
Une aide à la réingénierie d'un scénario pédagogique via la préconisation et la formalisation d'observables

Vincent Barré, Christophe Choquet

*LIUM (FRE 2730 du CNRS) – IUT de Laval
52, rue des docteurs Calmette et Guérin
53020 Laval Cedex 9, FRANCE
{Vincent.Barre, Christophe.Choquet}@univ-lemans.fr*

RÉSUMÉ. La démarche que nous proposons dans cet article s'inscrit dans une réflexion menée autour de la réingénierie des EIAH. Elle concerne plus précisément la réingénierie d'un scénario pédagogique en assistant les concepteurs dans la phase de définition des observables qui leur permettront de comprendre a posteriori le déroulement d'une session d'apprentissage. De ce point de vue, notre démarche présente plusieurs points forts. Tout d'abord, elle permet de préconiser des observables à partir de la structure du scénario construit par l'équipe des concepteurs. Ensuite, elle propose une façon de retranscrire ces observables dans un langage interprétable en conception, dans le but de pouvoir comparer le scénario conçu a priori à celui résultant des usages observés a posteriori. Enfin, cette démarche s'appuie sur une analyse au niveau du méta-modèle du langage de modélisation pédagogique retenu par les concepteurs et permet donc de s'abstraire du langage réellement utilisé.

MOTS-CLÉS : ingénierie, conception des EIAH, normes et standards, scénario pédagogique.

1. Introduction

La désynchronisation des deux rôles majeurs de l'enseignant – la conception de cours et le tutorat – dans un système de Formation Ouverte et A Distance (FOAD), pénalise l'optimisation itérative de la qualité du dispositif par l'insuffisance de la prise en compte des usages dans un objectif de réingénierie. Pour cette raison, nous avons proposé dans [CORBIERE & CHOQUET 04] une extension du modèle de méta-architecture IEEE-LTSA (Learning Technology Systems Architecture) [IEEE 01], [IEEE/LTSC 05]. Cette proposition intègre explicitement une étape d'observation et d'analyse des comportements du système FOAD et des acteurs du processus d'apprentissage dans un processus itératif, dirigé par les intentions de conception. Nous soulignons en particulier le besoin d'une description formelle du point de vue de conception sur le scénario, qualifié alors de *scénario prédictif*, d'une aide à l'analyse des usages par la comparaison des *scénarios descriptifs* ("décrivant a posteriori le déroulement effectif d'une situation d'apprentissage" [PERNIN & LEJEUNE 04a]) avec le scénario prédictif. Ceci produit de l'information, significative pour les concepteurs d'un point de vue pédagogique, lorsqu'ils effectuent une rétro-conception ou une réingénierie [CHIKOFSKY & CROSS II 90] de leurs systèmes.

Le rôle du concepteur dans un tel processus est alors non seulement d'établir le scénario prédictif de la situation d'apprentissage envisagée, mais également d'anticiper sur la construction des scénarios descriptifs en définissant des besoins d'observation de la situation permettant l'évaluation effective de l'activité des apprenants. Dans le cadre du projet REDiM (Réingénierie des EIAH Dirigée par les Modèles), nous nous intéressons plus particulièrement à supporter la réalisation de ces deux tâches dans un contexte où le concepteur utilise un langage de modélisation pédagogique centré activité, de la famille des EMLs (Educational Modelling Languages) [KOPER 01].

Bien qu'il existe actuellement une volonté internationale d'aboutir à un langage de modélisation pédagogique normalisé, plusieurs langages de scénarisation coexistent [RAWLINGS & al. 02], même si le langage *Learning Design* (IMS/LD) du consortium IMS [KOPER & al. 03] semble être proche du statut de standard. C'est pourquoi l'ensemble de nos travaux, bien que s'exemplifiant sur IMS/LD, veut être indépendant du langage de modélisation choisi par le concepteur.

