

VERIFICATION DE LA COHERENCE

1. Introduction

L'étape de composition d'un objet d'apprentissage peut se traduire par la définition de sa structure via un graphe orienté connexe et acyclique et de sa sémantique via la définition de ses métadonnées et en particulier les métadonnées sémantiques.

Nous proposons d'assister l'auteur durant cette phase pour vérifier les aspects structurels et sémantiques de l'objet composé. Cette vérification nécessite d'une part l'identification des contrôles à effectuer et d'autre part les moyens nécessaires pour y parvenir.

2. Vérification de la cohérence structurelle

A partir de l'ensemble d'objets d'apprentissage retenus lors de la première phase, l'auteur définit un nouvel objet en les combinant ensemble. Dans le cadre de l'approche SIMBAD cette combinaison d'objets d'apprentissage se manifeste sous forme d'un graphe nommé « graphe de composition abstrait ». La connexion des objets d'apprentissage réutilisés se fait par le biais des opérateurs : ALT, PAR et SEQ.

Cette étape de la phase de conception nécessite un certain nombre de vérifications que ce soit au niveau structurel ou sémantique [Bouzeghoub, 2009].

2.1. Règles de cohérence structurelle

Concernant les aspects structurels, dans le cadre de l'approche SIMBAD six règles de cohérence structurelles doivent être vérifiées au niveau du graphe de composition abstrait.

1^{ère} règle : Le graphe doit avoir un seul point d'entrée de type objet d'apprentissage.

En fait, l'idée est d'avoir un point d'entrée unique. A partir de celui-ci, et selon le profil de l'apprenant, un chemin particulier de navigation va être retenu. Cette idée est empruntée au cas des sites Web qui offrent un point d'entrée unique qui est la page d'accueil à partir de laquelle l'internaute commence la navigation dans le site Web.

Le point d'entrée unique doit être un objet d'apprentissage et il est caractérisé par le fait qu'il n'est pas à l'extrémité d'aucun arc.

Voici un exemple d'un graphe de composition abstrait non conforme puisqu'il comporte plusieurs points d'entrée de type objet d'apprentissage :

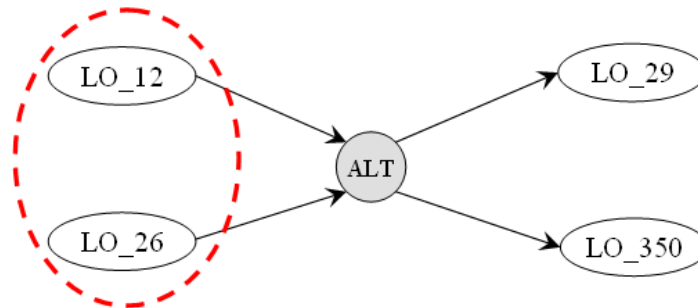


Figure 20 : Exemple de violation de la première règle de cohérence SIMBAD

2^{ème} règle : Chaque opérateur ALT ou PAR doit être à l'origine d'au moins deux arcs.

En fait les opérateurs ALT et PAR expriment respectivement la présence d'alternatives et de choix. L'alternative et le choix doivent se faire par rapport au moins deux chemins. Au niveau structurel ceci se manifeste par le fait que dès qu'il y a un opérateur PAR ou ALT, il y a présence d'au moins deux arcs dont il est la source

L'exemple suivant est un graphe de composition abstrait où cette règle est violée.

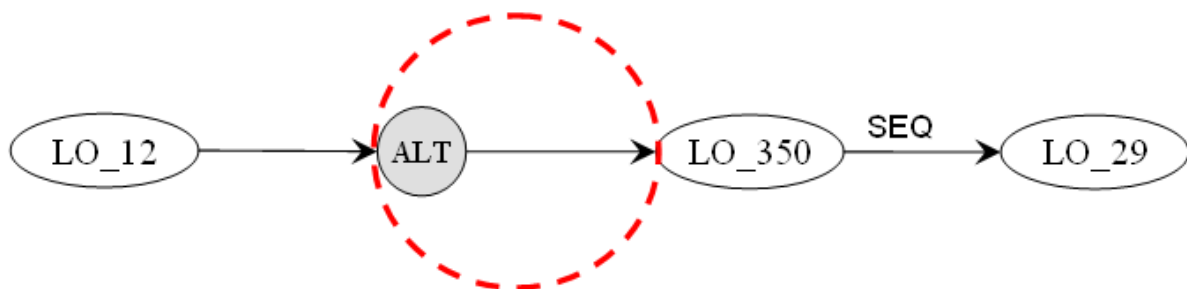


Figure 21 : Exemple de violation de la deuxième règle de cohérence SIMBAD

3^{ème} règle : Le graphe doit se terminer par un ou plusieurs sommets de type objet d'apprentissage.

