

# VUE D'ENSEMBLE DE NOTRE APPROCHE D'ASSISTANCE

## 1. Introduction

Dans ce chapitre nous allons présenter notre approche d'assistance aux auteurs des objets d'apprentissage composés de type SCORM ou SIMBAD. Cette approche comporte cinq phases d'assistance et d'aide aux auteurs, à savoir : l'assistance à la recherche des objets d'apprentissage, l'assistance à la composition, l'assistance à la compréhension, le calcul automatique des métadonnées éducatives et la vérification de la conformité à des patrons de conception. Après avoir donné la vue d'ensemble de notre approche, nous allons détailler les phases de recherche des objets à réutiliser et de calcul automatique des métadonnées. Nous terminons le chapitre par la présentation d'un algorithme de transformation d'un objet SCORM en un équivalent SIMBAD ce qui nous permet d'appliquer notre approche à un plus grand nombre d'objets et d'étendre la portée pratique de nos contributions.

## 2. Vue d'ensemble

### 2.1. Présentation

Nous proposons d'adopter une approche assistée de conception et de développement d'objets d'apprentissage composés [Farhat et al., 2009]. Cette approche comporte cinq phases. Lors de la première phase l'approche propose des mécanismes de recherche et de découverte d'objets d'apprentissage candidats à la réutilisation. L'auteur doit choisir un ensemble d'objets qui doivent couvrir le ou les concepts à traiter (ex. algèbre relationnelle) et les rôles éducatifs à couvrir (ex. exercice, exemple, description, introduction). Il peut ensuite passer à la deuxième phase dans laquelle il va concevoir le graphe de composition abstrait en utilisant les objets retenus lors de la première phase et les opérateurs d'assemblage. Suite à cette phase l'auteur peut revenir à la phase deux s'il souhaite faire des modifications sur l'ensemble des objets à réutiliser. Sinon, il passe à la phase trois dans laquelle une analyse automatique est effectuée sur l'objet créé afin de permettre à l'auteur de mieux comprendre le résultat de la composition.

L'analyse effectuée au niveau de la troisième phase a pour objectif de générer une cartographie détaillée sur l'objet en question. Il s'agit d'offrir à l'auteur des indicateurs divers et variés qui vont lui permettre d'avoir une meilleure vue sur les différentes facettes de l'objet d'apprentissage en cours de conception. En particulier, il aura une analyse de la vue système (graphe de composition concret) et de la vue apprenant (graphe affichable).

Une fois que les analyses sont satisfaisantes l'auteur passe à la quatrième phase. Celle-ci permet d'enrichir les métadonnées déjà introduites par l'auteur par des métadonnées complémentaires calculées automatiquement en se basant sur les métadonnées éducatives des objets utilisés dans la composition de l'objet.

La cinquième phase consiste à confronter l'objet d'apprentissage composé à des patrons de conception. Ceux-ci permettent de décrire certains critères structurels et sémantiques recherchés. La vérification de conformité doit permettre de détecter la nature et les causes de la non-conformité si c'est le cas. L'auteur sera donc en mesure de revenir à l'une des phases précédentes pour apporter les modifications nécessaires permettant à l'objet de s'aligner avec les directives. Tout ceci se déroule d'une façon itérative et incrémentale vers la mise au point de l'objet d'apprentissage composé.

