
Spécification de langages de scénarisation graphiques centrés sur les plateformes de formation à distance

Etude et expérimentation d'approches DSM pour Moodle

Esteban Loiseau*, Pierre Laforcade*

* LIUM, Université du Maine
Avenue Olivier Messiaen
72085 Le Mans cedex 9
esteban.loiseau@univ-lemans.fr
pierre.laforcade@univ-lemans.fr

RÉSUMÉ. Cet article présente un travail de recherche exploratoire de six mois ayant porté sur la spécification de langages de scénarisation pédagogiques. Les langages graphiques visés doivent répondre à la double exigence de proposer une expressivité dirigée vers des besoins spécifiques à des enseignants-concepteurs et de proposer un niveau de formalisation des scénarios/modèles suffisant pour garantir leur opérationnalisation complète sur une plateforme de formation à distance cible. La modélisation centrée métier, ou DSM pour Domain-Specific Modeling, sert de cadre méthodologique théorique et pratique pour la spécification de ces langages et leur outillage. Le périmètre de ce travail exploratoire est centré sur la plateforme Moodle. Il a impliqué une équipe d'ingénieurs pédagogiques de l'Université du Maine pour la définition des besoins et l'analyse des travaux. Trois approches DSM différentes ont été expérimentées. L'analyse finale de leurs mises à l'essai a permis d'orienter la suite des travaux de recherche qui prennent place actuellement dans le cadre du projet ANR GraphiT.

MOTS-CLÉS: scénarisation pédagogique, plateformes de formation à distance, domain-specific modeling, Moodle, ingénierie dirigée par les modèles

1. Introduction

De nombreuses institutions académiques mettent à disposition de leurs usagers, enseignants comme étudiants, des plateformes de formation à distance. Pourtant, leur usage n'est pas limité à la seule mise en place de formations ou modules destinés à des étudiants à distance. Elles sont aussi disponibles et utilisables par tout enseignant souhaitant les utiliser pour mettre en place diverses activités pédagogiques qu'elles soient complémentaires aux séances en présentiel ou bien qu'elles soient réalisées pour un face-à-face outillé. Leurs usages varient alors de la simple mise à disposition de ressources au support complexe d'un projet tuteuré sur plusieurs séances selon une pédagogie active par projet avec forts usages des outils de communication et de travail collaboratif proposés par la plateforme. Pour mettre en œuvre ces activités les enseignants doivent développer une bonne maîtrise, voire une forte expertise, de la plateforme (types d'outils, usages possibles de ces outils, etc.) souvent proposé par le biais de formations internes à l'institution. En revanche, en ce qui concerne la *scénarisation* de la situation d'apprentissage qui sera supportée par la plateforme (séquencement des activités, utilisation des groupes, relation avec les objectifs pédagogiques et didactiques, relation avec des approches pédagogiques spécifiques, etc.), peu ou pas d'accompagnement et de formations ne sont proposés. La grande disparité des approches pédagogiques, l'inclusion de la scénarisation du dispositif comme une sous-partie d'une conception pédagogique plus large (incluant par exemple les activités en présentiel), le manque d'outils, méthodes et autres outils-support dédiés à la plateforme, tendent à laisser l'enseignant développer seul en amont ses propres pratiques de conception (instrumentée ou pas). Il est également rare que l'enseignant se soit tourné, seul, vers l'exploitation de langages/éditeurs de scénarisation pédagogique, standardisés comme spécifiques, si ceux-ci ne sont pas encouragés et guidés par l'institution.

Dans de tels contextes, il serait pertinent d'aider les enseignants à utiliser, à s'appropriier et à concevoir spécifiquement pour ces plateformes. En plus d'améliorer leur connaissance de l'environnement mis à leur disposition, ceci encouragerait une approche réflexive des pratiques de conception, voire la formation de communautés de pratiques. Cette approche de conception centrée plateformes est à contre-courant des approches traditionnelles où le choix de la plateforme n'intervient que lors de la mise en œuvre du dispositif. Cette approche est celle choisie par le projet ANR GraphiT, présenté dans la section 2 prochaine. Cet article présente également plus spécifiquement les résultats d'une action de recherche exploratoire de 6 mois ayant pris place dans ce projet. L'objectif consistait à explorer des pistes d'application de techniques issues du Domain-Specific Modeling (DSM) pour la spécification de langages pédagogiques graphiques, et leurs outils-auteurs, centrés plateforme. Nous précisons en section 2 le périmètre de ce travail et la méthodologie que nous avons suivie. La section 3 porte sur l'analyse d'un premier niveau de besoins en termes d'expressivité pour les scénarios visés et en terme d'abstraction du métier de la plateforme : la brique de base est l'activité pédagogique au sens usages pédagogiques possibles des outils/services de la plateforme. La section 3 analyse également différentes correspondances « manuelles » possibles pour ces briques de base du langage. La section 4 étudie le cadre DSM de nos travaux d'un point de vue théorique pour identifier trois approches d'application pour répondre aux exigences visées. La section 5 détaille alors la réalisation concrète de ces trois approches dans un environnement DSM outillé. La section 6 est dédiée à la comparaison et à l'analyse des trois résultats.