Un nœud feuille par définition n'est pas à l'origine d'aucun arc. Et donc c'est absurde d'avoir un opérateur en tant que feuille dans un graphe de composition d'un objet d'apprentissage composé puisque l'opérateur n'a de sens que s'il est la source d'au moins deux arcs.

L'exemple suivant montre un graphe de composition qui viole cette règle à cause d'un nœud terminal de type opérateur.

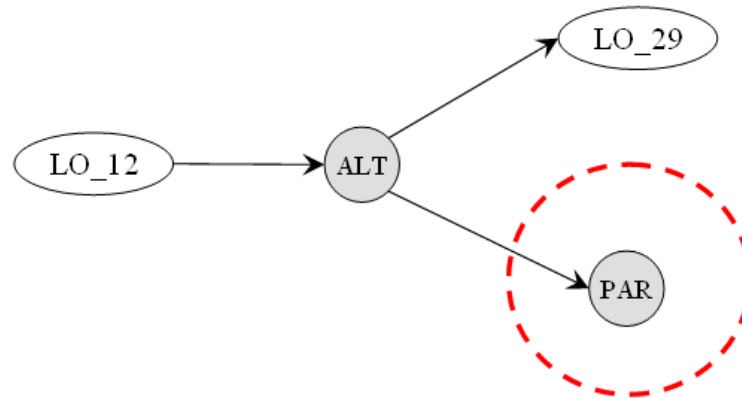


Figure 22 : Exemple de violation de la troisième règle de cohérence SIMBAD

4^{ème} règle : Chaque sommet de type objet d'apprentissage doit être à l'origine d'au plus un arc.

Puisqu'il n'y a pas d'interprétation possible du cas où un objet est à l'origine de plusieurs arcs, il est interdit d'avoir une telle structure. Si l'auteur veut exprimer une alternative il doit faire usage de l'opérateur ALT et s'il veut exprimer que le choix de l'ordre entre plusieurs chemin est libre il doit faire l'usage de l'opérateur PAR. Voici un exemple qui viole cette règle :

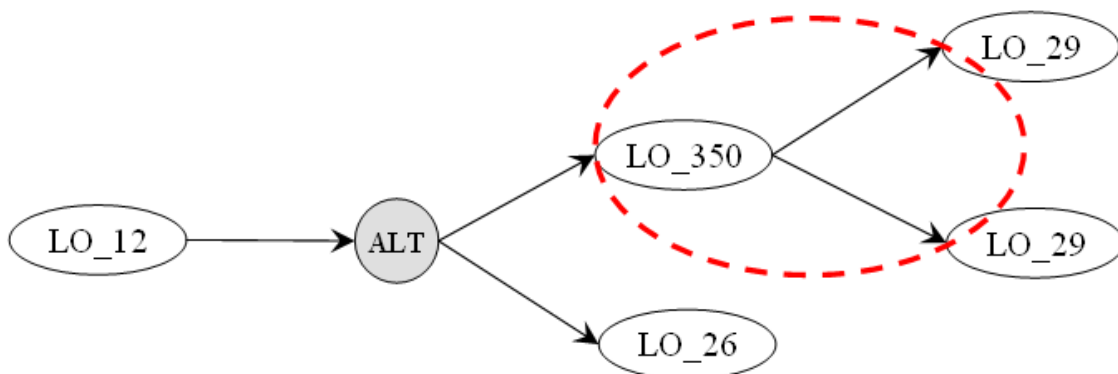


Figure 23 : Exemple de violation de la quatrième règle de cohérence SIMBAD

5^{ème} règle : Le graphe doit être acyclique.

Le graphe de composition traduit les différentes possibilités offertes par l'auteur pour avancer dans le contenu de l'objet d'apprentissage composé. Ainsi, un cycle dans le graphe ne peut pas refléter la notion d'avancement au sein d'un contenu.