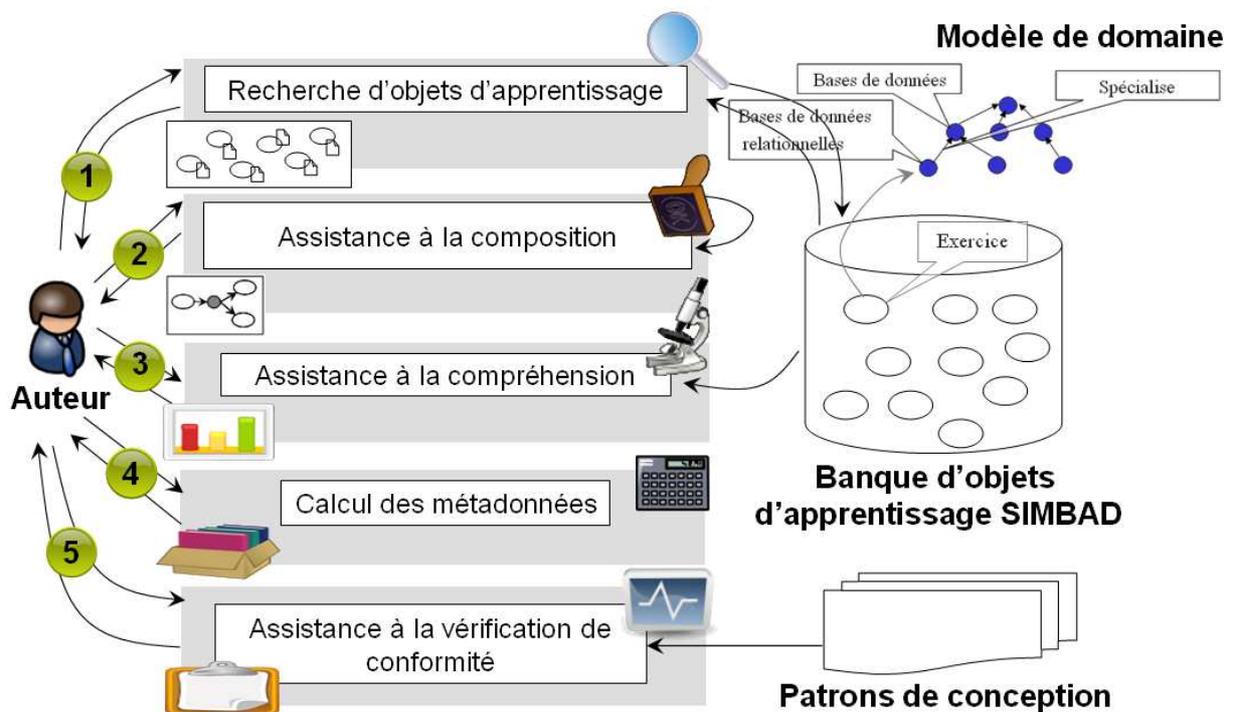


Figure 14 : Vue d'ensemble de notre approche d'assistance

La figure ci-dessus résume les différentes phases de notre approche d'assistance des auteurs lors de la conception d'objets d'apprentissage composés.



### **3. Présentation des phases recherche et calcul de métadonnées**

Dans cette thèse nous allons investiguer principalement les phases vérification de la cohérence, de l'analyse et de la vérification de la conformité. Toutefois, nous allons présenter certains aspects relatifs aux phases de recherche et de calcul automatique de métadonnées.

#### **3.1. Assistance à la recherche**

Lors de la conception d'un objet d'apprentissage composé il s'agit de réutiliser des objets d'apprentissage existant et de les combiner ensemble suivant une certaine logique. Ces objets sont traditionnellement stockés dans des banques d'objets d'apprentissage. Ces banques offrent le plus souvent un moteur de recherche permettant à l'utilisateur de trouver les objets d'apprentissage qui répondent à certains critères.

En effet, lors de l'ajout d'un objet d'apprentissage au niveau d'une banque d'objets d'apprentissage les métadonnées sont extraites et sauvegardées. Ainsi, la résolution de toute requête de recherche se base sur les informations disponibles au niveau des métadonnées éducatives des objets d'apprentissage.

Dans le cas de l'approche SIMBAD les métadonnées éducatives comportent en particulier des indications sur la sémantique et la structure de l'objet en plus des métadonnées du standard LOM. Comme il a été avancé dans la partie introductive de l'approche SIMBAD, les métadonnées sémantiques concernent en particulier le contenu, les pré-requis nécessaires et la fonction d'acquisition permettant de mettre à jour le profil de l'apprenant qui a validé le contenu de l'objet.

Nous définissons deux modes de recherche : la recherche intentionnelle et la recherche extensionnelle. La recherche intentionnelle est celle qui est utilisée principalement lors de la première itération lors de la conception d'un objet d'apprentissage composé. L'objectif est de repérer les objets d'apprentissage à réutiliser.

La requête est formulée via la description de l'intention de l'auteur en termes de contenu de pré-requis et de fonction d'acquisition.

Ainsi, l'auteur peut formuler ses requêtes pour récupérer les objets à réutiliser. Principalement, cette recherche est guidée par la sémantique. En effet, l'auteur fixera en premier lieu le concept auquel il s'intéresse. Ensuite, il précisera à chaque fois le rôle de l'objet recherché et son niveau de complexité. Il peut également faire entrer en jeu d'autres paramètres tels que le temps d'apprentissage, le média, la langue, la taille, etc.

