



HAL
open science

Design Patterns en EIAH : vers un langage de Patterns pour l'évaluation des apprenants

Elisabeth Delozanne, Françoise Le Calvez, Agathe Merceron, Jean-Marc Labat

► To cite this version:

Elisabeth Delozanne, Françoise Le Calvez, Agathe Merceron, Jean-Marc Labat. Design Patterns en EIAH : vers un langage de Patterns pour l'évaluation des apprenants. STICEF (Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation), 2007, 14, pp.45-80. hal-00696316

HAL Id: hal-00696316

<https://hal.science/hal-00696316v1>

Submitted on 11 May 2012

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.

Design Patterns en EIAH : vers un langage de Patterns pour l'évaluation des apprenants.¹

Élisabeth Delozanne [CRIP5, Paris, Françoise Le Calvez [LIP6, Paris]

Agathe Merceron [TFH, Berlin, Jean-Marc Labat [LIP6, Paris]

■ **RÉSUMÉ** : Dans cet article nous présentons un premier ensemble de 17 Design Patterns (DP) pour la conception d'EIAH qui évaluent des apprenants en situation de résolution de problèmes, par exemple des problèmes mathématiques, logiques ou de programmation. La démarche d'évaluation décrite par ces DPs comporte trois étapes : premièrement, recueil et analyse des informations sur un seul apprenant pour un seul exercice, deuxièmement construction d'une vue générale sur l'activité individuelle d'un apprenant sur un ensemble d'exercices et, troisièmement élaboration d'une vue générale sur l'activité de toute une classe sur un même ensemble d'exercices. Nous montrons comment cet ensemble de DPs mis au point pour capitaliser l'expérience de plusieurs projets menés par les co-auteurs rend compte également de la conception de l'évaluation des apprenants dans d'autres EIAH décrits dans la littérature. Nous soutenons que l'approche Design Pattern facilite la communication entre les disciplines intervenant dans la conception des EIAH et permet de capturer leur expérience de conception afin de faciliter la réutilisation dans des projets à venir, de cumuler les résultats de recherche et de participer à la formation des jeunes chercheurs du domaine. L'ensemble de DPs présentés ici est un premier pas en ce sens.

■ **MOTS CLÉS** : Design Patterns, Langage de Design Patterns, évaluation des apprenants, Résolution de problème, Diagnostic cognitif, Environnements Informatiques pour l'apprentissage Humain (EIAH)

■ **ABSTRACT** : This paper presents a structured set of 17 Design Patterns (DP for short). These DPs target the design of learning systems that assess learners while solving problems, for example problems in mathematics, logic or programming. Learners' assessment is a three step approach: first collect and analyse data while a learner solves a problem, second overview of a learner's activity across a set of exercises and third overview of the activity of a group of learners on a set of exercises. This set of DPs has emerged as a way of capitalizing on the work done by the co-authors while sharing their experience in this area. We show that these DPs do account for other experiences taken from the literature. In our opinion, writing sets of DPs is a way to facilitate communication in pluridisciplinary teams for the design of e-learning systems, to create a research memory, and to training and education in the field. The present set is a step in that direction.

■ **KEYWORDS** : Design Patterns, Design Pattern Language, Students' Assessment, Problem solving, Cognitive Diagnosis, Technology Enhanced Learning

- [1. Introduction](#)
- [2. L'approche Design Pattern](#)
- [3. Exemples de collections de DPs en EIAH](#)
- [4. Objectifs de recherche et méthodologie](#)
- [5. Expériences de référence](#)
- [6. Un ensemble structuré de Design Patterns](#)
- [7. Discussion](#)
- [8. Conclusion](#)
- [BIBLIOGRAPHIE](#)

1. Introduction

La recherche pour la conception de systèmes informatiques destinés à favoriser des apprentissages a maintenant plus de vingt ans. Depuis quelques années, de nombreux efforts sont entrepris pour capitaliser les expériences et définir des méthodes de conception plus systématiques. Le travail présenté ici va dans ce sens. Nous faisons l'hypothèse que l'approche Design Patterns (DP) est un moyen pour recueillir et partager des expériences, pour mener une réflexion en profondeur et pour capitaliser et partager des résultats obtenus sur des projets de recherches spécifiques.

C'est l'architecte Alexander ([Alexander et al., 1977](#)) qui a initié cette approche. L'intention d'Alexander était de faciliter la participation de tout un chacun à la conception de l'espace dans lequel il vit. La définition qu'Alexander

a donnée d'un DP fait encore référence aujourd'hui : « chaque pattern décrit un problème qui se pose de façon récurrente dans notre environnement puis il décrit l'essentiel de la solution de ce problème et ce de façon que l'on puisse réutiliser cette solution un million de fois sans jamais la reproduire à l'identique » (Alexander et al., 1977 p. x). Ainsi, un DP n'est pas une prescription mais un outil conceptuel pour aborder la résolution d'un problème. Un DP n'est jamais utilisé de façon isolée. Il fait partie d'un réseau de DPs qui crée ce qu'Alexander appelle un langage. « Il s'agit de présenter chaque pattern en relation avec les autres patterns, de façon à ce que l'on comprenne la collection des patterns comme un tout, un langage à partir duquel on puisse créer une infinie variété de combinaisons » (Alexander et al., 1977 p. xv). Dans l'optique de conception participative qui est celle d'Alexander, ce langage peut être utilisé par tout individu (particulier ou professionnel), qu'il construise lui-même sa propre maison ou qu'il travaille dans une équipe d'architectes concevant des bureaux ou des espaces publics.

Depuis 30 ans, l'approche DP a fait son chemin et est utilisée dans de nombreuses disciplines : en architecture (Alexander et al., 1977) mais aussi en Génie Logiciel et particulièrement en programmation objet (Gamma et al., 1995), (Buschmann et al., 1996), (Schmidt et al., 2000), (Larman, 2003), en Interaction Humain-Machine (Chung et al., 2004), (Deng et al., 2005), en conception de sites Internet (Van Duvne et al., 2002), (Van Welie, 2004) par exemple, et même des Patterns pour écrire des Patterns (Meszaros et Doble, 1998). Dans les domaines de la pédagogie et des EIAH, des collections de DPs sont maintenant disponibles ; par exemple, des collections de DPs ont été publiées dans le cadre des projets *Pedagogical Patterns Pr`ject* (PPP, 2000), *E-Learning Expertise Eur`pean Netw`rk* (E-LEN, 2003), (Avgeriou et al., 2003) et *Pattern Language f`r Architectures `f Intelligent Tut`ring systems*, *PLAIT* (Devedzic, 2001), (Devedzic et Harrer, 2005).

Dans le cadre du réseau d'excellence européen Kaléidoscope, le projet DPULS *Design Patterns f`r rec`rding and analysing Usage `f Learning Systems* (Choquet, 2004), avait pour objectif d'identifier un ensemble de DPs pour l'analyse des traces d'activité d'apprenants interagissant avec des EIAH. Comme dans E-LEN, l'objectif était de fournir des outils conceptuels pour systématiser la conception d'EIAH, mais aussi de fournir un support pour partager des expériences et pour construire des références communes dans un consortium regroupant des équipes de différentes nationalités, différentes disciplines et différentes cultures. Dans la continuité des travaux des projets PPP et E-LEN, le projet DPULS s'est lui centré sur le recueil et l'analyse de données sur l'activité des apprenants.

Dans cet article, nous présentons le sous-ensemble des DPs sur lequel l'équipe AIDA² a particulièrement travaillé dans le cadre du projet DPULS. Nous avons appelé cet ensemble : évaluation de l'Apprenant (DP EA). Il concerne l'analyse des traces des apprenants dans un EIAH pour détecter les savoirs, savoir-faire et stratégies mis en œuvre par des apprenants lors de la résolution de problèmes dans des domaines comme les mathématiques, la logique ou la programmation. Dans un premier temps, en nous appuyant sur l'expérience acquise par d'autres équipes dans la mise en évidence de DPs, en particulier en EIAH, nous précisons nos objectifs de recherche et la méthodologie que nous avons employée. Puis, après avoir présenté trois expériences de conception qui nous ont servi de références pour dégager des invariants et identifier des DPs, nous détaillons l'ensemble des DPs que nous avons mis au point. Ensuite une discussion montre la généralité de notre approche en la comparant avec des travaux similaires. Enfin nous abordons les perspectives ouvertes par ce travail.

2. L'approche Design Pattern

Les communautés de recherche qui ont adopté l'approche DP sont assez consensuelles sur la définition d'un DP comme une solution à un problème récurrent dans un contexte donné (Goodyear et al., 2004). Cependant, chaque communauté a adapté l'approche d'Alexander à ses besoins et les points de vue diffèrent sur de nombreux points, par exemple :

- *P`urqu`i créer des ensembles de DPs ?* Les objectifs varient d'un projet à l'autre : il s'agit (i) de créer un langage commun dans une équipe pluridisciplinaire ou multiculturelle, (ii) de capturer ou disséminer une expertise, (iii) de capitaliser des expériences passées, (iv) d'améliorer la qualité du produit (par exemple en standardisant les solutions apportées à un problème ou en automatisant certains processus récurrents) ou (v) d'enseigner la conception de systèmes. Dans le travail présenté ici, nous nous sommes limités, pour l'instant, aux trois premiers objectifs.

- *Qui s`nt les utilisateurs des DPs ?* Les ensembles de DPs sont mis au point pour assister des experts, des chercheurs, des membres d'une communauté de pratique ou d'une équipe pluridisciplinaire, les utilisateurs finaux du produit de la conception ou, même, des programmeurs chargés de la mise en œuvre de la solution. Par exemple, les DPs d'Alexander, dans une approche participative de la conception, avaient pour cible principale les habitants - non-architectes - des bâtiments à concevoir ; les célèbres DPs (Gamma et al., 1995) surnommés DPs de la « bande des quatre » (Gang of Four ou GoF) sont eux, destinés à des programmeurs professionnels. Dans notre travail, les utilisateurs visés sont d'une part, les différents membres d'une équipe pluridisciplinaire de conception afin de créer un langage commun et, d'autre part, les chercheurs en EIAH afin de capturer, de disséminer une expertise et de capitaliser des expériences passées.

- *C`mment créer des DPs ?* L'approche la plus fréquente est une approche ascendante qui consiste à identifier dans des expériences passées des solutions efficaces à des problèmes récurrents : les patterns ne sont pas inventés mais sont « découverts » (Devedzic et Harrer, 2005) dans une approche de Pattern Mining (Avgeriou et al., 2003). Certains auteurs recommandent une approche descendante (Hernandez-Leo et al., 2005) ou une approche ascendante dirigée par la théorie (E-LEN, 2003), c'est-à-dire qui consiste à étudier les expériences passées en se référant à des

approches théoriques (pour ces auteurs, il s'agit d'approches socioconstructivistes). Étant donné le contexte pluriculturel et pluridisciplinaire de notre travail, nous nous situons dans la première approche : nous avons cherché des invariants dans les solutions apportées à des problèmes récurrents.

- *Quel est le format d'un DP ?* Dans toutes les collections publiées de DPs, tous les DPs sont présentés selon un format prédéfini, mais selon les objectifs et les utilisateurs visés, les formats varient. Ils peuvent être informels quand ils sont destinés à être utilisés par des humains, semi-formels (par exemple sous forme de diagramme UML) ou formels (sous forme de fichier XML par exemple) quand ils sont destinés à être mis en œuvre par des machines. Leur structure reflète des variations autour d'un noyau commun : le format défini par Alexander. Ainsi un pattern est composé presque toujours d'un nom, de la description du problème, de la description des solutions, de la description d'exemples de conception réussie où ce pattern a été détecté ou a été utilisé et, enfin, d'une liste de patterns apparentés. Ensuite, selon les domaines d'autres composants sont ajoutés : par exemple, une description du contexte, des dessins ou des photos, une catégorie. Certains auteurs (E-LEN, 2003) (Landay et Van Duyne, 2004) soulignent qu'un des écueils est de passer trop de temps à discuter de la structure au détriment du contenu des DPs, alors que d'autres auteurs s'attachent à donner une définition formelle de la structure des DPs (Borchers, 2001). Les partenaires du projet DPULS ont mis au point une version formelle du format des DPs sous forme de schéma de fichier XML afin de permettre des traitements automatiques sur la collection de DPs par un navigateur de DPs (DPULS - 7, 2005). Dans ce papier, afin d'en faciliter la lecture, nous présentons une version simplifiée de la structure en nous limitant aux composants utiles pour la compréhension humaine.