L'exemple suivant présente un graphe de composition abstrait qui viole cette règle puisqu'il comporte un cycle.

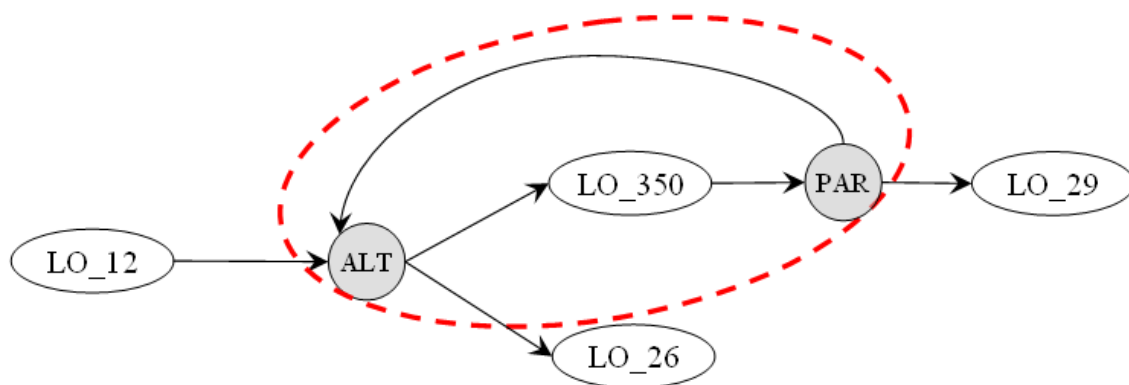


Figure 24 : Exemple de violation de la cinquième règle de cohérence SIMBAD

6^{ème} règle : Le graphe doit comporter une seule composante connexe.

En fait, le graphe de composition abstrait doit avoir une seule composante connexe. En effet, il est structuré de façon qu'à partir d'un seul point d'entrée (le sommet racine) on doit pouvoir accéder à tous les autres sommets du graphe. Cette propriété revient à dire que l'objet doit avoir une seule composante connexe.

L'exemple suivant présente un graphe de composition abstrait qui viole cette règle puisqu'il comporte trois composantes connexes. Dans le cas de la composante qui comporte le sommet LO_351 on se trouve déjà avec un graphe de composition abstrait avec deux points d'entrées possibles. Ceci revient à dire que le graphe de composition abstrait viole la première règle.

Pour la composante connexe qui comporte le sommet LO_352 on a un seul point d'entrée au niveau du graphe de composition abstrait mais avec un cycle. Ceci revient à dire que le graphe de composition abstrait viole la cinquième règle.

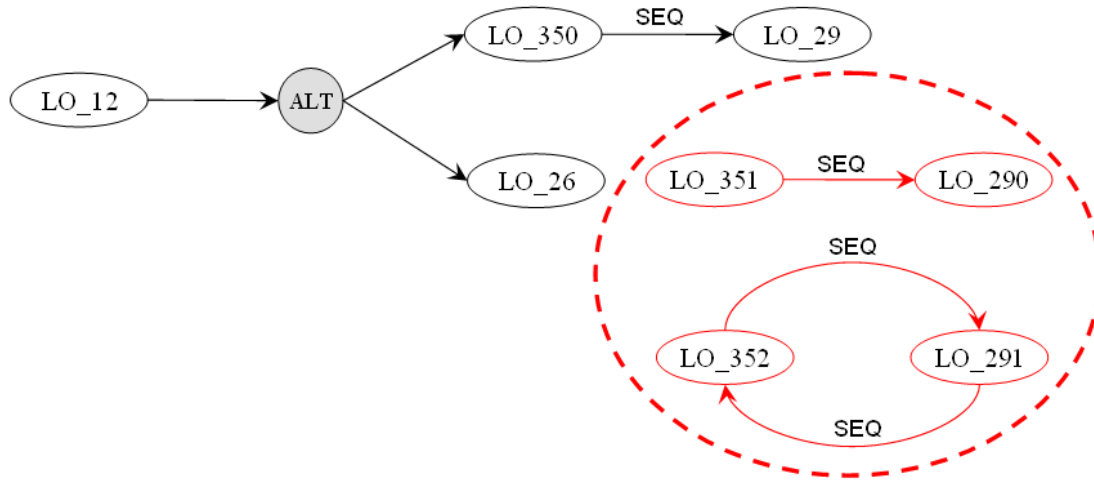


Figure 25 : Exemple de violation de la sixième règle de cohérence SIMBAD

2.2. Algorithme de vérification de la première règle

Nous allons considérer les quatre ensembles suivants :

O_i l'ensemble des objets d'apprentissage du graphe de composition de l'objet i .