Toutefois, nous prenons comme hypothèse que l'auteur a accès à un large corpus d'objets d'apprentissage disposant des métadonnées éducatives nécessaires pour assurer leur accessibilité. Cette hypothèse est à notre avis tout à fait réaliste en partant du principe qu'un grand nombre de ressources éducatives est déjà accessible sur Internet. D'autre part, nous soulignons que de plus en plus d'initiatives d'interconnexion et de partage entre banques d'objets d'apprentissage sont en train de voir le jour, tels que eduSource<sup>6</sup> ou VLORN<sup>7</sup>. Cependant, au cas où l'auteur se trouve avec des objets qui manquent il peut les ajouter lui-même. Ceci va permettre d'une part à l'auteur de ne s'investir que dans un contenu qui n'existe pas, ce qui présente un gain considérable en termes de temps. Et d'autre part, ceci doit encourager les auteurs à contribuer, et d'une façon distincte, à deux niveaux de granularité. La contribution à granularité fine consiste principalement en un effort de production de contenu sous forme d'exercice, de schéma illustratif, de démonstration, etc. La contribution à granularité large consiste principalement en un effort d'ingénierie pédagogique en organisant des éléments de contenu d'une façon particulière afin de permettre à l'apprenant d'atteindre de la façon la plus efficace possible un objectif pédagogique.

### **3.2. Calcul de métadonnées**

Malgré l'importance des métadonnées éducatives, les renseigner est considéré par les auteurs comme l'une des tâches les plus fastidieuses à accomplir. Ceci peut être expliqué par le nombre important de catégories et d'entrées dans les profils d'application LOM. Dans le cas de SIMBAD la tâche est encore plus difficile dans la mesure où il faut également renseigner les métadonnées sémantiques.

Dans le cadre de notre approche nous proposons d'assister l'auteur grâce au calcul automatique des métadonnées éducatives. En fait, dans le cas des objets d'apprentissage composés nous disposons déjà de certaines informations concernant les métadonnées grâce à la collecte et l'utilisation des métadonnées des objets qui le forment.

Il faut noter que certains champs de métadonnées peuvent facilement être calculés. C'est le cas par exemple des mots clés qui peuvent être générés en appliquant la formule suivante : les mots clés de l'objet d'apprentissage composé sont l'union des mots clés donnés par l'auteur, et les mots clés des objets d'apprentissage réutilisés.

---

<sup>6</sup> <http://edusource.netera.ca>

<sup>7</sup> <http://pre2005.flexiblelearning.net.au/projects/resources/2005/vlorn/>

Toutefois le calcul de certaines métadonnées est plus compliqué. C'est le cas du calcul automatique des pré-requis. Rappelant que les pré-requis sont décrits sous la forme d'un triplet  $\langle \text{Contenu}, \text{Rôle}, \text{Niveau} \rangle$ . Pour parvenir à produire automatiquement les pré-requis nous avons opté pour une approche ascendante par rapport au graphe de composition abstrait. Nous allons utiliser la notation suivante :

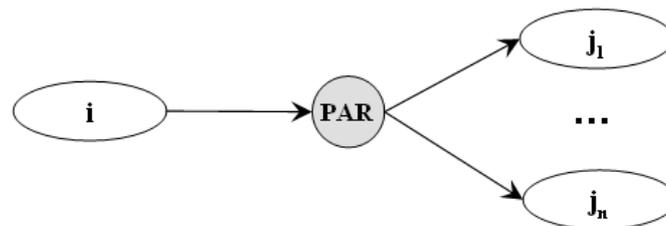
- $P_k$  est l'ensemble de triplets  $[c, r, l]$  ( $c$ : un concept,  $r$ : un rôle,  $l$ : un niveau) qui représente les pré-requis d'un objet d'apprentissage  $k$ .
- $A_k$  est l'ensemble de triplets  $[c, r, l]$  ( $c$ : un concept,  $r$ : un rôle,  $l$ : un niveau) qui représente la fonction d'acquisition de l'objet d'apprentissage  $k$ .

Pour calculer l'ensemble  $P_k$  il faut commencer à partir des feuilles du graphe de composition concret pour remonter vers la racine. Pour chaque itération de l'algorithme de parcours on peut se trouver dans l'un des trois cas :

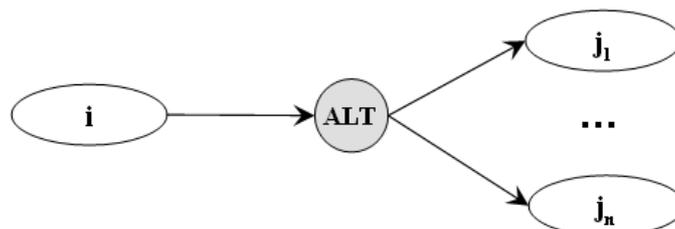
- Le premier cas correspond à la situation où deux objets  $i$  et  $j$  soient reliés dans le graphe de composition par un opérateur SEQ :



