



**HAL**  
open science

## Abstraction des fonctionnalités d'une plateforme de formation pour la mise en œuvre de langages de scénarisation

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

### ► To cite this version:

Esteban Loiseau, Pierre Laforcade, Nour El Mawas, Sébastien Iksal. Abstraction des fonctionnalités d'une plateforme de formation pour la mise en œuvre de langages de scénarisation. *STICEF (Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation)*, 2017, 24, 10.23709/sticef.24.1.3 . hal-01714525

**HAL Id: hal-01714525**

**<https://hal.science/hal-01714525>**

Submitted on 21 Feb 2018

**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 des fonctionnalités d'une plateforme de formation pour la mise en œuvre de langages de scénarisation

► **Esteban LOISEAU, Pierre LAFORCADE** (LIUM, Le Mans-Laval), **Nour EL MAWAS** (Télécom Bretagne, Brest), **Sébastien IKSAL** (LIUM, Le Mans-Laval)

---

---

■ **RÉSUMÉ** • Le projet GraphiT vise à aider les enseignants à spécifier des scénarios pédagogiques pertinents qui puissent être opérationnalisés en tant que cours sur un *Learning Management System* cible. Nous nous intéressons à l'abstraction des aspects opérationnalisation afin de mettre l'accent sur la dimension spécification de la scénarisation. Nous proposons une approche centrée Moodle pour abstraire les usages des fonctionnalités du LMS et spécifier des briques de conception pédagogique de plus haut-niveaux. Nous proposons un langage et un éditeur de scénarisation pédagogique graphique.

■ **MOTS-CLÉS** • Informatique, Scénario pédagogique, Méta-modélisation, Outil auteur, Système de gestion de cours

■ **ABSTRACT** • *The GraphiT project aims to help teachers in specifying of pedagogically sound learning scenarios that can be executed for automatically setting-up the related Learning Management System course. We intend to provide teachers with LMS-specific instructional design languages and editors. To achieve this goal, we have to raise the LMS semantics in order to enrich the pedagogical expressiveness of the produced models. We propose a specific LMS-centered approach for abstracting the low-level concepts and turning these semantics into higher-level pedagogical building blocks. We present and illustrate our propositions focused on the Moodle LMS.*

■ **KEYWORDS** • *Computer science, Pedagogical Scenario, Learning Scenario, Meta-modelling, Authoring Tool, Learning Management System*

## **1. Introduction**

De nos jours, les plateformes de formation ou *Learning Management Systems* (LMS) sont 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 (Garrison et Kanuka, 2004). Néanmoins, les résultats d'une enquête et d'entretiens que nous avons menés en 2014 auprès de 208 enseignants mettent en avant de nombreuses difficultés quant à l'appropriation et l'utilisation de ces plateformes (Podvin et Laforcade, 2014). 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 les différentes fonctionnalités de la plateforme afin de supporter l'objectif pédagogique fixé ? Ces compétences peuvent être acquises au cours de formations professionnelles, qui se concentrent davantage sur les aspects techniques liés aux fonctionnalités de la plateforme qu'à la conception de scénarios pédagogiques cohérents et adaptés à cette plateforme. Etant donné la multiplicité des théories éducatives (Ormrod, 2011) et des approches de conception, ainsi que 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 à leur disposition plutôt que de leur proposer des outils de conception, indépendants des plateformes, pédagogiquement expressifs, mais ayant des difficultés à faciliter la mise en œuvre des modèles de conception produits (tels que (Alario-Hoyos *et al.*, 2012) et (Katsamani *et al.*, 2012)) pour les travaux récents sur ce domaine). Notre approche consiste ainsi à s'intéresser à une plateforme donnée et à identifier son potentiel en termes d'expressivité pédagogique (c.-à-d. que l'on saura mettre en œuvre sans perte d'information). Nous cherchons donc à produire des langages de scénarisation, et leurs outils-auteurs associés, dédiés à la spécification et la mise en œuvre de situations d'apprentissage pour une plateforme donnée. À l'aide de tels outils nous cherchons à ce que les enseignants puissent plus facilement s'approprier leur plateforme et ainsi concevoir et mettre en œuvre des situations d'apprentissage plus évoluées sur le plan pédagogique.

Le projet GraphiT (financé par l'Agence Nationale de la Recherche) suit cette approche de conception centrée plateforme. 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 plateformes, 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 plateformes (Abedmouleh *et al.*, 2008). 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 plateforme et d'augmenter ainsi l'expressivité pédagogique. La proposition d'abstraction que nous présentons dans cet article concerne uniquement la plateforme Moodle.

## **2. Contexte de recherche**

### **2.1. Plateforme de formation et scénarisation pédagogique**

Un *Learning Management System* (LMS) est un environnement logiciel qui permet la mise en œuvre et le déroulement de processus d'apprentissage ou de parcours pédagogiques. A l'origine il s'agissait de plateformes dédiées à la formation ouverte et à distance. Ces dispositifs avaient pour premières finalités la consultation à distance de contenus pédagogiques, l'individualisation de l'apprentissage et le télé-tutorat. De nos jours, ces LMS (pour les plus répandus : Moodle, Claroline, Sakai, Dokeos...) sont largement déployés et utilisés dans les institutions de formation pour tous types d'apprentissage : à distance, en *blended learning*, comme en présentiel. Les LMS fournissent actuellement de nombreuses fonctionnalités et outils pour l'hébergement des contenus pédagogiques, le contrôle de l'accès aux ressources, la réalisation des activités pédagogiques, la mise en place et le suivi des apprentissages, ou des activités de tutorat, le pilotage de la formation, la gestion des communautés d'apprenants et d'enseignants, la gestion administrative des formations, la gestion technique de la plateforme, etc.

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 (Dougiamas et Taylor, 2003). Globalement, les LMS proposent une approche de conception exploitant l'agrégation et le séquençage de nombreux types d'outils (aussi appelées fonctionnalités ou, avec plus d'ambiguïté, activité pour la plateforme 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 exemple: simple consultation de ressource, mise en œuvre d'auto-évaluation, résolution de situations-problèmes collaboratives complexes, etc.).

Les standards centrés activités, tel que IMS-Learning Design (IMS-LD), n'ont pas réussi à s'intégrer dans les LMS actuels. Des travaux menés (Burgos *et al.*, 2007) ont montré que, en plus du coût d'ingénierie, cela nécessitait de faire évoluer le métier même de conception de la plateforme en lui ajoutant un « moteur d'exécution » dédié au standard. Les approches les plus récentes (Berggren *et al.*, 2005) (Katsamani *et al.*, 2012), ayant cherché à proposer des langages opérationnalisables sur les plateformes 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.

Moodle propose son propre format d'import pour les quiz. Notre idée est de généraliser cette approche à tous les aspects de la conception pédagogique (spécification et opérationnalisation). Le projet GraphiT part de l'hypothèse que si les LMS étaient capables d'explicitier formellement leur « format opérationnel » de scénarisation pédagogique, et par extension d'importer des modèles conformes à ce format, alors cela permettrait d'élaborer des langages de conception, dédiés à tel ou tel aspect de conception, à divers degrés d'abstraction selon les objectifs et le positionnement du langage développé.

## **2.2. Outils et langages de scénarisation pédagogique pour Moodle**

L'objectif à gros grain du projet est d'explorer les limites, en termes d'expressivité pédagogique, de langages de scénarisation pédagogique opérationnalisables pour un LMS cible. D'après la classification des *Visual Instructional Design Modeling Languages* proposée par (Botturi *et al.*, 2006), nous visons des langages de scénarisation de type **formels** (scénarios interprétables par la machine), comprenant un ensemble fini de concepts et de règles de construction des scénarios pédagogiques, aux niveaux **spécification** (description de la tâche sans prendre en compte la plateforme cible) et/ou **implémentation** (description de la mise en

œuvre de la tâche sur la plateforme cible), et qui proposeront une notation **visuelle** (scénarios compréhensibles par l'humain). Se pose donc la question de l'expressivité que peuvent offrir des langages de scénarisation permettant de s'abstraire de la dimension implémentation de la plateforme cible, tout en garantissant le fait que les scénarios produits soient bien implémentables sur cette plateforme. Le travail de recherche présenté dans cet article ne répond pas complètement à cette question, mais présente un premier résultat d'abstraction mis en œuvre pour la plateforme Moodle. Nous nous sommes intéressés aux langages de scénarisation existants dont les propriétés sont proches de celles que l'on recherche.