Nous proposons, dans la suite de ce papier, un ensemble de règles raisonnant sur le méta-modèle, *i.e.* le schéma XML, d'un EML (ici IMS/LD, mais tout autre EML décrit par un schéma XML pourrait convenir) et dont les inférences sur un scénario prédictif permettent de définir un ensemble d'observables (*i.e.* d'usages à observer) pertinents à capturer durant la session. Nous proposons, en outre, une méthode de représentation des scénarios descriptifs basée sur l'extension du langage de modélisation utilisé pour l'expression du scénario prédictif. Cette méthode a l'avantage (i) de conditionner les extensions du langage à l'inférence des règles de définition des observables, (ii) de faciliter l'interprétation des scénarios descriptifs par leur comparaison avec le scénario prédictif, les deux types de scénarios étant exprimés dans le même langage, et (iii) d'être indépendante du langage de modélisation utilisé, dès l'instant que son méta-modèle est exprimé à l'aide d'un

schéma XML, ce qui est le cas pour la plupart d'entre eux. Une seconde partie est consacrée à l'exposé de deux cas d'utilisation montrant la dynamique du processus et explicitant les règles introduites.

2. Utilisation du méta-modèle pour préconiser et représenter des observables pertinents

Nous présentons, dans cette partie, une méthode permettant de préconiser des observables, dans le but de pouvoir comparer les scénarios prédictif et descriptif. L'originalité de notre approche repose sur le fait que les préconisations sont faites via l'analyse du méta-modèle de l'EML permettant d'exprimer les scénarios prédictifs, plutôt que par une analyse des scénarios eux-mêmes ou de la structure des observés (*i.e.* des usages observés) fournis par la plate-forme de diffusion.

Nous limitons ici notre approche aux EMLs exprimés sous forme de schémas XML, pour lesquels les préconisations sont déduites de la structuration des *éléments* qu'ils définissent. Plus précisément, dans un schéma XML, deux types d'éléments peuvent être définis : ceux de type *complexe* (pouvant contenir des sous-éléments et/ou être qualifiés par des attributs) et ceux de type *simple*. Dans la perspective de mettre en relief les observables liés aux éléments d'un EML quelconque (par exemple IMS/LD), ces deux types d'éléments apportent des informations différentes et complémentaires : les observables liés aux éléments de type *complexe* sont définis à l'aide de règles (par rapport au méta-modèle et à ce que l'on est en droit d'espérer retrouver dans les traces pour ces éléments), tandis que ceux liés aux éléments de type *simple* sont simplement extraits des traces informatiques. Mais, quel que soit le type de l'élément, les règles permettant de produire les observables pertinents ne dépendront pas du langage décrit par le schéma (*i.e.* de l'EML retenu).

Cet ensemble de règles s'inscrit dans une démarche d'opérationnalisation de la réingénierie d'un scénario pédagogique telle que nous l'avons déjà abordée dans [IKSAL & al. 04] et permet d'apporter des réponses opérationnelles aux réflexions sur le cycle de vie d'un scénario pédagogique [PERNIN & LEJEUNE 04b]. Cependant, les règles que nous présentons ici constituent un premier ensemble qu'il conviendra d'enrichir par la suite.

La démarche que nous proposons consiste à « enrichir » le scénario prédictif par les observations effectuées à la demande des concepteurs. Un premier groupe de règles (règles 1 à 3 ci-après) permet un enrichissement par des informations synthétiques, concernant l'ensemble des parcours observés, tandis qu'un second groupe (règles 4 à 6) permet d'attirer l'attention des concepteurs sur des configurations pouvant être signifiantes du point de vue de la réingénierie. Enfin, bien que les règles que nous proposons actuellement ne soient pas conçues en ce sens, il serait intéressant de pouvoir produire des observables permettant de retranscrire les parcours individuels des apprenants.

Règle 1 : Pour un élément de type complexe, il est intéressant de connaître le nombre d'instances de cet élément retrouvées dans les traces.