- Le deuxième cas correspond à la situation où l'objet  $i$  précède plusieurs objets auxquels il est attaché via l'opérateur PAR :



- Le troisième cas correspond à la situation où l'objet  $i$  précède plusieurs objets auxquels il est attaché via l'opérateur ALT :



L'objectif est de calculer à chaque étape l'ensemble  $P'_i$  tel que  $i$  est le sommet de type objet d'apprentissage en cours d'analyse. Le dernier sommet  $i$  est la racine du graphe de composition de l'objet  $k$ .

L'ensemble des pré-requis de l'objet d'apprentissage composé  $k$  (noté  $P'_k$ ) est l'union de l'ensemble des pré-requis renseignés par l'auteur ( $P_k$ ) et de l'ensemble des pré-requis calculés automatiquement ( $P'_i$ ) :

$$P'_k = P_k \cup P'_i$$

Le calcul démarre au niveau des feuilles du graphe en remontant vers la racine. Pour les sommets feuilles il faut noter que  $P'_i = P_i$ , alors pour le reste des sommets il faut appliquer l'une des formules suivantes selon le cas.

Pour le premier cas de figure (l'opérateur est SEQ) l'ensemble des pré-requis est formé par l'ensemble des pré-requis de l'objet  $i$  auxquels il faut ajouter les pré-requis de l'objet  $j$  qui ne sont pas déjà satisfaits par la fonction d'acquisition de l'objet  $i$  :

$$P'_i = P_i \cup (P'_j - A_i)$$

Pour le deuxième cas de figure (l'opérateur est PAR) et pour le troisième cas de figure (l'opérateur est ALT) l'ensemble des pré-requis est formé par l'ensemble des pré-requis de l'objet  $i$  auxquels il faut ajouter les pré-requis de tous les objets suivants et qui ne sont pas déjà satisfaits par la fonction d'acquisition de l'objet  $i$  :

$$P'_i = P_i \cup \left( \bigcup_{x \in [1..n]} P'_x - A_i \right)$$

Le calcul des pré-requis s'arrête suite au calcul de l'ensemble  $P'_i$  lorsque  $i$  est le nœud racine.

Il faut noter également que nous proposons que le calcul automatique des métadonnées soit effectué suite à l'analyse sémantique et structurelle de l'objet composé. Rappelons que l'analyse sémantique est basée sur une utilisation des métadonnées sémantiques. Nous préférons que ces métadonnées soient celles renseignées explicitement par l'auteur. Ainsi, toute incohérence sémantique viendra soit de la façon avec laquelle l'objet a été assemblé (l'ordre suivant lequel les objets réutilisés ont été interconnectés), soit à cause d'un renseignement imprécis, erroné ou incomplet de la part de l'auteur. Il nous semble que calculer automatiquement les métadonnées avant cette phase peut masquer certains défauts alors que l'objectif est de justement les mettre en évidence. Une fois que l'objet est cohérent et qu'il respecte les règles qui seront énoncées dans le chapitre suivant, le processus de calcul

automatique de métadonnées permettra d'enrichir davantage la description de l'objet en profitant des métadonnées des objets réutilisés. Ces métadonnées seront présentées à l'auteur pour vérification, éventuellement modification et validation. Ceci présente un gain de temps et une simplification de la tâche par rapport à un renseignement explicite par l'auteur de toutes les métadonnées.

## **4. Transformation d'objets SCORM en objets SIMBAD**

### **4.1. Objectif de la transformation**

Actuellement la spécification SCORM<sup>8</sup> est considérée comme un standard de facto [Ostyn, 2003]. Il nous semble important de pouvoir transformer des objets SCORM en des objets SIMBAD. Ainsi, tout ce que nous allons définir pour SIMBAD sera donc applicable aux objets SCORM. En particulier, il sera possible d'appliquer notre approche d'assistance afin de composer un nouvel objet par réutilisation d'objets SCORM transformés en objets SIMBAD.