Le système *Glue!* associé à l'éditeur *Glue!PS* (Alario-Hoyos *et al.*, 2012) ainsi que l'éditeur de scénarios *CADMOS* (Katsamani *et al.*, 2012) sont deux exemples récents de travaux partageant nos objectifs. Ils proposent tous deux une solution indépendante de la plateforme, mais avec la possibilité de déployer les scénarios produits vers la plateforme la plus courante : Moodle. Dans les deux cas le déploiement se fait concrètement en exploitant la fonction de restauration de cours de Moodle. Ils génèrent des fichiers conformes au format d'un cours de la plateforme ; chaque concept du langage est associé à un équivalent dans le modèle de données de la plateforme (subjectif à chaque approche). Avec *CADMOS*, cette phase de traduction est réalisée par l'éditeur. Elle s'appuie sur des correspondances non modifiables, fixées par les auteurs. *Glue!* propose une approche différente où les scénarios produits sont traduits (via des *adaptateurs* dédiés) vers un langage pivot intermédiaire (non accessible aux enseignants-concepteurs) pour être indépendant du langage de scénarisation d'origine. Cette correspondance *abstraite* est alors à nouveau traduite (via de nouveaux *adaptateurs* spécifiques) en scénario spécifique à une plateforme telle que Moodle. Ce type d'approches mène à des adaptations et des pertes sémantiques durant les phases de « traduction » des scénarios spécifiés par les concepteurs. Ces pertes sémantiques sont liées à la trop grande dissimilarité entre le langage de conception et le modèle de données (et par extension, les fonctionnalités) de la plateforme. De plus, ces correspondances ne peuvent pas être adaptées par le concepteur (*CADMOS*) ou difficilement (pour *Glue!* cela revient à développer un nouvel *adaptateur*).

Le projet *Flexo* (Dodero *et al.*, 2010) propose également un langage de scénarisation pédagogique intégrant des éléments sémantiques liés à la description des activités ainsi qu'à leur séquençement. Sa particularité est

de proposer une représentation des scénarios sous une forme textuelle annotable par le concepteur (langage pivot spécifique à *Flexo*, contrairement au langage pivot abstrait de *Glue*). Les spécifications du scénario et les moyens de les opérationnaliser sur Moodle sont explicites et modifiables (approche *scripts*). Mais pour cela le concepteur devra maîtriser ce nouveau langage textuel. Les approches actuelles ont donc :

- (1) une expressivité limitée (types d'activités très proches des notions d'outils, structure du scénario, ...) qui ne représente pas des pratiques spécifiques à une communauté d'enseignants-concepteurs,
- (2) une prise en charge de l'opérationnalisation des scénarios faible (pas de considérations du paramétrage fin des outils et services de Moodle), s'appuyant sur des correspondances pas ou difficilement adaptables par un enseignant.

D'autres travaux (Abdallah *et al.*, 2008) ont montré que les techniques issues de *l'Ingénierie Dirigée par les Modèles (IDM)*, telles que les transformations de modèles, peuvent permettre de transformer un modèle de scénario centré concepteur en un modèle spécifique à une plateforme. Néanmoins, ils mettent en avant la complexité des transformations en jeu, et donc le coût de leur conception, ainsi que la perte sémantique inhérente.

### **2.3. Le projet GraphiT d'un point de vue IDM et DSM**

La méthodologie du projet consiste à explorer comment l'Ingénierie Dirigée par les Modèles (IDM) et particulièrement le *Domain Specific Modeling (DSM)* (Kelly et Tolvanen, 2008) peuvent permettre de développer des éditeurs de scénarios pédagogiques ayant les caractéristiques suivantes :

- spécifiques à un LMS ;
- suffisamment expressifs pour pouvoir s'abstraire du métier de conception du LMS ;
- opérationnalisables, *i.e.* pouvoir être exécutables.

L'IDM est une méthodologie d'ingénierie logicielle qui privilégie la définition et l'exploitation de modèles et de méta-modèles, plutôt que la production manuelle de code source (Schmidt, 2006). Les travaux de recherche en IDM s'intéressent notamment à la spécification, l'exécution, la transformation, la composition de ces (méta-) modèles et proposent de nombreux outils et langages pour supporter ces activités. Le DSM (Kelly et Tolvanen, 2008) peut être considéré comme un processus issu de l'IDM, il

visé à systématiser la conception et l'utilisation de langages de modélisation spécifiques à un domaine métier. Ces langages visent la plupart du temps des modèles formels à haut-niveau d'abstraction, le plus proche possible des constructions et de la sémantique du domaine métier visé, en opposition aux langages semi-formels dits *généralistes*, comme UML (*Unified Modeling Language*, qui demandent aux concepteurs un effort de spécification supplémentaire.

Notre approche originale est de proposer une architecture directement dépendante du LMS considéré, afin de concevoir un langage de scénarisation tout en prenant en compte, dès son élaboration, des problématiques de correspondances entre le métier de conception, à niveau *spécification* (à construire), et le métier de conception du LMS, à niveau *implémentation* (à expliciter et formaliser). 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, à niveau implémentation : ce métier est réifié au travers de certains outils/services, de certaines interfaces et paramétrages, du *workflow* ou *learning flow* sous-jacent, etc. Ce métier de conception doit être dans un premier temps identifié et formalisé sous forme de méta-modèle. Ce dernier servira alors de base à l'élaboration d'un schéma XSD (*XML Schema Definition*) 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 plateforme. Elle a pour fonction de traiter le fichier XML (*eXtensible Markup Language*) du scénario (issu de l'éditeur) afin d'ajouter les données adéquates à la base de données du LMS.

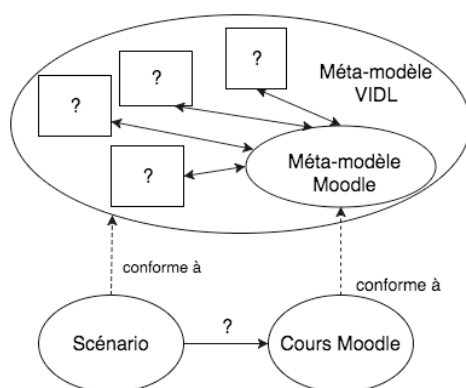
Dans le cadre du projet GraphiT, des travaux précédents se sont intéressés à la formalisation du méta-modèle de la plateforme (Abedmouleh *et al.*, 2008), expérimentée sur plusieurs plateformes dont Moodle. L'API a également été développée pour Moodle 2.4. Nous considérons dorénavant ces éléments comme existants et exploitables. Egalement, des résultats de travaux précédents (Loiseau et Laforcade, 2013) ont montré que la solution présentant le meilleur compromis entre expressivité pédagogique et compatibilité avec le LMS est d'étendre le méta-modèle de la plateforme. Cette solution a l'inconvénient de nécessiter une expertise importante en méta-modélisation afin de limiter le coût de développement induit par la nécessité de rendre les modèles conformes au méta-modèle de la plateforme (*i.e.* rétablir la compatibilité).



En effet, en étendant le méta-modèle de la plateforme, nous modifions la syntaxe abstraite de notre langage de modélisation et donc perdons la conformité avec le format d'import de la plateforme. La figure 1 schématise :

- (1) notre approche par extension du méta-modèle de Moodle pour élaborer le méta-modèle d'un *Visual Instructional Design Language* (VIDL),
- (2) la nécessité de contrôler les correspondances vers Moodle pour les nouveaux éléments, et
- (3) le besoin de remise en conformité du scénario produit.

Malgré le fait que notre approche soit spécifique à un LMS cible, l'abstraction du métier d'implémentation doit être dirigée vers des pratiques et besoins au niveau spécification que l'on cherche à couvrir. La prochaine section traite de cette collecte.



**Figure 1 • Schématisation de notre approche et des verrous à considérer**

### **3. Collecte et analyse des besoins**

Afin d'orienter nos propositions, nous avons mené à la fois une étude théorique sur plusieurs sources, notamment (Conole *et al.*, 2004), et une étude plus pratique en interrogeant des ingénieurs pédagogiques ainsi que des enseignants universitaires. Nous avons également conduit une enquête à plus grande échelle, à l'aide d'un questionnaire en ligne, suivie d'entretiens avec certains des répondants. L'objectif était de vérifier nos hypothèses et de recueillir des avis sur les orientations du projet. Cela fut également l'occasion de collecter des informations sur les pratiques, en

termes d'usage des LMS, et les besoins concernant les outils de conception pédagogique de nos potentiels utilisateurs finaux.

