



HAL
open science

Concevoir des Scénarios Pédagogiques Exécutables avec des Patrons de Conception Pédagogiques

César Olavo de Moura Filho, Alain Derycke

► **To cite this version:**

César Olavo de Moura Filho, Alain Derycke. Concevoir des Scénarios Pédagogiques Exécutables avec des Patrons de Conception Pédagogiques. Jun 2007. hal-00161475

HAL Id: hal-00161475

<https://hal.science/hal-00161475>

Submitted on 10 Jul 2007

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.

Concevoir des Scénarios Pédagogiques Exécutables avec des Patrons de Conception Pédagogiques

César Olavo de Moura Filho^{§,*}, Alain Derycke[§]

§ Laboratoire Trigone, Université des Sciences et Technologies de Lille, 59655, Villeneuve d'Ascq cedex, France.

** Centro Federal de Educação Tecnológica do Ceará
Av. 13 de Maio, 2081
60.000 Fortaleza Ceará Brasil*

RÉSUMÉ. Le papier propose une démarche de conception, pour les EIAH destinés à l'e-formation, qui s'appuie sur le concept de patrons pédagogiques de conception. Cette démarche part d'un constat quant aux difficultés du processus de conception lui-même, et aux risques de coupure qu'il peut y avoir entre pédagogues et développeurs de logiciels. Pour améliorer la capacité des équipes pédagogiques à pouvoir spécifier, et même implémenter, des scénarios pédagogiques, dans une approche basée sur les patrons, nous proposons de garder côte à côte et en coïncidence les descriptions en langage informelles, et les descriptions plus formelles et normatives, et d'offrir la possibilité d'effectuer des allers-retours à toutes les phases du cycle de vie du dispositif pédagogique. Notre proposition réside dans la conception d'un environnement informatique « auteur », qui utilisent les avancées récentes du génie logiciel, telles l'Ingénierie Dirigée par les Modèles et les Languages Oriented Programming (LOP), et sur l'existence d'un environnement ouvert et extensible pour fabriquer des ateliers de génie logiciels dédiés : ECLIPSE. Nous développons dans ce papier essentiellement la partie éditeur de scénarios pédagogiques textuels et les mécanismes de construction des modèles sous-jacents, permettant ensuite une formalisation, et dégageons enfin les intérêts d'une telle démarche de conception.

MOTS-CLÉS : informatique, pédagogie, e-formation, conception des EIAH, modélisation, scénario pédagogique, démarche de conception par patrons, patrons pédagogiques, ingénierie dirigée par les modèles, normes et standards, langages de descriptions, environnement de conception logiciel, génie logiciel et ECLIPSE pour le EIAH.

1. Introduction

La plupart des recherches récentes en EIAH, visant l'amélioration des façons de concevoir et d'implémenter des applications logicielles pour l'e-formation, ont été centrées sur l'exploitation de modèles de description standardisés, et plus ou moins formels. Friesen (2004) nous explique qu'en e-formation les standards sont souvent composés de plusieurs éléments : un modèle conceptuel donnant les fondements, accompagné d'un ou plusieurs éléments spécifiant comment ce modèle conceptuel est exprimé dans un idiome formel, le plus souvent dérivé du langage XML. Ce modèle conceptuel est enfin quelque fois également accompagné d'une spécification pour une interface de programmation ou de service (une API). Cette API définit des « contrats » standardisés pour la coopération avec les autres entités du système.

Ceci signifie qu'au cœur de ces EIAH, il y a des modèles conceptuels, formulés de manière plus ou moins standardisée, et décrivant les éléments clés des entités logicielles utilisées dans l'e-formation. Certains des standards pour la spécification, comme IMS - LD, concernent la modélisation des scénarios pédagogiques et donc de l'activité d'apprentissage. C'est ce domaine qui constituera le cœur de notre proposition.

Avec la généralisation des formalismes et langages informatiques dans le paradigme objet, les modèles conceptuels prennent encore plus de poids, car ils peuvent servir directement à la génération des applications informatiques ou des EIAH. Dans ce cas, il est possible de considérer l'application logicielle finale comme le produit d'une opération de transformation complexe dont la source serait le modèle conceptuel du dispositif pédagogique à instrumenter. Ainsi il semble qu'avec ce nouveau courant du Génie Logiciel, le processus global du développement des applications informatiques peut être raccourci et facilité, au point de permettre à des non-informaticiens – par exemple des concepteurs pédagogiques – de pouvoir contribuer de manière significative à l'effort de développement, dans un nouveau processus que Fowler (2005) appelle le *Lay Programming*.