- *Quel nom ad`pter p`ur un DP ?* Le nom doit permettre d'identifier rapidement le contenu du DP et devenir une expression du langage de conception (Alexander et al., 1977), (Meszáros et Doble, 1998). Plusieurs façons de nommer les DPs sont en concurrence : adopter une métaphore liée au domaine (par exemple dans PPP, le pattern Early Bird indique que, dans la mise en place d'un cours semestriel, il est conseillé d'évoquer les notions importantes le plus tôt possible), référer au problème ou référer à la solution. L'utilisation de métaphores nous est vite apparue inadaptée dans notre contexte multiculturel. Nous avons donc choisi d'utiliser le nom des problèmes ou des solutions. Comme dans (Van Duyne et al., 2002), nous leur avons associé une abréviation qui rend compte de leur catégorie (cf. section 6).

- *Comment évaluer un DP ?* Plusieurs façons de valider un DP sont présentées dans la littérature. Certains auteurs suggèrent qu'un patron peut être accepté par la communauté s'il vérifie « la règle de trois », c'est-à-dire s'il a été mis en évidence ou utilisé avec succès dans trois expériences autres que celle de son auteur. Cette règle est soutenue, en autres, par (Buschmann et al., 1996), (Landay et Van Duyne, 2004), (Chung et al., 2004), (Devedzic et Harrer, 2005). Certains auteurs distinguent la découverte d'un pattern (à partir de deux occurrences de la solution) de son utilité (évaluée par le nombre d'usages connus de ce pattern). Une fois un pattern « découvert », il fait l'objet de critiques par des pairs ou par des professionnels avant d'être publié (Landay et Van Duyne, 2004). Remarquons que l'expression « usage d'un DP » recouvre deux sens différents : d'une part, les « découvreurs » d'un DP reconnaissent le schéma de solution dans des expériences de conception et, d'autre part, une fois le DP publié, des concepteurs revendiquent son utilisation (par exemple, en modélisation objet, les concepteurs utilisent le DP « fabrique » de GoF). (Devedzic et Harrer, 2005) parlent de « l'évidence » d'un DP ou de sa « récurrence ». L'approche DP en EIAH en étant à ses balbutiements, il ne peut s'agir, dans ce texte, que de la première acception. Les DPs que nous proposons satisfont au moins à la règle de trois. Ils ont été validés en interne par une revue critique par au moins trois équipes différentes au sein du projet et ont été mis en évidence au moins une fois dans des travaux de chercheurs extérieurs au projet (cf. section 7).

- *Comment créer un langage de patterns ?* Alexander estimait que des patterns isolés n'étaient pas d'une grande utilité, mais qu'en connectant ensemble des patterns apparentés et en montrant comment les composer, on pourrait créer un véritable langage de conception. Ainsi, certains auteurs parlent de langage de Design Patterns pour désigner des collections structurées de DPs - par exemple, (Van Duyne et al., 2002), alors que d'autres parlent plus modestement d'ensembles ou de collections de patterns (par exemple GoF). Après avoir identifié des patterns dans un domaine, il faut les organiser soit en une hiérarchie, soit en un réseau. Les critères d'organisation sont très variés. Tous les auteurs mentionnent qu'il existe plusieurs façons d'organiser un ensemble de DPs, l'essentiel étant de faciliter leur repérage pour organiser une conception cohérente. (Gamma et al. 1995) classent leurs DPs selon deux critères : le rôle (créateur, structurel, comportement) et le domaine (classe ou instances). Toujours en génie logiciel, il est d'usage de distinguer des patterns d'analyse et des patterns de conception (Fowler, 1997), (Devedzic et Harrer, 2005), (Borchers, 2001) et (Van Duyne et al., 2002) font intervenir plusieurs dimensions dans leur classification : le niveau d'abstraction (tâche ou style), la fonction (perception, manipulation et navigation) et le rendu effectif (dimension spatiale ou temporelle). Cependant (Devedzic et Harrer, 2005) notent que si les noms des patterns et leur organisation sont importants pour aborder un langage de patterns et s'y orienter, dans la pratique, ce sont les contenus qui sont importants pour l'analyse, pour l'utilisation des patterns et pour l'étude d'alternatives ou de variantes. En ce qui nous concerne, l'approche DP en EIAH étant récente, nous préférons garder le terme de collection, plutôt que de parler de langage de DPs qui sous-entend un consensus qui n'existe pas encore. Quant à la structuration de la collection, elle a donné lieu à beaucoup de discussions au sein des membres du projet sans doute pour arriver à s'expliquer et à négocier un langage commun entre les différentes équipes et cultures. La structure que nous présentons résulte de l'obtention d'un consensus (cf. section 4). Elle a permis aux différentes équipes de reconnaître leurs expériences et de les situer par rapport aux autres. Dans le navigateur de patterns mis au point dans la suite du projet, c'est l'utilisateur qui crée sa propre structure en suivant les liens entre patterns apparentés. C'est un avantage de l'approche DP de permettre cette flexibilité dans la structuration.

3. Exemples de collections de DPs en EIAH

Dans les domaines de l'éducation et des EIAH, trois projets ont popularisé l'approche DP en publiant des collections de DPs : les projets PPP, E-LEN et PLAIT.

Le Pedagogical Patterns Project ([PPP, 2000](#)) a mis en ligne trois ensembles de DPs de scénarios pédagogiques. Ce sont des patterns assez généraux qui élicitent l'expérience d'enseignants. Ils sont exprimés selon le format proposé par Alexander et sont décrits en langage naturel sous forme narrative en s'adressant directement aux enseignants ou aux formateurs : les tournures impersonnelles sont remplacées par des formes actives : « vous voulez... », « votre problème... », « vous allez... ». Le premier ensemble de 14 DPs est présenté comme une première étape vers un langage de DP pour développer des cours d'informatique ; il est organisé suivant les thèmes suivants : Enseigner selon différentes perspectives, Apprentissage Actif, Rétroaction, Apprentissage par l'expérience, Tirer profit de différentes perspectives. Un second ensemble de 48 patterns concerne la mise en place de séminaires et enfin un ensemble de cinq patterns la mise en place d'un cours.

Le projet E-LEN ([Avgeriou et al., 2003](#)), ([Avgeriou et al., 2004](#)), ([Goodyear et al., 2004](#)), ([E-LEN, 2003](#)) a développé une banque de 40 DPs regroupés en quatre centres d'intérêt (SIG, Special Interest Groups) : Ressources d'apprentissage pour les plateformes du e-learning, Apprentissage tout au long de la vie, Apprentissage collaboratif et Apprentissage adaptatif. L'objectif de ce projet est de construire une base de connaissances pour les concepteurs pédagogiques. Cette équipe de chercheurs recommande l'approche DP pour recueillir et disséminer l'expertise de conception, pour partager l'expérience et favoriser le travail dans des équipes pluridisciplinaires. Ils ont mis au point un guide pour l'écriture de DPs pour le e-learning ([E-LEN, 2003](#)). Les utilisateurs cibles de leurs DPs sont les concepteurs de plateformes de e-learning. Leurs DPs sont aussi très généraux. Par exemple, le DP « Pister l'étudiant » suggère des fonctionnalités devant être implémentées dans un système, mais ne précise pas comment les implémenter. Le concepteur est responsable de la génération d'une solution adaptée à son contexte. Éventuellement, il définira un pattern plus spécialisé s'il détecte des invariants dans sa solution. « Un pattern suggère une solution plutôt qu'il ne la prescrit » ([E-LEN, 2003](#)).

Le projet PLAIT ([Devedzic, 2001](#)), ([Devedzic et Harrer, 2005](#)) propose des patterns concernant les architectures logicielles des ITS. Contrairement aux deux projets précédents qui s'adressent à des enseignants ou des pédagogues, les DPs de PLAIT s'adressent aux informaticiens qui mettent en œuvre des ITS. Les auteurs ont « découvert » ces patterns à partir d'une étude sur 96 articles dans les conférences et journaux de la communauté AIED-ITS et à partir de leur connaissance des DPs du génie logiciel. Leur objectif principal est de contribuer à une meilleure compréhension des architectures des différents ITS et de promouvoir une approche de conception plus systématique. Pour chaque type d'ITS étudié, ils présentent les modules de base et les variantes. Ils étudient, par exemple, l'architecture classique des ITS, l'architecture fondée sur différentes vues sur les modèles de connaissances, l'architecture de systèmes d'apprentissage collaboratif, et l'architecture de systèmes utilisant des agents pédagogiques ou des compagnons virtuels.

4. Objectifs de recherche et méthodologie

Dans le cadre du projet DPULS, l'équipe AIDA s'est focalisée sur l'analyse des traces pour l'évaluation des apprenants. L'objectif à long terme est, comme dans les projets précédents, de capitaliser l'expérience des membres de l'équipe, de fertiliser les recherches par le partage d'expériences et, plus généralement, de supporter d'autres concepteurs d'EIAH qui ont à évaluer des apprenants. Nous ne nous situons pas au niveau de la conception objet du logiciel, mais au niveau de la démarche générale pour aborder la conception d'un système d'évaluation. Plus précisément nous avons cherché à :

- Trouver des invariants dans les solutions expérimentées par les membres du projet pour résoudre des problèmes récurrents pour l'évaluation de l'apprenant ;
- Étudier dans quelle mesure ces invariants se rencontrent dans d'autres expériences pour résoudre des problèmes similaires ;
- Présenter un ensemble cohérent de DPs exprimant ces invariants.

A titre d'exemple, voici un scénario qui illustre le genre de problème qu'un concepteur peut avoir à résoudre pour concevoir un système qui rende compte de l'activité d'un apprenant dans un EIAH.

Scénario 1 : Sarah est une enseignante qui organise une séance de TP pour un groupe d'apprenants. A la fin de la séance, elle voudrait une vue générale de l'activité des apprenants pour adapter le cours suivant à leur évolution. Plus précisément, elle veut un rapport lui indiquant si certaines compétences sont acquises et avoir un aperçu des différentes stratégies de résolution utilisées par les apprenants.

L'approche que nous proposons dépasse les limites des approches de l'évaluation proposées par de nombreuses plateformes d'enseignement en ligne où l'évaluation consiste à affecter une note ou un taux de réussite après avoir évalué l'apprenant à travers un quizz, un questionnaire à choix multiples, des réponses numériques ou de simples clics, - cf. par exemple le DP *Management of on-line questionnaire* ([Avgeriou et al., 2003](#)). Nous avons pour

objectif d'évaluer aussi les productions des apprenants quand ils effectuent des tâches spécialement conçues pour donner du sens à la connaissance enseignée et où ils ont à réaliser des opérations cognitives complexes. De nombreux travaux en EIAH, par exemple nos propres travaux au sein de AIDA mais aussi ([VanLehn et al., 2005](#)), ([Nicaud et al., 2005](#)), montrent l'intérêt de ne pas se contenter d'analyser une simple réponse mais également le raisonnement qui a conduit à cette réponse. Ce type d'évaluation nécessite plus qu'une évaluation en termes de réponse « correcte/incorrecte ». Il nécessite une analyse multidimensionnelle fondée sur des études pédagogiques, didactiques ou psychologiques précises de l'activité de l'apprenant. Dans notre scénario, ceci peut être illustré de la façon suivante :

Scénari` 1 bis : Sarah est enseignante expérimentée. A partir de sa pr`pre expérience et d'études didactiques, elle est capable d'identifier les différentes interprétati`ns pers`nnelles que les apprenants se c`nstruisent généralement sur les pr`blèmes qu'elle leur p`se. Elle s'est f`rgée une typ`l`gie des rép`nses et veut gr`uper les apprenants sel`n le type de leurs rép`nses.