### **3.1. Analyse des résultats de l'enquête**

L'enquête (Podvin et Laforcade, 2014) a été réalisée sous la forme d'un questionnaire en ligne, diffusé au sein de la communauté francophone des enseignants du supérieur, sur une durée de quatre semaines. L'enquête visait en particulier les enseignants et ingénieurs pédagogiques utilisateurs de LMS. Le questionnaire comprenait 21 questions, permettant pour la plupart des réponses multiples. Les 8 premières questions (portant sur la conception globale des cours) étaient indépendantes du LMS utilisé par le répondant. Les autres questions n'étaient accessibles qu'aux personnes déclarant utiliser Moodle. Nous avons reçu et analysé 208 réponses. Voici quelques-uns des points les plus remarquables et pertinents dans le contexte de cet article.

Sur le contexte d'enseignement :

- 74 % des répondants utilisent un LMS en complément de leurs cours en présentiel (32 % uniquement pour cet usage) ;
- 52 % l'utilisent pour des cours à distance ;
- 37 % l'utilisent pendant les sessions en présentiel.

Sur les types d'usage de la plateforme :

- 91 % des répondants utilisent le LMS pour de la transmission de documents ;
- 52 % pour recueillir des devoirs ;
- 47 % pour supporter des activités collaboratives ;
- 47 % pour des évaluations ;
- 58 % pour des pratiques pédagogiques innovantes.

La moitié des répondants a déclaré avoir expérimenté sur la plateforme par eux-mêmes. Parmi ceux qui ne se considèrent pas comme novices (56 %), 73 % déclarent avoir amélioré leurs compétences par eux-mêmes quant à l'utilisation de la plateforme.

Bien que la moitié des utilisateurs de Moodle ayant répondu au questionnaire considèrent que l'interface utilisateur d'un cours est facile à manipuler, seuls 33 % pensent que les écrans de paramétrage à base de formulaires sont compréhensibles. D'un point de vue conception pédagogique, 38 % conçoivent entièrement (37 % partiellement) leur scénario avant de mettre en place le cours équivalent sur Moodle. 43 % de cette sous-population déclare rencontrer des difficultés lors de cette étape

de transition et se sont sentis contraints à modifier leur scénario initial et leurs intentions (12 % déclarent ne pas réussir à adapter leur scénario).

La majorité des utilisateurs de Moodle exploitent les fonctionnalités de base de la plateforme telles que l'indentation (64 %) ou le paramètre de visibilité (84 %). La moitié des répondants utilisent la fonctionnalité de notation des devoirs, ainsi que les groupes et groupements si nécessaire. 62 % utilisent la restriction d'accès, mais seulement 34 % le suivi d'achèvement. 15 des 22 fonctionnalités standards de Moodle ne sont pas bien connues par au moins 50 % des répondants. Le *forum* est largement préféré au *chat* pour les activités de communication. Pour mettre en œuvre des exercices, le *devoir* (47 %) et le *test* (37 %) sont préférés à *Hot Potatoes* (15 %) ou à la *leçon* (19 %). Le *wiki* est l'outil collaboratif le plus utilisé (23 %) (*journal* 8 %, *atelier* 8 %).

### **3.2. Analyse des entretiens**

Parmi les répondants ayant accepté de participer à un entretien complémentaire (téléphonique ou par visioconférence) à l'enquête, nous avons sélectionné 20 personnes en fonction de leur expertise quant à la scénarisation pédagogique et à l'utilisation de Moodle.

Les personnes interrogées validaient la pertinence de Moodle pour des besoins pédagogiques simples, mais reconnaissaient que, pour des situations d'apprentissage plus complexes, l'activité de conception devenait chronophage. Les écrans de paramétrage leur paraissaient complexes et difficiles à manipuler, notamment à cause d'un mélange entre paramètres d'ordre pédagogique et d'autres purement techniques. Il leur était nécessaire de tester des combinaisons de paramètres différentes et de vérifier le résultat afin de pouvoir atteindre leur objectif.

La plupart des personnes interrogées validaient l'idée d'un éditeur de scénario externe spécifique à Moodle et l'utilisation d'un *bloc* pour importer les scénarios dans l'interface du cours (l'aspect externe permettant de concevoir des scénarios en dehors de la plateforme, hors-ligne, et l'aspect graphique permettant de mieux visualiser le scénario dans son ensemble lors de la conception). Ils ont approuvé l'approche que nous proposons, en insistant sur l'intérêt de pouvoir manipuler des exemples d'usage d'outils de Moodle dans l'éditeur. Ils ont également mis en avant le besoin d'utiliser un langage/outil de conception qui couvre différents usages pédagogiques sans devenir pour autant trop générique. Certains ont exprimé le besoin de pouvoir continuer la conception avec

l'éditeur après l'import afin d'adapter le scénario, même s'ils avaient conscience que la modification de celui-ci à la fois avec l'éditeur et directement sur Moodle risquait de poser problème.

Cette étude a également montré que les enseignants n'ont pas de pratiques complexes communes, 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 plateforme qu'ils vont employer en fonction de l'usage pédagogique qu'ils visent. En effet, un grand nombre d'entre eux ont pointé 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 a semblé nécessaire d'avoir une vue plus abstraite en termes 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.

### **3.3. Analyse des besoins**

En ce qui concerne les 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 : une approche *top-down*, de la spécification vers l'implémentation, ne doit pas être imposée. 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 plateforme 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 plateforme.

Un autre point soulevé concernait la possibilité de déclarer dans le scénario des informations qui n'ont pas d'implémentation directe sur la plateforme 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. 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.

#### **4. Abstraction du méta-modèle de scénarisation**

Nous avons étudié comment abstraire le métier de conception pédagogique d'un LMS sous deux angles : l'un théorique, applicable à différents LMS, l'autre pratique, centré sur Moodle. Nous avons suivi une approche *bottom-up* en nous concentrant d'abord sur notre cas d'étude : Moodle. A partir des besoins des enseignants-concepteurs présentés précédemment, une abstraction possible est de s'appuyer sur des usages récurrents de fonctionnalités de la plateforme pour une activité pédagogique donnée. Du point de vue de la théorie de l'activité (Engeström, 1987) (Benson *et al.*, 2008), les activités pédagogiques impliquent de traduire également sur la plateforme les concepts de sujet, objet, outils/artefacts, communauté, division du travail, et règles. Les résultats des enquêtes et entretiens mettant davantage en avant le besoin de faciliter le paramétrage des outils sur la plateforme, nous avons décidé de nous concentrer sur cette problématique sous l'angle de la relation *sujet ↔ outils/artefacts*.

Les sections suivantes décrivent ces abstractions et leur formalisation dans le cas de la plateforme Moodle. Les éléments importants constituant ce méta-modèle sont expliqués dans les sous-sections suivantes. Nous reviendrons ensuite sur la vue générale en 4 niveaux du méta-modèle.

##### **4.1. L'activité pédagogique comme abstraction des outils**

Nous définissons une activité pédagogique comme *une abstraction du paramétrage d'un outil de la plateforme 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 l'implémentation la plus adéquate pour une activité

pédagogique. L'instanciation d'une activité pédagogique nécessite de renseigner un nom, une description (textuelle) et l'ensemble des critères au moment de la conception du scénario. 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.