Cependant nous soutenons que les descriptions informelles des entités, phénomènes, scénarios, etc. sont actuellement sous-représentées dans les différentes approches de la conception des dispositifs d'e-formation. Avec la représentation de cette source originelle de connaissances, capturée directement auprès des experts, pas nécessairement au fait des technologies informatiques, nous faisons l'hypothèse que ces modèles informels pourraient occuper la scène centrale, à côté des modèles plus formels et plus technologiques.

De nombreuses techniques d'analyse des requis, comme les *Use-cases* dans le langage de description UML, les scénarios, les narrations, etc. ont été conçues pour pouvoir décrire informellement le système à concevoir, et comme une première étape dans la conception des logiciels. Parmi ces techniques, les approches par patrons de conception (*Design Patterns*) sont fréquemment utilisées, comme une notation pratique pour capturer la connaissance des experts, et ceci dans des domaines aussi

variés que la conception architecturale, le génie logiciel... Nous nous sommes particulièrement intéressés à l'application de ces patrons dans le domaine des stratégies pédagogiques, c'est-à-dire aux Patrons Pédagogiques (PP dans la suite du texte).

Ce papier discute quelques uns des aspects impliqués dans la conception des scénarios d'apprentissage, et propose une approche et une implémentation informatique, dans lesquelles les modèles informels joueront un rôle majeur dans la démarche de conception de ces scénarios. Nous proposons donc un environnement logiciel de conception d'EIAH qui laisse le concepteur construire des descriptions plus formelles, sans pour autant perdre la centration sur des descriptions faites en langage naturel. Nous faisons l'hypothèse que les approches de conception centrées sur les patrons ne sont pas bien connues des lecteurs de ce papier. Aussi nous ferons une introduction assez importante, aux problèmes généraux de la conception des artefacts, auxquels la communauté EIAH n'échappe pas, et à la genèse de l'approche par patrons, à ses fondements, et aux langages de patrons.

Ensuite nous nous focaliserons sur l'approche fondée sur les Patrons Pédagogiques et sur l'instrumentation du processus de conception au travers d'un environnement complet de conception et génération d'applications informatiques de type EIAH. Nous décrirons plus particulièrement l'éditeur spécialisé de scénarios pédagogiques à base de patrons et les allers-retours que permet le système entre représentations informelles, proches du langage naturel, et les représentations plus formelles, proches des propositions de standardisation.

2. Sur les représentations informelles et leurs limites:

« *Il est acquis que les locuteurs veulent transmettre plus d'informations que celles contenues dans ce qu'ils disent* » (Jaszczolt, 2006). Certaines théories de la communication expliquent pourquoi et quand le locuteur, inconsciemment, décide quand les inférences contextuelles manquantes peuvent être reconstituées par l'interlocuteur, et sont donc omises de son discours. Mais bien sûr une reconstitution parfaite n'est pas toujours possible, et c'est l'un des traits fondamentaux du langage naturel. Des figures linguistiques subjectives, comme la métonymie, la métaphore ou la synecdoque, etc., ou alors juste un manque de précision dans la définition et l'utilisation des mots utilisés, ajoutent même plus de polysémie dans l'énoncé, et rendent son interprétation toujours incertaine.

C'est donc sans surprise que les personnes désirant créer des représentations de phénomènes, entités, etc. devant être supportés par l'informatique, se tournent alors des modèles plus formels et à la sémantique plus limitée et contrainte. Cependant la transition depuis une description en langage naturel de l'artefact à produire (une première étape de la conception), vers des modèles formels, tels que ceux requis par les ordinateurs, ne peut pas être automatisée. Mais il est possible d'offrir un environnement informatique de conception qui permette des allers-retours entre ces deux représentations, et une mise en coïncidence favorisant une pratique réflexive

chez le concepteur. Cela peut être un bon compromis entre expressivité et précision dans le cas de la conception des scénarios pédagogiques.

