



HAL
open science

Abstraction par méta-modélisation du métier de conception des plate-formes de formation

Esteban Loiseau, Nour El Mawas, Pierre Laforcade, Sébastien Iksal

► To cite this version:

Esteban Loiseau, Nour El Mawas, Pierre Laforcade, Sébastien Iksal. Abstraction par méta-modélisation du métier de conception des plate-formes de formation. 7ème Conférence sur les Environnements Informatiques pour l'Apprentissage Humain (EIAH 2015), Jun 2015, Agadir, Maroc. pp.324-335. hal-01405961

HAL Id: hal-01405961

<https://hal.science/hal-01405961>

Submitted on 30 Nov 2016

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.

Abstraction par méta-modélisation du métier de conception des plate-formes de formation

Esteban Loiseau, Nour El Mawas, Pierre Laforcade and Sébastien Iksal

LUNAM Université, Université du Maine, EA 4023, LIUM, 72085 Le Mans, France
prenom.nom@univ-lemans.fr

Résumé. Le projet GraphiT vise à aider des enseignants à se focaliser sur la conception de situations d'apprentissage tout en garantissant que les modèles produits sauront être exploités informatiquement pour faciliter la mise en œuvre sur la plate-forme de formation mise à leur disposition (opérationnalisation). Notre approche originale est centrée sur l'abstraction du métier de conception d'une plate-forme afin de spécifier des langages de conception spécifiques. Cette problématique est traitée dans le cadre méthodologique de l'Ingénierie Dirigée par les Modèles et concerne, à titre de cas d'étude, la plate-forme Moodle. Notre proposition consiste à augmenter l'expressivité pédagogique des modèles de conception par la spécification de blocs qui correspondraient à des usages récurrents des outils, ressources et paramétrages de la plate-forme. Nous proposons de capturer ces éléments de conception ainsi que leurs correspondances pour faciliter la conception et diriger la traduction lors de l'utilisation d'outils dédiés que nous développons.

Mots-clés. Conception pédagogique, Langage de scénarisation pédagogique graphique, Learning Management System, Plate-forme de formation, Méta-modélisation

Abstract. The funded GraphiT project aims to help teachers in focusing on the specification of pedagogically sound and technically executable learning designs dedicated to an existent Learning Management Systems (LMS). To this end, we propose to support teachers by providing an LMS-specific Visual Instructional Design Language (VIDL). This paper proposes a specific LMS-centered approach for raising the pedagogical expressiveness of its implicit learning design semantics. We discussed how the LMS low-level parameterizations could be abstracted in order to build higher-level building blocks. Based on the Moodle application, we present and illustrate our approach by formalizing the abstract syntax of a Moodle-dedicated VIDL.

Keywords. Instructional Design, Visual Instructional Design Language, Learning Management System, Modeling and Meta-modeling

1 Introduction

De nos jours, les plate-formes de formation ou *Learning Management Systems* (LMS¹) sont largement répandues dans les institutions académiques. Leur usage n'est plus limité à des formations à distance mais s'est étendu aux formations mixtes comme aux formations en présentiel [11]. Néanmoins, les résultats d'une enquête et d'entretiens menés en 2014 auprès de 203 enseignants mettent en avant de nombreuses difficultés quant à l'appropriation et l'utilisation de ces plate-formes [18]. S'ils souhaitent mettre en place des activités pédagogiques complexes, les enseignants doivent développer des compétences de haut-niveau quant à l'utilisation du LMS : comment et quand utiliser et combiner les différentes fonctionnalités de la plate-forme afin de supporter l'objectif pédagogique fixé ? Ces compétences peuvent être acquises au cours de formations professionnelles, qui se concentrent d'avantage sur les aspects techniques liés aux fonctionnalités de la plate-forme plutôt qu'à la conception de scénarios pédagogiques cohérents et adaptés à cette plate-forme. Dû à la multiplicité des théories éducatives [17] et des approches de conception, ainsi qu'à l'absence d'outils et de méthodes de scénarisation spécifiques aux LMS, les enseignants développent, de façon ad-hoc, leurs propres méthodes et outils de conception.

Dans un tel contexte, il semble pertinent d'aider les enseignants-concepteurs à mieux exploiter les LMS qui sont mis à leur disposition plutôt que de leur proposer des outils de conception, indépendants des plate-formes, pédagogiquement expressifs mais ayant des difficultés à faciliter la mise en œuvre des modèles de conception produits (tels que [3] et [13] pour les travaux récents sur ce domaine). Notre approche consiste ainsi à s'intéresser à une plate forme donnée et à identifier le potentiel en terme d'expressivité pédagogique qu'elle permet d'exprimer (c.-à-d. que l'on saura mettre en œuvre sans perte d'information). Nous cherchons donc à produire des langages de conception, et leurs outils-auteurs associés, dédiés à la conception et la mise en œuvre de situations d'apprentissage spécifiques pour une plate-forme donnée. À l'aide de tels outils nous cherchons à ce que les enseignants puissent plus facilement s'approprier leur plate-forme et ainsi concevoir et mettre en œuvre des situations d'apprentissage plus évoluées pédagogiquement.