Afin de pouvoir étudier la faisabilité de cette transformation il est nécessaire d'étudier les spécificités liées principalement à la navigation dans un contenu SCORM. Ceci doit nous permettre de définir un algorithme pour la transformation. En fait, cet algorithme doit pouvoir permettre de générer le graphe de composition abstrait de l'objet SCORM décrit conformément à l'approche SIMBAD. Concernant les métadonnées il n'y a pas de transformations particulières à faire. En fait, SCORM ne prévoit pas l'utilisation de métadonnées sémantiques et donc nous allons considérer qu'elles ne sont pas renseignées. Quant aux métadonnées éducatives informationnelles elles ne nécessitent pas de transformations particulières puisque SIMBAD utilise un profil d'application LOM [IEEE, 2002].

Il faut noter également que la version SCORM 2004 permet l'utilisation de la spécification IMS Simple Sequencing [IMS, 2003b] [ADL, 2004d]. Celle-ci est sensée permettre d'offrir à l'auteur les moyens nécessaires pour mettre en place certaines stratégies d'apprentissage. Le terme « Simple » fait référence au fait que le modèle ne prend en compte qu'un nombre limité de ces stratégies. Celles-ci doivent influencer la façon avec laquelle l'objet est parcouru. Toutefois, nous allons nous contenter des informations relatives au modèle d'agrégation de contenu IMS Content Packaging [IMS, 2004] [ADL, 2004b]. En fait, nous avons constaté que

---

<sup>8</sup> <http://www.adlnet.gov/Technologies/scorm/default.aspx>

l'IMS Simple Sequencing connaît des difficultés à se faire adopter vu la complexité de sa mise en œuvre et le nombre important de paramètres à fixer qu'elle comporte, et ceci sans toutefois qu'elle soit capable de supporter une large variété de stratégies d'apprentissage. Bien que certains systèmes annoncent leur conformité à SCORM 2004, nous n'avons pas trouvé (en Juin 2009) un système de gestion d'apprentissage qui supporte le « simple sequencing ». Ceci s'explique par le fait qu'il existe trois niveaux de conformité pour ces systèmes [ADL 2009]. La majorité des systèmes sont conformes au premier niveau (LMS RTE 1.0), alors qu'il faut être conforme au troisième niveau (LMS SN 1.0) pour pouvoir supporter le « simple sequencing ».

Ainsi, nous allons tout d'abord étudier les subtilités liées au modèle d'agrégation de contenu du standard SCORM. Puis nous allons proposer notre algorithme de transformation d'un objet SCORM en un objet SIMBAD équivalent.

## **4.2. Modèle d'agrégation de contenu**

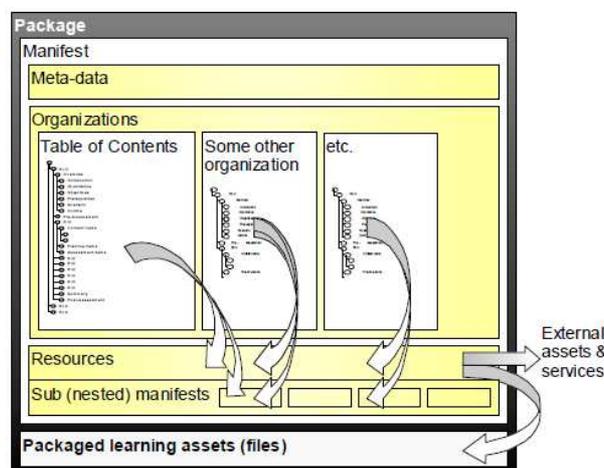
La spécification SCORM offre à l'auteur la possibilité de décrire plusieurs parcours d'un objet d'apprentissage en se basant sur la spécification de paquetage de contenu d'IMS [IMS 2004]. Cette spécification offre à l'auteur la possibilité de définir plusieurs parcours, au moins un, appelés organisations. Chaque organisation comporte une suite d'éléments qui se déclinent en deux catégories. La première catégorie concerne les éléments de type contenu alors que la deuxième concerne les conteneurs.

Les éléments contenus se déclinent à leur tour en deux sous catégories. D'une part on a les « assets » et d'autre part on a les SCO (Sharable Content Object). Les « assets » sont des média élémentaires qui peuvent être visualisés par un navigateur Web. Par exemple les fichiers HTML, les documents PDF, les animations flash et les applets Java sont considérés comme étant des « assets ». La deuxième sous catégorie de contenu est celle des SCO. Il s'agit d'un ou de plusieurs « assets » capables de communiquer avec un système de gestion de l'apprentissage conforme à SCORM et qui ne contient pas de liens vers un autre objet d'apprentissage. Concrètement, il faut rattacher à la ressource un code JavaScript permettant la communication avec l'API SCORM [ADL 2004c]. En fait, les SCO à leur tour se déclinent en deux sous-catégories. La première c'est les SCO minimaux. Il s'agit d'un contenu capable d'être lancé via un navigateur Web et utilisant l'API (Application Program Interface) SCORM pour une communication minimale avec le système de gestion de l'apprentissage. Ce dernier est ainsi capable de surveiller le temps entre le lancement du contenu et sa terminaison

normale. La deuxième sous-catégorie de SCO est les « data-enabled SCO ». Elle est différente de la première sous-catégorie dans la mesure où l'API SCORM est utilisée pour échanger des données avec le système de gestion de l'apprentissage. Ces données peuvent concerner l'apprenant, le contenu, etc.