2. Contexte et positionnement : le projet GraphiT

2.1. Présentation générale du projet

Le projet GraphiT [GRAPHIT 13] est un projet ANR JCJC débuté en février 2012. Son principal objectif est d'étudier les possibilités et limites en termes d'expressivité pédagogique de langages de scénarisation *opérationnalisables*, i.e. dont les scénarios produits peuvent être informatiquement exploités pour la mise en œuvre sur des plateformes de formation à distance. A travers les langages et éditeurs de scénarisation visuels visés par le projet, l'objectif à long terme est d'encourager et d'améliorer l'utilisation de ces plateformes. Pour cela le projet s'appuie sur les hypothèses suivantes : 1/ les plateformes embarquent implicitement leur propre *paradigme de conception pédagogique*, 2/ l'explicitation de ce *métier* permettra d'envisager son exploitation pour, entre autres, la traduction et l'opérationnalisation des scénarios pédagogiques. Ces deux hypothèses présupposent également que ce *métier* évolue suffisamment peu dans le temps (versions, extensions, intégration d'outils externes) pour garantir une pérennité suffisante des artefacts de conception pédagogique qui l'exploiteront. L'approche scientifique du projet est illustrée dans la figure 1. Pour chacune des plateformes étudiées, nous proposons a/ d'explicitier son métier de conception, b/ de l'étendre par une API chargée de l'import/export adaptatif de scénarios, c/ d'explicitier des pratiques/besoins de conception pour cette plateforme, d/ de spécifier et outiller des langages de scénarisation centrés sur la plateforme mais tendus vers les pratiques/besoins à prendre en compte. L'enjeu majeur du projet réside dans l'abstraction du « métier » de conception de la plateforme. En effet, il s'agit de dépasser les limites inhérentes à l'expressivité pédagogique concrète de la plateforme afin de proposer des mécanismes de scénarisation plus proches de l'expressivité des pratiques de conception existantes ou souhaitées. La traduction de ces scénarios en termes compatibles avec le métier de conception de la plateforme dirigera l'abstraction. La problématique scientifique sous-jacente est d'identifier la relation forte entre expressivité pédagogique et maîtrise de l'opérationnalisation. Le projet exploite un double cadre méthodologique : ingénierie des besoins et patrons pédagogiques pour la capture et l'explicitation des pratiques de conception, et ingénierie dirigée par les modèles et modélisation spécifique à un métier pour l'explicitation du métier de conception des plateformes et son exploitation pour la spécification de langages/éditeurs de scénarisation. L'explicitation des pratiques de conception (figure 1 gauche) et celle du métier des plateformes (figure 1 droite) ont donné lieu à des premiers résultats [CLAYER et al. 12] [ABEDMOULEH & LAFORCADE 12]. Le cœur du projet (cadre central) a fait pour le moment l'objet des premiers travaux de recherche exploratoire, relatés dans cet article, afin de guider les pistes qu'il faudra étudier et développer par la suite.

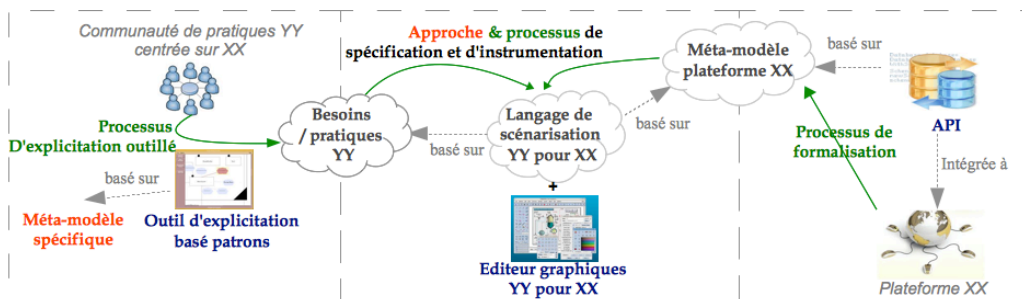


Figure 1. Contributions visées par le projet GraphiT (outils en bleu, processus/méthodes en vert, modèles/techniques en orange, mise en œuvre spécifique en gris) et périmètre du travail exploratoire de cette publication.

2.2. Les langages de scénarisation pédagogique visés

Le projet vise à spécifier et outiller des langages de scénarisation (1) *graphiques* (scénarios représentés par un formalisme visuel graphique) et (2) *opérationnalisables* (scénarios produits manipulables par les API - cf. figure 1). Le deuxième critère correspond au concept de *binding* visé par les EML (*Educational Modeling Language*) [KOPER & MANDERVELD 04]. Ces EML visent généralement à être *génériques* au sens où l'expressivité pédagogique qu'ils proposent est *indépendante* de celle de dispositifs spécifiques et *couvrante* au sens des diverses pratiques et approches de conception compatibles. Les EML misent sur la formalisation des scénarios afin qu'ils soient interprétables pour être opérationnalisés sur des dispositifs compatibles. L'expérience du standard *de facto* IMS-LD a montré que l'adaptation de plateformes de formation comme Moodle nécessitait une lourde ingénierie et une difficulté de mise en correspondance avec les fonctionnalités offertes par la plateforme [BURGOS et al. 07]. Les langages visés par le projet seront spécifiques aux plateformes. Ils nécessiteront tout de même un travail d'ingénierie, moins coûteux, pour le développement des API d'import/export adaptables. Le critère *graphique* correspond davantage aux VIDL (*Visual Instructional Design Language*) [BOTTURI & STUBBS 08]. Ces langages se focalisent davantage sur l'expressivité des scénarios et leur formalisation visuelle destinée en premier lieu à une interprétation humaine par les praticiens. Contrairement aux EMLs, ces langages cherchent à favoriser la compréhension, la réflexion, individuelle ou collective autour de l'activité même de conception pédagogique. Ils se focalisent alors sur des pratiques de conception spécifiques et moins sur un éventuel *binding* des scénarios produits : les scénarios sont généralement dans des formats propriétaires non destinés à une interprétation machine pour une opérationnalisation ; ils sont aussi indépendants de tout dispositif. Dans de rares cas [DODERO et al. 10] des exports partiels vers des standards comme IMS-LD sont proposés, ou bien vers les fonctionnalités offertes par les plateformes mais par des mécanismes de mise en œuvre trop génériques pour prendre en compte la totalité du scénario initial. Selon le critère d'*élaboration* de la classification des VIDL [BOTTURI et al. 06], les langages de scénarisation graphiques visés par le projet s'orientent vers un niveau intermédiaire entre les VIDL à niveau *implémentation* (conception exprimée selon ce que la plateforme permet) et ceux à niveau *spécification* (conception exprimée selon les pratiques de conception des enseignants). La difficulté consistera à étudier jusqu'à quel niveau d'expressivité visuelle et sémantique proche des pratiques de conception des enseignants il est possible de s'élever sur la base d'un formalisme et d'une sémantique spécifiques à une plateforme.