Les DPs sont généralement conçus et mis au point via un processus de collaboration assez long comprenant plusieurs étapes, en particulier avec des révisions critiques. L'ensemble de DPs produits dans le projet DPULS capture l'expérience d'équipes pluridisciplinaires et multiculturelles. Nous avons adopté une méthodologie en cinq étapes, chaque étape nécessitant des interactions entre les différents partenaires européens du projet. Chaque équipe participante a d'abord exposé ses expériences ([DPULS - 2, 2005](#)) et, parallèlement nous avons établi un état de l'art sur notre sujet ([DPULS - 3, 2005](#)). Puis, à partir des relations d'expériences, nous avons mis en évidence un ensemble de problèmes récurrents et chaque équipe a établi une liste de ses savoir-faire pour résoudre ces problèmes. Cette première liste était donc encore très spécifique aux expériences locales des équipes ([DPULS - 4, 2005](#)). Ensuite une étape, difficile, a consisté à travailler sur la description de l'énoncé des problèmes et des solutions en des termes à la fois assez généraux pour couvrir chaque expérience et assez concrets pour ne pas vider la solution de toute substance. Dans cette troisième étape, nous avons, en parallèle, négocié un format pour exprimer l'ensemble des DPs ([DPULS - 5, 2005](#)) et pour la conception d'un navigateur destiné à la consultation des DPs ([DPULS - 7, 2005](#)). Dans une quatrième étape, nous avons travaillé pour obtenir un consensus sur la structuration des patterns dégagés dans l'étape précédente et pour les intégrer en un ensemble cohérent. Cette étape de structuration a provoqué la réécriture de certains patterns, leur éclatement en plusieurs patterns plus spécifiques ou au contraire leur regroupement en un seul. De plus, certains aménagements ont été apportés au format de description des patterns. Des itérations ont donc été nécessaires entre les étapes trois et quatre. Dans la cinquième étape, une fois que nous sommes arrivés à un ensemble structuré assez stable pour rendre compte de différentes expériences traitant des problèmes similaires, les DPs ont été saisis dans le navigateur ([DPULS - 6, 2005](#)).

5. Expériences de référence

Ce qui caractérise un ensemble de DPs, ce n'est pas l'originalité des démarches qu'il propose : « Les patterns n'énoncent pas des idées nouvelles, ils codifient des principes de base très utilisés. Un expert les trouvera très basiques et très familiers. C'est leur objectif » (Larman, 2002, p. 239). En revanche, le mérite d'un ensemble de DPs est de rassembler et de structurer un ensemble de bonnes pratiques. Dans le travail présenté ici, nous avons d'abord généralisé à partir de l'expérience des membres de l'équipe AIDA dans la conception de l'évaluation des apprenants dans six EIAH : Combien? ([Combien?, 2000](#)), Diane ([Hakem et al., 2005](#)), Java-En-Ligne ([Duval et al., 2005](#)), le Logic-ITA ([Merceron et Yacéf, 2004](#)), Pépite ([Pépite, 2000](#)), Math Logs ([Vandebrouck et Cazes, 2005](#)). Dans la section suivante, nous présentons Combien?, Pépite et le Logic-ITA pour que les lecteurs puissent mettre des exemples assez différents sur le contenu des DPs.

5.1. Expérience 1: Le projet Combien? ([Combien?, 2000](#))

Combien? entraîne les étudiants à résoudre des exercices de dénombrement et à justifier leur solution. En dénombrement (compter le nombre de façons d'arranger des objets d'une façon donnée) l'essentiel du processus de résolution ne vient pas d'une façon astucieuse de mener des calculs ou d'enchaîner des inférences, mais réside dans la construction d'une représentation adaptée de la situation et en sa transformation en représentations équivalentes. Ce projet se fonde donc sur l'importance des représentations de situations dans la résolution d'un problème. Combien? s'appuie sur la « méthode constructive » ([Le Calvez et al., 2003](#)), une méthode de résolution mise au point par des enseignants et qui tient compte des conceptions usuelles des étudiants pour leur donner accès à la théorie mathématique de ce domaine.

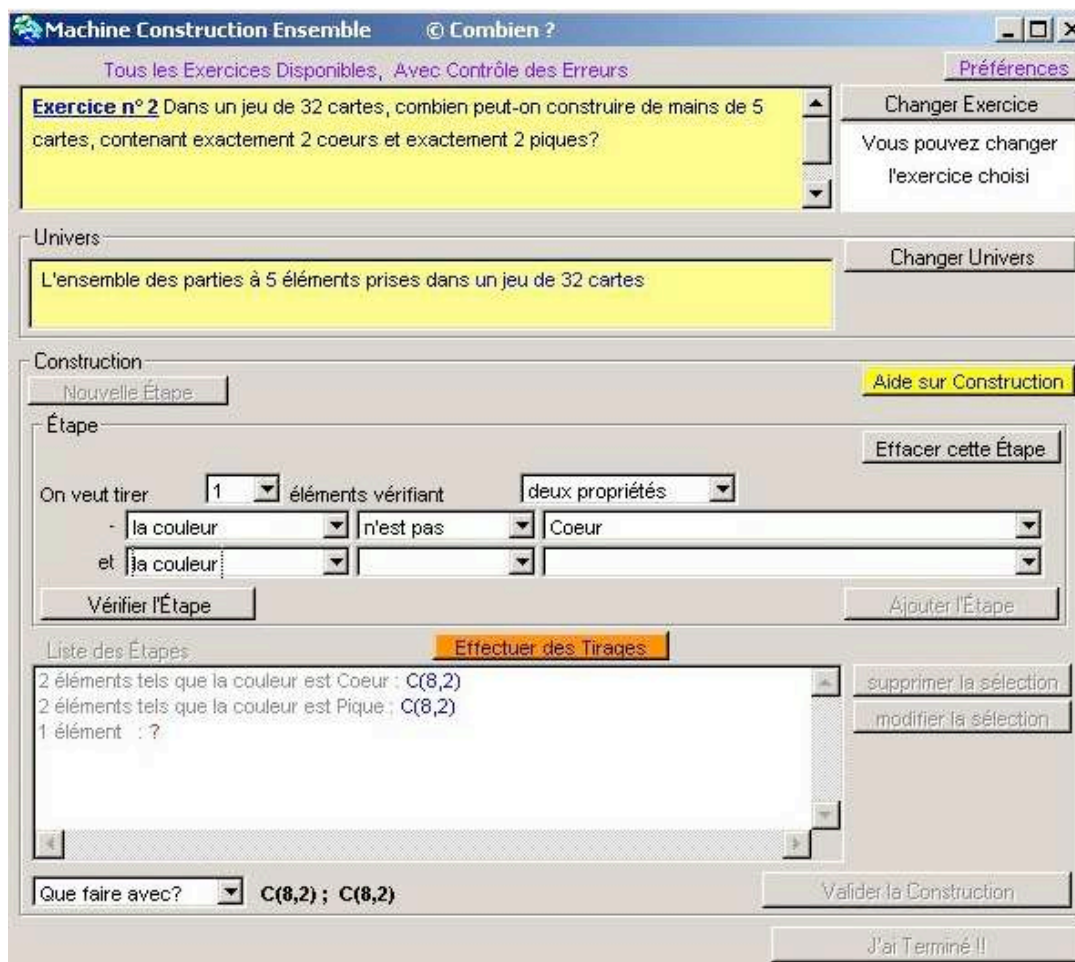


Figure 1: Une étape de résolution dans Combien?

Combien? demande aux étudiants de construire un élément générique (appelé configuration) de l'ensemble solution. Ils doivent décrire leur construction sous la forme d'un ensemble (appelé univers des solutions) et d'un ensemble de contraintes sur cet ensemble. Pour trouver la solution numérique, ils doivent raisonner sur leur construction. Les exercices sont classés par référence au type de leurs solutions ; Combien? propose une interface différente pour la résolution de chaque classe d'exercices.

A chaque étape de la construction, le système détermine automatiquement si la solution engagée par l'étudiant mène à une construction correcte ou non et, dans ce dernier cas, il fournit des indices pour aider l'étudiant à identifier ses erreurs. Tous les ans, une centaine d'étudiants de seconde année d'université utilisent le système en TP. Ils peuvent aussi utiliser Combien? pour s'entraîner par eux-mêmes en libre service ou chez eux. Toutes les actions des étudiants (tout ce qu'ils saisissent qu'ils aient validé ou non) ainsi que l'heure de l'action sont enregistrées afin de pouvoir rejouer la session de l'étudiant. Combien? détecte les erreurs, les classe et mémorise leur type et leur caractérisation. Toutes ces informations sont enregistrées dans des fichiers XML qui sont analysés a posteriori. Deux types d'analyse sont menés. Premièrement, chaque étudiant obtient un compte-rendu détaillé de sa session : pour chaque exercice, le temps passé, le nombre d'erreurs et leur type, le nombre d'hésitations, si l'exercice a été terminé ou abandonné et le nombre de consultations du support de cours. Ce premier type d'analyse concerne aussi bien l'apprenant que l'enseignant. Deuxièmement, les enseignants disposent des exercices classés selon leur niveau de difficulté et de groupes d'étudiants classés selon leur taux de réussite et les types d'erreurs qu'ils ont commises.

5.2. Expérience 2 : Le projet Pépite (Pepite, 2000)

Le logiciel Pépite recueille et analyse les réponses des élèves à une série d'exercices spécialement conçus pour permettre un diagnostic cognitif en algèbre élémentaire (Jean, 2000), (Delozanne et al., 2005). Ce travail s'appuie sur une analyse didactique (Grugeon, 1995) qui ne s'intéresse pas seulement aux erreurs des élèves mais détecte des cohérences dans les conceptions des élèves afin soit de les renforcer pour consolider un apprentissage en cours, soit de les déstabiliser quand ces conceptions sont inadaptées et font obstacle à l'apprentissage. Les réponses des élèves sont exprimées de diverses manières : par des expressions algébriques, par un raisonnement algébrique, par des explications librement formulées en français, par des choix multiples, des clics ou des glisser-déplacer. L'évaluation d'un élève se fait en trois étapes. Tout d'abord chaque réponse est codée en fonction d'un ensemble de 36 critères regroupés sur 6 dimensions (cf. Figure 2). Ainsi, Pépite commence par créer, pour chaque élève, un fichier XML contenant les réponses et le code attribué par le système à chaque réponse. Une interface graphique permet à l'enseignant de vérifier ou corriger ce codage automatique (Cf. Figure 2). Puis, pour obtenir un aperçu de

plus haut niveau sur l'activité de l'élève, le profil cognitif est construit en cumulant, sur l'ensemble des exercices les codes identiques ou en associant ou opposant certains codes. La méthode de cumul et d'association des critères est fondée sur une expertise didactique. Ce profil est exprimé par des taux de réussite, par les points forts et points faibles sur trois dimensions (usage de l'algèbre, calcul algébrique et traduction d'une représentation en une autre). Enfin, en utilisant ce profil, Pépite situe l'élève sur une échelle de compétence sur chacune de ces dimensions en la classant dans une classe de profils (appelée stéréotype).



Figure 2: Le diagnostic automatique de Pépite sur 6 dimensions, modifiable par un évaluateur humain.

Un enseignant dispose ainsi de trois types de résultats calculés par Pépite :

- un regroupement des élèves selon leurs points forts et leurs points faibles sur plusieurs dimensions qui permet à l'enseignant d'avoir une vue générale sur les compétences de la classe en situant chaque élève par rapport à une grille de compétence ;
- un rapport détaillé sur le profil cognitif de chaque élève (qui permet aussi au système de calculer les groupes) ;
- une liste d'activités d'apprentissage adaptées à chaque groupe.

Cinq cents élèves ont passé le test avec Pépite. Les utilisateurs de Pépite sont principalement des enseignants pour d'une part, organiser les activités d'apprentissage en classe et d'autre part, pour établir des bilans individualisés afin de favoriser une métaréflexion sur la compétence algébrique. Des chercheurs ont utilisé les corpus recueillis avec Pépite pour des recherches en linguistique, en didactique ou en informatique. Quant aux concepteurs, ils ont utilisé ces expériences pour améliorer la conception du logiciel et fiabiliser le diagnostic.

5.3. Experience 3 : Le Logic-ITA.

Le Logic-ITA (Merceron et Yacef, 2004) est un système d'enseignement en ligne qui permet à des étudiants de s'entraîner à la démonstration formelle en logique des propositions. Il s'appuie sur la théorie de la charge cognitive selon laquelle la pratique d'exercices permet aux étudiants de se construire des schémas de résolution de problèmes (Sweller et al., 1998).