3. Sur la conception elle-même et ses difficultés:

Concevoir n'est jamais une tâche humaine triviale. Les concepteurs sont toujours en train de négocier avec le « nouveau, l'imprévu », ou comme le dit Schön (1987) avec une situation embrouillée et indéterminée. Parce que *“la réalité humaine est trop complexe pour que nous puissions y agir sans la réduire”* (Meirieu, 1987), les concepteurs, y compris ceux des EIAH, sont en permanence en train d'essayer de rendre leur tâche plus tangible, en réduisant la complexité, et tout en préservant ce qu'ils considèrent comme essentiel. Un façon de réduire la complexité de la tâche de conception est de contraindre l'espace problème, et pour arriver à cela, il est souvent fait appel à la puissance de l'abstraction : *“One of the most important challenges of system design is dealing with complexity. We attack complexity with abstraction.”* (Coplien, 1996). Une voie scientifique, et commune, de se frotter à l'abstraction, est d'utiliser des modèles. Ces modèles autorisent la focalisation sur les aspects pertinents de la conception, en enlevant toutes les parties superflues, ou en d'autres mots en augmentant le niveau d'abstraction (Greenfield, 2003), suivant le vieux principe de séparation du bon grain de l'ivraie.

Dans ce papier nous intéressons plus particulièrement à l'idée d'une démarche de conception vue comme un choix de patron(s) de conception, c'est-à-dire comment les patrons nous aident à se frotter avec la complexité. Une preuve de cela peut être fournie, pour le génie logiciel, par le livre de (Gamma et al. 1995) qui prétend qu'une fois que l'on a trouvé le patron de conception adéquate, une bonne partie des décisions de conception suivront automatiquement. Ce qui signifie que, lorsque l'on connaît le patron à appliquer, l'espace solution devient particulièrement contraint, par la solution proposée dans le patron. Ainsi le problème devient alors comment savoir quel patron de conception appliquer dans un cas particulier ? Un thème que nous allons traiter dans la suite du papier.

4. Patrons de Conception:

4.1. Sur la genèse des patrons de conception dans la démarche de conception

C'est Ch. Alexander (1977), dans le livre *“A Pattern Language: towns, buildings, construction”*, qui le premier a proposé l'idée de patron, proposant un langage de patrons de conception architecturale, comprenant 253 patrons différents, pour décrire les bâtiments, et comment ils peuvent être conçus. Langage où chaque patron est *“ une règle à trois parties, qui exprime un relation entre un certain contexte, un problème et une solution”*.

Le patron peut être aussi vue comme une **micro-architecture** (un motif pouvant être récurrent). L'idée de base est que pour un problème particulier de conception, le patron est en général « plus petit » que le problème global, c'est-à-dire que le patron va contribuer seulement partiellement à la conception globale. Mais il n'est pas interdit, bien au contraire de faire appel dans un problème de conception à plusieurs patrons, et à les combiner. Dans le domaine du génie logiciel la démarche de conception basée sur les patrons en tant que micro-architecture a su également largement s'installer : « *[Design patterns] are micro-architectures: structures larger than objects but not large enough to be system-level organizing principles* ». (Coplien, 1996)

Les patrons de conception peuvent être vue aussi comme des supports pour encapsuler de la connaissance, au sens où : “*pattern names are abstractions that encapsulate implementation knowledge, creating a new and more powerful language*” (Greenfield *et al.*, 2004). On voit donc que ces patrons faciliteront la transmission des connaissances des concepteurs, qui au sein de communautés de pratiques vont cristalliser leurs expériences dans des éléments, les patrons, pouvant prendre sens dans une autre problème de conception, et une autre communauté de concepteurs.

Il vient donc une autre vision des patrons comme étant des « Bonnes Pratiques », une compilation des voies particulières et efficaces de faire les choses. En étant des écrits ayant « autorité », suite par exemple à la notoriété de leurs auteurs, des concepteurs reconnus et expérimentés, les patrons permettent aux concepteurs novices d'être, en quelque sorte, porter par ces experts. Le succès du mouvement sur les patrons dépend donc de la confiance que les concepteurs novices placent dans les auteurs des patrons (ou dans la crédibilité de la communauté qui les a produits) et dans l'assurance qu'il s'agit bien là des « meilleures » pratiques.

4.2. Les langages de Patrons

Comme nous venons de le voir précédemment, pour être utile dans la conception, les patrons se doivent de réduire la complexité. Cependant, même si les patrons peuvent être de bonnes solutions pour des problèmes de petites tailles, l'établissement de relations entre patrons de nature différente peut être utile pour adresser des problèmes plus complexes, dans une approche pas à pas. Pour s'attaquer à ce problème de liaison entre patrons, Alexander a proposé la notion de *Pattern Language* (langage de patrons) qui consiste en un jeu de patrons travaillant ensemble.