Règle 2 : Lorsqu'un élément est d'un type simple (par exemple, *integer*, *string*...), il est intéressant de connaître la (les) valeur(s) de cet élément (en tenant compte de ses différentes valeurs successives, le cas échéant).

Règle 3 : Lorsqu'un élément référence ou contient un élément pour lequel on a déterminé des observables, on importe les observables correspondants dans le contexte de l'élément. Ceci implique que tous les observables de l'élément inclus ou référencé sont accessibles, mais aussi qu'il est possible d'en obtenir une vision restreinte à ceux en rapport direct avec l'élément courant.

Règle 4 : Pour un élément de type complexe, contenant un choix parmi plusieurs éléments, il est intéressant de mettre en évidence les éléments apparaissant dans les traces.

Règle 5 : Pour un élément de type complexe, contenant une séquence d'éléments ayant une propriété *maxOccurs* non limitée, il est intéressant d'examiner le scénario descriptif afin de repérer les éléments non utilisés.

Règle 6 : Lorsqu'un élément du modèle est d'un type complexe vide, il n'y a pas d'observable correspondant.

Ces règles permettent de préconiser des observables à partir de l'analyse de la structure du schéma XML du langage utilisé par le concepteur (quel qu'il soit), nous allons maintenant montrer comment représenter ces observés dans ce même langage. Plus précisément, nous souhaitons produire un scénario « enrichi » par les informations provenant des observés [BARRE & al. 04]. Pour ce faire, l'idée est de partir du scénario prédictif, contenant l'information sur les activités proposées et le déroulement théorique de la séance, et d'y ajouter les informations extraites des traces sur la demande des concepteurs. Concrètement, pour chacune de nos règles, les observables obtenus sont ajoutés à la définition de l'élément sur lequel la règle s'applique (de sorte à enrichir le scénario prédictif). Ceci passe par la transformation du schéma XML décrivant le langage des concepteurs de sorte à y incorporer nos extensions. Soulignons que ce nouveau schéma constitue une extension du schéma originel, et que, par conséquent, le scénario prédictif est également un document valide dans notre schéma étendu.

3. Cas d'utilisation des règles et représentation des observables

Depuis 3 ans, nous analysons l'usage d'une unité d'enseignement par un public d'une cinquantaine d'étudiants de DUT Services et Réseaux de Communication à l'IUT de Laval (niveau Bac+2) [BARRE & al. 03]. Cette unité d'apprentissage, bien que supportée par un dispositif informatique dédié à la formation, *FreeStyle Learning* [BROCKE & al. 00], est diffusée en présentiel et vise à former les étudiants aux architectures logicielles des services sur réseaux. Elle comprend 7 activités : *introduire l'unité d'enseignement* (une vidéo de présentation), *approfondir les concepts* (des textes de référence), *synthétiser les concepts* (deux diaporamas), *s'exercer pour comprendre* (des exercices à résoudre), *s'auto-évaluer* (deux QCM), *apprendre en faisant* (mise en pratique de l'ensemble des connaissances du domaine et modification, étape par étape, du code Java d'un

serveur web) et *utiliser pour produire* (activité collective de conception d'un site Web utilisant le serveur HTTP précédemment développé).

Bien qu'un scénario prédictif ait été présupposé par les concepteurs, le dispositif utilisé laisse les étudiants libres de naviguer comme ils le souhaitent entre les ressources des 6 premières activités. Après chaque session, les traces obtenues sont analysées et comparées au scénario prédictif afin de l'enrichir.

3.1. Observables liés à l'usage des QCM présents dans le scénario prédictif

Dans ce premier cas d'utilisation, nous allons mettre en évidence les observables liés à l'usage des propriétés d'IMS/LD niveau B (la spécification IMS/LD comprenant une version de base, le niveau A, pouvant être enrichie par l'ajout de propriétés et de structures pour constituer le niveau B, qui peut lui-même être enrichi par un mécanisme événementiels pour constituer le niveau C de la spécification). Les propriétés apportent un mécanisme permettant de conserver des informations sur les apprenants, sur les groupes d'apprenants, ... elles permettent aussi d'individualiser les parcours selon le profil des apprenants ou les résultats qu'ils ont précédemment obtenu. Elles sont principalement utilisées et mises à jour dans la section *method* (représentant la dynamique du scénario) de deux façons : (1) lorsqu'un acteur achève un élément du scénario (activité, *act*, *play*...) et/ou (2) dans une règle conditionnelle (si telle condition est vérifiée alors telle propriété doit être mise à jour).