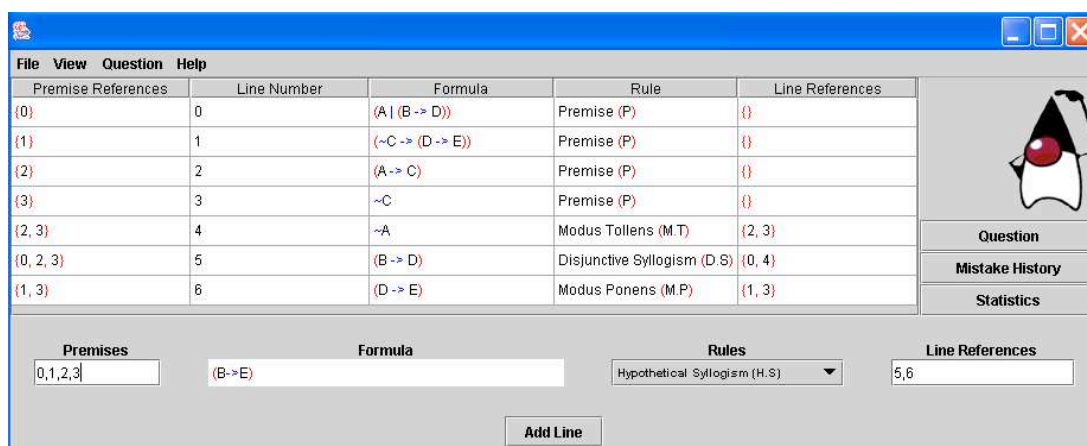


Figure 3: Copie d'un écran du Logic-ITA quand un étudiant approche de la conclusion.

Le Logic-ITA a été utilisé par environ huit cents étudiants de seconde année à l'université de Sydney où il était à la disposition des étudiants en complément d'un cours et de TD traditionnels. Le système se compose d'une base d'exercices, d'un générateur d'exercices et les étudiants peuvent également entrer leurs propres formules à prouver. Une preuve formelle consiste à dériver une formule appelée conclusion à partir d'un ensemble de formules appelées prémisses. Un exercice consiste ainsi à appliquer, pas à pas, des règles logiques pour produire des formules nouvelles jusqu'à l'obtention de la conclusion. Le système dispose d'un module expert qui évalue automatiquement chaque étape de solution proposée par l'étudiant. En particulier, il vérifie si la règle choisie est applicable sur la (ou les) formule(s) sélectionnée(s) par l'étudiant et il fournit une rétroaction avec éventuellement une indication en cas d'erreur. Les étudiants s'exercent quand ils le souhaitent, à leur rythme et ils choisissent l'ordre et le nombre d'exercices qu'ils traitent ; les sessions ne sont pas notées et n'interviennent pas dans l'évaluation.

Une base de données conserve pour chaque exercice et pour chaque étudiant la date, le temps passé, les règles correctement appliquées, les erreurs, le nombre de pas pour arriver à la conclusion et si l'étudiant a ou non terminé l'exercice. La base de données a été interrogée par des experts humains en utilisant soit des requêtes classiques de bases de données soit des techniques de fouille de données pour obtenir des informations à valeur pédagogique ayant pour but d'améliorer l'enseignement. Par exemple, une simple requête permet de connaître les règles de logique avec lesquelles les étudiants commettent le plus d'erreurs. La technique de fouille de données appelée règles d'association permet de trouver des corrélations entre les erreurs commises. Ces résultats ont conduit à changer la présentation des différentes règles de logique en cours.

6. Un ensemble structuré de Design Patterns

Dans cette section nous présentons l'ensemble de DPs que nous avons mis au point à partir de notre expérience. Le format défini dans le projet DPULS avait pour objectif de permettre la mise au point d'un outil pour faciliter la navigation et la recherche dans l'ensemble de DPs. Nous avons adopté ici un format simplifié pour faciliter la lecture. Notre ensemble de DPs est structuré en cinq hiérarchies dont la racine décrit un problème d'évaluation et les descendants décrivent des approches alternatives ou complémentaires pour résoudre ce problème. Chaque DP a un nom qui décrit à la fois le problème et la solution (Alexander et al., 1977), (Meszaros et Doble, 1998).

En combinant les noms, on obtient un langage, i.e. un concepteur peut utiliser les noms des DPs pour décrire une solution du problème d'évaluation dans son contexte (Van Duyne et al., 2002, chapitre 2). Nous ne prétendons pas que l'ensemble de DPs présenté, en l'état actuel, constitue un langage, mais le scénario suivant - qui reprend le problème de Sarah présenté dans les scénarios 1 et 1 bis - illustre comment il pourrait servir de langage de conception, i.e. comment les DPs peuvent être combinés entre eux pour donner une démarche de conception.

Scénario 2 : Aminata est un ingénieur travaillant sur l'adaptation de l'environnement numérique de travail au contexte de l'établissement où enseigne Sarah. Pour fournir à Sarah une vue générale de l'activité de son groupe de TP, Aminata considère d'abord le DP « Bilan d'un groupe d'apprenants sur un ensemble d'exercices » (DP EA4, racine de la quatrième hiérarchie). En lisant la description de la solution, elle est amenée à considérer plusieurs types de bilans (DP descendants) en fonction des objectifs des utilisateurs du bilan. Elle pense qu'elle va pouvoir utiliser des méthodes de « classification automatique » (DP EA 4. 1). Elle étudie les prérequis de ce DP : il utilise les résultats du DP « Bilan individuel sur un ensemble d'exercices » (DP EA2, racine de la deuxième hiérarchie). De même, plusieurs solutions sont envisageables pour présenter un bilan (DP descendants) : le DP « points forts et points faibles d'un apprenant » lui semble adapté à la situation de Sarah. En fait ce DP nécessite d'abord une « Analyse multidimensionnelle de la solution d'un apprenant à un seul exercice » (DP EA1, racine de la première hiérarchie). Comme précédemment ce DP propose plusieurs approches pour obtenir des informations pertinentes à partir des traces recueillies sur l'activité de l'apprenant : « Analyse par comparaison avec des schémas de solutions » (DP EA 1.2), « Analyse par utilisation de l'outil spécifique » (DP EA 1.2), « Analyse

semi-automatique supervisée par un humain » (DP EA1.3).

La Figure 4 montre les noms d'un premier ensemble de DPs concernant l'évaluation de l'apprenant³. Dans la section suivante, nous détaillons seulement certains patterns, ceux que nous avons cités dans le scénario 2. Chaque DP est illustré à l'aide des expériences Combien?, Pépité, le Logic-ITA.

DP EA1 Analyse multidimensionnelle de la solution d'un apprenant à un seul exercice
DP EA1.1 Analyse par comparaison avec des schémas de solutions
DP EA1.2 Analyse par utilisation de logiciel spécifique
DP EA1.3 Analyse semi-automatique supervisée par un humain
DP EA2 Bilan individuel sur un ensemble d'exercices
DP EA2.1 Points forts et points faibles d'un apprenant
DP EA 2.2 Position d'un apprenant sur une échelle de compétences
DP EA2.3 Progression d'un apprenant sur un ensemble d'activités individuelles
DP EA2.4 Autonomie d'un apprenant sur un ensemble d'activités individuelles
DP EA2.5 Performance d'un apprenant sur un ensemble d'activités individuelles
DP EA3 Bilan d'un groupe d'apprenants sur un seul exercice
DP EA4 Bilan d'un groupe d'apprenants sur un ensemble d'exercices
DP EA4.1 Classification automatique
DP EA4.2 Relations entre erreurs ou succès et types d'usage
DP EA4.2.1 Règles d'association
DP EA5 Musarder dans les exercices
DP EA5.1 Musarder dans les QCM

Figure 4: Les DPs sur l'évaluation de l'apprenant(en gras ceux qui sont présentés ci-dessous)

6.1. DP EA1 Analyse multidimensionnelle de la solution d'un apprenant à un seul exercice

Résumé Ce patron décrit plusieurs approches pour évaluer automatiquement la solution d'un apprenant à un problème résolu en ligne. Vous pouvez simplement valider la solution ou enrichir l'évaluation sur d'autres dimensions, par exemple : catégoriser les erreurs, évaluer la stratégie de résolution utilisée, les savoir-faire ou compétences mis en œuvre, leur degré de maîtrise etc.

C`ntexte L'apprenant a répondu à une seule question ou a résolu un seul exercice. Sa réponse peut être simple (e.g. choix multiple, réponse numérique) ou complexe (e.g. programme, raisonnement algébrique, langage naturel). Le système a enregistré la réponse ainsi que des données d'usage (temps passé, actions, hésitations, demandes d'aide, etc.). Vous voulez mettre au point une analyse locale de la réponse en temps réel (cas d'un système qui donne un retour immédiat à l'apprenant) ou différé (cas d'un système de diagnostic).

Pr`blème Comment évaluer automatiquement la solution d'un apprenant à un exercice (ou à une question d'un exercice) ou bien, si ce n'est pas possible, comment assister un évaluateur humain ? Comment votre système peut-il analyser, corriger, commenter ou classifier la réponse de l'apprenant ? Si votre système demande aux apprenants de résoudre des problèmes complexes, il permet aussi probablement aux apprenants de construire leur propre solution ; dans ce cas, il est souvent impossible d'anticiper la forme exacte d'une réponse en raison de l'explosion combinatoire des possibilités et de la diversité cognitive des apprenants.

S`luti`n Si votre objectif est d'évaluer la validité de la solution, alors vous pouvez générer un indicateur comme une note ou un code (par exemple correct, partiellement correct, incorrect ou 0, 5, 10). Si votre objectif est de fournir une rétroaction ou un diagnostic cognitif, vous avez besoin d'une caractérisation plus approfondie de la réponse de l'apprenant. Par exemple vous pouvez avoir besoin d'information sur la stratégie suivie par l'apprenant, sur les savoir-faire mis en évidence par sa réponse, sur les types d'erreurs qu'il a commises ou sur ses hésitations. Une caractérisation multidimensionnelle de la réponse dépend en général du domaine. Votre système va construire un indicateur composite qui rend compte de cette analyse multidimensionnelle de la réponse. C'est souvent une suite de codes identifiant les caractéristiques cognitives de l'apprenant ou une liste des erreurs qu'il a commises.

Résultats Si vous implémentez ce patron, vous caractériserez la réponse de l'apprenant avec certains des items suivants :

- Une note. Une note peut-être une lettre comme A, B, C ou D ou un code ou un message comme correct, partiellement correct, incorrect.
- Des caractéristiques cognitives.
- Un code ou un ensemble de codes pour identifier le processus de résolution.
- Une liste d'erreurs.

Discussi`n Vous devrez vous adjoindre un ou plusieurs experts (enseignant expérimenté, psychologue, didacticien) pour définir différents modèles (comme un modèle de compétences, un modèle des tâches ou une typologie des erreurs) qui fonderont l'analyse mise en œuvre par votre système.

Exemples

- Pépite calcule automatiquement un code pour évaluer une réponse selon plusieurs dimensions : la validité, la signification des lettres en algèbre (inconnue, variable, nombre généralisé, abréviation, unités), le calcul algébrique (usage des parenthèses, règles de calcul etc.), la traduction entre plusieurs représentations (graphique, géométrique, algébrique, langage naturel), le type de justification (par l'exemple, algébrique, par explication, par règles incorrectes) et le calcul numérique (cf. Figure 2). Par exemple, une réponse qualifiée de « partiellement correcte, usage incorrect des parenthèses et justification algébrique » est codée « T2 M31 R1 ». De façon similaire, « T3 M4 R3 » signifie « incorrect, identification incorrecte de + et x, justification par exemple numérique ».

- Combien? fournit un retour immédiat à l'apprenant en fonction de la validité de la réponse ou du type d'erreur. Pour chacune des quinze classes de problèmes, Combien? prend en compte en moyenne vingt types d'erreurs. Combien? enregistre aussi des données d'usage : action, heure, type d'action (entrée de l'étudiant, demande d'aide, demande de tirage au hasard etc.), et paramètres d'action.

- Le Logic-ITA fournit un retour immédiat à l'étudiant sur la validité de sa réponse et sur le type d'erreur. Il enregistre le temps passé à résoudre le problème, la liste des erreurs commises ainsi que la liste des règles correctement utilisées et calcule les statistiques correspondantes (nombre d'utilisations correctes pour chaque règle, nombre d'occurrences des erreurs etc.).

Patr`ns v`isins : DP EA1.1, DP EA1.2, DP EA1.3.