Greenfield *et al.* nous explique que: “*Any language defines a set of abstractions, which form its vocabulary, and a set of rules, which form its grammar, for combining those abstractions to create expressions. Both the vocabulary and the grammar of a pattern language can be expressed as patterns*” (Greenfield *et al.*, 2004)

5. Les Patrons Pédagogiques:

Si le livre de Alexander jeta les principes du mouvement « patrons » dans les années 70, c'est que très récemment que la portée des patrons s'est étendue à de nombreux domaines de la conception, dans des domaines aussi divers que le Travail Coopératif Assisté par Ordinateur, le Génie Logiciel, les IHM... voire à l'éducation et à la conception des dispositifs pédagogiques, avec et sans instruments informatisés. C'est une approche qui n'a pas encore fait l'objet de beaucoup d'intérêt de la part de la communauté francophone des EIAH. Dans le monde anglophone, les *Pedagogical Patterns* (PP) ont commencé à susciter un intérêt et la création d'une communauté vivace, en parallèle d'ailleurs avec son pendant informatique, comme le témoigne les ateliers PP qui ont en général lieu pendant la grande conférence ACM PLoPs consacrée aux patrons de conception.

Pour Bergin (2001): "*The intent [of pedagogical patterns] is to capture the essence of the practice in a compact form that can be easily communicated to those who need the knowledge*". Mais il faut reconnaître aussi que cette approche de conception des dispositifs pédagogiques, n'a pas la maturité de celle existante en génie logiciel, peut être pour des raisons culturelles, et que la communauté éducative est toujours loin de mettre les PP dans leur boîte à outils de tous les jours. Pour une vue non exhaustive d'un catalogue de PP se référer à www.pedagogicalpatterns.org. Le tableau ci-dessous présente un extrait du patron *Challenge Understanding*, tiré du langage Feedback Patterns (Bergin et al., 2002) :

CHALLENGE UNDERSTANDING ** (This pattern was written by Helen Sharp.)
You are teaching a course or seminar. You want students' understanding to be challenged so that you and they will know how well they have learned a topic.
□□□

Students may think they understand a topic when in fact they do not. If they think they understand, they will not ask questions and will not know the gaps in their knowledge. In addition it is not possible to cover all nuances of a topic in notes ...One way in which students learn is to ask questions of you and their peers, but....
□□□

Therefore, give the participants exercises, tasks or activities that challenge their understanding. These can be in the form of exercises, which help the student to apply techniques by Try it Yourself, or Groups Work [EBS] during...
□□□

This pattern is quite abstract, and other patterns are needed to refine, extend, and implement it. However we do not refine it in this language, although it is

6. Notre prototype pour l'instrumentation de la conception de scénarios pédagogiques à base de patrons:

Si nous considérons que les stratégies pédagogiques sont conçues par des personnes, il est raisonnable de penser qu'elles vont d'abord l'exprimer en langage naturel, dans une notation textuelle. Le processus de transformer cela dans une

représentation plus formelle, entraîne toujours une déperdition dans l'intentionnalité, et implique nécessairement deux moments critiques : en premier le concepteur développe une interprétation particulière de l'intention du créateur (prescripteur ?), et ensuite il fixe cette interprétation dans un modèle, en utilisant un langage formel. Le problème est que ce processus « interprétation/fixation » laisse seulement une place limitée à l'adaptation (limitée par la sémantique et la syntaxe du langage de description sélectionné), dans le cas où la stratégie pédagogique ne répondrait pas précisément aux besoins.

En se fondant sur toutes les considérations exposées ci avant, nous avons construit un éditeur logiciel qui permet, premièrement, l'édition de patrons pédagogiques, et deuxièmement, l'extraction de modèles pédagogiques formels à partir des scénarios d'apprentissage édités sur notre environnement. Ceci a été conçu en suivant le principe que, à chaque fois que possible, nous garderons les représentations formelles et informelles côte à côte. Ainsi si la conception instrumentée se focalise sur la production de représentations formelles, permettant ainsi d'être injectées dans une approche d'ingénierie des EIAH fondée sur l'Ingénierie Dirigée par les Modèles, ou IDM, comme dans (Caron et al, 2005), nous souhaitons également conserver ces représentations du « phénomène » original dans une notation informelle. Et ceci afin de laisser une certaine liberté d'action pour l'interprétation et la négociation, permettant d'autres traductions futures vers de nouveaux modèles formels.