Dans notre mise à l'essai, les propriétés sont utilisées pour retranscrire les scores que les apprenants obtiennent aux QCM :

```
<imsld:learning-activity identifier="LA5-AutoEval-QCM1">
  <imsld:title>S' auto-évaluer (QCM1)</imsld:title>
  <imsld:environment-ref ref="LO3-QCM1.html" />
  <imsld:activity-description>
    Tester ses connaissances sur les protocoles du Web
  </imsld:activity-description>
  <imsld:complete-activity>
    <imsld:user-choice />
  </imsld:complete-activity>
  <imsld:on-completion>
    <imsld:change-property-value property-ref="P-QCM1-score"
      property-value="score"/>
  </imsld:on-completion>
</imsld:learning-activity>
```

Figure 1. Utilisation d'une propriété pour stocker le résultat d'un QCM (extrait du scénario prédictif)

Cette activité est associée aux apprenants de type *R-learner* via un élément *role-part* (ici *RPI-QCM* pour apprenants) :

```

<imsld:role-part identifiant="RPl-QCM pour apprenants">
  <imsld:role-ref ref="R-learner" />
  <imsld:learning-activity-ref ref="LA5-AutoEval-QCM1" />
</imsld:role-part>

```

Figure 2. Association entre l'activité "QCM" et les apprenants devant la réaliser (extrait du scénario prédictif)

Lorsque l'on examine *a posteriori* les traces d'une session d'apprentissage, il est possible d'en extraire la valeur prise par la variable *score* lors de son affectation. Néanmoins cette information n'a pas de réelle valeur si elle reste « brute ». Elle sera beaucoup plus informative si elle peut être contextualisée, par exemple rapprochée du QCM auquel correspond le score de l'apprenant l'ayant obtenu ou encore de la position du QCM par rapport aux autres activités. Examinons donc ce que nos règles permettent d'apporter dans ce cas.

Tout d'abord, l'application de la première règle permet de retranscrire le nombre d'instances du QCM retrouvées dans les traces. Pour ce faire, il faut tout d'abord modifier la définition de l'élément '*learning-activityType*' dans le schéma définissant IMS/LD afin de lui adjoindre un nouvel attribut '*observedOccurs*' :

```

<xs:complexType name="learning-activityType">
  <xs:sequence>
    <xs:element ref="title" minOccurs="0"/>
    <xs:element ref="learning-objectives" minOccurs="0"/>
    <xs:element ref="prerequisites" minOccurs="0"/>
    <xs:element ref="environment-ref" minOccurs="0"
      maxOccurs="unbounded"/>
    <xs:element ref="activity-description"/>
    <xs:element ref="complete-activity" minOccurs="0"/>
    <xs:element ref="on-completion" minOccurs="0"/>
    <xs:element ref="metadata" minOccurs="0"/>
  </xs:sequence>
  <xs:attributeGroup ref="attr.identifier.req"/>
  <xs:attributeGroup ref="attr.isvisible"/>
  <xs:attributeGroup ref="attr.parameters"/>
  <xs:attribute
    name="observedOccurs"
    type="xs:nonNegativeInteger"/>
</xs:complexType>

```

Figure 3. Modification de la définition de l'élément '*learning-activityType*' (dans le schéma XML définissant IMS/LD)

L'activité '*LA5-AutoEval-QCM1*' peut alors être enrichie dans le scénario descriptif :

```

<imsld:learning-activity identifiant="LA5-AutoEval-QCM1"
  observedOccurs="10">
  <imsld:title>S' auto-évaluer (QCM1)</imsld:title>
  . . .
</imsld:learning-activity>