6.2. DP EA1.1 Analyse par comparaison avec des schémas de solutions

Résumé Dans certains cas, l'approche par comparaison avec des schémas de solutions peut vous aider à évaluer automatiquement la solution de l'apprenant.

C`ntexte Même contexte que DP EA1. Cette analyse est pertinente quand vous disposez d'une grille d'analyse de l'exercice qui donne des modèles de réponse. C'est en particulier le cas pour les questions à choix multiples ou pour les expressions arithmétiques ou algébriques.

Pr`blème Comment utiliser une approche par comparaison avec des schémas de solutions pour vous aider à évaluer automatiquement la solution de l'apprenant ?

Prérequis Vous avez besoin de créer un indicateur à partir d'une analyse pédagogique, didactique ou cognitive. Vous disposez par exemple :

- D'une grille de réponses correctes. Si la solution correcte s'exprime de façon unique, une grille d'analyse fournit les réponses correctes et incorrectes. Cette grille de réponses doit être fournie au système.
- Des schémas de réponses correctes ou de réponses courantes. Quand il existe plusieurs façons d'exprimer la solution, un schéma peut donner un modèle général de solutions.

S`luti`n Pour une classe de problèmes, une analyse didactique, cognitive ou pédagogique vous précise les critères pour caractériser un schéma de solution. Ainsi si vous instanciez un schéma de solution avec la réponse de l'apprenant, vous la caractériserez en utilisant les critères correspondants.

Résultats Voir DP EA1.

Discussi`n Pour des questions ouvertes, définir un schéma de solution est un travail délicat. Cependant, c'est possible dans certains cas : par exemple en programmation on définit des schémas pour les itérations, en algèbre on définit des règles de réécriture pour caractériser des transformations correctes ou incorrectes.

Exemples Voici un exemple simple tiré du système Pépite. Dans un exercice l'élève doit traduire algébriquement la phrase « il y a six fois plus d'élèves que de professeurs » : $E = 6 \times P$ est une réponse correcte ; le système accepte aussi $E = 6P$, $P = E / 6$ etc. Ainsi $E = 6 \times P$ est un schéma de réponse correcte. Toute réponse équivalente à l'expression algébrique $E = 6 \times P$ est correcte et évaluée par le système comme cette réponse « correct, traduction français vers algébrique, usage correct des lettres ». Par contre $P = 6 E$ ou bien $E + P = 6$ sont incorrectes et évaluées par le système comme « incorrect, traduction par abréviation, lettres utilisées comme des labels ».

Patr`ns v`isins Ce patron est une spécialisation de DP EA1.

6.3. DP EA1.2 Analyse par utilisation de logiciel spécifique

Résumé Pour vous aider à analyser la solution de l'apprenant, vous pouvez utiliser des logiciels spécifiques au domaine.

C`ntexte Même contexte que DP EA1. Dans certains domaines spécifiques comme la programmation, la logique formelle ou les mathématiques, un logiciel spécifique (par exemple un compilateur, un résolveur de problèmes, un

système expert, un logiciel de calcul formel) peut évaluer la validité de la réponse et, le cas échéant, donner des informations sur les erreurs.

Pr`blème Comment utiliser un logiciel spécifique pour analyser la solution de l'apprenant ?

S`luti`n Vous pouvez utiliser un logiciel spécifique pour vérifier la validité de la solution. Si ce logiciel donne des messages d'erreurs, vous pouvez les utiliser pour caractériser la solution.

Résultats Voir DP EA1.

Exemples Combien? et le Logic-ITA utilisent un logiciel spécifique.

- Avec Combien? l'apprenant construit une solution pour un exercice de combinatoire. Chaque étape de la solution est analysée par le système selon une « méthode de détection ciblée » (Giroire et al., 2002). Si la construction de l'apprenant ne peut pas aboutir à la solution, Combien? affiche un message adapté à l'erreur détectée pour guider l'étudiant. Combien? enregistre dans la trace la réponse de l'étudiant et le type éventuel de l'erreur détectée.

- Un exercice du Logic-ITA est une preuve formelle. L'apprenant dérive pas à pas la conclusion à partir des prémisses en utilisant des règles de logique, ce qui l'amène à dériver des formules intermédiaires. Un système expert examine chaque formule intermédiaire entrée par l'étudiant et donne un retour adéquat. S'il n'y a pas d'erreur, le système enregistre l'usage correct de la règle de logique. En cas d'erreur, une aide appropriée est fournie à l'apprenant pour l'aider à entrer une formule correcte. L'erreur et son type sont enregistrés par le système.

Patr`ns v`isins Ce patron est une spécialisation de DP EA1.

6.4. DP EA1.3 Analyse semi-automatique supervisée par un humain

Résumé Soit l'enseignant (ou un chercheur) évalue lui-même la réponse, soit il complète, vérifie ou corrige l'évaluation automatique.

C`ntexte Même contexte que DP EA1. Cette analyse est à l'heure actuelle la seule possible quand, sur des questions très ouvertes, la réponse de l'apprenant est composite ou si l'apprenant répond en utilisant ses propres termes, ou bien si le diagnostic n'a pas encore été complètement formalisé ou si le diagnostic automatique est en cours de validation ou encore pour des besoins de formation des enseignants.

Pr`blème Comment évaluer la solution de l'apprenant s'il n'y a pas de diagnostic automatique ou si ce dernier n'est pas complètement fiable ?

S`luti`n Votre système fournit une interface pour qu'un assistant humain complète, vérifie et corrige le diagnostic automatique. Cette interface fournit une liste des exercices où le diagnostic automatique a échoué ou est non fiable.

Résultats Voir DP EA1.

Discussi`n Si votre système doit traiter un grand nombre d'apprenants ou si les enseignants sont très occupés (ce qui est souvent le cas) cette solution n'est réaliste que dans la mesure où le nombre d'échecs du diagnostic est faible. Cependant elle est indiquée pour obtenir une évaluation très précise des apprenants dans un contexte de recherche ou de formation des enseignants.

Exemples Pépité évalue 80 % des réponses mais n'évalue pas entièrement les solutions dans lesquelles l'apprenant utilise le langage naturel ; Pépité fournit une interface pour que l'enseignant puisse corriger, vérifier ou compléter le diagnostic automatique.

Patr`ns v`isins Ce patron est une spécialisation de DP EA1.

6.5. DP EA2 Bilan individuel sur un ensemble d'exercices

Résumé Ce patron décrit plusieurs approches pour présenter un bilan du travail d'un apprenant sur un ensemble d'exercices. Ces approches dépendent des objectifs des destinataires de ce bilan.

C`ntexte L'apprenant a résolu en ligne des problèmes complexes et le système a recueilli et analysé ses réponses. Pour diverses raisons on vous demande de fournir un bilan du travail de cet apprenant : constituer des groupes de remédiation, faire prendre conscience à l'apprenant de ses points forts et de ses points faibles, situer un apprenant sur une échelle de compétences, évaluer l'évolution d'un savoir-faire au cours du temps etc.

Pr`blème Comment le système peut-il établir un bilan du travail de l'apprenant, de ses succès et échecs ou son profil cognitif ? Dans un système de diagnostic, le bilan est un instantané des compétences de l'apprenant. Dans un système d'apprentissage l'évolution de l'apprenant dans le temps peut être analysée. Si le destinataire du bilan doit prendre des décisions stratégiques à partir du bilan, celui-ci doit contenir une description de haut niveau de l'activité de l'apprenant. Par exemple pour concevoir des situations d'apprentissage personnalisées, les enseignants ont besoin d'une vue synthétique de l'activité d'un apprenant ou d'un bilan de son évolution. Le classement d'un apprenant sur une échelle de compétences peut être utile pour organiser des groupes de travail ou pour les apprenants eux-mêmes pour se situer par rapport aux compétences attendues. Vous voulez ainsi définir les traits essentiels qui résument les compétences d'un apprenant.

Prérequis Voir résultats DP EA1.

S`luti`n Pour résoudre ce problème vous avez d'abord recueilli les réponses d'un apprenant à un ensemble de questions ou d'exercices qui portent sur un thème donné ou sur un cours tout entier. Puis vous avez analysé chaque réponse pour chaque question ou chaque exercice. Maintenant, en fonction des objectifs des différents destinataires du bilan, vous devez mettre au point (avec des enseignants et des chercheurs) les dimensions nécessaires et calibrer l'agrégation des indicateurs locaux en des indicateurs plus généraux. Vous pouvez vous contenter de cumuler les résultats locaux. Vous pouvez aussi déterminer les points forts et les points faibles de l'apprenant, situer l'apprenant sur une échelle de compétences ou établir un bilan de l'évolution durant un cours etc.

Résultats Un bilan du travail de l'apprenant.

Discussi`n Il est essentiel de définir les dimensions du bilan qui sont très dépendantes du domaine et de tenir compte des besoins des personnes qui utiliseront ce bilan. Une autre difficulté consiste à savoir comment agréger les indicateurs locaux en indicateurs généraux.

Exemples

- Combien? et Le Logic-ITA présentent le temps total passé, le nombre d'exercices essayés, réussis, ratés, les erreurs commises, les règles correctement utilisées etc.
- Pépite construit un profil cognitif de l'apprenant en algèbre en le situant sur une échelle de compétences et en précisant, sur chaque dimension, ses points forts et ses points faibles (voir résultats de DP EA2.1).

Patr`ns v`isins DP EA1, DP EA2.

6.6. DP EA2.1 Points forts et points faibles d'un apprenant

Résumé Une façon fréquente de présenter un bilan du travail de l'apprenant consiste à mettre en lumière ses points forts et ses points faibles manifestés sur un ensemble d'exercices.

C`ntexte Même contexte que DP EA2. L'apprenant a résolu plusieurs exercices mettant en jeu différentes compétences.

Pr`blème Comment définir les points forts et les points faibles de l'apprenant ?

Prérequis Voir résultats DP EA1.

S`luti`n Après avoir analysé chaque réponse pour chaque exercice, vous voulez mener une analyse sur une session entière ou sur une certaine période. D'abord vous calculez le taux de succès pour chaque compétence en jeu et le taux d'apparition des erreurs. Parfois il vous faut aussi tenir compte du contexte où les erreurs ont été commises ou bien où les compétences ont été mises en évidence. Dans ce cas, vous aurez besoin d'établir des catégories d'exercices et d'erreurs.

Résultats

- Listes de compétences et taux de succès sur ces différentes compétences.
- Listes d'erreurs, taux d'occurrences et/ou contexte d'occurrence de ces erreurs.

Discussi`n Il est essentiel d'avoir aussi les points forts et pas seulement les faiblesses de l'apprenant. Si vous établissez un rapport, mettre en lumière ses points forts encourage l'apprenant et, secondairement, aussi l'enseignant. Si vous voulez aider l'apprenant dans ses progrès, les stratégies d'enseignement peuvent différer selon ses points forts sur lesquels vous pourrez vous appuyer pour faire évoluer ses conceptions.

Exemples Pépite offre un bilan en trois dimensions (usage de l'algèbre, traduction d'une représentation à une autre, calcul algébrique) et, pour chaque dimension, indique d'une part les taux de réussite sur les types d'exercices concernant ces dimensions et, d'autre part, les points forts (liste de compétences acquises ou en cours d'acquisition avec taux de succès et contexte dans lesquelles elles ont été mises en évidence) et les points faibles (liste d'erreurs typées avec fréquence et contexte d'occurrence).

Patr`ns v`isins Ce patron est une spécialisation de DP EA2.

6.7. DP EA4 Bilan d'un groupe d'apprenants sur un ensemble d'exercices

Résumé Ce patron permet de concevoir un bilan sur le travail d'un groupe d'apprenants sur un ensemble d'exercices.

C`ntexte Un ensemble d'apprenants a résolu un ensemble d'exercices et votre système a analysé toutes leurs réponses.

Pr`blème Comment le système peut-il construire un bilan du travail d'un groupe d'apprenants sur un ensemble d'exercices ? Si le destinataire du bilan doit prendre des décisions stratégiques à partir du bilan, celui-ci doit contenir une vue synthétique du travail des apprenants pour une activité complète. Par exemple cette vue peut aider à organiser des sessions adaptées au niveau d'une classe ou à organiser des groupes de travail dans une classe. Elle

peut aussi aider les enseignants à réfléchir à la qualité et à la pertinence des exercices et des supports de cours dans le cas où votre système repèrerait que certaines erreurs sont très fréquentes ou qu'elles se produisent souvent ensemble. Ainsi elle fournit aux concepteurs et aux enseignants des informations susceptibles de les aider à évaluer ou à améliorer leur conception ou leur enseignement (voir aussi DP MV1 ([DPULS - 6, 2005](#))).