V_i est l'ensemble des sommets du graphe de composition abstrait de l'objet i .

T_i est l'ensemble des arcs du graphe de composition abstrait de l'objet i .

t_{ij} est l'arc qui a comme source le sommet i et comme extrémité le sommet j .

Rappelons que la première règle indique que le graphe de composition d'un objet d'apprentissage composé doit avoir un seul point d'entrée qui est un sommet de type objet d'apprentissage.

Algorithme 3 : Vérification de l'existence d'un seul point d'entrée de type objet d'apprentissage

Booléen VerifierR1 (G : Graphe)

Début

Si $\exists i \in O / (\exists t_{ji} \in T, \forall j \in V)$ **Alors**

// Pas de sommet racine de type objet d'apprentissage

Retourner (Faux)

Sinon Si $\exists \{i, i'\} \subset O / (\exists t_{ji} \in T \text{ et } \exists t_{j'i'} \in T, \{j, j'\} \in V)$ **Alors**

// Il existe au moins deux sommets racine

Retourner (Faux)

Fin Si

Retourner (Vrai)

Fin

2.3. Algorithme de vérification de la deuxième et troisième règle

La deuxième règle à suivre, pour être structurellement cohérent avec le modèle SIMBAD, concerne les feuilles du graphe de composition abstrait. Ceux-ci doivent être de type objet d'apprentissage.

La troisième règle consiste à indiquer que tout opérateur PAR ou ALT est à l'origine d'au moins deux arcs. Il faut noter que cette troisième règle permet de garantir les propriétés demandées au niveau de la deuxième règle.

Algorithme 4 : Vérification des opérateurs

Booléen VerifierR2R3 (G : Graphe)

Début

Si $(\exists i \in V / (\exists \{t_{ij}, t_{ik}\} \subset T, \{j, k\} \subset V))$ ET $(i \notin O)$ **Alors**

Retourner (Faux)

Fin Si

Retourner (Vrai)

Fin

2.4. Algorithme de vérification de la quatrième règle

La quatrième règle précise que chaque sommet de type objet d'apprentissage doit être à l'origine d'au plus un arc.

Algorithme 5 : Vérification des objets d'apprentissage

Booléen VerifierR4 (G : Graphe)

Début

Si $(\exists i \in O)$ ET $(\exists \{t_{ij}, t_{ik}\} \subset T / \{j, k\} \subset V)$ **Alors**

Retourner (Faux)

Fin Si

Retourner (Vrai)

Fin

2.5. Algorithme de vérification de la cinquième règle

La cinquième règle impose le fait que le graphe de composition abstrait doit être acyclique. Pour y parvenir il faut utiliser un algorithme de parcours de graphe orienté avec deux marqueurs, un marqueur pour dire qu'on a visité déjà le nœud et l'autre est un jeton pour

marquer le chemin en cours de traitement. Si tous les sommets sont marqués et que l'algorithme trouve dans un chemin deux fois le même sommet alors le graphe est certainement connexe (tous les sommets sont marqués et donc visité depuis le point d'entrée) et sans cycles.

L'algorithme est récursif. Le premier appel se fait avec l'identifiant du sommet qui représente le point d'entrée du graphe et l'identifiant du graphe de composition abstrait. Dès que l'algorithme détecte un cycle il retourne « faux ».