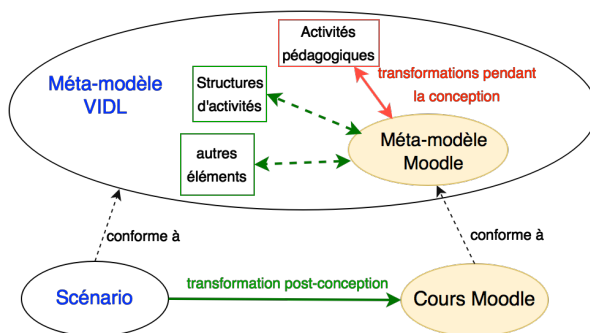
#### **4.2. Structures d'activités**

Selon (Gedera et Williams, 2013), la mise en place d'un cours en ligne nécessite un séquençement minutieux des activités, intégrant différents types de structures lors de la conception. Afin d'aider les enseignants-concepteurs à mettre en place des combinaisons d'activités et de ressources complexes, nous proposons d'ajouter des éléments structurels, habituels dans les VIDL (séquence, sélection, etc.). Ces structures peuvent être composées d'activités ou d'autres structures. Dans le cas de Moodle ces structures se traduiront par une combinaison d'*étiquettes*, indiquant son nom et son type, et d'indentation du contenu de la structure. Au moment de l'export, les paramètres des éléments contenus liés au suivi d'achèvement, à la restriction d'accès, à la visibilité, etc. seront fixés en fonction du type de structure. Les correspondances (spécification vers implémentation) de ces éléments sont fixées et non adaptables. Bien que celles-ci soient discutables, car issues de notre expertise de Moodle, l'intérêt est davantage porté sur la manière dont nous formaliserons et exploiterons ces correspondances dans la conception de l'éditeur final plutôt que sur leur pertinence objective.

#### **4.3. Retour sur la résolution des verrous de remise en conformité**

Les modèles produits (scénarios) par l'éditeur doivent être conformes au méta-modèle initial de la plateforme pour pouvoir être importés via l'API que nous avons développée. Nous proposons de restaurer cette compatibilité à l'aide de 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 pour les activités pédagogiques 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 parfaitement conforme au méta-modèle de la plateforme. Tous les éléments du méta-modèle du VIDL qui ne sont pas des activités

pédagogiques sont concernés par cette traduction. Les correspondances seront donc prises en compte dès l'identification et l'ajout de ces éléments, de manière similaire aux correspondances évoquées dans la section précédente pour les structures d'activités. Le propos de cette communication concerne davantage l'identification, la formalisation et l'exploitation des correspondances des activités pédagogiques. Les autres correspondances seront toutefois évoquées, ainsi que leur mise en œuvre technique (cf. section sur l'éditeur).

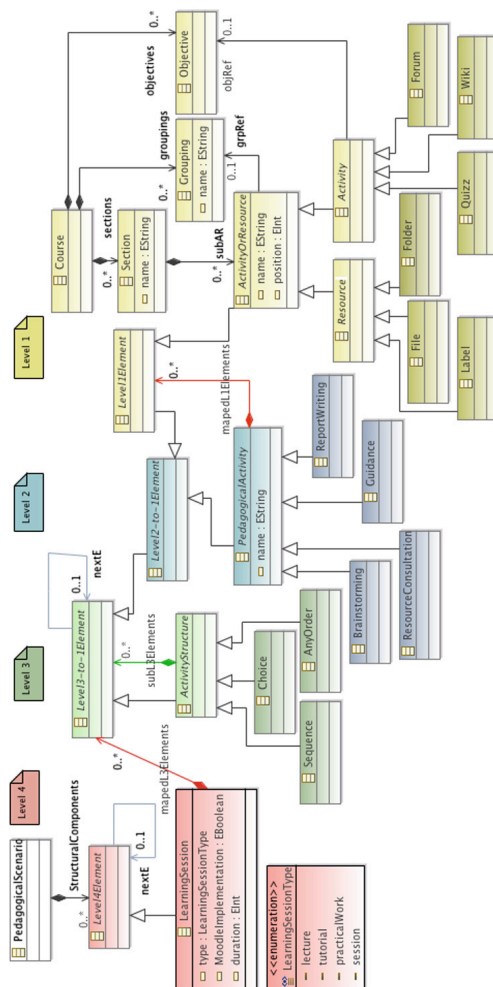


**Figure 2 · Les deux types de transformations**

La section suivante présente le méta-modèle du VIDL que nous proposons pour Moodle.

#### **4.4. 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.



**Figure 3 • Syntaxe abstraite (partielle) du langage de scénarisation intégrant le méta-modèle de Moodle**

La figure 3 illustre notre proposition à l'aide d'une représentation graphique du modèle *Ecore* du domaine (le format utilisé par *Eclipse Modeling Framework* pour les méta-modèles). Le niveau 1 correspond au méta-modèle de Moodle. 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 (**utilisation partielle directe**). Ce niveau comprend également les



concepts de *cours* et de *sections*, *groupes/groupements*, *objectifs*, indispensables à l'opérationnalisation du scénario. La transformation post-conception (lors de l'export du scénario) prendra en charge la création et l'ordonnancement de ces éléments Moodle afin de produire un scénario cohérent et compatible avec l'API que nous avons développée (**génération complète indirecte**).

Le niveau 2 comprend les briques abstraites du langage : les activités pédagogiques. 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. La figure 3 illustre seulement quelques exemples pour ces activités. La version complète du méta-modèle contient l'ensemble des activités pédagogiques que nous avons identifiées (cf. section suivante) et leurs propriétés pédagogiques discriminantes.

Le niveau 3 propose les structures d'activités (séquence, au choix, etc.), qui sont reliées aux éléments de niveau 2 ou 1 afin que les concepteurs puissent pendant la scénarisation regrouper sans distinction activités pédagogiques et outils Moodle.

Enfin, le niveau 4 est d'ordre organisationnel et correspond au découpage global de la session d'apprentissage en *sessions* plus fines de natures diverses. Ces *sessions* contiennent les éléments des niveaux 1 à 3. Elles se traduiront en *sections* dans Moodle, qui est le seul concept qui permette la structuration. Néanmoins, la fonctionnalité d'indentation du contenu dans Moodle (la position horizontale des éléments) sera exploitée pour représenter visuellement la notion de contenance entre les éléments de niveaux 4 et les autres niveaux, ainsi qu'entre les structures d'activités et leurs contenus (éléments de niveau 3 composés d'éléments de niveaux 1 et 2). Ces correspondances seront réalisées lors de l'export du scénario afin de positionner tous les éléments du scénario, quel que soit leur niveau, en éléments inter-reliés conformes à l'unique niveau 1 (celui précisant comment implémenter des scénarios spécifiques, les espaces-cours de Moodle).

Les méta-classes en fin d'arbre d'héritage (sur le diagramme de la figure 3), sont des exemples d'éléments de chacun des niveaux ; leurs attributs ne sont pas représentés ici, mais chacun possède des propriétés.

Concrètement, dans l'éditeur, l'utilisateur aura accès dans un premier temps aux éléments de niveau 4 dans la palette d'outil. Ces éléments,

équivalents aux sections de Moodle, sont indispensables à la structure du cours. Il est également possible d'intégrer des sessions qui ne reposent pas sur le LMS, spécifiées à l'aide d'un attribut booléen *opérationnalisable*, afin de pouvoir représenter un cours complet au sein du scénario. Lorsque l'utilisateur double-cliquera sur un élément de niveau 4 opérationnalisable, un sous-diagramme s'ouvrira, 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.

Les 4 niveaux présentés dans le diagramme représentent donc les 4 niveaux d'éléments proposés inter-reliés par 3 relations de « contenance » possibles (spécifiées par des relations de composition dans la figure 3). Le niveau des sessions peut contenir des éléments issus des 3 autres niveaux, le niveau des structures d'activités peut contenir également des éléments des niveaux 1 à 3 (une structure d'activité peut contenir d'autres structures d'activités). Le niveau des activités pédagogiques contient les éléments de niveau 1 traduisant la correspondance associée au paramétrage actuel des activités pédagogiques. Nous proposons pour les spécifications de l'éditeur graphique (cela reste discutable) :

- (1) qu'il y ait un diagramme pour spécifier les éléments de niveau 4 (les sessions),
- (2) que chaque session puisse avoir un diagramme dans lequel seront spécifiés, à la convenance du concepteur, les structures activités, les activités pédagogiques, et les outils Moodle,
- (3) que chaque activité pédagogique puisse être « ouverte » pour visualiser un diagramme spécifiant, en termes d'éléments de Moodle, la correspondance (modifiable) qui est réalisée dynamiquement pendant la scénarisation.

Le méta-modèle que nous avons présenté décrit la structure d'imbrication de nos 4 niveaux. Pour autant, les correspondances des éléments de niveaux 4 (sessions) et 3 (structures d'activités) ne sont pas capturées dans ce méta-modèle. Ces spécifications ont été directement formalisées et mises en œuvre sous la forme d'un modèle de transformation à l'aide du framework ATL (*ATLAS Transformation Language*) (Jouault *et al.*, 2006), elles ne sont pas davantage présentées dans

cet article. En revanche, les correspondances entre activités pédagogiques et implémentation Moodle ne sont pas triviales à identifier et capturer.