Le projet GraphiT (financé par l'ANR) est basé sur cette approche de conception centrée plate-forme. Son objectif principal est d'étudier les techniques liées à l'Ingénierie Dirigée par les Modèles (IDM) et le *Domain Specific Modeling* (DSM) afin d'explicitier le métier de conception des plate-formes, puis de l'abstraire afin de concevoir des langages de scénarisation pédagogique graphiques de plus haut niveau d'expressivité sans perte d'information lors de la mise en œuvre. Des précédents travaux se sont intéressés à la méta-modélisation pour identifier et explicitier le métier de conception de plusieurs plate-formes [2]. Cet article s'intéresse quant à lui à l'exploitation de techniques de méta-modélisation afin de traiter l'abstraction du métier de conception de la plate-forme et d'augmenter ainsi l'expressivité pédagogique. La proposition d'abstraction que nous proposons dans cet article ne concerne uniquement la plate-forme Moodle.

Nous détaillons dans la section 2 de cet article notre contexte de recherche. La section 3 est dédiée à la présentation de notre proposition d'abstraction par méta-modélisation pour le cas de la plate-forme Moodle. La section 4 illustre l'utilisation de ce méta-modèle pour spécifier un exemple de scénario. Nous présentons également des exemples de correspondances entre fonctionnalités de la plate-forme et activités

¹ L'acronyme LMS et le terme « plate-forme » seront utilisés par la suite, indifféremment, pour évoquer les plate-formes de formation.

pédagogiques, ainsi que notre proposition à base de tissage de modèles pour les spécifier et les intégrer dans notre prototype d'éditeur. Enfin, nous terminons cet article en tirant les premières conclusions de ce travail d'abstraction spécifique à la plate-forme Moodle.

2 Contexte de recherche

2.1 Plate-forme de formation et conception pédagogique

Le développement d'un LMS suit, explicitement ou non, certains courants éducatifs et intègre des approches pédagogiques spécifiques. Par exemple, Moodle adopte officiellement une approche socio-constructiviste [8]. Globalement, les LMS proposent une approche de conception basée sur l'agrégation et le séquençement de nombreux types d'outils (aussi appelées fonctionnalités ou, avec plus d'ambiguïté, *activité* pour la plate-forme Moodle) et de ressources. Les enseignants-concepteurs ont alors la charge de combiner ces divers éléments pour mettre en œuvre des activités pédagogiques à différents grains selon leurs besoins, leurs approches de conception, et leurs connaissances et compétences vis-à-vis de la plateforme (par exemples : simple consultation de ressource, mise en œuvre d'auto-évaluation, résolution de situations-problèmes collaboratives complexes, etc.).

Les standards centrés activités, tel que IMS-LD, n'ont pas réussi à s'intégrer dans les LMS actuels. Des travaux menés [6] ont montré, en plus du coût d'ingénierie, que cela nécessitait de faire évoluer le métier même de conception de la plate-forme en lui ajoutant un « moteur d'exécution » dédié au standard. Les approches les plus récentes [5][13], ayant cherché à proposer des langages opérationnalisables sur les plate-formes existantes sans les dénaturer n'ont pas réussi à fournir de solutions robustes pour éviter d'appauvrir les modèles conçus lors du passage à la mise en œuvre.

Pourtant, Moodle propose son propre format pour les quiz. Notre idée est de généraliser cette approche à tous les aspects de la conception pédagogique. Le projet GraphiT est basé sur l'idée que si les LMS étaient capables d'explicitement formellement leur « format » de scénarisation pédagogique, et par extension d'importer des modèles conformes à ce format, alors cela permettrait de spécifier de nombreux langages de conception, plus ou moins spécifiques à tel ou tel aspect de conception, à divers degrés d'abstraction selon les objectifs et le positionnement du langage développé.

2.2 Le projet GraphiT d'un point de vue IDM

L'objectif principal du projet est d'étudier les limites, en termes d'expressivité pédagogique, des langages de scénarisation pédagogique opérationnalisables sur une plate-forme donnée. En d'autres termes, il s'agit d'étudier la capacité d'expressivité pédagogique de langages de conception contraints par l'assurance première de l'opérationnalisation complète des modèles produits. La méthodologie du projet consiste à explorer en quoi l'Ingénierie Dirigée par les Modèles peut être utile et pertinente pour supporter la spécification de techniques et le développement d'outils permettre de traiter cet objectif.