Prérequis Voir résultats DP EA2.

S`luti`n Elle est fonction de vos objectifs. Pour situer les apprenants sur une échelle de compétences ou en fonction de la réalisation d'objectifs pédagogiques, vous pouvez proposer une cartographie de la classe. Une telle carte classe les apprenants en différents groupes. Pour cela, vous pouvez par exemple utiliser des stéréotypes prédéfinis. Un stéréotype est un moyen d'identifier des groupes d'apprenants ayant les mêmes caractéristiques. Ces caractéristiques peuvent être aussi simples qu'une note. Dans ce cas, vous obtiendrez des groupes en fonction de leur réussite, par exemple : le groupe des bons, le groupe des moyens et le groupe des faibles. Un stéréotype peut être plus précis et décrire les apprenants ayant acquis certaines compétences mais pas d'autres ou bien ayant acquis certaines compétences mais présentant encore telles et telles faiblesses. Puis en fonction du bilan de compétence personnel de l'apprenant, vous lui attribuerez un stéréotype.

Pour définir ces stéréotypes, une première solution consiste à faire appel à des experts. Vous pouvez aussi laisser le système grouper lui-même les apprenants par exemple en utilisant des algorithmes de classification de fouilles de données.

Si votre objectif est d'obtenir un bilan des résultats, vous pouvez produire des statistiques et diagrammes qui regroupent les exercices par chapitre ou sujet.

Si votre objectif est de détecter des associations telles que « si les apprenants font l'erreur A, alors ils font aussi l'erreur B » ou « si les apprenants ratent l'exercice E, alors ils ratent aussi l'exercice F » vous pouvez utiliser des algorithmes de règles d'association de fouilles de données.

Résultats Si vous implémentez ce patron, vous caractériserez le bilan de votre groupe d'apprenants avec certains des items suivants.

- Un cahier de notes et de statistiques sur les performances des apprenants.
- Une carte de la classe groupant les apprenants par compétences, par type d'erreur, par stéréotypes.
- Une catégorisation des erreurs ou des habiletés co-occurentes.

Discussi`n Les stéréotypes peuvent être simples (élèves bons, moyens, faibles), multidimensionnels (plaçant les apprenants sur une grille de compétences) ou peuvent décrire des usages (joueur, apprenant systématique, papillon etc.).

Exemples

- Pépite utilise des stéréotypes définis à partir d'une analyse didactique pour classer les apprenants par groupes d'habiletés et pour présenter une carte cognitive de la classe en vue d'adapter l'enseignement à chacun des groupes.
- Le Logic-ITA utilise une classification automatique pour regrouper les apprenants en fonction des erreurs qu'ils ont commises afin d'adapter l'enseignement pour chacun de ces groupes.

Patr`ns v`isins Ce patron est spécialisé par les patrons DP EA4.1 et DP EA4.2. Si votre objectif est d'améliorer le cours, consultez aussi DP MV1 ([DPULS - 6, 2005](#)).

6.8. DP EA4.1 Classification automatique

Résumé Les méthodes de classification automatique permettent de dresser un bilan de l'activité d'un groupe d'apprenants sur un ensemble d'exercices.

C`ntexte Même contexte que DP EA4. Si vous caractérisez chaque apprenant par un ensemble d'attributs comme le nombre d'exercices réussis ou ratés, le nombre d'erreurs commises, les compétences manifestées ou non, vous pouvez utiliser la classification automatique pour partitionner le groupe d'apprenants en sous-groupes homogènes.

Pr`blème Comment construire des groupes d'apprenants en utilisant la classification automatique ? Est-il mieux de faire des groupes homogènes ou hétérogènes ? Ceci est une question pédagogique. Dans tous les cas, il vous faut connaître les groupes homogènes, c'est-à-dire des groupes composés d'apprenants avec des attributs similaires. Cependant vous n'avez pas de stéréotypes prédéfinis ou vous voulez essayer d'autres formes de groupements. Vous pouvez avoir recours à la classification automatique.

Prérequis Comme pour DP EA4.

S`luti`n Votre système a analysé chaque réponse de chaque apprenant. Vous agrégez ces analyses pour obtenir une caractérisation de chacun. Par exemple, chaque apprenant peut être caractérisé par les exercices qu'il a réussis et ratés. Une autre caractérisation simple est le nombre total d'erreurs commises sur tous les exercices. Parmi ces caractéristiques (ou attributs), vous sélectionnez celles que vous voulez utiliser pour la classification automatique. L'algorithme de classification automatique produit des classes d'apprenants. Les apprenants dans une même classe

sont similaires par rapport aux caractéristiques retenues.

Résultats Groupes homogènes d'apprenants.

Discussion Pour utiliser la classification automatique de façon sensée, vous devez avoir un minimum d'expertise dans le domaine de la fouille de données. Construire et sélectionner les caractéristiques ou attributs pertinents est crucial pour le résultat et n'est pas nécessairement évident.

Exemples La classification automatique a été utilisée avec le Logic-ITA pour voir si les apprenants en situation d'échec (ceux qui commencent des exercices sans arriver à les finir correctement) peuvent être regroupés de façon homogène. En utilisant comme attribut le nombre d'erreurs, la classification a produit deux groupes, les étudiants faisant beaucoup d'erreurs et ceux en faisant peu. Au sein du premier groupe les chercheurs ont pu identifier les apprenants qui utilisaient une stratégie « par devinette » qui consiste à systématiquement utiliser toutes les règles de logique du système jusqu'à rencontrer la bonne pour dériver la prochaine formule.

Patrons voisins Ce patron est une spécialisation de DP EA4.

7. Discussion

Comment évaluer un ensemble de patrons ? C'est un sujet de discussion parmi les promoteurs de l'approche DP ([Avgeriou et al., 2004](#)), ([Buschmann et al., 1996](#)), ([Chung et al., 2004](#)), ([Salingaros, 2000](#)), E-LEN project ([E-LEN, 2003](#)), ([Todd et al., 2004](#)). La plupart des patrons décrits dans la littérature ont été validés par une révision entre pairs. Certains auteurs ([Avgeriou et al., 2003](#)), ([Devedzic et Harrer, 2005](#)) suggèrent qu'un patron peut être accepté par la communauté s'il vérifie « la règle de trois ». Dans notre travail, nous avons vérifié la validité de nos patrons en trois étapes. Dans une première étape, les patrons ont été discutés au sein de l'équipe AIDA. Dans une seconde étape, les patrons ont été discutés et raffinés au sein du projet DPULS. Dans une troisième étape, nous avons pris en compte les contributions à l'atelier AIED'05 Usage Analysis ([Choquet et al., 2005](#)) ayant trait à l'évaluation et avons examiné dans quelle mesure les expériences décrites étaient en accord avec nos patrons. Remarquons que sur les 6 contributions étudiées dans cet atelier, une seule faisait partie du projet DPULS.

Les patrons proposés dans notre article généralisent des expériences fructueuses réalisées au sein de l'équipe AIDA, à savoir les trois expériences mentionnées dans la section 2 et également trois autres: Diane, Math Logs et Java-En-Ligne.

Diane est un système de diagnostic. Il identifie les stratégies adéquates ou erronées ainsi que les mécanismes cognitifs impliqués dans la résolution de problèmes d'arithmétique chez les élèves de l'école primaire (8-10 ans) ([Hakem et al., 2005](#)). DP EA1 et DP EA2 décrivent l'approche de Diane pour le diagnostic cognitif.

Math Logs ([Vandebrouck et Cazes, 2005](#)) apporte aux chercheurs des informations concernant l'utilisation de la plateforme Wims pour des TD de Mathématiques par des étudiants. Les expressions mathématiques entrées par les étudiants sont vérifiées par un logiciel de calcul formel tel que MUPAD, PARL ou Maxima, approche proposée par DP EA1.2. Math Logs affiche à l'enseignant les notes moyennes, le temps moyen passé par type d'exercice et des indicateurs sur l'évolution des notes dans le temps. Il détecte des stratégies d'usage suivies par certains étudiants telles que se concentrer sur des exercices faciles ou préférer les exercices avancés (DP EA2, DP EA3 et DP EA4). Il permet une classification automatique des exercices en fonction de deux indicateurs, à savoir la difficulté et le rendement (DP MV1, amélioration des supports, ([DPULS - 6, 2005](#))).

Java-En-Ligne est un cours en ligne d'introduction à la programmation avec le langage Java. Les étudiants écrivent des (morceaux de) programmes et les soumettent pour compilation et exécution. Les erreurs signalées par le compilateur (DP EA1.2) sont enregistrées par le système. Pour chaque exercice et chaque étudiant, le système affiche un bilan comprenant le nombre de fois où l'exercice a été consulté, essayé, réussi, le nombre d'erreurs commises, le temps passé. Il fournit également un bilan par chapitre et pour le cours entier (DP EA2).

Dans l'atelier AIED'05 "Usage Analysis", six contributions concernaient l'évaluation des apprenants.

([Kumar, 2005](#)) présente un Tuteur pour la programmation en C++. Chaque exercice est défini à partir d'un schéma d'exercice qui concerne exactement un objectif d'apprentissage. Le système vérifie automatiquement si la réponse de l'étudiant est correcte, partiellement correcte, incorrecte, absente ou si l'exercice n'a pas été abordé (DP EA1). Ensuite un bilan individuel indique pour chaque objectif d'apprentissage le taux de réussite, de réussite partielle etc. (DP EA2). La moyenne de la classe pour chaque exercice (DP EA3) et la performance moyenne de la classe pour chaque objectif d'apprentissage (DP EA4) sont calculées. Les concepteurs utilisent ces résultats pour affiner les schémas et générer automatiquement des exercices en accord avec les objectifs (DP MV1, amélioration des supports ([DPULS - 6, 2005](#))).

([Feng et Heffernan, 2005](#)) présentent Assistment, un tuteur en ligne pour les mathématiques qui fournit à l'enseignant un bilan sur l'activité des apprenants et aux apprenants une assistance quand ils sont bloqués sur une tâche. Cette assistance est donnée sous forme de questions non seulement pour aider l'étudiant mais aussi pour aider le système à discerner les éléments de connaissance impliqués dans le blocage quand la tâche mets en jeu plusieurs éléments (DP EA1). Ensuite le système produit un cahier de notes qui présente à l'enseignant tous les résultats des apprenants (DP EA2) ainsi qu'un bilan de la classe (DP EA4). Une analyse des points abordés aide à déterminer les difficultés des apprenants dans le but d'améliorer l'enseignement et les supports de cours (DP MV1,

amélioration des supports (DPULS - 6, 2005).

(Nicaud et al., 2005) modèlent le comportement des apprenants en calcul algébrique dans Aplusix, un système pour apprendre l'algèbre élémentaire. Le diagnostic est un processus en deux phases. La première phase est une analyse locale des transformations d'une expression algébrique proposées par un élève. En utilisant une bibliothèque de règles correctes et incorrectes, le système détermine la séquence de règles utilisées par l'apprenant lors d'une transformation. Le système détermine aussi le type d'exercice et le contexte algébrique dans un *Local Behaviour Vector* (LBV) (DP EA1). La deuxième phase utilise un réseau de concepts pour construire un bilan des compétences acquises par l'apprenant à partir de différents LBV résumant un ensemble de transformations algébriques (DP EA2). Aplusix offre à l'enseignant une carte des compétences de la classe entière (DP EA4).