## **5. Un concept central : l'activité pédagogique**

### **5.1. Identification des activités pédagogiques et de leurs correspondances**

Afin d'identifier les outils les plus appropriés à l'implémentation d'une activité pédagogique, nous avons suivi une méthode en trois étapes :

- analyse, pour chacun des outils proposés par Moodle, de ses usages récurrents (méthode *bottom-up*) ;
- identification d'outils permettant des usages identiques (*top-down*) ;
- spécification des critères discriminants permettant la sélection de l'outil le plus adéquat.

Moodle, dans sa version 2.4, propose 7 types de ressources (*Livre, Page, Étiquette, Paquetage IMS, Fichier, Dossier et URL*) et 13 activités (*Forum, Base de données, Glossaire, Devoir, Leçon, Test, Atelier, Paquetage SCORM, Outil externe, Sondage, Consultation, Wiki et Feedback*). Après avoir étudié les usages récurrents de chacun de ces outils, nous avons remarqué que certains d'entre eux pouvaient avoir des usages détournés. Par exemple, un *forum* peut être utilisé pour discuter autour d'une thématique, mais peut également servir de Foire Aux Questions (FAQ), ou permettre aux étudiants de partager des fichiers. Dans un second temps, nous avons identifié quels outils avaient des usages en commun, par exemple, pour une FAQ il est possible d'utiliser un *forum*, un *wiki*, ou un *glossaire*.

Nous avons pu alors définir des critères discriminants afin d'aider l'enseignant à décider quel outil utiliser lorsque le choix se présente. Il est possible de représenter ces critères dans une matrice de décision qui se construit selon les règles suivantes :

- **R1.** Le nom de l'activité pédagogique est formulé du point de vue de l'étudiant (sauf si l'activité leur est invisible, et dans ce cas c'est le point de vue de l'enseignant qui est pris), exemple : « Répondre à un sondage » plutôt que « Créer un sondage ».
- **R2.** Les outils permettant de mettre en œuvre l'activité pédagogique considérée sont représentés sur les colonnes.
- **R3.** Les critères discriminants sont représentés sur les lignes.
- **R4.** Les critères discriminants doivent exprimer, le plus possible, une caractéristique pédagogique et doivent être formulés comme des questions fermées (oui/non).

- **R5.** Les cellules à l'intersection d'un outil et d'un critère doivent contenir toutes les valeurs possibles de critères qui permettent de choisir cet outil.
- **R6.** Un critère discriminant doit permettre de discriminer au moins un outil.
- **R7.** La matrice est complète s'il n'y a pas de combinaisons de critères parfaitement identiques menant à deux outils.

Une matrice incomplète nécessite d'ajouter des critères supplémentaires, jusqu'à satisfaire la règle **R7**. Le tableau 1 ci-dessous présente un exemple pour l'activité « Répondre à un sondage », pour laquelle quatre outils sont disponibles : *test* (O1), *sondage* (O2), *feedback* (O3) et *consultation* (O4). Nous proposons 7 critères discriminants :

- (C1) Il y a-t-il plusieurs questions ?
- (C2) Il y a-t-il uniquement des questions à choix multiples ?
- (C3) Voulez vous utiliser des questionnaires prédéfinis ?
- (C4) Est-ce en temps limité ?
- (C5) Est-ce anonyme ?
- (C6) Est-ce noté ?
- (C7) Il y a-t-il un feedback après la validation du sondage ?

Selon la matrice du tableau 1, l'outil *consultation* (O4) par exemple, permet de sonder sur plusieurs questions, proposant des choix multiples et des questionnaires prédéfinis. Pour cet outil, il n'est pas possible d'imposer une limite de temps pour répondre, ni d'anonymiser les réponses ou de les noter. On ne peut pas non plus donner de feedback à l'apprenant.

**Tableau 1 • Exemple de matrice de décision**

<b>AP</b>	<b>O1</b>	<b>O2</b>	<b>O3</b>	<b>O4</b>
<b>C1</b>	Oui/Non	Non	Oui/Non	Oui
<b>C2</b>	Oui/Non	Non	Oui/Non	Oui
<b>C3</b>	Non	Non	Non	Oui
<b>C4</b>	Oui/Non	Non	Non	Non
<b>C5</b>	Non	Non	Oui/Non	Non
<b>C6</b>	Oui	Non	Non	Non
<b>C7</b>	Oui	Non	Oui	Non
	<i>Paramètres O1</i>	<i>Paramètres O2</i>	<i>Paramètres O3</i>	<i>Paramètres O4</i>

Cette matrice doit également être complétée par des informations sur les paramètres de l'outil sélectionné, qu'ils soient généraux (peu importe les valeurs des critères), ou contextuels (la valeur du paramètre dépend

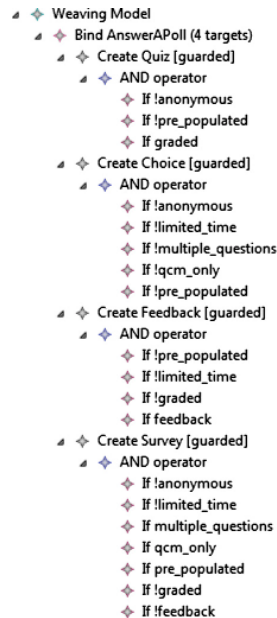
d'une réponse à un des critères). Ces précisions optionnelles sont importantes pour encapsuler, dans les correspondances, l'implémentation Moodle détaillée de l'activité pédagogique considérée.

## **5.2. Formalisation des correspondances par tissage de modèles**

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 sont exécutées à la demande, durant le processus de modélisation du scénario lorsque le concepteur double-clique sur une activité pédagogique qu'il a paramétrée. Cela permet d'ajouter automatiquement au scénario des éléments d'implémentation (niveau 1) qui seront représentés graphiquement dans un sous-diagramme. Ces transformations de modèles sont coûteuses à produire, en raison notamment du nombre d'éléments à implémenter et de la complexité des règles de correspondance. Afin de pallier ce problème nous proposons d'utiliser le tissage de modèles, présenté dans (Loiseau *et al.*, 2014), afin de formaliser les règles de correspondances à l'aide d'un modèle de tissage et de générer le code source des transformations de modèles. D'un point de vue pratique, à l'aide de la matrice construite par un expert de la plateforme telle que présentée dans la section 5.1, un ingénieur modélise les règles de correspondances entre une activité pédagogique et des outils de la plateforme afin de produire un modèle de tissage. Il utilise ensuite une transformation de haut niveau (une transformation de modèle spécifique) afin de générer le code des transformations de modèle permettant effectivement d'opérer les correspondances. Ces dernières peuvent alors être intégrées à l'éditeur et s'exécuteront à la demande, pendant l'utilisation de l'éditeur par l'enseignant-concepteur.

Le modèle de tissage peut être spécifié à l'aide d'un langage de tissage utilisant un méta-modèle de tissage générique que nous proposons. Ce méta-modèle de tissage définit la syntaxe du modèle de tissage (de correspondances). Chaque correspondance (ou *binding*) référence un élément **source** (l'activité pédagogique) et un ou plusieurs éléments **cibles** (l'outil) tous issus du méta-modèle du langage de scénarisation. Il est possible de poser des conditions sur « l'instanciation » d'une cible et de donner des valeurs à ses attributs (également avec des conditions). La figure 4 présente un exemple de modèle de tissage issu de la matrice de décision du tableau 1. Ce modèle définit les valeurs des critères pour

lesquelles un outil Moodle doit être instancié en les combinant avec des opérateurs ET/OU/NON. Les informations sur les paramètres des outils doivent être déduits des indications données par l'expert (non représentés sur la figure 4).



**Figure 4 · Exemple de modèle de tissage**

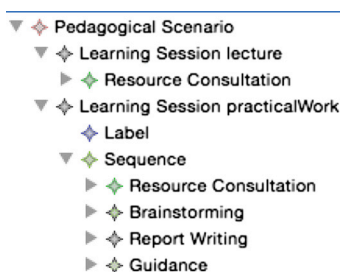
Nous exploitons les outils et langages du projet Epsilon (Paige *et al.*, 2009) afin de construire un *framework* de tissage de modèles correspondant à nos besoins. Nous utilisons le format *Ecore*, comme précédemment pour formaliser les méta-modèles. Les modèles de tissage sont édités à l'aide de *ModeLink*, un éditeur de modèles à trois panneaux qui permet de présenter les méta-modèles nécessaires au tissage sur les côtés gauche et droit, tandis que le modèle de tissage sera spécifié au centre. Les transformations de modèles exécutées pendant l'utilisation de l'éditeur afin de réaliser les correspondances sont exprimées à l'aide de l'*Epsilon Object Language* (EOL) et sont générées par une transformation *Model to Text*, faisant office de transformation de haut niveau, à l'aide du langage *Epsilon Generation Language* (EGL).