En fait l'éditeur dont nous parlons, et appelé *Pedagogical Patterns Editor* (ou PP Editor), est une partie d'un environnement de conception des EIAH plus large, le projet MDEduc (Moura et al, 2007) destiné à la production de scénarios d'apprentissage dans des spécifications informelles et formelles, multi-standards pour ces dernières, et incluant la génération de codes logiciels. Plus précisément MDEduc est construit comme un jeu de *plug-in* (ou modules) pour l'environnement de génie logiciel, largement utilisé dans le monde, ECLIPSE (Gamma & Beck, 2003). Modules qui, mis ensemble, doivent couvrir tout le cycle de vie dirigé par les modèles, pour produire du code depuis un modèle génératif, en partant d'une approche patron.

Le PP Editor est un éditeur de textes spécialisé (dirigé par la syntaxe) qui facilite l'édition de scénarios d'apprentissage en respectant la syntaxe des patrons pédagogiques. Cette simplification de la tâche du concepteur est obtenue en proposant des assistants jouant sur le contenu en cours d'édition : une décoration en couleur du texte en se fondant sur la syntaxe retenue pour les PP, des *Templates* spécifiques, parmi d'autres fonctionnalités, pour que ceux, qui ne sont pas accoutumés à une approche de type PP, puissent y arriver sans un effort trop important. Dans la figure 1, nous pouvons voir que certains mots ou membres de phrase, ont un rendu visuel différent en accord avec les conventions syntaxiques usuelles des PP, par exemple les liens vers d'autres PP sont colorés en rouge, la début de la section Solution est en gras... Sur la droite de la figure nous pouvons aussi voir une vue Plan qui présente seulement les mots mis en valeurs et disposés dans une arborescence. De plus pour aider à l'édition des PP, le PP Editor, en

interagissant avec les autres modules de notre environnement MDEduc, facilite la tâche de création de meta-modèles dans le formalisme pivot ECORE (Budinsky et al., 2003) en vigueur dans ECLIPSE pour supporter l'IDM, pour supporter la génération de modèles formels de scénarios d'apprentissage. La figure 1 montre aussi les actions possibles des utilisateurs sur ECLIPSE pour la création du métamodèle. Le modèle textuel que nous montrons est dérivé d'un PP appelé « *Peer Feedback* » (Bergin et al., 2002).