Concernant les éléments conteneurs au sein d'une organisation de contenu c'est soit une référence à un « sub-manifest », soit un nœud qui à des nœuds fils. Dans le cas d'un « sub-manifest » il s'agit d'un ensemble d'organisations (au moins une) avec les mêmes principes d'organisation que l'organisation qui lui fait appel. Ces organisations, contiennent à leur tour des éléments de type contenu ou conteneurs. Toutefois, les feuilles de l'arbre qui représente l'agrégation du contenu dans un « sub-manifest » sont obligatoirement des nœuds de type contenu, donc des SCO ou des « assets ». Pour ce qui est un conteneur qui est un nœud avec des fils, son seul rôle est de permettre une meilleure structuration de contenu sans que ça influence réellement la façon avec laquelle un auteur désire que le contenu soit parcouru. Toutefois dans la version SCORM 1.2 ce nœud peut faire référence à une ressource qui joue généralement le rôle d'une page de garde. Dans la version SCORM 2004 seules les feuilles peuvent faire référence à une ressource.

SCORM spécifie comment concevoir un paquetage mais sans préciser comment un système de gestion de l'apprentissage doit utiliser les différentes organisations de contenu disponibles. Toutefois, la spécification précise qu'une fois qu'une organisation est choisie, par l'apprenant ou par le système de gestion de l'apprentissage, l'apprenant doit pouvoir accéder à tout objet (SCO) défini dans l'organisation. Cependant chaque organisation de contenu reflète la vision auteur concernant la façon avec laquelle l'auteur désire que l'apprenant navigue dans le contenu.



**Figure 16** : Vue d'ensemble d'un paquetage SCORM

### 4.3. Algorithme de transformation

Dans le cas de l'approche SIMBAD c'est le graphe de composition abstrait qui permet à un auteur d'objet d'apprentissage composé de décrire les parcours possibles d'un objet d'apprentissage. Pour transformer les organisations proposées à un auteur d'objets d'apprentissage SCORM en un objet composé SIMBAD nous proposons un algorithme qui opère en deux étapes.

Lors de la première étape il transforme toute organisation dans le paquetage en un chemin élémentaire dont le nœud racine est le premier élément de l'organisation et le dernier nœud est le dernier élément de l'organisation. Ainsi tous les nœuds sont reliés les uns aux autres par l'opérateur SEQ fidèlement au parcours proposé initialement par l'auteur. Les éléments de type SCO ou Asset sont considérés comme des objets d'apprentissage atomiques, alors que les éléments qui font référence à un « sub-manifest » sont considérés comme des objets d'apprentissage composés. Pour ceux-ci leur graphe de composition abstrait est déduit par le même algorithme. Pour y parvenir nous utilisons un algorithme de parcours d'arbre en profondeur à gauche pour chaque organisation.

Lors de la deuxième étape il vérifie s'il y a plusieurs organisations dans le paquetage. Dans ce cas, il définit un nœud supplémentaire de type objet d'apprentissage atomique. Celui-ci est un fichier (par exemple HTML) qui contient quelques éléments extraits des métadonnées (par exemple le titre, la description et les noms des auteurs). Cet objet sera le nœud racine du graphe de composition. Il sera relié à chacun des nœuds racine des organisations via un opérateur ALT. Ceci traduit la sémantique décrite dans la spécification SCORM concernant l'interprétation de l'existence de plusieurs organisations. Ci-après l'algorithme définit d'une façon plus formelle.

Soient :

- NO le nombre d'organisations dans un paquetage SCORM.
- NE la liste de nombres d'éléments par organisation, avec NE [j] est le nombre d'éléments dans la  $j^{\text{ième}}$  organisation.
- EP les éléments du paquetage avec EP [j] [k] est le  $k^{\text{ième}}$  élément de la  $j^{\text{ième}}$  organisation.
- T l'ensemble des transitions du graphe de composition abstrait, avec T [1] [j] est le nœud origine de la  $j^{\text{ième}}$  transition et T [2] [j] est le nœud extrémité de cette transition.