```

Figure 4. Représentation du nombre d'occurrences observées dans le scénario descriptif

Ensuite, l'application de la règle 2 permet de retranscrire les valeurs prises par la propriété *P-QCMI-score*. Par rapport à l'exploitation directe des traces, cela présente l'avantage de pouvoir retranscrire les valeurs successives prises par une propriété (par exemple le QCM peut avoir été utilisé dans un premier temps pour évaluer les connaissances à acquérir, puis, dans un second temps, pour évaluer celles effectivement acquises). Nous voulons donc enrichir la description de l'élément '*initial-value*' (`<xs:element name="initial-value" type="xs:string"/>`) de façon à retranscrire les valeurs successives prises par la propriété. Pour ce faire, nous commençons par créer un nouveau type complexe pour décrire cet élément, *i.e.* `<xs:element name="initial-value" type="initial-valueType"/>`, le type '*initial-valueType*' étant quant à lui défini comme une extension du type de base '*xs:string*' :

```
<xs:complexType name="initial-valueType">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="history" use="optional">
        <xs:simpleType>
          <list itemType="xs:anySimpleType"/>
        </xs:simpleType>
      </xs:attribute>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

Figure 5. Définition du type '*initial-valueType*' prenant en compte l'historique des valeurs manipulées (modification du schéma XML définissant IMS/LD)

Examinons ensuite l'élément *change-property-value* et en particulier l'élément *property-value* qu'il met en œuvre :

```
<xs:complexType name="property-valueType" mixed="true">
  <xs:annotation>
    <xs:documentation xml:lang="en">The element property-value
    should contain either text, or langstring, or calculate,
    or property-ref.</xs:documentation>
  </xs:annotation>
  <xs:choice minOccurs="0">
    <xs:element ref="langstring"/>
    <xs:element ref="calculate"/>
    <xs:element ref="property-ref"/>
  </xs:choice>
</xs:complexType>

<xs:element name="langstring" type="xs:string"/>

<xs:element name="property-ref">
  <xs:complexType>
    <xs:attributeGroup ref="attr.ref.req"/>
  </xs:complexType>
</xs:element>
```

Figure 6. Définition de l'élément '*change-property-value*' (extrait du schéma XML définissant IMS/LD)

Cet élément pourra contenir, au choix, un élément de type *xs:string*, une référence vers une propriété ou le « résultat » d'un calcul. En ce qui concerne cette dernière possibilité, l'examen de l'élément *calculate* montre qu'il s'agit d'une définition récursive aboutissant à une référence sur un élément propriété, un élément *property-value*, ou à une nouvelle expression (*i.e.* une sous-expression). Quel que soit le cas, notre méthode permettra de préconiser un certain nombre d'observables :

- Si *property-valueType* correspond à une chaîne de caractères (type de base), alors l'application de la seconde règle permet de préconiser l'observation de la valeur de la propriété et éventuellement de retranscrire ses valeurs successives.
- Si *property-valueType* correspond à une référence vers une propriété, alors la troisième règle préconise d'associer les observables liés à la propriété référencée et de contextualiser cette information (ce qui n'a pas d'intérêt particulier ici, puisque seule les valeurs prises par la propriété semblent intéressantes).
- Enfin, si *property-valueType* correspond à la définition récursive précédemment évoquée, alors nos règles permettent dans tous les cas (référence à un élément propriété, élément *property-value*, nouvelle expression) de préconiser des observables.

Nous pouvons ensuite appliquer notre troisième règle afin d'importer les observables liés à une propriété dans le contexte de l'élément ayant déclenché sa modification. Ceci permet, dans un premier temps, d'importer les observables liés à la propriété *P-QCMI-score* dans le contexte de l'activité d'apprentissage *LA5-AutoEval-QCMI*, puis, dans un second temps, dans le contexte du *role-part* liant cette activité avec le rôle *R-learner*.

Les observables « importés » sont liés à l'élément inclus ou référencé, il n'y a donc rien de particulier à prévoir pour les représenter au niveau du schéma. Il faut, par contre, pouvoir représenter l'information contextualisée. Ceci peut s'effectuer de deux façons différentes selon que l'élément contient ou référence un autre élément : dans le cas de l'inclusion d'un élément, il faudra enrichir la description de cet élément, tandis que dans le cas d'un élément référencé, il faudra enrichir la définition de l'élément contenant la référence. Néanmoins, quel que soit le cas, l'enrichissement de la définition se fera d'une façon similaire et le même attribut pourra être utilisé.