Des travaux similaires récents ont adopté des approches plus conventionnelles. Par exemple, le système *Glue!* associé à l'éditeur *Glue!PS* [3], ainsi que l'éditeur *CADMOS* [13] ont adopté une approche indépendante de la plate-forme de mise en

œuvre tout en proposant une fonctionnalité de déploiement des scénarios sur Moodle [16]. Dans les deux cas, cette fonctionnalité repose sur le mécanisme d'import de cours de Moodle : le système génère un fichier similaire à ce que la plate-forme exporte lorsque l'utilisateur souhaite sauvegarder un cours. Les données du scénario sont alors mises en correspondance avec les données nécessaires à Moodle. Les approches comme celle-ci mènent à des modifications et des pertes sémantiques lors du processus de mise en correspondance des données, ceci est dû à la grande dissimilarité entre le langage de scénarisation et le modèle de données du LMS. D'autres travaux [1] montrent que les techniques de transformation de modèles issues de l'IDM peuvent être utilisées pour transformer un scénario centré concepteur vers un scénario centré plate-forme. Néanmoins, ils mettent aussi en avant la complexité de modélisation des transformations, la difficulté à décrire le méta-modèle du LMS, les pertes sémantiques suite à la transformation, et aussi la nécessité d'avoir une solution technique pour exploiter le modèle obtenu et le mettre en œuvre, sans intervention manuelle, dans le LMS.

Notre approche est différente : nous proposons une architecture directement dépendante du LMS considéré, afin de concevoir un langage de scénarisation tout en prenant en compte les problématiques de correspondances de métier de conception qui vont en découler. Nous ne cherchons pas à étendre les fonctionnalités du LMS avec de nouveaux *plugins* qui ajouteraient de nouveaux outils pédagogiques, ou de nouveaux « moteurs d'exécution ». Notre objectif est de supporter la spécification de scénarios pédagogiques en accord avec le métier de conception du LMS : ce métier est réifié au travers de certains outils/services, de certaines interfaces et paramétrages, du workflow sous-jacent, etc. Cela ne restreint pas pour autant l'expressivité des conceptions pédagogiques que l'on cherche à proposer à ce métier mais cela restreint cette expressivité au périmètre de ce que l'on saura traduire selon les termes du métier du LMS. La contrainte principale est de garantir l'opérationnalisation de scénarios sans perte sémantique. En revanche, notre approche a pour principal inconvénient que les langages et outils seront limités à un LMS, voire à une version spécifique de ce LMS. Dans le cadre de nos travaux, nous n'avons pas considérés les outils et services *tiers* qui pourraient être ajoutés aux versions *standards* des LMS.

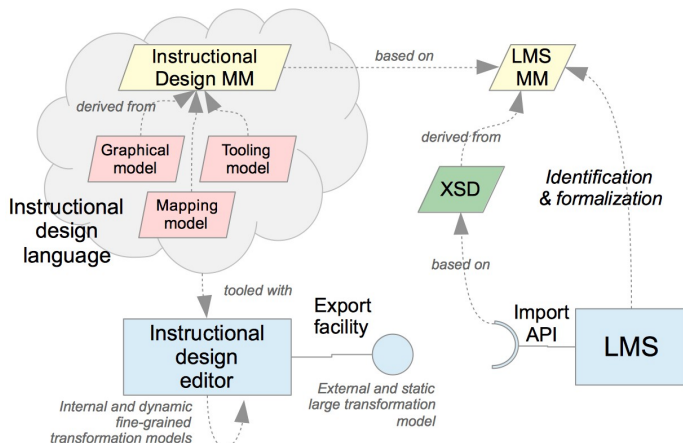


Fig. 1. Architecture globale du projet GraphiT

La figure 1 présente l'architecture générale de notre proposition. Le métier de conception de la plate-forme est dans un premier temps identifié et formalisé sous forme de méta-modèle. Ce dernier sert alors de base à l'élaboration d'un schéma XSD qui définit le format de fichier compatible avec l'API d'import à développer. Cette API

sera accessible aux enseignants-concepteurs à travers l'interface utilisateur de leur espace-cours sur la plate-forme. Le méta-modèle de la plate-forme sert également de base pour la conception du langage de scénarisation pédagogique graphique. Suivant les principes de l'approche *Domain Specific Modeling*, ce langage est composé d'une syntaxe abstraite, le méta-modèle, ainsi que d'une syntaxe concrète, la notation graphique. Selon la solution technique adoptée, cette syntaxe concrète est définie à l'aide d'un ou plusieurs modèles s'appuyant sur le méta-modèle du domaine afin de spécifier les différents outils présents dans la palette de l'éditeur ainsi que la représentation graphique de chacun des concepts. Un des apports de l'approche DSM est l'exploitation de générateurs de code, qui réduisent la tâche et donc le coût d'ingénierie pour le développement.

Les scénarios produits par l'éditeur doivent être conformes au méta-modèle initial de la plate-forme, afin d'être compatibles avec l'API d'import. Nous proposons d'exécuter deux phases de transformation de modèles. La première est exécutée durant l'utilisation de l'éditeur à une granularité fine : elle propose à l'utilisateur des correspondances en termes de fonctionnalités de la plateforme. La seconde transformation agit comme un système d'export, dans une phase post-conception du scénario. Elle permet la production d'un modèle conforme au méta-modèle de la plate-forme.