(Heraud et al., 2005), (Stefanov et Stefanova, 2005), ainsi que (Mühlenbrock, 2005) ne sont pas centrés sur l'évaluation de l'apprenant. Ils décrivent des approches pour construire un bilan de l'apprenant selon plusieurs perspectives. Ils considèrent des traces d'apprenants collectées dans des plates-formes d'enseignement en ligne. Les traces rendent compte, par exemple, des réussites, des tâches achevées, du temps passé, des consultations d'aide (DP EA1). Dans Virtuoso, Stefanov et Stefanova utilisent une approche « objet pédagogique » pour fournir aux apprenants, enseignants et concepteurs des statistiques sur les points forts et points faibles des étudiants (DP EA2.1), sur les taux de réussite et d'échec pour chaque objet pédagogique (DP EA3) et pour améliorer le cours (hiérarchie DP MV (DPULS - 6, 2005)). (Mühlenbrock, 2005) utilise des arbres de décision pour classer les apprenants en trois catégories : faible, moyen et fort (DP EA4). Pour obtenir le bilan de l'activité d'un apprenant sur une session, (Heraud et al., 2005) comparent la trace de l'apprenant avec le scénario d'apprentissage prescrit. Leur système affiche une comparaison graphique du temps passé pour chaque tâche avec le temps prescrit. Une « barre d'ombre » indique une différence entre le temps enregistré dans la trace et le temps prescrit. Alerté par cette barre d'ombre, l'évaluateur humain, appelé le « compositeur de trace », peut alors consulter d'autres sources d'informations comme les logs du serveur ou de la station de travail de l'apprenant ou encore un observateur humain. Cette expérience suggère d'affiner notre hiérarchie en prenant en compte des données d'usage alors que nous sommes plutôt concentrés sur des données cognitives précises jusqu'à maintenant.

Dans cette section nous avons étudié dans quelle mesure l'ensemble de DPs pour l'évaluation de l'apprenant que nous avons identifié décrit des invariants dans des systèmes existants. Notre objectif était de capitaliser des expériences afin d'aider des concepteurs à construire des systèmes de qualité ; d'autres expériences de conception d'EIAH prenant en compte l'évaluation des apprenants pourront amener à développer ce premier ensemble de DPs.

La revue de systèmes que nous venons de présenter montre que notre premier objectif est atteint au moins en ce qui concerne les domaines des mathématiques et de la programmation. Une étude plus approfondie serait nécessaire pour étudier la pertinence de notre approche dans d'autres domaines. Toutefois les tests standards de compétence en langue étrangère (e.g. TOEFL, TOEIC, IELTS, TFI) (IELTS, 2006) utilisent une approche similaire. En s'appuyant sur les résultats de l'apprenant, ils le classifient sur une échelle prédéfinie de compétences sur plusieurs dimensions : écouter, lire, écrire, parler.

Pour l'instant il est prématuré de dire si notre objectif d'assister les concepteurs est atteint. Notre ensemble de DPs est accessible sur Internet (DPs-DPULS, 2006) et sera utilisé en particulier dans le cadre de l'école doctorale virtuelle du réseau Kaléidoscope ce qui devrait nous permettre d'avoir des retours intéressants et sans doute d'enrichir ce premier ensemble de DPs concernant l'évaluation des apprenants. Plus généralement, l'aide à la conception d'EIAH est un problème très actuel. L'approche IMS-LD (Koper et al., 2003) attire beaucoup de chercheurs et commence à donner lieu à des réalisations intéressantes. Dans un article récent (McAndrew et al., 2006) proposent une intéressante discussion sur le positionnement réciproque des approches Design Pattern et IMS-LD et illustrent leur propos par le développement d'une activité d'apprentissage collaboratif en ligne. Afin d'aider les concepteurs, en particulier des enseignants non spécialistes, ils proposent d'intégrer les deux approches : l'approche DP permettant plus facilement un partage des conceptions par les humains et une grande flexibilité dans l'adaptation à des contextes particuliers et l'approche LD fournissant une description plus formelle pour des traitements automatiques par des machines. Dans un article prospectif (Brouns et al., 2005) envisagent deux idées d'utilisation complémentaires de ces deux approches. Tout d'abord, ils suggèrent d'utiliser des techniques d'induction automatique pour détecter des patterns d'apprentissage à partir de traces d'usage effectif de cours conçus en utilisant les spécifications IMS-LD ; ensuite ils envisagent la perspective d'utiliser les patterns d'apprentissage pour des ingénieries pédagogiques ultérieures.

8. Conclusion

Dans cet article, nous avons proposé un ensemble de dix-sept DPs concernant l'évaluation en ligne des apprenants. Cet ensemble représente une partie des DPs élaborés dans le cadre du projet européen (DPULS - 6, 2005). Un des résultats de notre travail est la mise en évidence de principes de base pour la conception de systèmes qui ont pour objectifs de mettre à disposition un « bilan sur l'activité d'un groupe d'apprenants sur un ensemble d'exercices », DP EA4. Une étape essentielle est de mener une « Analyse multidimensionnelle de la solution d'un apprenant à un seul exercice » (DP EA1). Ce DP est spécialisé en plusieurs DPs qui décrivent chacun une approche pour mener cette analyse. Ces approches peuvent être adoptées soit pour obtenir un « Bilan individuel sur un ensemble d'exercices » (DP EA2) soit pour obtenir « Bilan d'un groupe d'apprenants sur un seul exercice » (DP EA3).

Ces DPs sont assez généraux. Une perspective que nous envisageons consiste à définir des DPs plus spécifiques

pour décrire comment mettre en œuvre les différentes approches. Par exemple nous pourrions préciser comment établir des bilans sur un groupe d'apprenants (DP EA4 et DP EA3) à l'aide des techniques de fouilles de données. Il est nécessaire d'étudier plus précisément ce qui, dans l'évaluation est indépendant du domaine d'activité pour identifier des fonctionnalités qui pourraient être proposées dans les plates-formes de formation : par exemple un cahier de notes augmenté comme présenté dans Assessment ([Feng et Heffernan, 2005](#)) ou un drapeau de retro-action (un drapeau vert pour une bonne réponse et rouge pour une mauvaise) comme le suggère ([VanLehn et al., 2005](#)). Plus généralement, des DP sur les interfaces de présentation des bilans ou de travail sur les bilans ou sur l'évolution du travail de groupe d'apprenants ([Guéraud et al., 2004](#)) enrichiraient l'ensemble que nous proposons ici. Nous estimons aussi que des patterns spécifiques à des domaines de formation seraient une aide efficace pour les concepteurs par exemple sur l'analyse de programmes ou d'expressions algébriques.

Ce travail nous a montré que l'approche DP est très fructueuse pour amener des équipes hétérogènes à partager des expériences et à les capitaliser. Mettre en évidence des DP est une technique d'intégration efficace en ce sens qu'elle oblige à distiller les expériences sur des projets spécifiques pour ne garder que les invariants des conceptions de qualité. « Les patterns sont vivants et évoluent » ([Alexander et al., 1977](#)). Dans cet esprit nous attendons des retours d'usages de nos DP pour développer et enrichir cette première proposition de DP sur l'évaluation des apprenants en ligne.

Pour conclure, bien que les DP présentés ici ne soient pas des DP de génie logiciel, nous faisons nôtre la conclusion du GoF (Gamma et al., 1995, p. 409) : « on peut prétendre que ce livre n'a pas apporté grand-chose. Après tout, il ne propose aucun algorithme ou technique de programmation qui n'ait été déjà utilisé (...) Il ne fait que documenter des conceptions existantes. (...) Nous espérons que ce sera le début d'un mouvement pour la documentation du savoir-faire des praticiens. »

BIBLIOGRAPHIE

- ALEXANDER C., ISHIKAWA S., SILVERSTEIN M., JACOBSON M., FIKSDAHL-KING I., ANGEL S. (1977). *A pattern language: towns, buildings, construction*. New York: Oxford University Press.
- AVGERIOU P., PAPASALOUROS A., RETALIS S., SKORDALAKIS M. (2003). Towards a Pattern Language for Learning Management Systems, *Educational Technology & Society*, 6(2), p.11-24.
- AVGERIOU P., VOGIATZIS D., TZANAVARI A., RETALIS S., (2004). Towards a pattern language for adaptive web-based educational systems, *Advanced Technology for Learning*, 1(4), ACTA Press.
- BORCHERS J., (2001). *A Pattern Approach to Interaction Design*. Wiley, Chichester, England.
- BROUNS F., KOPER R., MANDERVELD J., VAN BRUGGEN J., SLOEP P., VAN ROSMALEN P, TATTERSALL C., VOGTEN H. (2005). A first exploration of an inductive analysis approach for detecting learning design patterns. *Journal of Interactive Media in Education* ([Advances in Learning Design](#), Special Issue, eds. Colin Tattersall, Rob Koper), 2005/03. ISSN:1365-893X
- BUSCHMANN F., MEUNIER R., ROHNERT H., SOMMERLAD P., STAL M. (1996). *Pattern-Oriented Software Architecture, Volume 1: A System of Patterns*. John Wiley & Sons.
- CHOQUET C., LUENGO V., YACEF K., Eds. (2005). Proceedings of « *Usage Analysis in Learning Systems* » workshop, held in conjunction with the *12th Conference on Artificial Intelligence in Education*, Amsterdam, The Netherlands, Disponible sur internet à : <http://hcs.science.uva.nl/AIED2005/W1proc.pdf> (Consulté en mars 2006)
- CHOQUET C. (2004). JEIRP Design Patterns for Recording and Analysing Usage of Learning Systems proposal. Disponible sur internet à <http://www.noe-kaleidoscope.org/pub/activities/jeirp/activity.php?wp=33> consulté en mars 2006.
- CHUNG E.S., HONG J.I., LIN J., PRABAKER M.K., LANDAY J.A., LIU A. (2004). Development and Evaluation of Emerging Design Patterns for Ubiquitous Computing, *In Proceedings of Designing Interactive Systems (DIS2004)*, Boston, MA. p. 233-242.
- COMBIEN? Le site du projet Combien?, <http://combien.lip6.fr>
- DELOZANNE E., PREVIT D., GRUGEON B., JACOBONI P. (2003). Supporting teachers when diagnosing their students in algebra, *Workshop Advanced Technologies for Mathematics Education, supplementary Proceedings of Artificial Intelligence in Education*, Sydney, July 2003, IOS Press, Amsterdam, p.461-470.
- DELOZANNE E., VINCENT C., GRUGEON B., GELIS J.-M., ROGALSKI J., COULANGE, L. (2005). From errors to stereotypes: Different levels of cognitive models in school algebra, *Proceedings of E-LEARN05*, Vancouver, 24-28 October 2005.
- DENG J., KEMP E., TODD E.G. (2005). *Managing UI Pattern Collections, CHINZ'05*, Auckland, NZ.
- DEVEDZIC V. (2001). A Pattern Language for Architectures of Intelligent Tutors. IEEE Learning Technology Newsletter 3/3, consultable sur: http://ltf.ieee.org/learn_tech/issues/july2001/index.html#8.
- DEVEDZIC V., HARRER A., Software Patterns in ITS Architectures, in *JAIED*, 15, (ISSN 1560-4292), p. 63-94.
- DPULS- Design Patterns Browser : <http://lucke.univ-lemans.fr:8080/dpuls/login.faces> (Consulté en mars 2007).

DPULS deliverables (2005). Disponible sur internet à <http://www.noe-kaleidoscope.org/intra/activity/deliverables/> (Consulté en janvier 2006)

deliverable 2: MERCERON, A., Report on partners' experiences

deliverable 3: DAVID J.P., State of art of tracking and analyzing usage

deliverable 4: POZZI F. The set of recurrent problems and description of solutions

deliverable 5: VERDEJO M.F., CELORRIO C., The design pattern language structure

deliverable 6: DELOZANNE E., LABAT J.-M., LE CALVEZ F., MERCERON, A., The structured set of design patterns

deliverable 7: IKSAL S., ALEXIEVA A., BEALE R., BYRNE W., DUPONT F., LONSDALE P., MILEN P., The design pattern browser

DUVAL P., MERCERON A., SCHOLL M., WARGON L. (2005). Empowering Learning Objects: an experiment with the Ganesha platform. In Proceedings of the *World Conference on Educational Multimedia, Hypermedia and Telecommunications ED-MEDIA 2005*, Montreal, Canada, P. Kommers and G. Richards Ed., p. 4592-4599, AACE Digital Library (<http://www.aace.org>)

E-LEN Le site des patterns du projet E-LEN : http://www2.tisip.no/E-LEN/patterns_info.php (Consulté en décembre 2005)

FENG M., HEFFERNAN N. (2005). Informing Teachers Live about Student Learning: Reporting in Assistent System.25-32, Proceedings of « *Usage Analysis in Learning Systems* » workshop, held in conjunction with the *12th Conference on Artificial Intelligence in Education*, Amsterdam, The Netherlands, p.25-32. Disponible sur internet à <http://hcs.science.uva.nl/AIED2005/W1proc.pdf> (Consulté en décembre 2005)

FOWLER, M. (1997). *Analysis Patterns: Reusable Object Models*. Reading, MA: Addison-Wesley.