```
<imsld:role-part identifier="RPI-QCM pour apprenants">
  <imsld:role-ref ref="R-learner" observedOccurs="11"/>
  <imsld:learning-activity-ref ref="LA5-AutoEval-QCMI"
                                observedOccurs="12"/>
</imsld:role-part>
```

Figure 7. Utilisation de l'attribut 'observedOccurs' pour enrichir la définition d'un élément quelconque (extrait de notre scénario descriptif)

L'exemple de la Figure 7 indique par exemple que l'activité *LA5-AutoEval-QCMI* (le QCM proprement dit) a été instanciée 12 fois dans le contexte de l'élément *RPI-QCM pour apprenants*, et que 11 apprenants (*R-learner*) ont été associés à ce *role-part*. Ces données peuvent ensuite être comparées aux nombres

totaux d'apprenants dans le rôle *R-learner* et d'instances de l'activité *LA5-AutoEval-QCM1*.

Dans notre situation expérimentale, l'activité 'LA5-AutoEval-QCM1' est en fait l'un des 3 questionnaires proposés aux apprenants. Pour représenter le nombre d'instances de chaque QCM, comme nous l'indique la règle 4, il suffit d'étendre la définition de l'élément '*activity-structureType*' d'une façon analogue à la règle 1 :

```
<imsld:activity-structure identifieur="AS5-AutoEval"
structure-type="selection" number-to-select="1">
  <imsld:title>Auto-évaluation</imsld:title>
  <imsld:learning-activity-ref ref="LA5-AutoEval-QCM1"
                                observedOccurs="7"/>
  <imsld:learning-activity-ref ref="LA5-AutoEval-QCM2"
                                observedOccurs="0"/>
  <imsld:learning-activity-ref ref="LA5-AutoEval-QCM3"
                                observedOccurs="3"/>
</imsld:activity-structure>
```

Figure 8. Utilisation de la définition étendue de l'élément '*activity-structureType*' (extrait de notre scénario descriptif)

La cinquième règle permettant enfin d'attirer l'attention du concepteur sur le fait que l'activité 'LA5-AutoEval-QCM2' n'a été effectuée par aucun des apprenants au cours de la session.

3.2. Cas de l'ordonnement des activités

Ce second cas met en évidence le parti que l'on peut tirer de la structuration arborescente des scénarios IMS/LD lorsque l'on souhaite enrichir (ou contextualiser) les observables recueillis sur les feuilles lors de leur remontée dans l'arbre. L'élément *role-part* du langage IMS/LD permet d'associer les rôles aux activités qu'ils ont à réaliser. C'est donc tout naturellement qu'il va jouer un rôle important dès lors que l'on cherchera à contextualiser des informations.

```
<xs:element name="role-part" type="role-partType"/>
<xs:complexType name="role-partType">
  <xs:sequence>
    <xs:element ref="title" minOccurs="0"/>
    <xs:element ref="role-ref"/>
    <xs:choice>
      <xs:element ref="learning-activity-ref"/>
      <xs:element ref="support-activity-ref"/>
      <xs:element ref="unit-of-learning-href"/>
      <xs:element ref="activity-structure-ref"/>
      <xs:element ref="environment-ref"/>
    </xs:choice>
    <xs:element ref="metadata" minOccurs="0"/>
  </xs:sequence>
  <xs:attributeGroup ref="attr.identifieur"/>
</xs:complexType>
```

Figure 9. Définition de l'élément *role-partType* dans le schéma XML définissant IMS/LD

L'application de la règle 3 indique que les observables liés à une *learning-activity* peuvent être remontés et enrichis au niveau d'un *role-part*. Ces observables ayant au préalable été produits au niveau de la *learning-activity* par application de la première règle.