Des travaux précédents se sont intéressés à la formalisation du méta-modèle de la plate-forme [2] (expérimentée sur plusieurs plate-formes dont Moodle). L'API a été également développée pour Moodle 2.4. Cet article s'intéresse plus précisément à la syntaxe abstraite d'un langage de scénarisation dédié à Moodle.

2.3 Syntaxe abstraite du langage de scénarisation

La principale problématique étudiée dans ce projet est de s'abstraire suffisamment de la sémantique de conception pédagogique de la plate-forme afin de proposer aux enseignants des « briques » de conception pédagogique de plus haut niveau (au sens abstraction des détails techniques de mise en œuvre). Pour cela, il est nécessaire de s'affranchir des limites d'expressivité de la plate-forme afin de proposer aux enseignants des éléments de conception plus proches de leurs pratiques.

Nous avons mené un travail exploratoire [14] qui a permis d'identifier l'approche de méta-modélisation la plus adéquate à la mise en œuvre de cette abstraction : le méta-modèle du langage de conception pédagogique doit être élaboré comme une « extension » du méta-modèle de la plate-forme (meilleur rapport expressivité pédagogique / compatibilité avec la plate-forme). Il mettait également l'accent sur l'importance de la prise en compte des correspondances au moment de la conception même par extension du méta-modèle du langage.

2.4 Analyse des besoins de conception

Afin de connaître les orientations pédagogiques pour la réalisation de l'extension par méta-modélisation du métier de la plate-forme, nous avons collecté et analysé les besoins d'enseignants-concepteurs à propos de la plate-forme Moodle (plate-forme la plus répandue). Pour cela, nous avons mené à la fois une étude théorique, sur la littérature [7], une étude plus concrète au travers d'une enquête auprès de 203 enseignants et ingénieurs pédagogiques, ainsi que des entretiens individuels. L'enquête et les entretiens portaient sur les usages, la conception et l'appropriation de cette plate-forme. L'analyse des différentes sources nous a permis de recenser des pratiques mais aussi de recueillir des besoins fonctionnels pour l'éditeur de scénario à produire. Cette étude a également montré que les enseignants n'ont pas de pratiques

complexes communes à capturer, à cause de l'hétérogénéité de leurs niveaux d'expertises et de leurs approches pédagogiques. Néanmoins, ils ont en commun de réfléchir aux outils de la plate-forme qu'ils vont employer en fonction de l'usage pédagogique qu'ils visent. En effet, un grand nombre d'entre eux pointe le problème d'interface utilisateur de Moodle : le nombre de paramètres nécessaires à la mise en place d'une activité est trop élevé. Il leur semble nécessaire d'avoir une vue plus abstraite en terme d'usages pédagogiques afin de les guider dans le choix du bon outil et des bons paramètres pour mettre en place l'activité pédagogique qu'ils conçoivent.

En termes de besoins fonctionnels pour un outil-auteur graphique dédié à Moodle, les enseignants ont évoqué le besoin de ne pas être contraints dans leur méthode de scénarisation : le fait d'avoir accès à différents niveaux d'abstraction ne doit pas imposer une approche *top-down* (de la spécification vers l'implémentation). Ainsi, ils souhaitent pouvoir **mixer les concepts de spécification** (des briques pédagogiques abstraites) **et ceux d'implémentation** (les briques de base issues du métier de la plate-forme comme les outils, ressources, etc.). Un autre besoin identifié était de pouvoir visualiser une implémentation possible (traduite dans le *métier* de conception de Moodle) d'une brique pédagogique sans avoir à la spécifier eux-mêmes (**implémentation par défaut**), tout en ayant la possibilité de la modifier manuellement (**adaptation de l'implémentation**). Cette approche de conception devrait aider les concepteurs à s'approprier les concepts pédagogiques et à maîtriser leurs traductions en éléments de la plate-forme.

Un autre point soulevé concernait la possibilité de déclarer dans le scénario des informations qui n'ont pas d'implémentation directe sur la plate-forme ou qui ne seront pas visibles par les étudiants : indications pour une session en présentiel ; précisions sur des objectifs pédagogiques ; informations sur les étudiants, précisions sur les activités durant le déroulement de la session d'apprentissage etc. (**contrôle sur la visibilité de certaines informations**). Enfin, un besoin de conception que nous avons identifié est celui de pouvoir séquencer les activités au sein de **structures avancées** (séquences, activités au choix, etc.) pour lesquelles le contenu ne serait dévoilé qu'après la réalisation de l'activité précédente. Cette possibilité est offerte par Moodle dans sa version actuelle, mais nécessite un travail de paramétrage assez complexe (suivi d'achèvement et restriction d'accès) que les enseignants apprécieraient de ne pas avoir à faire manuellement.