**Début**

j :=1

// Commencer par traduire les organisations en des transitions

**Pour i de 1 à O Faire**

k :=1

**Tant que k < NE [i] Faire**

T [1] [j] := EP [i] [k]

T [2] [j] := EP [i] [k+1]

k++

j++

**Fin tant que****Fin pour**

// Traiter le cas où on a plusieurs organisations

**Si O > 1 Alors**

// Créer un opérateur, ici c'est un ALT

op := créerOpérateur(ALT)

// Créer un nœud de type objet d'apprentissage qui contient quelques

//métadonnées, tel que le titre et les auteurs

root := créerRoot()

T [1] [j] := root

T [2] [j] := op

j++

**Pour i de 1 à O Faire**

T [1] [j] := op

T [2] [j] := EP [i] [1]

j++

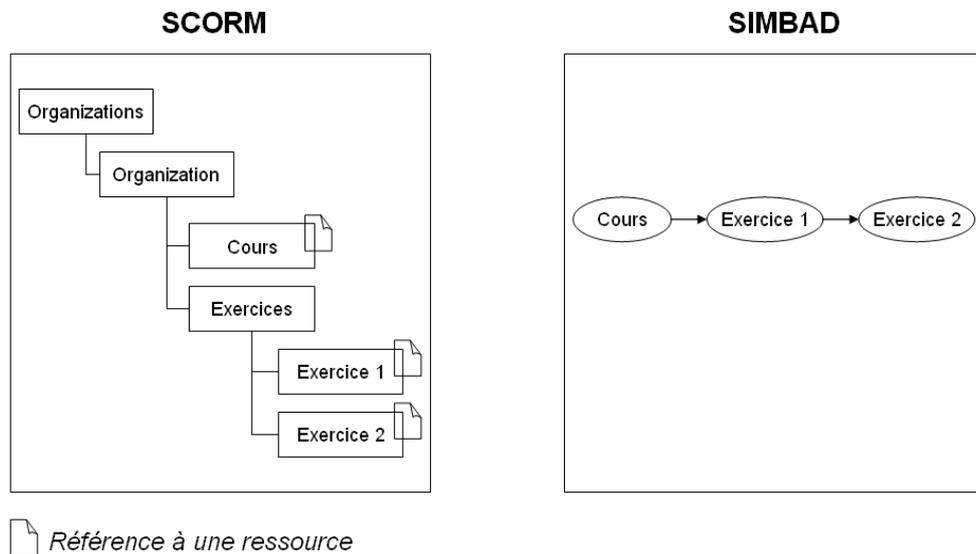
**Fin pour****Fin si****Fin**

---

#### 4.4. Exemples de transformation

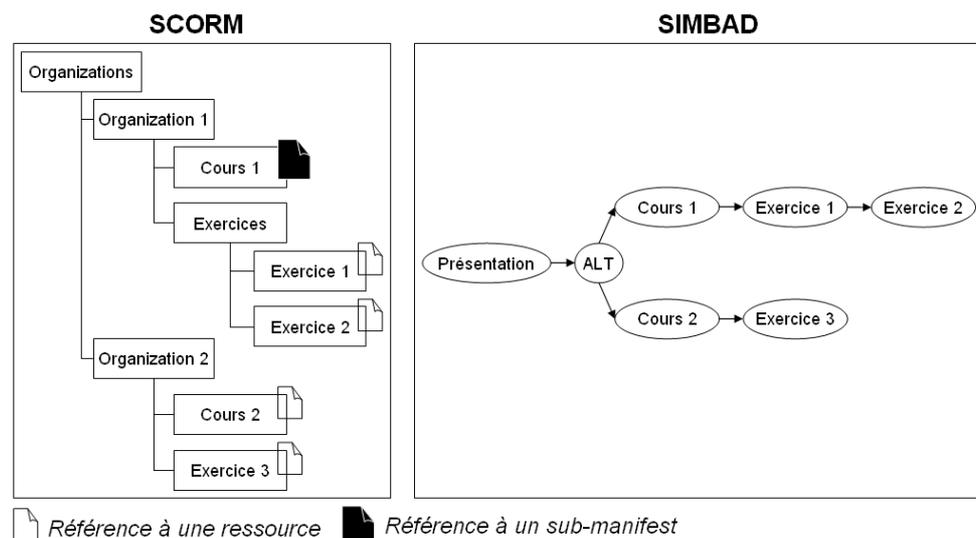
Afin d'illustrer l'algorithme nous proposons deux exemples. Le premier concerne un objet SCORM contenant une seule organisation qui ne comporte pas d'éléments de type « sub-

manifest ». L'application de l'algorithme à cet objet permet de produire un graphe de composition abstrait conforme au modèle SIMBAD. La seule organisation proposée par ce contenu est traduite par un graphe comportant trois sommets de type objet d'apprentissage et deux opérateurs SEQ indiquant l'ordre de parcours désiré par l'auteur de l'objet.