```
<xs:element name="act" type="actType"/>
<xs:complexType name="actType">
  <xs:sequence>
    <xs:element ref="title" minOccurs="0"/>
    <xs:element ref="role-part" maxOccurs="unbounded"/>
    <xs:element ref="complete-act" minOccurs="0"/>
    <xs:element ref="on-completion" minOccurs="0"/>
    <xs:element ref="metadata" minOccurs="0"/>
  </xs:sequence>
  <xs:attributeGroup ref="attr.identiflier"/>
</xs:complexType>
```

Figure 10. Définition de l'élément *act* dans le schéma XML définissant IMS/LD

L'application de la règle 3 indique ensuite que les observables provenant du *role-part* peuvent être enrichis au niveau de l'*act*. Nous pourrions ensuite continuer à appliquer la règle 3 et faire ainsi remonter les observables au niveau du *play* puis de la *method*. Plus concrètement, dans notre expérimentation le scénario prédictif des concepteurs était le suivant :

```
<imsld:method>
  <imsld:play identiflier="PLAY-ACM">
    <imsld:title>Activité Collective : Projet Web</imsld:title>
    <imsld:act identiflier="ACT1-définition">
      <imsld:role-part identiflier="RP1-learner-répart. des roles">
        <imsld:role-ref ref="R-learner" />
        <imsld:learning-activity-ref ref="LA1-répart. roles" />
      </imsld:role-part>
      <imsld:role-part identiflier="RP2-learner-assistance tech">
        <imsld:role-ref ref="R-learner" />
        <imsld:support-activity-ref ref="SA1-Assistance tech" />
      </imsld:role-part>
      <imsld:role-part identiflier="RP3-learner-def. proj">
        <imsld:role-ref ref="R-learner" />
        <imsld:activity-structure-ref ref="AS4-définition proj" />
      </imsld:role-part>
      <imsld:complete-act>
        <imsld:when-role-part-completed ref="RP3-learner-def. proj" />
      </imsld:complete-act>
    </imsld:act>
  </imsld:play>
</imsld:method>
```

Figure 11. Scénario descriptif de l'étape 7 de notre expérimentation (extrait)

Nous souhaitons pouvoir l'enrichir grâce aux informations présentes dans les traces. La première règle nous indique que la définition de l'élément *learning-activity* doit être enrichie pour retranscrire le nombre d'instances observées de l'activité d'apprentissage. De la même façon, la définition du rôle devra être étendue pour indiquer le nombre d'acteurs tenant ce rôle.

L'application de la troisième règle permet ensuite de resituer ces informations dans le contexte d'un *role-part* particulier, par exemple de retranscrire la proportion des apprenants de type *R-learner* ayant effectivement effectué la structure d'activité *AS4-definition* (cf. Figure 11). Nous pourrions alors continuer à faire remonter l'information dans la hiérarchie en la contextualisant à chaque niveau (*act, play puis method*).

4. Conclusion

La démarche que nous proposons dans cet article présente plusieurs avantages. Tout d'abord, elle permet de préconiser des observables à partir de la structure du scénario construit par le concepteur. Ensuite, elle propose une façon de retranscrire ces observables dans un langage interprétable en conception, dans le but de pouvoir comparer le scénario *prédicatif* avec les scénarios descriptifs résultant des usages observés *a posteriori*. Enfin, cette démarche s'appuie sur une analyse au niveau du méta-modèle du langage de modélisation pédagogique retenu par les concepteurs et permet donc de s'abstraire du langage réellement utilisé.