3. Extension du méta-modèle de la plate-forme

En relation avec les besoins fonctionnels précédents, nous proposons de réaliser l'extension du méta-modèle de la plate-forme Moodle en ajoutant des concepts et relations formalisant une forme abstraction, de la sémantique de conception de la plate-forme, au niveau des **usages de type activité**. D'autres abstractions complémentaires sont à l'étude mais ne font pas l'objet de cet article : abstraction des concepts et relations permettant de spécifier et de référencer des objectifs/pré-requis, abstraction des concepts et relations permettant de définir et préciser des restrictions à des groupes/rôles, etc.

La Figure 2 illustre le résultat de cette formalisation (méta-modèle partiel pour raison de clarté). Nous utilisons le format *Ecore* afin de pouvoir exploiter les outillages *EMF* et *GMF* du projet Eclipse [10].

3.1 Activités pédagogiques fines

La première abstraction que nous proposons est l'activité pédagogique de granularité fin. Elle peut être définie comme *l'abstraction d'un paramétrage donné associé à un outil de la plate-forme dans le cadre d'un usage pédagogique spécifique*. A l'aide d'un seul « outil », par exemple un forum, il est possible de concevoir plusieurs usages pédagogiques, qui dépendent de la configuration de l'outil : forum de nouvelles aux étudiants, mise en place de groupes, activité de revue par les pairs, etc.

Du fait de la multiplicité des outils disponibles pour un même usage, il est nécessaire de trouver des critères discriminants qui permettent d'identifier la bonne implémentation. Ce type de brique nécessite un nom, une description (textuelle), et un ensemble de propriétés, correspondant aux critères précédents, que l'utilisateur pourra remplir au moment de la conception. Ces derniers seront utiles à l'identification de l'implémentation par défaut. Par exemple, une activité de *type échange entre étudiants* pourrait être implémentée à l'aide d'un chat ou d'un forum, le choix dépendant du caractère synchrone/asynchrone de la communication envisagée.

3.2 Activités pédagogiques à gros grains

Le deuxième niveau d'abstraction comprend deux types de briques. Nous proposons d'adapter et d'intégrer au langage des patrons pédagogiques issus de la littérature [4] [12] sous forme de briques de haut-niveau que l'utilisateur peut instancier tel quel et combiner avec d'autres activités afin de concevoir des sessions d'apprentissages basées sur des stratégies pédagogiques spécifiques : situations-problèmes, apprentissage basé sur l'exploration et l'expérimentation, jeux de rôles... L'intégration de ce type d'activités au langage permettrait d'encourager leur application et de favoriser l'emploi de diverses approches pédagogiques. Ce type d'activités peut être implémenté à l'aide d'une combinaison de structures, d'activités plus fines, ou directement à l'aide d'outils de la plate-forme.

Afin de faciliter l'assemblage des différentes activités et ressources, nous proposons d'intégrer au langage un ensemble d'éléments de structures (séquence, sélection, activités conditionnelles etc.). Ces briques peuvent contenir d'autres briques du même niveau ou de niveaux inférieurs. De nombreux langages de modélisation proposent ce genre d'éléments. Dans le cas de Moodle, ces structures seront concrètement traduites en une combinaison de labels (ou étiquettes), précisant les instructions éventuelles concernant la structure (pour le choix par exemple) et de décalages (indentation du contenu) afin de représenter l'imbrication des éléments compris dans la structure. Des règles d'achèvement et de restrictions seront également produites, lorsque nécessaires, pour les éléments composant ces structures.

3.3 Une syntaxe abstraite à quatre niveaux

La syntaxe abstraite que nous proposons pour le langage de scénarisation centré sur Moodle est composée de quatre niveaux, illustrés dans la Figure 2. Le premier niveau correspond à celui du méta-modèle de Moodle (représentation partielle dans la figure 2). Les éléments de niveau 1, limités aux outils Moodle peuvent être directement instanciés dans le scénario par le concepteur qui devra ensuite remplir les paramètres associés. Ce niveau comprend également les concepts de *cours* et de *sections*, indispensables à l'opérationnalisation du scénario. La transformation post-conception prendra en charge leur création et leur ordonnancement afin de produire un scénario cohérent et compatible avec l'API que nous avons développée.

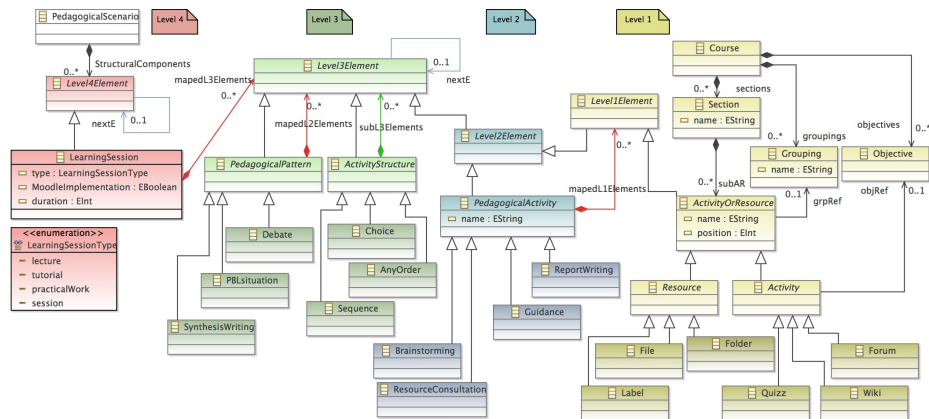


Fig. 2. Syntaxe abstraite du langage de scénarisation intégrant le méta-modèle de Moodle.