Notre approche de spécification des correspondances exploite pour le moment l'outillage technique de l'IDM. Cette proposition technique

montre que le codage des correspondances n'est plus nécessaire: leur formalisation peut être réalisée sous la forme d'un modèle simple spécifié en se basant sur les données décrites dans les matrices d'identification précédemment exposées. Il serait donc envisageable prochainement de déduire automatiquement ces modèles sur la base d'un éditeur dédié réifiant la sémantique des matrices d'identification.

## **6. Exemple de scénario pédagogique**

Nous proposons d'illustrer notre approche en formalisant un scénario pédagogique simple, mais représentatif, visant le LMS Moodle. Nous présentons dans un premier temps une description textuelle du scénario puis illustrons sa formalisation à l'aide du méta-modèle présenté dans la section 4. La figure 5 est une capture d'écran du scénario exemple, ouvert dans un éditeur en arbre EMF (il ne s'agit pas de l'éditeur graphique final, mais d'un éditeur de base permettant de spécifier des modèles conformes, par construction, avec leur méta-modèle).



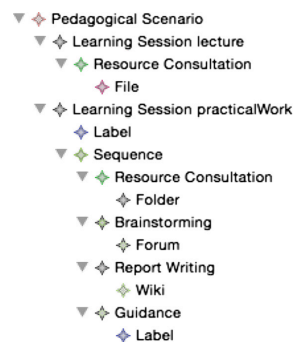
**Figure 5 • Exemple de scénario pédagogique composé d'éléments des quatre niveaux**

### **6.1. Description et formalisation**

Le scénario exemple est composé de deux sessions d'apprentissage. La première est un *cours magistral* pour lequel l'enseignant souhaite donner accès aux ressources utilisées en présentiel (*Resource consultation*). Cette activité pédagogique possède un attribut *quantity* valant *one* et un attribut *location* valant *local* (une seule ressource qui sera disponible directement sur le LMS). Ces propriétés permettront au mécanisme d'**implémentation par défaut** de sélectionner l'outil *File* (fichier) et de l'ajouter au scénario (voir la figure 6).

La seconde session d'apprentissage est de type *travaux pratiques* et se déroulera en présentiel dans une salle équipée d'ordinateurs. L'enseignant souhaite utiliser Moodle pour supporter une séquence d'activités comprenant quatre éléments. Le premier est une autre activité *Resource consultation*, dont les attributs *quantity* et *location* valent respectivement *many* et *local*. L'implémentation choisie cette fois est le dossier (*folder*). Le second élément de cette séquence est une activité de *brainstorming* qui, selon son attribut *orientation*, sera implémentée à l'aide d'un forum. De façon similaire, l'activité suivante (*Report Writing*) exploitera un wiki à cause de la dimension *collaborative*. Le dernier composant de cette séquence, *Guidance*, permet à l'enseignant de créer un mémo lui rappelant d'évaluer les productions des étudiants. Selon sa propriété *public* valant *tutor*, il se traduira sur Moodle en une *étiquette (Label)* ayant la propriété *visible* à faux, de manière à n'être visible que par l'enseignant.

La figure 6 présente le scénario une fois les implémentations de chacune des activités pédagogiques ajoutées automatiquement.



**Figure 6 • Exemple de scénario pédagogique intégrant les implémentations automatiques**

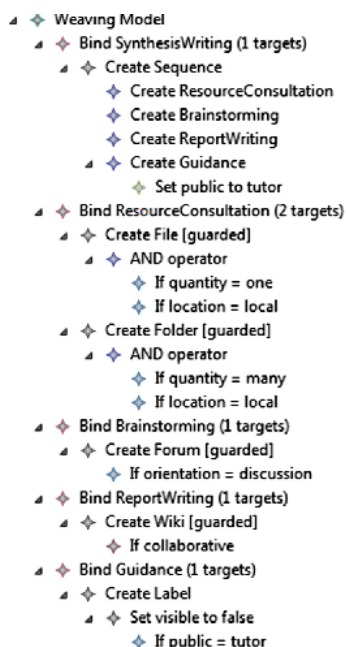
Cette représentation en arbre permet de bien illustrer la hiérarchie de « nœuds » du modèle :

- le scénario est la racine, il est composé de sessions ;
- les sessions contiennent soit des activités pédagogiques (comme *resource consultation*), soit des structures d'activités (comme *sequence*), soit directement un outil Moodle (comme le premier *Label* visible) ;
- les structures d'activités peuvent à nouveau être composées de ces 3 types d'éléments (1 seul visible dans l'exemple) ;



- les activités pédagogiques sont composées d'outils Moodle (comme *File* sous *Resource Consultation*).

La figure 7 présente un modèle de tissage spécifiant les correspondances des 5 activités pédagogiques impliquées dans l'exemple du scénario.

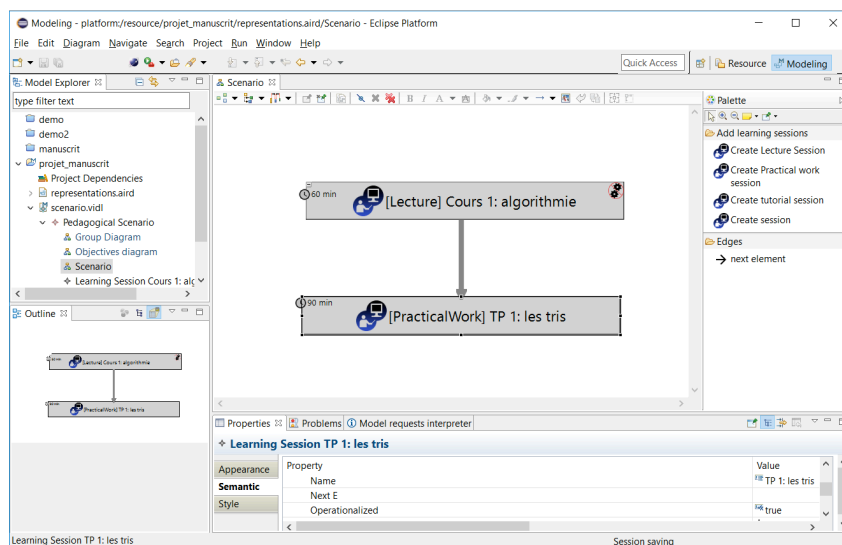


**Figure 7 · Exemple de modèle de tissage spécifiant les correspondances des activités pédagogiques impliquées dans l'exemple de scénario**

## **6.2. Prototype d'éditeur de scénarios réifiant les propositions**

Nous avons développé un prototype d'éditeur de scénarios. Il réifie notre proposition de langage et intègre la mise en œuvre des correspondances (celles au *runtime*, pour visualiser les implémentations Moodle des activités pédagogiques, et celles à l'export du scénario, pour restaurer sa conformité avec Moodle). Ce prototype nous permettra de vérifier la spécification d'un scénario en conformité avec les besoins identifiés par le travail exploratoire (enquête et entretiens) et la bonne exécution des différentes transformations de modèles.

