillir des informations concernant les échanges effectués entre les acteurs humains (tuteur et étudiants) en précisant (à l'aide d'une grille) leur contenu et leur type. Pour chaque interaction, nous avons par conséquent identifié les acteurs impliqués, relevé le contenu des échanges, les activités concernées ainsi que le moment de l'intervention. L'observateur avait aussi pour tâche de relever le comportement des étudiants, cette observation visant à mieux appréhender la manière dont chacun avait pu agir face au dispositif.

Nous avons également mis en œuvre un relevé de traces numériques afin de recueillir des informations sur l'activité des étudiants. L'apprenant, en interaction avec le dispositif, effectue un certain nombre de tâches. Des mécanismes de génération de traces intégrées au dispositif gardent alors un historique des événements liés à cette activité. Dans la phase de développement logiciel des supports d'activité, l'informaticien intègre ces mécanismes de capture au regard des spécifications des observables à obtenir. Nous nous sommes ici appuyés sur des méthodes empruntées à l'ingénierie de la fouille de données [BAZSALICZA et al. 01].

3. Description de l'unité d'enseignement et de ses activités

L'unité d'enseignement proposée lors de notre expérimentation a pour objectif de former les étudiants à la programmation des architectures logicielles des services sur réseaux. Celle-ci se décompose en 6 activités.

3.1. Objectifs des activités

S'il existe bien un scénario prédictif, rien n'est imposé : l'apprenant construit son propre parcours. Nous avons défini nos activités en nous appuyant sur des verbes d'action permettant à l'apprenant d'appréhender chaque objectif cognitif. Dans ce contexte, et pour l'aider à se créer un modèle mental de navigation, une activité est associée à un seul type de média [CROZAT 02].

– Introduire l'unité d'enseignement : Sous la forme d'une vidéo, les principes généraux sur le serveur web sont exposés. Cette activité a pour objectif de présenter l'ensemble des concepts nécessaires à la réalisation de l'unité d'enseignement pour inciter et motiver l'apprenant.

– Approfondir les concepts : Cette activité présente un contenu textuel et quelques schémas décrivant les principaux protocoles du web et les outils de programmation à utiliser.

– Synthétiser les concepts : Deux diaporamas sont présentés : l'un exposant les concepts liés au serveur web, l'autre décrivant l'exécution, étape par étape, du code Java d'un serveur.

– S'exercer pour comprendre : L'étudiant est amené à résoudre un ensemble de problèmes portant sur la programmation d'un serveur HTTP.

– S'auto-évaluer : Deux QCM permettent à l'apprenant de s'évaluer.

– Apprendre en faisant : L'apprenant met en pratique l'ensemble des connaissances du domaine et modifie, étape par étape, jusqu'à sa réalisation fonctionnelle, le code d'un serveur web.

Nous avons choisi de présenter plus précisément cette dernière activité. Elle nous apparaît la plus pertinente à exposer ici du fait notamment de nos observations de ses usages.

3.2. Modélisation EML de l'activité « Apprendre en faisant »

Cette activité se définit en précisant le contexte d'utilisation, les entités informatiques la supportant et les objectifs pédagogiques pour les trois acteurs. *Les apprenants* - au maximum 14, ils travaillent dans une même salle pendant une durée de deux heures, chacun devant un poste informatique ; *le tuteur* – il intervient auprès de l'apprenant en priorité de manière réactive (en réponse à une sollicitation d'un apprenant) mais également proactive lorsqu'il estime ce dernier en difficulté ; *le concepteur* – il est le prescripteur de l'activité et l'auteur de sa description.