Le niveau 2 comprend les premières briques abstraites du langage : les activités pédagogiques fines. Elles sont composées d'éléments de niveau 1 (outils et ressources de Moodle) qui seront dynamiquement déduits, instanciés et ajoutés pendant l'utilisation de l'éditeur lorsque les propriétés des éléments de niveau 2 seront modifiées (implémentation par défaut). Ces éléments de niveau 1 sont modifiables par la suite lors de la conception. Le niveau 3 intègre le second type d'activités pédagogiques, à plus gros grain, ainsi que les structures d'activité. Le premier type se composera, dynamiquement d'éléments de niveaux 1 à 3 de manière similaire aux activités pédagogiques fines. Enfin, le quatrième niveau sert de contexte, il contient la structure globale de la session d'apprentissage.

Les éléments du niveau 4 seront directement traduits *en sections* sur Moodle. Les relations de composition en rouge indiquent que le contenu ne sera pas visuellement représenté comme dans un conteneur, mais sera visible dans un sous-diagramme. La relation en vert, au contraire indique une représentation graphique sous forme de contenant/contenu dans un même diagramme. Enfin, la relation *nextE* permet de séquencer les différents éléments d'un même diagramme. Ces relations sont nécessaires dans la formalisation du méta-modèle car les différents outils *Domain Specific Modeling* s'appuient sur des techniques de spécification de la syntaxe concrète par dérivation d'éléments de la syntaxe abstraite.

L'éditeur associé à ce langage présentera dans un premier temps une palette d'outils correspondant au niveau 4. Ces éléments sont nécessaires afin de créer les *sections* dans le cours sur Moodle. Des sessions n'exploitant pas Moodle peuvent tout de même être décrites si l'enseignant le juge nécessaire, afin d'avoir une vue globale de son module par exemple. Lorsque l'utilisateur double-clique sur un élément de niveau 4, un sous-diagramme s'ouvre, permettant alors d'ajouter des éléments des niveaux 1 à 3 disponibles dans la palette. Ainsi, le concepteur a toute liberté de choisir son approche et le niveau de description désiré lors de la sélection des différents éléments. A l'exception des structures, tous les autres éléments des niveaux 2 et 3 ouvriront un sous-diagramme présentant une implémentation par défaut correspondant à l'élément parent et à son paramétrage. Cette implémentation peut alors être modifiée librement.

4. Illustration par l'exemple

4.1 Exemple de scénario

Nous proposons d'illustrer notre approche par un exemple de formalisation d'un scénario pédagogique simple mais représentatif, à destination de Moodle. La description textuelle de ce scénario sera accompagnée d'une capture d'écran (figure 3) illustrant sa formalisation à l'aide du méta-modèle que nous proposons.

Le scénario exemple est composé de deux sessions d'apprentissage : la première est un cours magistral, pour lequel l'enseignant ne souhaite proposer qu'un accès à des ressources qui constituent le support du cours. Cette activité *Resource Consultation* (consultation de ressources) possède plusieurs paramètres: *quantity* fixée à *one* et *location* à *local*. Cette combinaison de paramètres conduira le système d'implémentation par défaut à proposer un outil Moodle de type *File* (fichier).

La deuxième est une session de type travaux pratiques, en présentiel dans une salle équipée d'ordinateurs. L'enseignant souhaite exploiter Moodle pour mettre en place une activité de type *Synthesis Writing* (rédaction d'une synthèse). Cette activité à gros grain est automatiquement implémentée à l'aide d'une séquence contenant quatre sous-éléments. Le premier est une autre activité de consultation de ressources, avec cette fois-ci un paramètre *quantity* fixé à *many* (ce qui produira un outil *Folder*, dossier en français). Le suivant est une activité de type *Brainstorming*, dont le paramètre *orientation* fixé à *discussion* mènera le système d'implémentation par défaut à suggérer un outil *Forum* de Moodle. Le troisième sous-élément de cette séquence est une activité pédagogique de type *Report Writing* (rédaction d'un rapport) qui sera implémenté à l'aide d'un *Wiki*, du fait du paramètre *collaborative* de l'activité fixé à *true* (vrai). Le quatrième et dernier élément est de type *Guidance* (indication) qui rappellera à l'enseignant de corriger la synthèse produite dans le Wiki. Le paramètre *public* fixé à *tutor* provoquera la création d'un élément *Label* (étiquette) ayant la propriété de ne pas être visible par les étudiants (*visible=false*).