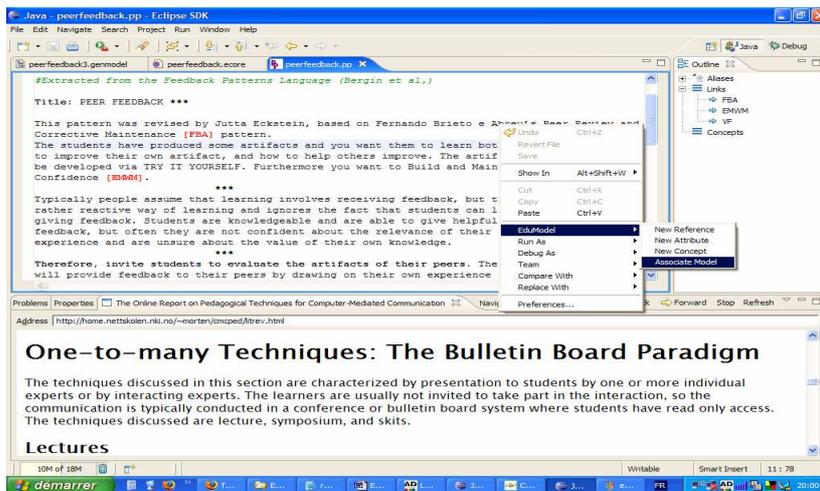


Figure 1 : éditeur de patrons pédagogiques

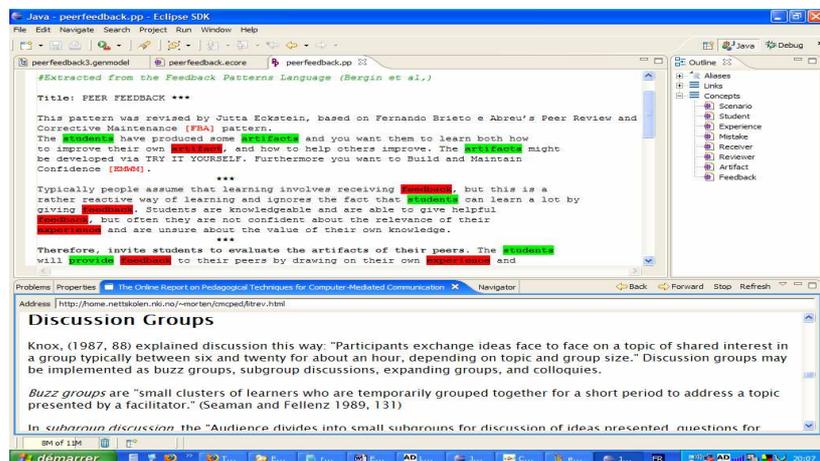


Figure 2 : la construction de la vue plan (en haut, à droite)

La figure 2 montre, dans la vue Plan, le métamodèle résultant, après quelques interactions avec le concepteur pour la création de nouveaux concepts, leurs attributs et références, et les mêmes termes mis en valeur dans le texte correspondant (concepts en rouge, références en vert, et les attributs, non représentés ici, en bleu).

Même si le modèle lui-même, avec ses vues et actions, dépendent de modules différents dans ECLIPSE, ceux-ci restent synchronisés entre eux, au travers du mécanisme proposé par ECLIPSE dans son intergiciel (les *extension points*) : Tous les modules se déclarant comme « auditeurs » seront avertis lors d'un changement dans un modèle – et ainsi ce changement se répercutera automatiquement dans tous les modules intéressés.

Enfin, même si ce n'est pas présenté ici, il faut savoir que l'environnement MDEduc propose un moyen de générer le code, par exemple en Java, nécessaire à l'exécution du modèle (son opérationnalisation). Cela rejoint une approche parallèle développée également dans notre laboratoire, en partant de nos outils supports à l'IDM comme MODIX et GENDEP (Caron et al, 2006).

7. Comparaison des différentes approches:

Dans cette section nous allons donner un aperçu des travaux de recherche et des projets qui proposent eux aussi l'utilisation des patrons comme artefacts de conception des EIAH. Pour rendre plus pédagogique cette vue comparée nous allons classer les différentes approches utilisées par les concepteurs en Pre-EML EML-Based, et Post-EML, par rapport à la place que revêt le langage de modélisation pédagogique (le Educational Modeling Language) dans le processus global.

Les approches dites Pre-EML sont basées sur des modèles pédagogiques prescriptifs, largement utilisés dans les courants de la conception des technologies de l'instruction (ISD, comme par exemple le modèle ADDIE – Grafinger (2007)). En fait ces approches sont de base des descriptions de type processuels, et le modèle doit être entendu comme quelque chose d'idéal, une prescription, qu'il faut imiter, et non dans le sens où nous avons utilisé le mot modèle comme étant conceptuel. Dans ce cas les techniques pédagogiques sont profondément enracinées, imbriquées, avec celles de la conception. Cela signifie que les approches pre-EML ne séparent pas clairement les requis et objectifs pédagogiques de ceux de la technologie. Quelques uns des projets que nous présentons brièvement ici, peuvent être qualifiés de pre-EML.

E-LEN (2006) est un projet visant la documentation de l'expérience de conception, sous la forme de patrons de conception, pour une plate-forme d'e-formation (un LMS) au travers du développement d'un langage de patrons spécifique. Ceci est utilisé pour l'implémentation de conseillers (Wizards) qui guident les administrateurs de ces plates-formes dans la configuration du système pour des instances sélectionnées de PP (Kolas and Staupe, 2004).

En contraste avec les approches présentées précédemment, les approches dites EML-Based mettent l'accent sur le travail au niveau de la modélisation conceptuelle. Par exemple **E²ML**, proposé par Botturi (2003) est un langage visuel servant à décrire des patrons pédagogiques, aussi bien que des nouvelles solutions pédagogiques. Il peut être utilisé pour créer des illustrations qui serviront d'outils de

communication entre les ingénieurs pédagogiques et les développeurs. Dans son travail Derntl (2005) propose une approche basée sur les patrons pour capturer, décrire, modéliser et réutiliser des conceptions de e-formation centrées sur l'apprenant. Dans cette approche, après qu'un patron ait été élicité, un concepteur peut commencer à modéliser conceptuellement sa structure et le « *learnflow* », en utilisant la notation UML comme en génie logiciel.