Cette activité propose à l'apprenant de compléter le code d'un serveur HTTP, d'ajouter la gestion du protocole MIME et de gérer l'affichage de l'arborescence du serveur. Chacune de ces sous-activités est structurée en quatre étapes :

```
<Unit-of-study Type="interactif" EML-version="1.0" Reusability="Reusable">
  <Activity-structure Id="str-http" Default-visibility="Show">
    <Metadata>
      <Title>Compléter le code d'un serveur HTTP</Title>
    </Metadata>
    <Content>
      <Activity Id="act-http1">
        <Metadata><Title>Editer un fichier</Title></Metadata> ...
      </Activity>
      <Activity Id="act-http2">
        <Metadata><Title>Compiler un fichier</Title></Metadata> ...
      </Activity>
      <Activity Id="act-http3">
        <Metadata><Title>Exécuter l'application serveur</Title></Metadata> ...
      </Activity>
      <Activity Id="act-http4">
        <Metadata><Title>Tester l'application serveur</Title></Metadata> ...
      </Activity>
    </Content>
  </Activity-structure>
</Unit-of-study>
```

De plus, le concepteur définit, pour chaque étape, un ensemble d'objectifs pédagogiques en précisant le type d'habileté à développer (savoir, savoir-faire, compétence, comportement,...) :

```
<Unit-of-study Type="interactif" EML-version="1.0" Reusability="Reusable">
...
  <Activity Id="act-http2">
    <Metadata><Title>Compiler un fichier</Title></Metadata>
    <Objectives>
      <Learning-objectives>
        <Learning-objective>
          <Objective-description>
            <P>Demander de valider le code du programme informatique</P>
          </Objective-description>
          <Objective-type><Attitude /></Objective-type>
        </Learning-objective>
        <Learning-objective>
          <Objective-description>
            <P>Corriger le code du programme</P>
          </Objective-description>
          <Objective-type><Knowledge /></Objective-type>
        </Learning-objective>
      </Learning-objectives>
    </Objectives>
  </Activity>
...
</Unit-of-study>
```

Pour chaque objectif, le concepteur associe un environnement supportant l'activité. Ici, des logiciels libres sont utilisés pour les sous-activités : *Editer un fichier*, *Compiler un fichier* et *Exécuter une application serveur*.

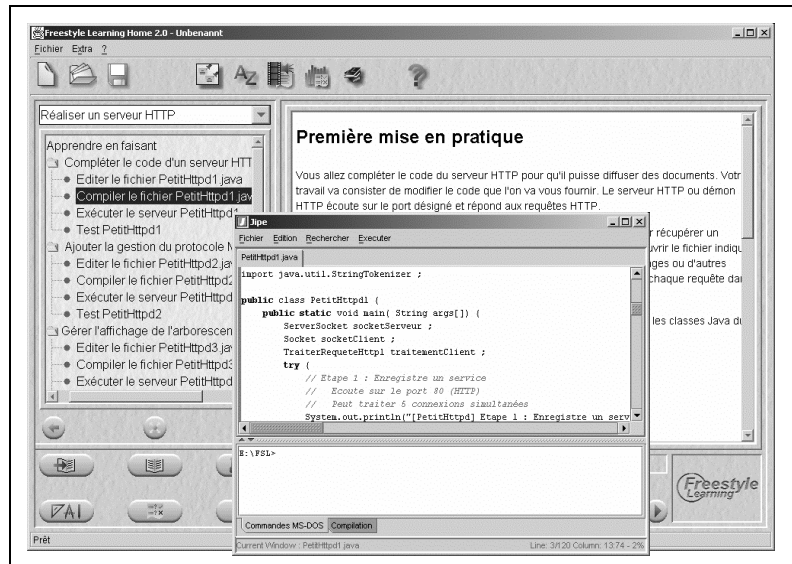


Figure 1. Activité « Apprendre en faisant »

Dans l'unité d'enseignement, le travail de conception permet d'identifier différentes séquences d'activité et de les mettre en relation avec des objectifs pédagogiques. Ces relations, spécifiées dans la description EML, sont utilisées en session d'apprentissage pour valider les différents parcours effectués par l'apprenant.

Le même principe est appliqué au niveau des ressources : un modèle de navigation (cf. figure 2) décrit les interactions possibles pour chaque environnement. Ce modèle est alors un outil permettant d'interpréter (i.e. associer une sémantique) les séquences de traces informatiques de manière à valider la réalisation des objectifs pédagogiques.