Pour mettre en œuvre l'éditeur nous avons tout d'abord spécifié une notation graphique (syntaxe concrète au sens DSM) sur la base du méta-modèle de scénarisation présenté dans la section 4. Cette notation décrit les représentations graphiques souhaitées pour les éléments du méta-modèle (concepts, propriétés et relations). Nous avons pour cela utilisé l'outillage DSM du projet Sirius (Sirius, 2016) qui permet de créer des éditeurs de modèles de façon plus rapide et demandant moins de compétences techniques qu'avec d'autres outillages comme GMF (*Graphical Modeling Framework*) (Eclipse Modeling Project, 2016), traditionnellement utilisé pour ce type de besoin. La syntaxe concrète est définie dans Sirius à l'aide d'un seul modèle (*Viewpoint Specification Model*) qui sera ensuite « interprété » à l'exécution par un plugin Eclipse dédié. Cette méthode sans génération de code permet de réduire significativement les coûts de développement d'un éditeur graphique et favorise le prototypage. Plusieurs diagrammes peuvent être spécifiés et articulés ensemble (support pour des *langages en couches*). Un élément de modèle peut avoir plusieurs représentations visuelles dans un même diagramme (support pour des représentations *multiplés*). Sirius prend en charge également la spécification *des palettes d'outils* présentant, à côté d'un diagramme, les éléments de modélisation disponibles, des menus contextuels, des actions réalisables sur les éléments de modèle, etc. Sirius propose un mécanisme pour appeler un service externe : nous l'avons exploité afin d'intégrer à l'éditeur les transformations de modèle nécessaires aux mécanismes **d'implémentation par défaut** et de **remise en conformité finale**.

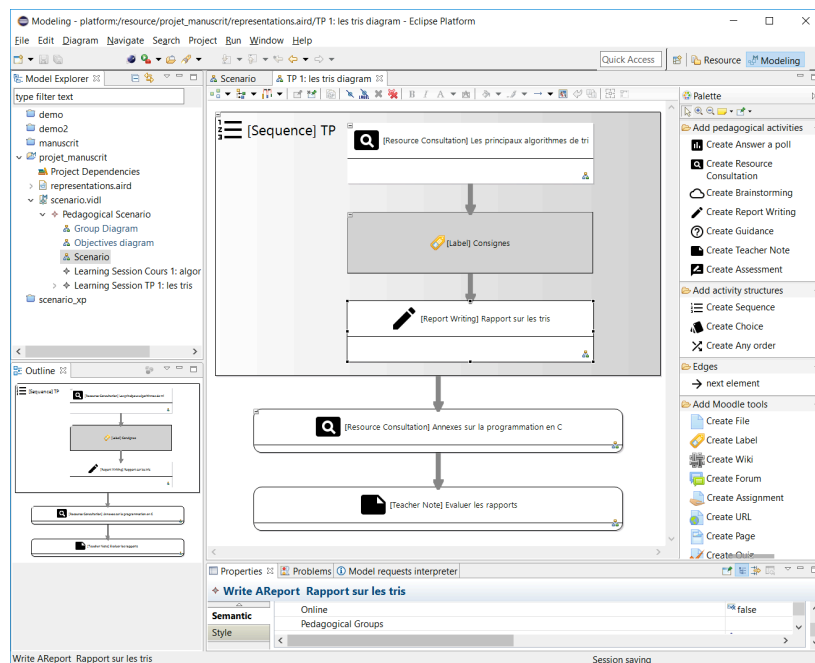


**Figure 8 • Capture d'écran d'un diagramme de sessions d'apprentissage**

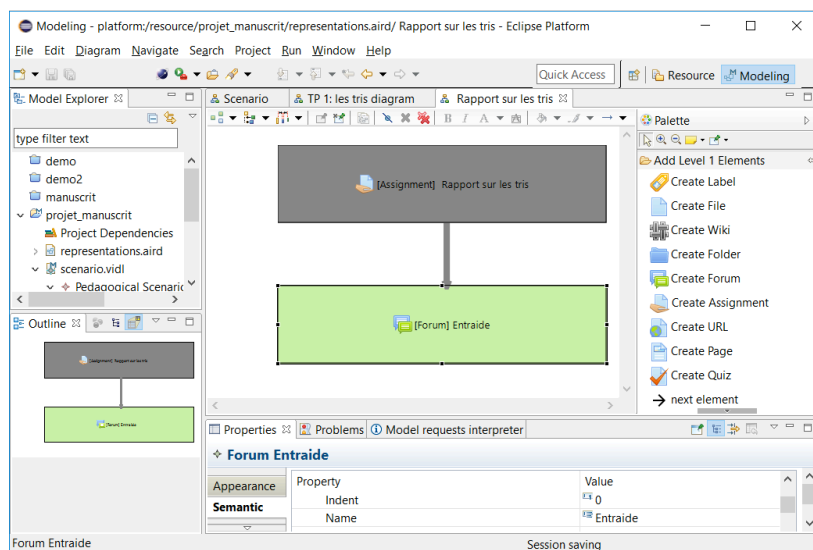
Cette notation graphique n'a pas fait l'objet d'une étude approfondie, au sens des préconisations d'étude de (Moody, 2009). Elle vise simplement à permettre de distinguer les éléments composant les diagrammes (couleurs, formes, emboitements). Elle reste subjective et discutable. Nous présentons ci-après cette notation à travers les 3 types de diagrammes que nous proposons.

Le premier diagramme est celui des *sessions d'apprentissage* (voir figure 8). Il permet de spécifier un premier niveau de découpage du scénario nécessaire, étant donné que chaque session correspondra à une *Section* dans l'espace-cours de Moodle. Lorsque l'utilisateur double-clique sur l'une des sessions d'apprentissage de ce diagramme, un nouveau diagramme des *activités* s'ouvre, dans lequel il est possible d'ajouter des éléments des niveaux 1 à 3 du méta-modèle (figure 9) afin de spécifier les activités à réaliser pour la session concernée. A ce niveau, il est possible de fixer les propriétés des éléments de niveau 2 (les activités pédagogiques). Lorsque l'utilisateur double-clique à nouveau sur l'un de ces éléments, l'exécution des transformations de modèles applicables à cet élément est réalisée, ajoutant au nouveau diagramme l'implémentation par défaut qui convient. Ce dernier diagramme d'*implémentation* (figure 10) contient donc des éléments de niveau 1 (outils de Moodle) qui peuvent être

modifiés, supprimés ou bien complétés par d'autres éléments du même niveau.



**Figure 9 · Capture d'écran d'un diagramme d'activités précisant les activités pédagogiques et les structures d'activités pour une session donnée**



**Figure 10 · Capture d'écran d'un diagramme d'implémentation (le contenu d'une activité pédagogique)**

## **7. Conclusion**

### **7.1. Abstraction des usages d'outils de la plateforme**

Nous proposons une approche centrée plateforme spécifique à un LMS afin d'augmenter l'expressivité pédagogique du métier de scénarisation, à niveau opérationnalisation de ces plateformes. Nous avons montré comment il était possible d'abstraire les usages de fonctionnalités (outils) de la plateforme, associés à leur paramétrage, afin de proposer des briques de conception de plus haut niveau appelées « activités pédagogiques ». Nous avons également proposé une méthode permettant aux experts de la plateforme de spécifier les correspondances entre activités pédagogiques et outils du LMS. Nous proposons une solution outillée de tissage de modèles afin de formaliser ces correspondances. Les modèles de tissages spécifiés par les experts peuvent alors être traités par une transformation de haut niveau qui produira un ensemble de transformations de modèles qui sont ensuite intégrées à un éditeur et seront exécutées pendant la phase de conception du scénario pédagogique. Nous avons présenté et illustré la syntaxe abstraite à 4 niveaux d'un langage de scénarisation pédagogique spécifique à Moodle. Enfin, nous avons présenté un prototype d'éditeur permettant de

spécifier des scénarios à travers 3 types de diagrammes (sessions, activités et implémentation).

L'ensemble de nos propositions (le langage de scénarisation, le prototype d'outil-auteur le réifiant, les différentes techniques d'identification, de formalisation et de mise en œuvre techniques par transformations) ont permis de valider, par construction, notre approche générale de conception d'un VIDL centré plateformes de formation. Les résultats obtenus ne sont qu'une forme d'abstraction possible. Bien que la syntaxe abstraite de notre VIDL ait été élaborée en prenant en compte des besoins spécifiques identifiés à travers des enquêtes et entretiens, d'autres approches d'abstraction répondant à d'autres besoins peuvent être envisagées. Le résultat principal de ces recherches est qu'il est possible d'élaborer des langages de scénarisation dirigés vers des besoins spécifiques issus de communautés de praticiens pour une plateforme de formation donnée, à condition de contrôler (identifier, formaliser et manipuler) les correspondances entre *éléments de spécification* et *éléments d'implémentation*. La subjectivité de ces correspondances peut être appréhendée en permettant l'adaptation des correspondances par défaut.