2.3. Le Domain-Specific Modeling comme cadre conceptuel théorique et technique

Le Domain Specific Modeling (DSM) [KELLY & TOLVANEN 08] est une méthode de développement logiciel qui consiste à définir un langage de modélisation spécifique à un domaine métier (en opposition aux langages généralistes type UML). Cette méthodologie vise à réduire les coûts de développement en automatisant le processus de conception du logiciel qui exploitera les modèles produits par ce langage. Les outils DSM reposent sur la génération automatique de code à partir d'un méta-modèle du domaine et divers paramétrages. Les usages potentiels et défis à relever concernant l'application du DSM au domaine de la scénarisation pédagogique ont été discutés dans [LAFORCADE 10]. Le projet GraphiT explore l'exploitation du DSM comme cadre conceptuel *théorique*, pour la spécification des langages, et *technique* pour le développement de l'éditeur graphique. En effet, la plupart des outils DSM guident le développement de cet éditeur par des processus de modélisation dérivés du langage métier initial, réduisant ainsi les coûts de développement. Dans le cadre du projet GraphiT une première application du DSM a été réalisée en proposant un langage de conception spécifique à Moodle [ABEDMOULEH & LAFORCADE 12]. Ce langage exploite directement le méta-modèle issu du processus

d'identification et propose une notation concrète graphique similaire à celle visible sur la plateforme : pas de réelle valeur ajoutée à la conception *intra*-Moodle mais a permis d'expérimenter la pertinence de l'approche outillée DSM.

2.4. Objectifs et périmètre du travail exploratoire

Le travail exploratoire entrepris avait pour objectif principal d'explorer les différentes approches DSM possibles pour la spécification de langages à un premier niveau d'abstraction du métier de conception de la plateforme. Il se situe dans le cadre du périmètre central de la figure 1. Pour cela nous avons fixé l'étude à une plateforme de formation donnée et à un premier besoin en terme de pratiques pédagogiques à intégrer par abstraction du métier de la plateforme (défini dans la section suivante). Pour ce dernier ainsi que pour l'API chargée de l'opérationnalisation, nous nous sommes appuyés sur les premiers résultats obtenus concernant cette plateforme [ABEDMOULEH 12]. Toutefois, afin de ne pas être dépendant de la subjectivité intrinsèque à tout (méta-)modèle, nos résultats mettront l'accent sur la sémantique des concepts, relations et propriétés du métier de conception de la plateforme et non sur leurs représentations syntaxiques, quelque peu contraintes, utilisées pour leurs formalisations sous forme de méta-modèles.

3. Analyse des besoins

3.1. La plateforme Moodle

Le choix de la plateforme de formation à distance s'est porté sur Moodle [MEHRABI & MASOUMEH 12]. Elle est actuellement déployée et proposée aux enseignants de l'Université du Maine pour être exploitée en relation avec les formations présentiellees. Moodle est une plateforme de formation très répandue dans les institutions publiques et incontournable dans le domaine de la formation à distance [MOODLE 13]. Elle possède une importante et active communauté d'utilisateurs. Il s'agit d'une application web PHP open-source et modulaire pour laquelle il est possible d'intégrer de nouvelles fonctionnalités. Moodle revendique une approche pédagogique socio-constructiviste [DOUGIAMAS & TAYLOR 03]. Concrètement, elle propose des ressources mais aussi des activités au sens outils et services de collaboration, de travail en groupe, etc. (forum, chat, wiki).

3.2. Un besoin d'activités pédagogiques

Afin d'analyser leurs besoins, nous avons rencontré des intervenants du *Pôle Ressources Numériques* (PRN) de l'Université du Maine. Le PRN prend en charge la gestion des différentes instances de Moodle et propose des formations aux enseignants. Plus récemment, le PRN sensibilise et forme les enseignants à la scénarisation pédagogique. Il assure également l'opérationnalisation manuelle des scénarios pédagogiques fournis par des enseignants-concepteurs de formations à distance. Le personnel du PRN possède donc des compétences et une expérience importante quant à l'utilisation de la plateforme, et constitue un premier interlocuteur pertinent pour notre travail exploratoire. Le PRN nous a fourni les supports de leur formation à destination des enseignants concepteurs, ainsi qu'un accès à des cours en ligne spécifiques. L'étude de ces diverses sources nous a permis d'identifier une première abstraction des activités/outils de la plate-forme. En effet, pour un même outil, par exemple un forum, on peut distinguer plusieurs *usages pédagogiques* selon le paramétrage réalisé. Nous avons alors constitué une liste représentative de ces *activités pédagogiques*, complétées également par des usages recensés dans la littérature du domaine [CONOLE et al. 04]. Enfin, nous avons relevé des éléments structurels récurrents (activités aux choix, en

séquence, conditionnelle, etc.) des VIDL et EML afin de compléter les éléments cibles à prendre en compte pour l'élaboration des langages de scénarisation visés.