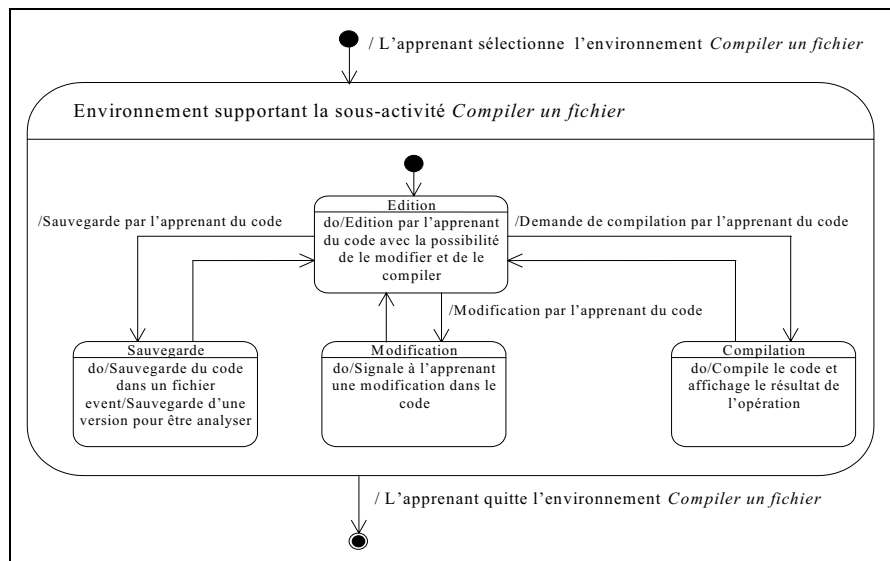


Figure 2. Diagramme d'états UML (Unified Modelling Language) de navigation de l'environnement associé à l'activité « Compiler un fichier »

3.3. Déroulement de l'activité « Apprendre en faisant »

L'activité se décompose en trois étapes (cf. figure 1). Aucune consigne n'a été précisée aux étudiants quant à l'utilisation des trois sous-activités à réaliser, l'un des objectifs de cette expérimentation étant de déterminer le parcours effectif réalisé par les étudiants.

4. Résultats

Pour mieux cibler notre propos, nous nous concentrons ici sur l'un des dysfonctionnements identifiés dans l'activité « Apprendre en faisant ». Nous avons en effet constaté que certains étudiants n'ont pas pu réaliser correctement les actions de modification, compilation et validation du code comme il était prévu. L'activité

des apprenants, durant ce moment particulier, est ici décrite au travers des trois méthodes d'analyse que nous avons choisi d'utiliser : entretiens, observations *in situ*, traces informatiques.

4.1. Entretiens

Les étudiants qui ont été confrontés aux problèmes de modification, compilation et validation du code se sont exprimés sur leurs difficultés.

Certaines remarques concernent l'utilisation de l'environnement de programmation considéré par les apprenants comme trop éloigné de celui employé habituellement. Les remarques nous montrent que l'appréciation des usages existants est déterminante et combien le contexte socio-technique des acteurs doit être appréhendé en amont pour mieux être pris en compte lors de la conception.

D'autres réflexions portent sur le contenu pédagogique et l'organisation globale de l'unité d'enseignement. Nous avons constaté (comme nous le verrons plus loin) que le scénario suivi par la grande majorité des apprenants est conforme à celui imaginé par le concepteur. Or, si le passage entre les activités « Introduire l'unité » et « Approfondir les concepts » ne semble pas avoir posé de difficulté majeure, la réalisation de l'activité « Apprendre en faisant » a été jugée trop en décalage avec l'ensemble de l'unité d'enseignement. Ce passage a véritablement été vécu comme une cassure dans le cheminement pédagogique. Ainsi, certains se sont exprimés sur la difficulté de réaliser l'activité dans la durée impartie. Les étudiants interrogés expliquent ce manque de temps par leur faible connaissance d'un langage qu'ils considèrent d'une trop grande complexité.