L'enseignant peut à tout moment changer la valeur des paramètres de chacun des éléments, ce qui peut mener à d'autres implémentations par défaut. Il peut également modifier l'implémentation en supprimant, ajoutant et réordonnant des éléments. La figure 3 propose une vue générale du scénario intégrant les implémentations par défaut produites en fonctions des valeurs des différents paramètres (non-représentés).

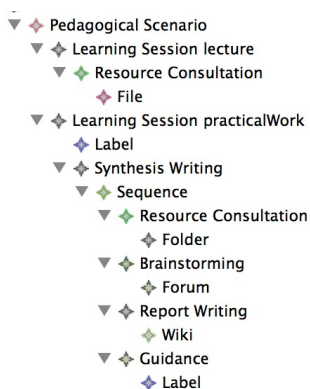


Fig. 3. Exemple de scénario pédagogique composé d'éléments des 4 niveaux

4.2 Technique de tissage de modèle pour les correspondances par défaut

Conformément à notre approche dirigée par les modèles, nous exploitons les transformations de modèles pour mettre en œuvre le mécanisme d'implémentation par défaut. Ces transformations peuvent être exécutées à la demande, durant le processus de modélisation du scénario, afin d'y ajouter automatiquement des éléments d'implémentations qui seront représentés dans des sous-diagrammes. Ces transformations de modèles, sont coûteuses à produire, en raison notamment au nombre d'éléments à implémenter et à la complexité des règles de correspondance. Afin de pallier à ce problème nous proposons d'utiliser le tissage de modèles.

D'un point de vue pratique, le processus à mettre en place est le suivant : un ingénieur, sur les indications d'un ou des représentants d'une communauté d'enseignants, va modéliser (à l'aide d'un éditeur de tissage) les règles de correspondance entre les éléments du langage de scénarisation et leurs implémentations. Sur ce modèle de tissage, va être appliquée une transformation de haut niveau (*High Order Transformation*) afin de générer les transformations de modèles qui seront concrètement exécutées dans l'éditeur de scénario.

Les modèles de tissages sont conformes à un méta-modèle de tissage que nous avons conçu et développé à l'aide du projet Eclipse Epsilon [9]. Cette contribution est détaillée dans [15]. La figure 4 illustre un modèle de tissage correspondant à l'exemple de la section 4.1.

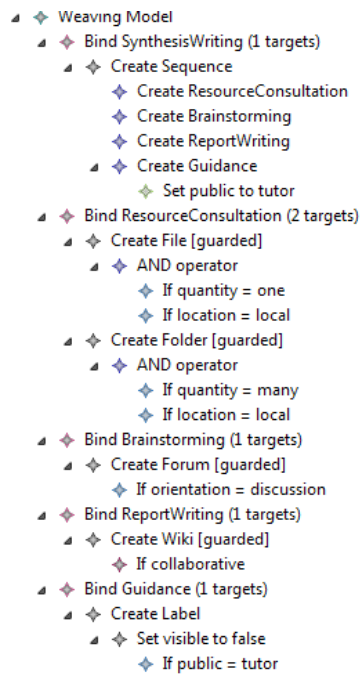


Fig. 4. Exemple de modèle de tissage (en relation avec l'exemple de la figure 3.

5. Conclusion

Cet article propose une approche centrée plate-forme afin d'augmenter l'expressivité pédagogique du métier de conception des plate-formes de formation. Nous proposons de construire des scénarios pédagogiques à l'aide de briques de haut-niveau basées sur l'abstraction des usages récurrents des outils et paramétrages de la plate-forme. Sur l'exemple de la plate-forme Moodle, nous avons présenté et illustré la syntaxe abstraite d'un langage de scénarisation pédagogique à quatre niveaux spécifique au LMS. Cette abstraction du métier de la plate-forme, semble être une approche pertinente afin de développer une nouvelle génération de langages de scénarisation centrés plate-forme qui permettra aux enseignants de concevoir des scénarios pédagogiquement riches et opérationnalisables.

L'abstraction d'activité pédagogique que nous avons proposée n'est qu'un premier exemple d'abstraction du métier de conception de la plateforme. Nous travaillons actuellement à compléter la syntaxe abstraite afin de permettre l'ajout d'objectifs pédagogiques aux différents éléments. Ces objectifs pourront être traduits en *objectifs* Moodle, rattachés au cours global et référencés par les éléments de niveau 1. Il reste également à ajouter une abstraction se rapprochant du concept de rôle ou sujet, qui pourront se traduire en *groupes* et *groupements* de Moodle. Ces éléments seront à mettre en relation avec les abstraction d'activités et d'objectifs.

Notre approche centrée plate-forme confère un caractère spécifique aux modèles de tissage, pour capturer les correspondances, ainsi qu'à la formalisation de l'extension du méta-modèle de conception de la plate-forme. Toutefois, le langage de tissage est exploitable pour d'autres plate-formes. De même les propositions de briques abstraites que nous avons définies dans cet article sont potentiellement réutilisables, mais non encore vérifiées, pour d'autres plate-formes.