3.3. Correspondances manuelles

Afin d'établir des spécifications pour la formalisation des éléments du langage de scénarisation à réaliser sur la base du métier de conception de Moodle, nous avons établi des mises en correspondance représentatives *manuelles* avec les outils proposés par Moodle. Les activités pédagogiques d'évaluation tel que l'*auto-évaluation*, l'*évaluation sommative* ou l'*évaluation diagnostique* reposent toutes sur la fonctionnalité de quizz (ou test). Les modalités de réponse au test distinguent ces trois usages par des paramétrages différents de l'outil (exemples : le nombre de tentatives autorisées, l'activation du mode « adaptatif »). Certains usages peuvent donc être retranscrits de plusieurs façons sur la plate-forme. Nous proposons alors des critères, valorisés par l'utilisateur lors de la conception du scénario, qui pour chaque instance de l'activité définiront sa traduction sur la plateforme. Ainsi, l'activité *Débat* peut se dérouler de manière synchrone, avec un *chat*, ou asynchrone, avec un forum. L'activité *rédaction d'un compte-rendu* est un exemple plus complexe à trois critères (collaboratif ou individuel, travail en ou hors ligne et rédaction itérative ou définitive) exploiter 4 outils (paramétrés différemment selon les 8 cas) de Moodle : le *texte en ligne*, le *wiki*, le *journal* ou le *dépôt de fichier*. Pour implémenter des éléments structurant le scénario pédagogique, nous avons dû passer outre les limitations de la plate-forme et chercher à détourner certains éléments de Moodle : elle ne propose pas par défaut (dans la version 1.9 déployée à l'Université du Maine) de fonctionnalité permettant de contrôler le déroulement du cours. Ainsi, pour indiquer des structures d'activités de type *séquence* ou *choix*, il est possible d'exploiter les *étiquettes* pour donner *a minima* une précision textuelle des modalités. Aussi, Moodle ne propose pas de branchement conditionnel avant la version 2.3 (*conditional activities*) : possibilité d'utiliser les *groupes* et *groupements* pour restreindre l'affichage d'une activité à un groupe donné et d'attribuer un groupement à chaque activité alternative. La condition à tester ne peut pas être vérifiée de manière automatique : une *étiquette, non visible* par les étudiants, pourrait rappeler à l'enseignant comment et quand réaliser le remplissage des groupes prédéfinis.

4. Conception des trois approches

Les langages de scénarisation graphiques visés sont des cas particuliers de langages de modélisation au sens DSM. Ceux-ci sont définis comme étant le tuple $\{AS, CS^*, M^{*ac}, SD, Mas\}$ où AS est la syntaxe abstraite, CS^* est la (les) syntaxe(s) concrète(s), M^{*ac} est l'ensemble des mappings de la syntaxe abstraite vers la (les) syntaxe(s) concrète(s), SD est le domaine sémantique et Mas est le mapping de la syntaxe abstraite vers le domaine sémantique. La syntaxe abstraite (AS) d'un langage de modélisation exprime, de manière structurelle, l'ensemble de ses concepts et leurs relations. Elle est concrètement représentée et formalisée par un méta-modèle. Ce méta-modèle assure aussi la spécification formelle des futurs modèles (*binding*). Les langages de scénarisation graphiques visés doivent répondre à une double exigence. Ils doivent tout d'abord proposer une notation graphique cohérente en relation avec l'interprétation humaine souhaitée pour les futurs scénarios : syntaxe concrète graphique spécifique. Ils doivent également garantir que le format de persistance pour les scénarios soit conforme au méta-modèle de Moodle, ceci afin de garantir qu'ils pourront être pris en charge par l'API spécifique déjà développée. Une première approche consiste donc à fixer la syntaxe abstraite du langage de scénarisation comme correspondant directement au méta-modèle de Moodle (considéré comme préexistant dans le périmètre de ce travail exploratoire) afin d'assurer prioritairement l'objectif d'opérationnalisation des futures modèles. La syntaxe concrète sera donc dérivée du méta-modèle de Moodle. Cette approche devra donc chercher à dépasser les limites

d'expressivité des modèles uniquement par le biais de l'expressivité issue des éléments visuels et graphiques. La seconde approche consiste à étendre la syntaxe abstraite, initialement correspondante à celle du méta-modèle de Moodle, par de nouveaux éléments syntaxiques dont la sémantique souhaitée (en relation avec les pratiques de conception visées) n'est pas considérée comme couverte par la syntaxe abstraite initiale. Ces extensions ne garantissant plus la conformité totale des futurs modèles au méta-modèle initial, cette approche devra étudier les moyens de retrouver cette conformité, dans un *temps* d'utilisation de l'outil-auteur (i.e. les modèles ne devront pas nécessiter un traitement *post*-conception), ceci afin d'étudier le potentiel des outillage DSM en la matière. La troisième approche s'oppose à la première en priorisant la définition d'un langage dont la sémantique couvre parfaitement les pratiques de conception visées : la syntaxe abstraite est spécifiée sans relation avec le méta-modèle de Moodle. Cette approche aboutira toutefois à des modèles globalement non conformes et donc non opérationnalisables. L'enjeu consistera alors à obtenir un modèle conforme par le biais de mécanismes de transformations.