Dans le cadre de l'activité « Apprendre en faisant », et essentiellement sur les étapes de modification, compilation et validation du code, les entretiens font donc émerger la nécessaire prise en compte du contexte d'usage dans lequel l'EIAH doit être plongé. Ils mettent aussi en lumière un déséquilibre entre les activités proposées, lié au degré de difficulté perçu par les apprenants.

4.2. Observations en salle

Les observations en salle nous ont permis d'identifier des comportements qui n'avaient pas été envisagés. Deux modalités d'interactions ont été observées : celles entre le tuteur et l'apprenant et celles entre les apprenants eux-mêmes.

Dans cette activité, le rôle du tuteur, pensé *a priori*, devait être de répondre aux sollicitations des apprenants sur des questions liées au langage de programmation (erreurs de compilation, questions de syntaxe, compréhension du code, etc.), l'objet même de l'activité. Nous avons pu observer dans les faits que le tuteur était amené à interagir de manière différente (de façon proactive ou réactive) et sur des problèmes dépassant la seule question des contenus. Ainsi a-t-il dû intervenir, sans sollicitation directe et verbale, mais en réaction à un comportement des apprenants manipulant le dispositif informatique. Le tuteur s'est par exemple décidé à s'adresser à un apprenant constatant qu'il progressait plus lentement que le reste du groupe. Cette

sollicitation lui a permis de comprendre que le programme était constamment compilé sans avoir été préalablement sauvegardé. Au-delà de ce constat, cette situation a aussi permis de révéler qu'un comportement qui semble n'être lié qu'à une mauvaise utilisation du dispositif, d'une défaillance ergonomique, peut en fait masquer un problème plus essentiel de compréhension et/ou d'inadéquation du scénario pédagogique.

Dans la même situation, d'autres apprenants en difficulté ont demandé l'assistance d'un de leurs collègues, cherchant à connaître leur avancement dans l'activité, à avoir une confirmation de la solution de programmation et/ou à comparer les réactions du dispositif. Ces échanges avaient donc pour objet d'obtenir des explications d'un autre apprenant et d'avoir une perception de son activité, forme de tutorat entre pairs, la nature de la demande portant autant sur le contenu de l'activité que sur le besoin de se situer par rapport au groupe [BARNIER 01].

4.3. Traces informatiques

Les traces numériques, à l'aide de l'outil de fouille de données WUM (Web Utilisation Miner) [SPILIOPOULOU 00], ont confirmé des comportements attendus par rapport au modèle de navigation. Nous avons formalisé la détection de séquence pour valider ou invalider l'objectif pédagogique : « **demander de valider le code du programme** » (cf. figure 3). La recherche systématique des comportements attendus peut donc être automatisée en s'appuyant sur les modèles de navigation (cf. figure 1).

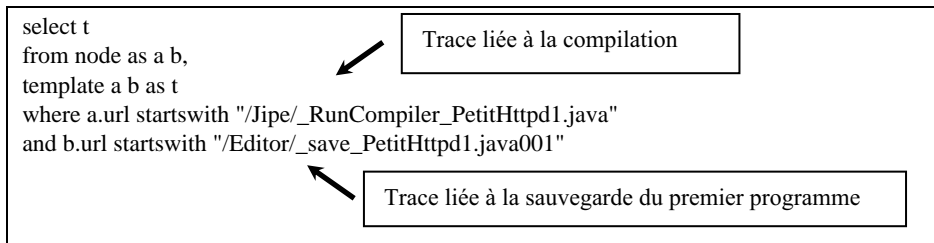


Figure 3. Exemple de script d'analyse des traces.

Par ailleurs, éclairé par les observations en salle, ce même outil d'analyse de trace, a permis de détecter différents comportements non inscrits dans la description initiale. Par exemple, le comportement d'un étudiant ayant compilé le code avant d'avoir sauvegardé ses modifications a été identifié à plusieurs reprises dans l'ensemble des traces. Ce même outil d'analyse a pu également nous informer sur le comportement d'un étudiant qui a compilé dix fois son programme avant de sauvegarder.