Comparée avec toutes ces propositions, où les patrons sont des lignes directrices conduisant un processus (pre-EML) ou un EML intégré, nous pouvons dire que notre proposition (qualifiée de post-EML, c'est-à-dire aussi au-delà de IMS-LD) présente une portée plus large, car les patrons de MDEduc peuvent piloter le cycle complet de conception conforme à l'IDM, comme dans la proposition MDA. Ce cycle commence avec un éditeur ad hoc de patrons, et finit avec une, ou plus, applications exécutables. Derntl, propose bien lui aussi des « *Web templates* » comme résultats de son processus de conception, mais ceux sont obtenus à la main, et non pas générés automatiquement depuis le modèles sous-jacent.

8. Conclusion

Avec l'émergence du paradigme LOP en génie logiciel, les représentations de scénarios, formelles et informelles, peuvent servir pour la génération d'applications informatiques complètes, en autorisant les concepteurs pédagogiques à participer à l'effort de développement. Le cycle de vie traditionnel de développement du logiciel, qui part de spécifications informelles, les besoins des usagers et prescripteurs, vers une application en action, est accessible aux seuls professionnels de l'informatique. Les solutions dites LOP-Based cherchent, quant à elles, à adresser cette complexité de la conception avec des cadres structurants qui cachent la plupart des difficultés techniques, laissant aux concepteurs la tâche de base de création du modèle conceptuel de leur domaine d'application. Modèle à partir duquel le programme exécutable, ou la configuration opérationnelle d'un LMS donné, seront automatiquement générés. Dans ce cas la totalité du cycle de développement pourra être couverte par la même équipe pédagogique experte du domaine de formation.

Dans un tel scénario nous considérons que ces descriptions informelles peuvent encore être utiles, même après que le dispositif de formation avec EIAH ait été déployé, et peuvent être continuellement revues, de telle façon que de nouvelles interprétations (de nouveaux métamodèles) puissent orienter l'affinement, l'amélioration du système. Nous pouvons donc justifier le fait que le stockage de stratégies pédagogiques sous la forme de Patrons Pédagogiques, peut apporter une plus-value à une archive partagée de scénarios, favorisant la réutilisation et le partage de compétences. Et ceci parce que, si nous ne sommes pas satisfaits d'une conception pédagogique donnée, nous pouvons toujours revenir à sa source première, et dériver un nouveau modèle à partir de celle-ci. Ainsi en gardant les patrons pédagogiques nous pouvons retarder, pour un temps plus lointain, le moment où les choses seront « figées », quand il arrivera qu'un nouveau modèle émerge,

peut-être après quelques considérations, faites après coup, au sujet du choix d'un formalisme pour la représentation plus formelle (à la IMS-LD, à la MISA...), c'est-à-dire des réflexions au niveau du métamodèle ou du modèle.

Le chemin à parcourir pour aller dans cette direction est encore long. Nos travaux portent sur la réalisation d'un environnement complet d'aides à la conception de dispositifs d'e-formation. Au sein de notre laboratoire ceci est en cohérence avec les travaux menés parallèlement (Caron et al 2005), en utilisant tout le potentiel de l'IDM, afin de faciliter l'implication directe et totale des pédagogues, dont les enseignants, dans le processus complexe et imbriqué de conception/ajustements/usages des EIAH. Nous n'avons développé ici que la dimension conception à base de patrons pédagogiques. Notre outil MDEduc est en cours de finalisation et devrait pouvoir être mis prochainement à disposition de la communauté EIAH (Moura, 2007). Pour que cette proposition devienne très effective il nous semble que nous devons encore travailler dans deux directions :

- Une mise à l'épreuve expérimentale de notre environnement de conception, après avoir au préalable nourri la base de référence pour les patrons pédagogiques ;
- Une contribution au mouvement international PP, en proposant de nouveaux PP plus appropriés à la conception de scénarios d'activité d'apprentissage avec instruments. Pour cela nous regardons du côté des rapports entre Patrons de conception et Théorie de l'Activité (Guy, 2004). De même le processus même de conception dans l'approche proposée devrait faire l'objet de patrons de processus documentés en tant que bonnes pratiques, suivant en cela la voie ouverte par les chercheurs en génie logiciel dans une approche de conception par patrons (Rieu, 1999)