5. Mise en œuvre outillée

5.1. Choix de l'outillage DSM

Nous avons choisi d'utiliser des frameworks complémentaires issus du projet open-source [ECLIPSE 2013] : EMF pour la définition de la syntaxe abstraite (méta-modèle) et la génération du code métier, GMF pour la spécification du langage de modélisation graphique et pour la génération de code de l'éditeur, ATL pour la spécification des règles de transformation et pour leur exécution. Le framework GMF reprend de façon outillée, la définition d'un langage de modélisation au sens du DSM. Ainsi les différentes syntaxes et mappings sont des modèles modifiables au travers d'un éditeur intégré à l'environnement de développement Eclipse : la syntaxe abstraite consiste en un méta-modèle conforme au méta-méta-modèle *Ecore* ; la syntaxe concrète est définie au travers de deux modèles : *gmfgraph*, qui définit la notation graphique, et *gmftool*, qui précise les concepts et les relations accessibles par la palette d'outils de l'éditeur. Le mapping entre les syntaxes abstraites et concrètes (Mac) est réalisé au travers d'un modèle *gmfmap*. Le domaine sémantique ainsi que son mapping avec la syntaxe abstraite ne sont pas représentés dans leurs propres modèles, mais peuvent être exploités au travers de règles OCL associées au méta-modèle du domaine (syntaxe abstraite). Deux modèles supplémentaires interviennent dans le processus de génération de l'éditeur : le modèle *genmodel*, qui paramètre la génération du code métier chargé entre autres de la persistance du modèle et le modèle *gmfgen* qui paramètre la génération du code de l'éditeur final. Les trois approches à instrumenter se sont basées sur un méta-modèle de départ formalisant le métier de conception de Moodle, issu des travaux passés [ABEDMOULÉH et al. 12] sur l'application d'un processus d'explicitation et de formalisation. Bien que subjectif et discutable dans les choix de méta-modélisation et les éléments formalisés, ce méta-modèle capture toutefois, entre autres, les éléments visibles de la plateforme intervenants dans les correspondances manuelles à prendre en compte : outils/ressources et leurs propriétés, groupes/groupements, sections ordonnées, etc.

5.2. Instrumentation de l'approche 1

Cette approche consiste à spécifier la syntaxe concrète du langage, modèles *gmfgraph* et *gmftool* sur la base du méta-modèle de Moodle, conservé sans modification en tant que syntaxe abstraite (partie gauche figure 2). Les figures, icônes, labels, propriétés, etc. représentant la palette et les éléments graphiques peuvent être définis de manière à cacher les concepts de la plate-forme sous-jacents, ceci afin de proposer une représentation offrant une signification davantage pédagogique en accord avec les éléments que le langage doit

offrir. Cette approche convient pour les cas tel que l'activité d'auto-évaluation, où un seul outil de la plate-forme est employé (*mapping* 1 à 1), mais s'avère problématique lorsque les critères de sélection de l'outil de la plate-forme sont nécessaires. GMF propose un mécanisme d'initialisation qui permet d'instancier plusieurs concepts du domaine en réponse à l'utilisation d'un élément de la palette d'outils. Néanmoins, cette instanciation intervient au moment de l'application de l'élément sur le diagramme et ne permet donc pas de faire varier les concepts instanciés en fonction de paramètres dynamiques. Afin de conserver le niveau d'expressivité du langage, il est nécessaire de définir un élément de la palette d'outils pour chaque combinaison des paramètres d'une activité pédagogique. Ainsi, l'activité de *rédaction d'un compte-rendu* devra être déclinée en 8 variantes, selon les valeurs associées aux 3 critères proposés, accessibles dans la palette de l'éditeur.

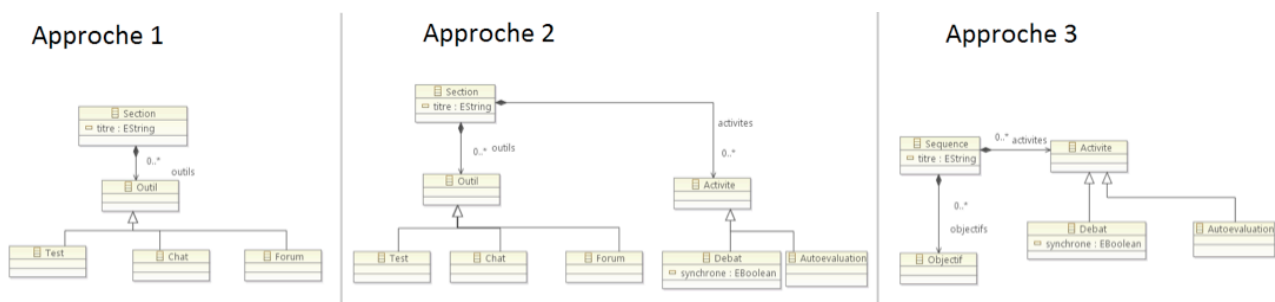


Figure 2. Représentations partielles illustrant la formalisation de la syntaxe abstraite selon les 3 approches