5. Discussion

Au cours de l'expérimentation, les trois outils d'observation mis en place ont été complémentaires. Le croisement des informations recueillies nous a permis de donner sens à certaines situations et comportements (à l'image de ceux que nous abordons dans cet article) et d'envisager une ré-ingénierie des scénarios pédagogiques et de ce qui les composent. Ce processus de ré-ingénierie s'appuie sur la complémentarité entre trois éléments qui pilotent une situation pédagogique : (1) le point de vue du concepteur exprimé par la description prédictive de l'activité, (2) le point de vue de l'apprenant qui vit directement la situation et (3) le point de vue du tuteur qui trouve, « à chaud », des solutions entre la prescription du concepteur et les attentes de l'apprenant. Cette perspective a des implications sur les cycles de conception et de tutorat.

5.1. Implications sur le cycle de conception

L'analyse guidée par les observables et le modèle de navigation de l'environnement présenté à l'apprenant permet de communiquer au concepteur de nouveaux observables pour d'éventuelles modifications, soit des documents supports de l'activité, soit dans leurs descriptions normalisées.

Concernant l'exemple où des apprenants essayent de compiler le code d'un programme sans l'avoir préalablement sauvegardé, situation dans laquelle ils souhaitent valider un code qui n'est pas le leur, trois solutions peuvent être envisagées par le concepteur. La première consiste à ajouter un nouvel objectif pédagogique lié à l'activité choisie, où l'acte de « **sauvegarder les dernières modifications** » doit être intégré à l'environnement, par exemple en modifiant l'ergonomie de l'interface. Par ailleurs, le déclenchement automatique de la sauvegarde du fichier avant chaque opération de compilation peut être considéré comme une deuxième solution. Enfin, une dernière proposition peut être de modifier le scénario pédagogique de l'unité d'enseignement en intégrant, par exemple, une intervention automatique du tuteur ciblé sur ce point particulier.

5.2. Implications sur le cycle de tutorat

A partir de l'analyse des résultats précédemment décrits, nous avons déterminé un ensemble de besoins spécifiques des acteurs humains (tuteur et étudiants) qu'il est possible d'instrumenter. Plus précisément, nous avons identifié les besoins liés à l'activité du tuteur ainsi que les attentes et demandes exprimées par les apprenants pendant le déroulement de l'activité « Apprendre en faisant ».

5.2.1. L'activité du tuteur

Le tuteur a besoin d'intervenir « juste à temps » auprès de l'apprenant. Ceci est difficile à mettre en œuvre compte tenu que les étudiants n'ont pas toujours conscience de leurs difficultés [DESPRES 01]. L'intervention du tuteur peut être

prescrite dans le scénario pédagogique en fonction de l'avancement de chacun des apprenants. L'exploitation automatique des traces nous permet d'instrumenter, à l'aide d'un capteur, la progression de l'activité du groupe et de l'apprenant, sachant que tous ne progressent pas au même rythme. En session de formation, cet outil génère des observables permettant de comparer l'avancement prévu avec celui d'un apprenant ou du groupe.

Cependant, l'analyse de l'activité de l'apprenant ne peut se limiter à une observation automatique. L'expertise du tuteur doit être sollicitée. Nous souhaitons définir des interventions du tuteur, soit de manière planifiée à la suite d'une activité particulière, soit en fonction de l'activité de l'apprenant.

5.2.2. *Le besoin des apprenants*

L'analyse des observations nous confirme l'importance du tutorat entre pairs [GOODLAD et HIRST 89] et du besoin des apprenants de se situer par rapport au groupe. Nous pouvons utiliser le capteur défini pour le tuteur, mais cette fois comme indicateur à destination des apprenants.

Par ailleurs, en nous appuyant sur les caractéristiques du public et sur les observations automatiques, nous proposons de scénariser les échanges entre apprenants. Par exemple, le dispositif peut proposer à un apprenant en difficulté de contacter un apprenant ayant déjà réalisé la même activité.