## **7.2. Evaluation des propositions**

Afin de valider globalement la proposition, nous avons réalisé une *vérification* sous la forme de mises à l'essai (processus interne) de l'outillage proposé. L'objectif était de spécifier différents scénarios de manière à vérifier les différents besoins initiaux : mixer éléments de spécification et éléments d'opérationnalisation Moodle, proposer une implémentation par défaut des activités pédagogiques selon le renseignement des critères pédagogiques, permettre la modification des implémentations proposées au *runtime*, vérifier la remise en conformité lors de l'export final du scénario, etc.

Une *validation* impliquant l'intervention de participants externes au projet (processus externe) aura prochainement lieu. L'objectif de cette validation ne sera pas de valider ou d'invalider les besoins et exigences initiales : il n'est pas envisageable de faire intervenir à nouveau des personnes ayant déjà participé à l'enquête ou aux entretiens. L'objectif consistera plutôt à s'assurer que l'environnement outillé que nous proposons répond correctement, selon le point de vue d'enseignants-concepteurs, aux besoins et exigences initiaux. Bien que l'API développée pour Moodle (dans le cadre global du projet GraphiT) n'entre pas directement dans le périmètre des travaux présentés, il nous paraît

nécessaire de l'exploiter afin de montrer aux participants que le fichier obtenu par le service d'export, à la fin de leur activité de scénarisation avec l'éditeur graphique, spécifie bien le contenu d'un espace-cours Moodle correspondant à leur scénario.

L'expérimentation consistera à demander à plusieurs enseignants-concepteurs de spécifier un scénario, donné initialement sous une forme textuelle, à l'aide de notre éditeur graphique, pendant une durée maximale commune prédéterminée. Chaque participant devra alors proposer une scénarisation pédagogique subjective, mais cadrée de manière à assurer qu'il ait la possibilité d'utiliser les principaux concepts, propriétés et relations de notre VIDL, ainsi que les fonctionnalités principales de l'éditeur (i.e. celles répondant à une exigence fonctionnelle identifiée). L'utilisation post-scénarisation, par nos soins, de la fonctionnalité d'export puis de l'utilisation de l'API d'import développée pour Moodle, nous permettra ensuite de montrer à chaque participant le résultat de sa conception pédagogique sous forme d'espace-cours équivalent. Ceci nous permettra alors de recueillir auprès des participants leurs opinions (adéquation entre les intentions initiales implicites dans leur scénario et la correspondance Moodle). Pour collecter ces données *qualitatives* nous pourrions utiliser, dans un premier temps, un questionnaire individuel, puis dans un second temps, proposer aux participants d'échanger à l'occasion d'une discussion collective guidée par nos soins, en exploitant le fil conducteur du questionnaire.

## **BIBLIOGRAPHIE**

Abdallah, F., Toffolon C. et Warin, B. (2008). Models transformation to implement a project-based collaborative learning (PBCL) scenario: Moodle case study. Dans P. Díaz, Kinshuk, I. Aedo et E. Mora (dir.), *Proceedings of the 8th IEEE International Conference on Advanced Learning Technologies* (p. 639-643). Santander, Espagne : IEEE Computer Society.

Abedmouleh, A., Oubahssi, L., Laforcade, P. et Choquet, C. (2008). An analysis process for identifying and formalizing LMS instructional language. Dans S. Hammoudi, M. van Sinderen et J. Cordeiro (dir.), *Proceedings of the 7th International Conference on Software Paradigm Trends* (p. 218-223). Rome, Italie : ScitePress.

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. et Dimitriadis, Y. (2012). GLUE! - GLUE!-PS: An approach to deploy non-trivial collaborative learning situations that require the integration of external tools in VLEs. Dans S. Retalis et M. Dougiamas (dir.), *Proceedings of the 1st Moodle Research Conference* (p. 77-85). Heraklion, Grèce : Moodle.

Benson, A., Lawler, C. et Whitworth, A. (2008). Rules, roles and tools: Activity theory and the comparative study of e-learning. *British Journal of Educational Technology*, 39, 456-467.

Berggren, A., Burgos, D., Fontana J.M., Hinkelman, D., Hung V., Hursh, A. et Tielemans, G. (2005). Practical and Pedagogical Issues for Teacher Adoption of IMS Learning Design Standards in Moodle LMS. *Journal of Interactive Media in Education*, 2005(1).

Botturi, L., Derntl, M., Boot, E. et Figl, K. (2006). A classification framework for educational modeling languages in instructional design. Dans *Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies* (p. 1216-1220). Kerkrade, Pays-Bas : IEEE Press.

Burgos, D., Tattersall, C., Dougiamas, M., Vogten, H. et Koper, R. (2007) A First Step Mapping IMS Learning Design and Moodle. *Journal of universal computer science*, 13, 924-931.

Conole, G., Dyke, M., Oliver, M. et Seale, J. (2004). Mapping pedagogy and tools for effective learning design. *Computers & Education*, 43, 17-33.

Dodero, J.M., Martinez del Val, A.A. et Torres, J. (2010). An extensible approach to visually editing adaptive learning activities and designs based on services. *Journal of Visual Languages & Computing* 21(6), p. 332- 346.

Dougiamas, M. et Taylor, P. (2003). Moodle: Using Learning Communities to Create an Open Source Course Management System. Dans D. Lassner et C. McNaught (dir.), *Proceedings of ED-MEDIA 2003 - World Conference on Educational Multimedia, Hypermedia & Telecommunications* (p. 171-178). Waynesville, NC : Association for the Advancement of Computing in Education.

Site officiel du projet EMP. <http://www.eclipse.org/modeling/>

Engeström, Y. (1987). *Learning by Expanding: An Activity Theoretical Approach to Developmental Research*. Helsinki, Finlande : Orienta-Konsultit Oy.

Garrison, D.R. et Kanuka, H. (2004). Blended learning: Uncovering its transformative potential in higher education. *The Internet and Higher Education*, 7, 95-105.

Jouault, F., Allilaire, F., Bézivin, J., Kurtev, I. et Valduriez, P. (2006). ATL: A QVT-like Transformation Language. Dans *Proceedings Companion to the 21st ACM SIGPLAN Conference OOPSLA'06* (p. 719-720). Portland, OR : ACM.

Katsamani, M., Retalis, S., Boloudakis, M. (2012). Designing a Moodle course with the CADMOS learning design tool. *Education Media International*, 49, 317-331.

Kelly, S. et Tolvanen, J.-P. (2008). *Domain Specific Modeling: Enabling full code generation*. Hoboken, NJ : Wiley.

Loiseau, E. et Laforcade, P. (2013). Specification of learning management system-centered graphical instructional design languages - a DSM experimentation about the Moodle platform. Dans J. Cordeiro, D. Marca et M. van Sinderen (dir.), *Proceedings of the 8th International Joint Software Conference* (p. 504-511). Reykjavik, Islande : Scitepress.

Loiseau, E., Laforcade, P. et Iksal, S. (2014). Model weaving and pedagogy mapping abstraction levels in instructional design languages. Dans A. Holzinger, J. Cardoso, J. Cordeiro, M. van Sinderen et S. Mellor (dir.), *Proceedings of the 9th International Joint Software Conference* (p. 81-86). Vienne, Autriche : Scitepress.



**Esteban LOISEAU, Pierre LAFORCADE, Nour EL MAWAS, Sébastien IKSAL**

Moody, D. (2009). The Physics of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. *IEEE Transactions on Software Engineering*, 35(6), 756-779. doi : 10.1109/TSE.2009.67.

Ormrod, J.E. (2011). *Human Learning*. Upper Saddle River, NJ : Pearson.

Paige, R.F., Kolovo, D.S., Rose, L.M., Drivalors, N. et Polack F.A.C. (2009). The design of a conceptual framework and technical infrastructure for model management language engineering. Dans *Proceedings of the 14th IEEE International Conference on Engineering of Complex Computer Systems* (p. 162-171). Postdam, Allemagne : IEEE Computer Society.

Podvin, H. et Laforcade, P. (2014). *Analyse des pratiques et des besoins de conception pédagogique centrés plateformes de formation* (livrable D3-4 du projet GraphiT). Récupéré du site du projet GraphiT: [http://www-lium.univ-lemans.fr/~laforcad/graphit/?page\\_id=166](http://www-lium.univ-lemans.fr/~laforcad/graphit/?page_id=166)

Schmidt, D.C. (2006). Model-driven engineering. *IEEE Computer*, 39, 25-31.

Site officiel du projet Sirius. <https://www.eclipse.org/sirius/>