5.3. Instrumentation de l'approche 2

Les techniques utilisées dans l'approche 1 peuvent être exploitées à nouveau pour traiter les cas simples de correspondance. Afin de palier aux cas menant à une surcharge de l'interface, il est possible d'étendre le méta-modèle du domaine métier initial avec des nouveaux concepts (par exemple l'activité pédagogique *Débat* avec la propriété booléenne *synchrone* cf. partie centrale figure 2) et de lui associer des représentations graphiques. En revanche, le framework GMF s'appuie sur ce méta-modèle pour gérer la persistance des futurs modèles créés à l'aide de l'éditeur. L'extension du méta-modèle initial entraîne la non-conformité des futurs modèles, et empêche donc leur opérationnalisation potentielle par l'API dédiée. Toutefois, nous pouvons intervenir à deux niveaux : au niveau de la définition même de ces extensions en positionnant un attribut « *transient* » sur chaque concept ajouté au méta-modèle de Moodle, afin d'empêcher sa persistance (exploitation des possibilités offertes par les langages de spécification de méta-modèle), et au niveau du code généré par l'outillage DSM. Ainsi, pour l'activité *débat* par exemple, si la propriété *synchrone* est vraie, l'élément *Débat* sera ignoré lors de la persistance mais remplacé par un élément *chat*.

5.4. Instrumentation de l'approche 3

Cette approche s'appuie sur la formalisation d'un méta-modèle spécifique aux besoins discutés en section 3.2. Ce nouveau méta-modèle spécifie ainsi les différentes activités pédagogiques retenues, les éléments de structuration/ordonnancement de ces activités, et permet également la définition d'objectifs pédagogiques (extrait en partie droite de la figure 2). Ces derniers ont été ajoutés arbitrairement afin de montrer que cette approche ne cherche pas à réaliser, dans un premier temps, de correspondances avec les services offerts par la plateforme. La seule limite de cette spécification est celle des besoins à formaliser. Similairement aux autres approches, le processus outillé avec GMF consiste ensuite à spécifier les différents modèles pour la définition de la notation graphique, la palette, etc. Les futurs modèles/scénarios produits par l'éditeur n'ont aucune conformité avec le méta-modèle initial de l'approche 1, et donc aucune assurance d'être opérationnalisable sur

Moodle. Le méta-modèle représentatif du métier de conception de Moodle n'étant pas inclus dans ce méta-modèle spécifique, il est difficile d'opérer, comme dans l'approche 2, par modification du code gérant la persistance. La transformation de modèles, via des règles ATL par exemple, est une solution envisageable mais qui nécessite une étape supplémentaire, *a posteriori* de l'activité de scénarisation et *a priori* de l'opérationnalisation. Les modèles produits par les éditeurs conçus avec GMF sont conservés au format XML, ce qui rend également applicable des techniques de transformations orientées graphe de type XSLT. Dans nos travaux nous avons expérimenté les règles ATL. Celles-ci sont plus ou moins complexes selon le niveau de correspondances à prendre en compte. Aussi, ces dernières doivent avoir été explicitées et précisées. Il est possible également que des éléments faisant sens au niveau de la conception pédagogique ne trouvent pas d'éléments correspondants sur la plateforme, même par « détournement » des usages (traductions partielles produisant des scénarios incomplets / inconsistants), ou, à l'inverse, que les éléments sources soient insuffisants pour déterminer des choix de projection. Par exemple, les objectifs pédagogiques de notre cas d'étude *pourraient* ne pas faire sens sur Moodle et seraient alors « perdus » au moment de l'opérationnalisation. Selon l'importance de ces éléments les pertes peuvent conduire à rendre les autres éléments traduits incohérents.

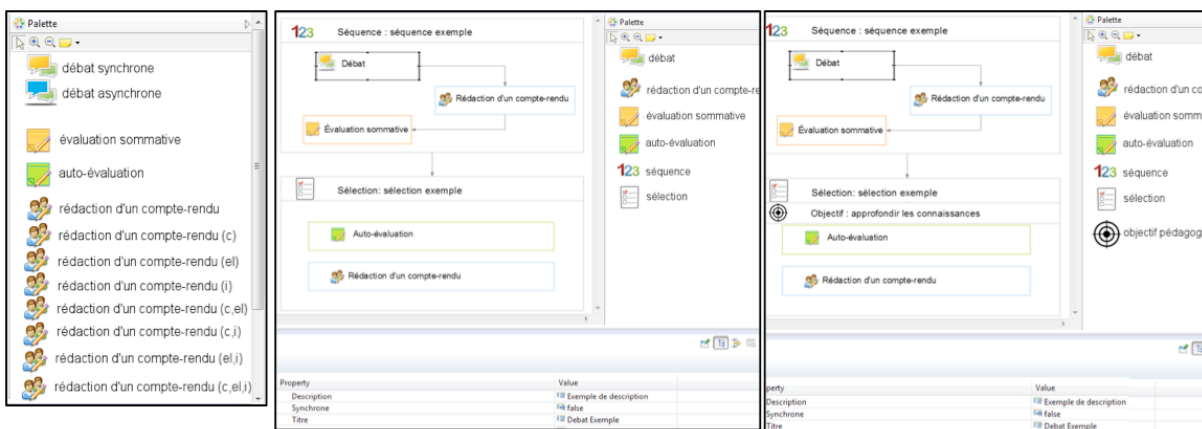


Figure 3. Captures d'écran partielles des éditeurs obtenus avec l'approche 1 (gauche), 2 (centre) et 3 (droite)