6. Conclusion

Dans cette expérimentation, nous voulions déterminer les écarts existants entre scénarios prescrits et comportements émergents. Nous avons analysé les interactions entre les acteurs humains du point de vue des stratégies développées et des contenus des interventions. Nos conclusions confortent l'idée qu'il émerge de l'activité des différents acteurs des scénarios non conformes aux prévisions. Considérant cet élément comme une donnée inhérente à tout usage d'EIAH, nous pensons que les efforts de normalisation doivent prendre en compte cette dimension. Il est nécessaire, dans un processus de ré-ingénierie, de penser les modalités techniques et méthodologiques d'une adaptation des scénarios aux usages émergents, pendant et en fin d'activité, plutôt que de trouver les moyens de contraindre les usages.

Dans cette perspective, il s'agit, d'une part, de mieux caractériser les activités de tutorat et d'autre part, de mettre en œuvre des processus de ré-ingénierie visant à adapter les scénarios prescrits. Pour cela, nous utilisons (1) une méthodologie d'observation du scénario émergent reposant à la fois sur le recueil de traces informatiques, l'observation en situation et l'interview des acteurs, (2) une analyse des écarts constatés afin de dégager des éléments significatifs permettant (3) d'enrichir le scénario prescrit.

7. Bibliographie

- [BARNIER 01] BARNIER G., Le tutorat dans l'enseignement et la formation, L'Harmattan, 2001, p 287.
- [BAZSALICZA et al. 01] BAZSALISCZA M., NAIM P., Data Mining pour le Web. Editions Eyrolles, 05/2001.
- [BROCKE et al. 00] BROCKE J.V., NORMAN L., FreeStyle Learning A new way to Computer Assisted Learning (CAL). Vancouver 2000.
- [CROZAT 02] CROZAT S. Eléments pour la conception industrialisée des supports pédagogiques numériques, Thèse de doctorat, Université Technologique de Compiègne, 2002.
- [DE LA PASSARDIERE et al 01] DE LA PASSARDIERE B., GIROIRE H., XML au service des applications pédagogiques, Actes des journées EIAO'2001, revue Sciences et Techniques Educatives Vol. 8, 2001, pp 99-112.
- [DESPRES 01] DESPRES C., Modélisation et conception d'un environnement de suivi pédagogique synchrone d'activités d'apprentissage à distance, 2001, Thèse de doctorat.
- [GOODLAD & HIRST 89] GOODLAD S., HIRST, B., Peer tutoring: a guide to learning by teaching, London, Kogan Page, 1989.
- [PAQUETTE 96] PAQUETTE G., La modélisation par objets typés : une méthode de représentation pour les systèmes d'apprentissage et d'aide a la tâche, Sciences et Techniques Educatives, 3(1), 1996, pp9-42.
- [PAQUETTE 00] PAQUETTE G., Construction de portails de télé-apprentissage Explor@, une diversité de modèles pédagogiques, Science et Techniques Educatives 7(1), 2000, pp 207-226.
- [KOPER 01] KOPER R., HERMANS H.J.H., LOEFFEN A., MANDERVELD J.M. et RUSMAN E.M., Edubox-EML Reference Manual (Beta version), 15/12/2000. <http://eml.ou.nl/eml/docs/edubox1.0.html>.
- [RAWLINGS et al. 02] RAWLINGS A., ROSMALEN P., KOPER R., RODRIGUEZ-ARTACHO M. et LEFRERE P., Survey of Educational Modelling Languages (EMLs), CEN/ISSS WS/LT Learning Technologies Workshop, 2002, pp 80.
- [SPILIOPOULOU 00] SPILIOPOULOU M., Web Usage Mining for Web Site Evaluation. Communication of the ACM, Vol 43, Number 8, 2000, pp 127-134.
- [TCHOUNIKINE 02], TCHOUNIKINE P., (2002), Pour une ingénierie des Environnements Informatiques pour l'Apprentissage Humain, In: Revue I3 "information interaction intelligence", Vol. 2, n°1, pp. 59-95.

Remerciement

Nous remercions Benoît Lemaire pour la relecture de cet article.