Notons, par ailleurs, que les règles que nous présentons ici identifient des opportunités d'observation, ce qui amène nécessairement le concepteur à une réflexion sur le réel besoin d'observation. Ceci participe à notre sens à une meilleure intégration des acteurs du cycle de vie d'un EIAH, à une mise en relation du besoin d'observation prescrit par le concepteur avec le moyen d'observation, généralement mis en place par un autre acteur. A ce titre, nous avons défini des outils logiciels dans [IKSAL & al. 04] qui, une fois développés, permettront d'analyser les données recueillies en se basant à la fois sur le scénario prédictif et sur une description formelle des éléments à observer. Ils construiront des représentations formelles du comportement des usagers sur la base de l'énoncé des besoins d'observation.

5. Bibliographie

- [BARRE & al. 03] Barré V., Choquet Ch., Corbière A., Cottier Ph., Dubourg X. & Gounon P., « MOCA, une approche expérimentale de l'ingénierie des EIAH », *Environnements Informatiques pour l'Apprentissage Humain EIAH 2003*, Strasbourg 2003.
- [BARRE & al. 04] Barré V., Choquet Ch., Iksal S., Corbière A., «Usage analysis in an e-learning system: LD representation significance», *IEEE International Conference on Advanced Learning Technologies ICALT'2004*, 30 août-1^{er} septembre 2004, Joensuu (Finlande), p. 570-574.
- [BROCKE & al. 00] BROCKE J.V., NORMAN L., *FreeStyle Learning A new way to Computer Assisted Learning (CAL)*. Vancouver 2000.
- [CORBIERE & CHOQUET 04] Corbière A., Choquet C., «Designer integration in training cycles: IEEE LTSA model adaptation», *International Conference on Computer Aided Learning in Engineering Education CALIE'04*, Grenoble, 16-18 février 2004, p. 51-62.
- [CHIKOFSKY & CROSS II 90] Chikofsky E.J., Cross II J.H., «Reverse engineering and design recovery: A taxonomy», *IEEE Software*, 1990, 7(1), p. 13-17.

- [IEEE 01] IEEE P1484.1/D9, 2001-11-30 Draft Standard for Learning Technology - Learning Technology Systems Architecture (LTSA), IEEE Computer Society, 2001.
- [IKSAL & al. 04] Iksal S., Barré V., Choquet C., Corbière A., «Comparing prescribed and observed for the re-engineering of elearning systems», *IEEE Sixth International Symposium on Multimedia Software Engineering MSE*, 13-15 décembre 2004, Miami (USA).
- [KOPER 01] Koper R., «Modeling units of study from a pedagogical perspective: the pedagogical meta-model behind EML», Open University of the Netherlands, 06/2001.
- [KOPER & al. 03] Koper R., Olivier B., Anderson T. (Eds.), «IMS Learning Design Information Model», *IMS Global Learning Consortium, Inc.*, version 1.0, 20/01/2003.
- [PERNIN & LEJEUNE 04a] Pernin J-P., Lejeune A., « Dispositifs d'apprentissage instrumentés par les technologies : vers une ingénierie centrée sur les scénarios », *actes du colloque TICE 2004*, Compiègne, octobre 2004, p. 407-414.
- [PERNIN & LEJEUNE 04b] Pernin J-P., Lejeune A., « Scénarios d'apprentissage : quelles stratégies de réutilisation pour les enseignants ? », *actes du colloque TICE Méditerranée*, Nice, novembre 2004, p. 48.
- [RAWLINGS et al. 02] Rawlings A., Rosmalen P., Koper R., Rodriguez-artacho M., Lefrere P., « Survey of Educational Modelling Languages (EMLs) », *CEN/ISSS WS/LT Learning Technologies Workshop*, version 1, 19/09/2002, p 80.

6. Références sur le WEB

- [IEEE/LTSC 05] <http://ltsc.ieee.org/>
- [IMS/LD 05] <http://www.imsglobal.org/learningdesign/>