6. Analyses des 3 approches

L'application de l'outillage DSM selon les trois approches a donné lieu à une analyse selon quatre critères : 1/ l'expressivité visuelle, correspondant au potentiel d'expressivité sémantique des modèles tel que perçu par les praticiens au travers de la manipulation de la notation visuelle ; 2/ l'expressivité abstraite, correspondant à la sémantique concrète des modèles, en relation avec la syntaxe abstraite sous-jacente, non perceptible directement par les praticiens ; 3/ le potentiel d'opérationnalisation des modèles produits ; 4/ la sémantique du scénario une fois opérationnalisé (i.e. une fois traduit dans le dispositif en termes de configuration, structuration des outils/services/concepts/propriétés en relation avec le métier de conception de la plateforme). Le critère 1 est le seul qui ne nécessitait pas d'expertise et de compétences informatiques et qui relevait d'un point de vue usager. Pour ce critère nous avons recueilli, sous la forme d'un entretien informel participatif, des retours qualitatifs de la part des membres du PRN impliqués. La table 1 synthétise les résultats obtenus, annotés +, / ou - selon leur rapport aux deux objectifs de l'étude (expressivité selon le métier de conception de la plateforme et selon les pratiques à proposer). Du point de vue de l'expressivité perçue au travers du formalisme graphique, les approches 2 et 3 sont mieux adaptées car elles nécessitent moins de choix et réflexions sur la conception lors de la sélection des éléments de la palette. Par exemple, l'activité *débat* impose avec l'approche 1

de faire le choix *synchrone/asynchrone* sur la modalité de réalisation dès la sélection de l'activité ; un changement futur nécessitera de supprimer l'activité choisie et de la remplacer par l'autre). La palette de sélection peut ainsi se focaliser sur des éléments de conception dont les propriétés pourront être précisées plus tard. L'approche 3, occultant les aspects conformité au métier de la plateforme, propose alors des éléments/propriétés et relations de conception directement corrélés aux besoins. Elle se révèle donc mieux adaptée sur ce point que l'approche 2. L'expressivité abstraite des modèles produits est directement corrélée aux choix initiaux définis en relation avec les 3 approches. La mise en conformité des modèles produits était directe pour l'approche 1 alors qu'elle nécessitait des adaptations pour les deux autres approches. Dans l'approche 2 nous avons exploité la formalisation même des méta-modèles selon l'outillage DSM avec EMF/GMF (par exemple la propriété *transient* sur les *Eclass* pour indiquer la non persistance du concept) ainsi que la modification du code généré de l'éditeur pour modifier la persistance du modèle. L'approche 3 nécessite en revanche une mise en correspondance totale avec le méta-modèle de la plateforme a posteriori de la spécification du modèle. Les techniques de l'approche 2 ne sont plus adaptées. Nous avons procédé par de lourdes et complexes transformations de modèles. Concernant la sémantique du scénario obtenu en conformité avec le métier de conception de la plateforme, l'approche 3 ne peut pas garantir que celui-ci sera consistant et cohérent. Des travaux abordant la transformation de modèles entre métier des concepteurs et métier des plateformes relèvent également ces faiblesses [NODENOT et al. 08] [ABDALLAH et al. 08]. L'approche 2 garantit le maintien de la sémantique initiale par l'élaboration conjointe de l'extension du méta-modèle et des techniques de mise en conformité ; le maintien est donc fortement dépendant du *tissage* réalisé pour étendre le méta-modèle initial de la plateforme. L'approche 1 garantit en revanche le maintien de la sémantique par exploitation directe du méta-modèle initial.

Critères	Approche 1	Approche 2	Approche 3
Expressivité visuelle	Trop d'éléments (-)	Nombre d'éléments cohérent avec besoins praticiens mais contraint (/)	Nombre d'éléments cohérent avec besoins praticiens (+)
Expressivité abstraite	Limitée à celle capturée de la plateforme (-)	Limitée à un périmètre proche de la sémantique capturée de la plateforme (/)	Non limitée (+)
Conformité au méta-modèle de la plateforme	Directe (+)	Nécessite traduction pendant la scénarisation (/)	Nécessite lourde transformation a posteriori de scénarisation (-)
Sémantique du scénario après mise en conformité	Conservée (+)	Conservée mais contrainte (/)	Scénario dispositif peut être inconsistant/incomplet (-)

Table 1. Comparatif des résultats d'analyse des 3 approches

Globalement, l'approche 3 correspond à l'approche traditionnelle d'élaboration d'un VIDL, avec le principal avantage (l'expressivité) mais aussi défaut (difficulté d'opérationnalisation). L'approche 1 a montré les limites d'expressivité de la syntaxe concrète lorsque définie par dérivation de la syntaxe abstraite : cette approche permet de donner une à plusieurs représentations différentes pour un même concept et/ou relation issue du métier de la plateforme mais est contrainte à conserver ce lien fort de dérivation (contrainte intrinsèque au processus de spécification de langage selon l'approche DSM). L'approche 2 est finalement une approche intermédiaire sur l'ensemble des critères : meilleur rapport expressivité / conformité avec le métier de la plateforme. Elle requiert toutefois une forte expertise en méta-modélisation pour réduire le coût du codage nécessaire pour garantir la conformité des modèles. En revanche elle met en évidence que la sémantique et la notation du langage/éditeur doivent être spécifiées en relation forte avec la capacité de traduction vers le métier identifié et formalisé de la plate-forme. La correspondance des besoins/pratiques en termes d'éléments *métiers* de la plateforme ne peut se limiter à une

traduction seulement informatique et doit impliquer les praticiens : elle doit être également explicitée.