Le méta-modèle *étendu* que nous proposons est exploité afin de définir la syntaxe concrète de notre langage (notation graphique) et de développer l'outil-auteur. Nous utilisons pour cela les *frameworks* EMF et SIRIUS. Un prototype opérationnel a été produit afin de mettre à l'essai nos propositions. Ce prototype intègre également les mécanismes internes de transformation de modèles automatique sur la base des modèles de tissage que nous avons présentés. Le développement se poursuit actuellement afin de dépasser le cadre du prototypage et de produire une première version d'éditeur exploitable dans le cadre des expérimentations prévus dans les prochains mois dans le cadre du projet GraphiT.

Aussi, les scénarios conformes au méta-modèle que nous proposons ne sont pas directement produits au format compatible avec le LMS, néanmoins, nous travaillons au développement d'une transformation de modèle « globale » qui agira comme une fonction d'export des scénarios depuis l'outil-auteur.

Remerciements. Cet article présente des travaux de recherche menés dans le cadre du projet GraphiT, financé par l'Agence Nationale pour la Recherche (ANR). Nous tenons à remercier toutes les personnes qui ont accepté de participer à l'enquête et aux entretiens.

Références bibliographiques

1. Abdallah, F., Toffolon, C., Warin, B.: Models transformation to implement a Project-Based Collaborative Learning (PBCL) Scenario : A Moodle case study. 8th IEEE International Conference on Advanced Learning Technologies, IEEE Computer Society, Washington DC (2008) 639–643.

2. Abedmouleh A., Oubahssi L., Laforcade P., Choquet C.: An analysis process for identifying and formalizing LMS instructional language. 7th International Conference on Software Paradigm Trends, ScitePress (2008) 218–223
3. Alario-Hoyos, C., Munoz-Cristobal, J.A., Prieto-Santos, L.P., Bote-Lorenzo, M.L., Asensio-Perez, J.I., Gomez-Sanchez, E., Vega-Gorgojo, G., Dimitriadis, Y.: GLUE! - GLUE!-PS: An approach to deploy non-trivial collaborative learning situations that require the integration of external tools in VLEs. 1st Moodle Research Conference, Moodle Research Conference, Heraklion (2012) 77–85
4. Bergin, J., Eckstein, J., Manns, M.L., Sharp, H., Chandler, J., Marquardt, K., Wallingford, E., Sipos, M., Völter, M.: Pedagogical Patterns: Advice For Educators. Joseph Bergin Software Tools (2012)
5. Berggren, A., Burgos, D., Fontana, J.M., Hinkelman, D., Hung, V., Hursh, A., Tielemans, G.: Practical and Pedagogical Issues for Teacher Adoption of IMS Learning Design Standards in Moodle LMS. Journal of Interactive Media in Education, special issue: Advances in Learning Design (2005)
6. Burgos, D., Tattersall, C., Dougiamas M., Vogten, H., Koper, R.: A First Step Mapping IMS Learning Design and Moodle. Journal of universal computer science 13, J.UCS Consortium, Graz (2007) 924–931
7. Conole, G., Dyke, M., Oliver, M., Seale, J.: Mapping pedagogy and tools for effective learning design. Computers & Education 4, (2004) 17–33
8. Dougiamas, M., Taylor, P.: Moodle: Using Learning Communities to Create an Open Source Course Management System. World Conference on Educational Multimedia, Hypermedia and Telecommunications, Association for the Advancement of Computing in Education, Waynesville (2003) 171 178
9. Eclipse Epsilon Project Official Website, <http://www.eclipse.org/epsilon/>
10. Eclipse Modeling Project Official Website, <http://www.eclipse.org/modeling/>
11. Garrison, D.R., Kanuka, H.: Blended learning: Uncovering its transformative potential in higher education. The Internet and Higher Education 7, (2004) 95–105
12. Heathcote, Elizabeth A.: Learning design templates - a pedagogical just-in-time support tool. In: Minshull, G., Mole, J. (eds.) Designing for Learning. JISC Development Group (2006) 19–26
13. Katsamani, M., Retalis, S., Boloudakis, M.: Designing a Moodle course with the CADMOS learning design tool. Educational Media International 49, (2012) 317–331
14. Loiseau, E., Laforcade, P.: Specification of learning management system-centered graphical instructional design languages - A DSM experimentation about the Moodle platform. 8th International Joint Conference on Software Technologies, Scitepress (2013) 504–511
15. Loiseau, E., Laforcade, P., Iksal, I.: Model Weaving and Pedagogy – Mapping Abstraction Levels in Instructional Design Languages. 9th International Joint Conference on Software Technologies, Scitepress (2014)
16. Moodle Official Website <https://moodle.org>
17. Ormrod, J.E.: Human Learning. Pearson College Division, Upper Saddle River (2011)
18. Podvin H., Laforcade P.: Analyse des pratiques et des besoins de conception pédagogique centrés plateformes de formation. Livrable D3-4 projet GraphiT (2014). Accessible à http://www-lium.univ-lemans.fr/~laforcad/graphit/?page_id=166