GAMMA E., HELM R., JOHNSON R., VLISSIDES J. (1995) *Design Patterns: Elements of Reusable Object-Oriented Software*, ADDISON WESLEY

GIROIRE H., LE CALVEZ F., TISSEAU G., DUMA J., URTASUN M. (2002). "Un mécanisme de détection incrémentale d'erreurs et son application à un logiciel pédagogique", *Actes de RFA'2002*, Angers, p. 1063-1072.

GOODYEAR P., AVGERIOU P., BAGGETUN R., BARTOLUZZI S., RETALIS S., RONTELTAP F., RUSMAN E. (2004). Towards a Pattern Language for Networked Learning, Proceedings of *Networked Learning 2004*, Lancaster UK.

GRUGEON B. (1995) Étude des rapports institutionnels et des rapports personnels des élèves à l'algèbre élémentaire dans la transition entre deux cycles d'enseignement : BEP et Première G, Thèse de doctorat, Université Paris VII.

GUÉRAUD V., ADAM J.-M., PERNIN J.-Ph., CALVARY G., DAVID J.-P. (2004) L'exploitation d'Objets Pédagogiques Interactifs à distance : le projet FORMID, *Revue STICEF*, Volume 11, mis en ligne le 25/05/2004, <http://sticef.org>

HAKEM K., SANDER E., LABAT J.-M. (2005). DIANE, a diagnosis system for arithmetical problem solving. Proceedings of *Artificial Intelligence in Education 2005*, Looi, McCalla, Bredeweg, Breuker eds, IOS Press, Amsterdam, Holland, p. 258-265.

HERAUD J.M., MARTY J.C., FRANCE L., CARRON T. (2005). Helping the Interpretation of Web Logs: Application to Learning Scenario Improvement, Proceedings of « *Usage Analysis in Learning Systems* » workshop, held in conjunction with the *12th Conference on Artificial Intelligence in Education*, Amsterdam, The Netherlands, p. 41-48, Disponible sur internet à : <http://hcs.science.uva.nl/AIED2005/W1proc.pdf> (Consulté en décembre 2005)

HERNANDEZ-LEO D., ASENSIO-PEREZ J. I., DIMITRIADIS Y. (2005). Computational Representation of Collaborative Learning, *Flow Patterns using IMS Learning Design*. *Educational Technology & Society*, 8 (4), p.75-89.

IELTS International Language Testing System (British Council ielts), <http://www.britishcouncil.org/learning-ielts.htm> (Consulté en mars 2006)

JEAN S., (2000) PÉPITE : un système d'assistance au diagnostic de compétences, Thèse de doctorat de l'Université du Maine, 21 janvier 2000

KALEIDOSCOPE Le site du réseau Kaleidoscope, <http://www.noe-kaleidoscope.org/> (Consulté en mars 2006)

KOPER R., OLIVIER B., ANDERSON T. eds. (2003). *IMS Learning Design. Information Model, Best Practice and Implementation Guide*, Binding document, Schemas. Consulté en octobre 2003 à <http://www.imsglobal.org/learningdesign/index.cfm>

KUMAR A. (2005). Usage Analysis in Tutors for C++ Programming, Proceedings of « *Usage Analysis in Learning Systems* » workshop, held in conjunction with the *12th Conference on Artificial Intelligence in Education*, Amsterdam, The Netherlands, p.57-64. Consulté en décembre 2005 à : <http://hcs.science.uva.nl/AIED2005/W1proc.pdf>

LANDAY J., VAN DUYNE D.K. (2004). Design Patterns for Customer-Centred Web Design, tutorial in the CHI 2004 international conference, Vienna, Austria.

LARMAN C. (2003). *UML et les Design Patterns* Campus Press. Paris.

LE CALVEZ F., GIROIRE H., DUMA J., TISSEAU G., URTASUN M. (2003). Combien? a Software to Teach Learners How to Solve Combinatorics Exercises. *Workshop Advanced Technologies for Mathematics Education*, supplementary Proceedings of *Artificial Intelligence in Education*, Sydney, Australia, IOS Press, Amsterdam, p. 447-453.

- McANDREW P., GOODYEAR P., DALZIEL J., (2006). Patterns, designs and activities: unifying descriptions of learning structures, *International Journal of Learning Technology 2006 - Vol. 2*, No.2/3 p. 216 - 242
- MERCERON A., YACEF K. (2004). Mining Student Data Captured from a Web-based Tutoring Tool: Initial Exploration and Results, in *Journal of Interactive Learning Research Special Issue on Computational Intelligence in Web-Based Education*, 15(4), p.319-346.
- MESZAROS G., DOBLE J. (1998). A Pattern Language for Pattern Writing, in *Pattern Languages of Program Design 3 (Software Patterns Series)*, Addison-Wesley, Consulté en décembre 2005 à <http://hillside.net/patterns/writing/patterns.htm>
- MÜHLENBROCK M. (2005). Automatic Action Analysis in an Interactive Learning Environment, Proceedings of « *Usage Analysis in Learning Systems* » workshop, held in conjunction with the *12th Conference on Artificial Intelligence in Education*, Amsterdam, The Netherlands, p.73-80. Consulté en décembre 2005 à : <http://hcs.science.uva.nl/AIED2005/W1proc.pdf>.
- NICAUD J.-F., CHAACHOUA H., BITTAR M., BOUHINEAU D. (2005). Student's modelling with a lattice of conceptions in the domain of linear equations and inequations Proceedings of « *Usage Analysis in Learning Systems* » workshop, held in conjunction with the *12th Conference on Artificial Intelligence in Education*, Amsterdam, The Netherlands, p.81-88. Consulté en décembre 2005 à : <http://hcs.science.uva.nl/AIED2005/W1proc.pdf>.
- PÉPITE Le site du projet Pépite, <http://pepite.univ-lemans.fr>
- PPP The Pedagogical Patterns Project site, <http://www.pedagogicalpatterns.org/> (Consulté en décembre 2005)
- SALINGAROS N. (2000). The Structure of Pattern Languages. *Architectural Research Quarterly*, 4, p.149-161.
- SCHMIDT D., STAL M., ROHNERT H., BUSCHMANN F. (2000). Pattern-Oriented Software Architecture, vol.2, *Patterns for Concurrent and Networked Objects*. Wiley.
- STEFANOV K., STEFANOVA E. (2005). Analysis of the Usage of the Virtuoso System, Proceedings of « *Usage Analysis in Learning Systems* » workshop, held in conjunction with the *12th Conference on Artificial Intelligence in Education*, Amsterdam, The Netherlands, p. 97-104. Consulté en décembre 2005 à : <http://hcs.science.uva.nl/AIED2005/W1proc.pdf>
- SWELLER J., VAN MERRIENBOER J.G., PAAS, F.G.W.C. (1998). Cognitive architecture and Instructional Design. *Educational Psychology Review*, 10(3), p. 251-295.
- TODD E., KEMP E., PHILIPS C. (2004). What makes a good user interface pattern language? The *5th Australasian User Interface Conference*, Dunedin, Australian Computer Society, p. 91-100.
- VANDEBROUCK F., CAZES C. (2005) Analyse de fichiers de traces d'étudiants : aspects didactiques, *Revue STICEF*, Volume 12, ISSN : 1764-7223, mis en ligne le 18/01/2006, <http://sticef.org>
- VAN DUYNE D. K., LANDAY J., HONG J. I. (2002) The Design of Sites: Patterns, Principles, and Processes for Crafting a Customer-Centered Web Experience. ADDISON WESLEY
- VANLEHN K., LYNCH C., SCHULZE K., SHAPIRO J.A., SHELBY R., TAYLOR L., TREACY D., WEINSTEIN A., WINTERSGILL M. (2005). The Andes Physics Tutoring System: Lessons Learned. *International Journal of Artificial Intelligence and Education*, 15 (3) p.147-204.
- VAN WELIE M. (2004). *Patterns in Interaction Design*. Extrait en 2006 de <http://www.welie.com/index.html>.
- Wims Le site de la plateforme Wims <http://wims.unice.fr/wims>

¹ Cet article est une version française profondément remaniée de l'article É. Delozanne, F. Le Calvez, A. Merceron, J.-M. Labat, (2007), A Structured set of Design Patterns for Learners' Assessment, *Journal of Interactive Learning Research (JILR)*, (2007) 18(2), 309-333.

² AIDA (Approche Interdisciplinaires pour des Dispositifs informatisés d'Apprentissage) est un PPF (Programme PluriFormation) financé par le Ministère français de la Recherche et regroupant sept équipes de recherche de différents laboratoires de la région parisienne.

³ La version complète du premier ensemble de DP mis au point dans le cadre du projet DPULS est présenté dans [\(DPULS-6, 2005\)](#).

■ A propos des auteurs

Elisabeth DELOZANNE est docteur en informatique de l'université du Maine depuis 1992. Chercheuse en EIAH, elle travaille sur des projets pluridisciplinaires qui visent à produire des logiciels pour assister les enseignants dans la prise en compte de la diversité cognitive de leurs élèves, spécialement en mathématiques. Membre du Conseil National des Universités, elle est actuellement Maître de Conférences à l'Université Pierre et Marie Curie. Elle est très concernée par la dissémination des résultats de la recherche dans l'enseignement secondaire et universitaire.

Adresse : L'UTES, Université Pierre et Marie Curie, 4 place Jussieu 75270 Paris cedex 05

Courriel : Elisabeth.Delozanne@upmc.fr

Toile : www.math-info.univ-paris5.fr/~delozanne/

Françoise LE CALVEZ est maître de conférences à l'université Paris 6 et chercheur dans l'équipe MOCAH (Modèles et Outils en ingénierie des Connaissances pour l'Apprentissage Humain). Ces dernières années ses recherches ont porté sur la méthodologie de conception de différents composants d'un EIAH dans le cadre du groupe Combien? (<http://combien.lip6.fr>). Elle s'intéresse aussi à la définition et à l'indexation de ressources pédagogiques et au diagnostic des activités de l'apprenant en situation de résolution de problèmes.

Adresse : LIP6, Université Pierre et Marie Curie, 104 avenue du président Kennedy, 75016 Paris

Courriel : Francoise.Le-Calvez@lip6.fr

Toile : <http://webia.lip6.fr/~lecalvez/>

Agathe MERCERON est professeur d'informatique à la Technische Fachhochschule Berlin (University of Applied Sciences TFH Berlin). Elle a un doctorat en 3ème cycle spécialité sciences de l'information et de la documentation de l'université Paris 7 et un doctorat d'Etat spécialité sciences informatiques de l'université Paris XI. Ses recherches en EIAH concernent particulièrement les fouilles de données éducationnelles. Elle est associée honoraire de l'Université de Sydney

Adresse : Technische Fachhochschule Berlin, Luxemburgerstrasse 10, 13353 Berlin, Germany

Courriel : merceron@tfh-berlin.de

Toile : <http://www.tfh-berlin.de/~merceron/>

Jean-Marc LABAT a soutenu une thèse en intelligence artificielle (90) portant sur les environnements informatiques pour l'apprentissage humain, puis une habilitation à diriger des recherches (97), il a ensuite été promu professeur d'informatique à l'université René Descartes où il a dirigé pendant 4 ans le laboratoire Crip5. Depuis février 2005, il a rejoint l'université Pierre et Marie Curie. Il est responsable d'une équipe de recherche au sein du Lip6 et dirige L'UTES, un service commun de l'université. Il est l'actuel président de l'ATIEF. Ses travaux portent sur la modélisation et le suivi de l'apprenant, et plus généralement sur la modélisation cognitive de l'utilisateur en situation de résolution de problèmes.

Adresse : L'UTES, Université Pierre et Marie Curie, 4 place Jussieu 75270 Paris cedex 05

Courriel : Jean-Marc.Labat@upmc.fr

Toile : <http://www.lutes.upmc.fr/labat/index.html>

Référence de l'article :

Élisabeth Delozanne, Françoise Le Calvez, Agathe Merceron, Jean-Marc Labat, Design Patterns en EIAH : vers un langage de Patterns pour l'évaluation des apprenants, *Revue STICEF*, Volume 14, 2007, ISSN : 1764-7223, mis en ligne le 7/12/2007, <http://sticef.org>

© Revue Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation, 2007