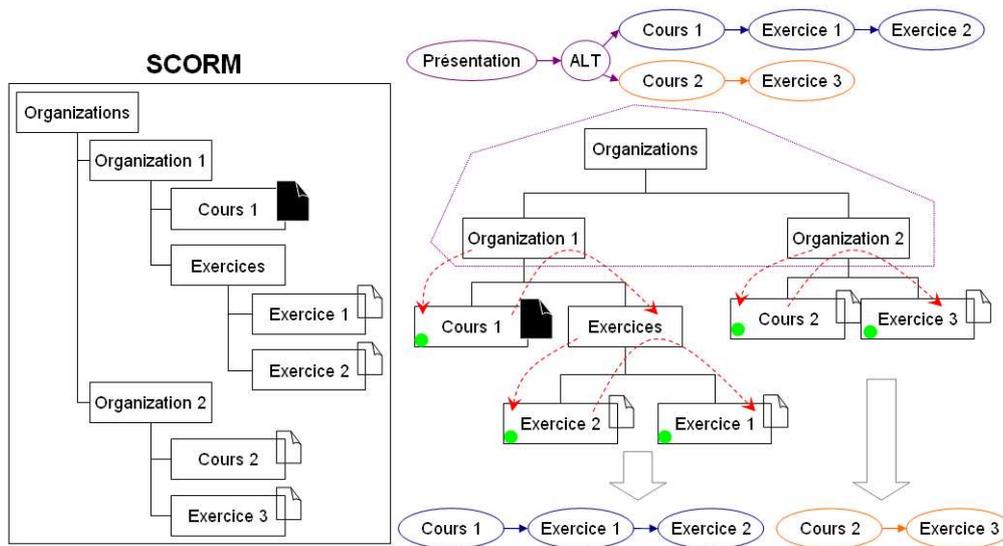
**Figure 17 :** Exemple de transformation d'un contenu SCORM simple en SIMBAD

Le deuxième exemple illustre la transformation d'un objet SCORM contenant deux organisations dont l'une contient un « sub-manifest ». Chaque organisation est traduite tout d'abord en un chemin. Un sommet racine est créé. Il contient une présentation sommaire de l'objet dont le contenu est issu des métadonnées LOM. Ce sommet racine est à l'origine d'une transition dont l'extrémité est un opérateur ALT qui donne accès aux deux chemins qui représentent les deux organisations proposées par l'auteur.



**Figure 18 :** Exemple de transformation d'un contenu SCORM complexe en SIMBAD

La figure suivante illustre le déroulement de l'algorithme :



**Figure 19 :** Transformation d'un contenu SCORM en SIMBAD

En conclusion, nous avons montré dans cette section que tout objet SCORM peut se transformer en un objet SIMBAD équivalent. Le contraire n'est cependant pas vrai puisque SIMBAD permet de définir des graphes de composition plus complexes, non supportés par SCORM (par l'utilisation de l'opérateur PAR ou le fait que les opérateurs peuvent être combinés récursivement).

## 5. Conclusion

Dans ce chapitre nous avons commencé par donner une vue d'ensemble de notre approche d'assistance aux auteurs d'objets d'apprentissage composés. Cette description a mis en évidence les différentes phases et la logique d'assistance proposée.

Nous avons détaillé deux phases de l'approche d'assistance : la recherche et le calcul automatique des métadonnées. Ces deux phases ont été présentées à ce niveau afin de détailler ensuite les phases qui comportent plus d'éléments à analyser.

Nous avons, ensuite proposé un algorithme de transformation d'un objet SCORM en un objet SIMBAD. Cette transformation permettra de profiter de l'approche d'assistance pour la mise au point d'un objet SCORM sans investissement au niveau de l'approche même. De plus, ceci permettra aussi de récupérer des objets SCORM pour leur ajouter la couche sémantique et construire ainsi une banque d'objets SIMBAD.

Le chapitre suivant sera ainsi consacré à la phase de vérification automatique de la cohérence structurelle et sémantique d'un objet d'apprentissage composé.