7. Conclusion et perspectives

Cet article a présenté un travail exploratoire de six mois dont l'objectif était d'étudier plusieurs approches de *modélisation centrée métier* pour la spécification/développement d'un langage/éditeur de scénarisation. Ce langage, restreint mais représentatif, avait deux exigences : expressivité tournée vers des pratiques de conception, et formalisation conforme au métier de conception d'une plateforme de formation donnée. Les résultats obtenus par les trois approches étudiées ont été analysés et comparés. L'approche offrant le meilleur compromis a consisté à étendre le méta-modèle initial de la plateforme afin d'y intégrer (syntaxes abstraites et concrètes) des éléments de pratiques de conception au niveau des usages des outils de la plateforme, le premier niveau d'abstraction choisi pour ce travail.

Ce travail s'inscrivait dans le cadre du projet ANR GraphiT dont l'approche originale est à contre-courant des approches de conception actuelles. Il va être poursuivi, pour la même double exigence, en élargissant le périmètre des études à plusieurs plateformes de formation existantes et à des pratiques de conception à différents niveaux d'abstraction de celui permis par les plateformes. Le cadre méthodologique du *Domain-Specific Modeling* mais aussi plus largement de l'Ingénierie Dirigée par les Modèles sera approfondi à commencer par les pistes mises en lumière par ce travail exploratoire : le tissage entre le métier des praticiens et celui de la plateforme doit être explicité et formalisé afin que des praticiens puissent vérifier, voire définir, la sémantique associée à la correspondance du scénario pédagogique en terme de conception/configuration de la plateforme.

Remerciements

Ce travail exploratoire a été réalisé dans le cadre du projet ANR GraphiT. Nous remercions les partenaires du *Pôle Ressource Numérique* de l'Université du Maine pour leur contribution à cette étude.

8. Bibliographie

- [ABDALLAH et al. 08] Abdallah F., Toffolon C., Warin B., « Models transformation to implement a Project-Based Collaborative Learning (PBCL) Scenario : Moodle case study », *Proceedings IEEE ICALT 2008*, Santander, p. 639-643
- [ABEDMOULEH et al. 12] Abedmouleh A., Oubahssi L., Laforcade P., Choquet C., « Identification of LMSs Instructional Languages: An Analysis Process », *Proceedings IEEE ICALT 2012*, Rome, 2012, p. 367-368
- [ABEDMOULEH & LAFORCADE 12] Abedmouleh A., Laforcade P., « Improving the design of courses thanks to graphical and external dedicated languages : a Moodle experimentation », *Proceedings 1st Moodle Research Conference*, Heraklion, 2012, p. 94-101
- [BOTTURI & STUBBS 08] Botturi L., Stubbs T. *Handbook of Visual Languages in Instructional Design: Theories and Practices*. Hershey, Information Science Reference 2008
- [BOTTURI et al. 06] Botturi L., Derntl M., Boot E., Figl K., « Classification framework for educational modeling languages in instructional design », *Proceedings IEEE ICALT 2006*, Kerkrade, 2006, p. 1216-1220.
- [BURGOS et al. 07] Burgos D., Tattersall C., Dougiamas M., Vogten H., Koper R., « Mapping IMS Learning Design and Moodle. A first understanding », *Journal of universal computer science*, volume 13, n°7, 2006

12 Environnements Informatiques pour l'Apprentissage Humain, Toulouse 2013

[CLAYER et al. 12] Clayer J-P., Toffolon C., Choquet C., « A pattern based and teacher-centered approach for learning design », *Proceedings CATE 2012*, Napoli, 2012

[CONOLE et al. 04] Conole G., Dyke M., Oliver M., Seale J., « Mapping pedagogy and tools for effective learning design », *Computers & Educations*, volume 43, n°1-2, p. 17-33.

[DODERO et al. 10] Doderio JM., Martinez del Val A., Torres J., « An extensible approach to visually editing adaptive learning activities and designs based on services », *Journal of Visual Languages & Computing*, volume 21, n°6, 2010, p. 332-346

[DOUGIAMAS & TAYLOR 03] Dougiamas M., Taylor P.C., « Moodle: Using Learning Communities to Create an Open Source Course Management System », *Proceedings EDMEDIA*, Honolulu, 2003

[KELLY & TOLVANEN 08] Kelly S., Tolvanen J-P., *Domain-Specific Modeling: Enabling Full Code Generation*, Wiley-IEEE Society Press, 2008

[KOPER & MANDERVELD 04] Koper R., Manderveld J., « Educational modelling language: modelling reusable, interoperable, rich and personalised units of learning », *British journal of educational technology*, volume 35, n°5, 2004, p. 537-551

[LAFORCADE 10] Laforcade P., « A Domain-Specific Modeling approach for supporting the specification of Visual Instructional Design Languages and the building of dedicated editors », *Journal of Visual Languages & Computing*, volume 21, n°6, 2010, p. 347-358

[MEHRABI & MASOUMEH 12], Mehrabi J., Masoumeh sadat A., « Teaching with Moodle in Higher Education », *Procedia - Social and Behavioral Sciences*, volume 47, 2012, p.1320-1324

[NODENOT et al. 08] Nodenot T., Caron P-A., Lepallec X., Laforcade P., « Applying Model Driven Engineering Techniques and Tools to the Planets Game Learning Scenario », *Journal of Interactive Media in Education*, volume 23

9. Références sur le WEB

[ECLIPSE 13] Site du projet Eclipse Modeling, <http://www.eclipse.org/modeling/>, 2013

[GRAPHIT 13] Site du projet GraphiT, <http://www-lium.univ-lemans.fr/~laforcad/graphit/>, 2013

[MOODLE 13] Statistiques sur le déploiement de Moodle, <https://moodle.org/stats/>, 2013