Il est clair dans cette contribution, que nous n'avons pas fait volontairement de croisement avec les démarches dominantes de conception tournant autour du standard IMS-LD, pour la scénarisation de l'activité d'apprentissage. C'est que nous inscrivons en creux par rapport aux présupposés de telles démarches vues comme une ingénierie lourde, quasi-industrielle de la formation, et que notre réflexion de fond sur l'activité humaine créative de conception (Moura, 2007), nous entraîne dans une autre voie, plus pluraliste, voire plus émancipatrice pour les pédagogues.

9. Bibliographie

- Alexander, C., Sara Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I. and Angel. S. (1977). *A pattern language: town, buildings, construction*. New York: Oxford University Press.
- Bergin, J. (2001). Mining Pedagogical Patterns. Retrieved March, 2005 from <http://www.pedagogicalpatterns.org/meetus/ ecoop2001.html>.
- Bergin, J., Eckstein, J., Manns, M.L. and Sharp, H. (2002). Retrieved March, 2006 from <http://pedagogicalpatterns.org/current/feedback.pdf>

- L. Botturi (2004), "E2ML - Educational Environment Modeling Language", *EDMEDIA 2003* (vol.1), AACE Press, 304-311.
- Budinsky, F., Steinberg, D., Merks, E., Ellersick, R., and Grose, T. (2003) *Eclipse Modeling Framework*, Addison-Wesley.
- Caron PA, Derycke A., Le Pallec X (2005), "Bricolage and Model Driven Approach to design distant course", *E learn 2005, world conference on E-learning in corporate Government, Healthcare & higher education*, Vancouver, AACE press, 2856- 2864
- Coplien, (1996) *J. Software Patterns*, SIGS Publications, New York, NY.
- Derntl, M. (2005). "Patterns for Person-Centered e-Learning," PhD thesis, Faculty of Computer Science, University of Vienna, Austria.
- E-LEN (2006). E-LEN project. Accessed online on sep. 2006 at: <http://www2.tisip.no/E-LEN/>
- Fowler, M. (2005). Language Workbenches: The Killer-App for Domain Specific Languages? Accessed online from: <http://www.martinfowler.com/articles/languageWorkbench.html>
- Friesen N. (2004) *Canadian Journal of Learning and Technology* Volume 30(3) Fall / automne Editorial - A Gentle Introduction to Technical E-learning Standards
- Frizell, S. & Hubscher, R. (2002). Aligning theory and web-based instructional design practice with design patterns. Paper presented at the World Conference on E-Learning in Corp., Govt., Health. & Higher Ed.
- Gamma, E., Beck, K.,(2003). *Contributing to Eclipse: Principles, Patterns, and Plug-Ins*. Addison –Wesley.
- Grafinger, Deborah J. (1988). Basics of instructional systems development. INFO-LINE Issue 8803.Alexandria: American Society for Training and Development.
- Greenfield, J., Short, K., Cook, S., Kent, S. and Crupi, J. (2004) *Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools*, Wiley, NY.
- Guy, E.S. (2004). Appropriating Patterns for the Activity Theory Toolkit. Proceedings of the *First International Workshop on Activity Theory Based Practical Methods for IT Design*. Copenhagen, Denmark, September 2-3, 2004.
- Jaszczolt, K. M., (2006) "Defaults in Semantics and Pragmatics", *The Stanford Encyclopedia of Philosophy*, Edward N. Zalta (ed.), URL = <<http://plato.stanford.edu>>
- Kolas, L., & Staupe, A. (2004). Implementing delivery methods by using pedagogical design patterns. Proceedings of *ED-MEDIA 2004*, AACE press,
- Meirieu, P. (1987) *Apprendre...oui mais comment*, ESF, Paris, 1987
- Moura, C.O. Cheuoua, J. Derycke, A (2007) A Multi-model Approach to Representing Pedagogical Scenarios. Soumis à *IFETS, Educational Technologies & Society*, 20pp.
- Rieu D., Girardin J.P., Saint-Marcel C., Front- Conte A., Des operations et des relations pour les patrons de conception, *Actes de Inforsid 1999*, Hermes, Paris.
- Schön, D. A. (1987). Educating the reflective practitioner. San Francisco, CA: Jossey-Bass.