Algorithme 6 : Détection de cycle dans un graphe

Booléen verifierR5R6(i : Sommet, G : Graph)

Début

Si (existeJeton(i) = Vrai) **Alors**

// S'il existe un jeton alors il y a un cycle car le sommet est visité deux

// fois lors du parcours du même chemin

Retourner (Faux)

Sinon

mettreMarqueur (i) // Pour dire que i est déjà visité

mettreJeton(i) // Pour dire que i appartient au chemin en cours de traitement

S = suivants(i) // S est la liste des sommets qui peuvent être visités depuis i

Pour j de 0 à taille-1 Faire

Si (verifierR5R6(S[j], G) == Faux) **Alors**

Retourner (Faux)

Sinon

j = j+1

Fin Si

Fin Pour

supprimerJeton(i)

// Le chemin auquel appartient le sommet i est visité sans trouver un cycle

Fin Si

Retourner (Vrai)

Fin

Il suffit après de vérifier que tous les sommets du graphe G sont marqués pour dire que le graphe est connexe.

3. Vérification de la cohérence sémantique

3.1. Présentation

La sémantique propre à l'objet d'apprentissage en cours de développement est définie à deux niveaux. Le premier niveau correspond à la description donnée par l'auteur de la sémantique de l'objet. Le deuxième niveau correspond aux descriptions de la sémantique des objets d'apprentissage réutilisés pour sa mise en place. Rappelons que cette description couvre trois aspects : les pré-requis, le contenu et la fonction d'acquisition. Cette description sémantique est renseignée au niveau des métadonnées de l'objet d'apprentissage.

Cette sémantique doit être cohérente. Pour pouvoir assister l'auteur de l'objet à vérifier la cohérence sémantique, nous avons défini des règles de cohérence sémantique qui touchent les différentes facettes de la description sémantique.

Par la suite nous allons détailler chacune des règles, les anomalies qu'elles permettent de découvrir et les algorithmes nécessaires pour les implémenter.

3.2. Satisfaction des pré-requis

3.2.1. Principe de satisfaction des pré-requis

Un objet d'apprentissage composé comporte un graphe de composition abstrait qui décrit une stratégie de navigation entre les objets réutilisés. Pour qu'un apprenant puisse accéder au contenu d'un objet il faut vérifier qu'il a le niveau nécessaire pour l'aborder. Si cette condition est satisfaite, il faut qu'il soit en mesure de parcourir la totalité du contenu de l'objet. C'est-à-dire, il ne doit pas se trouver bloqué à cause d'un objet d'apprentissage dont les pré-requis ne sont pas satisfaits. Ce cas de figure est le signe d'une mauvaise conception que la phase de vérification doit être en mesure de découvrir.

En fait, il faut que les pré-requis de chaque objet du graphe de composition soient satisfaits.

Un pré-requis est satisfait dans l'un des deux cas suivants :

- Il est satisfait s'il est couvert par les pré-requis de l'objet composé. C'est-à-dire si le profil de l'apprenant lui a permis d'accéder à l'objet d'apprentissage composé alors il va lui permettre également d'accéder au contenu de cet objet d'apprentissage réutilisé.
- Il est satisfait s'il est couvert par une fonction d'acquisition d'un des objets qui le précèdent dans le graphe de composition abstrait. C'est-à-dire si l'objet a un pré-requis qui ne figure pas dans les pré-requis de l'objet composé, il est certainement satisfait par le

profil de l'apprenant si un objet qui le précède (donc déjà validé par l'apprenant) contient une fonction d'acquisition qui couvre ce pré-requis.

Si ni l'un ni l'autre des cas de figure ci-dessus est vérifié ceci suppose que l'objet d'apprentissage composé tel qu'il est conçu présente une anomalie d'ordre sémantique.

3.2.2. Exemple

Afin d'illustrer le principe de satisfaction de pré-requis nous allons considérer l'objet d'apprentissage composé lo_903, ayant le graphe de composition suivant :

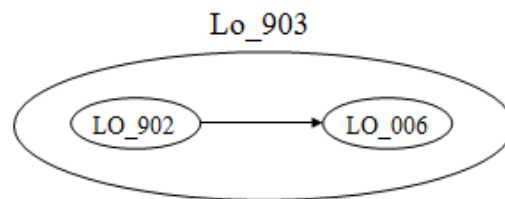


Figure 26 : Graphe de composition abstrait de l'objet d'apprentissage composé lo_903

Supposons que l'objet lo_903 est décrit par les métadonnées sémantiques suivantes :

Pré-requis : {

<"Théorie relationnelle", "principes", "intermédiaire">,
 <"Bases de données relationnelles", "Introduction", "Débutant">}

Contenu : {

<"Bases de données relationnelles", "principes">,
 <"Bases de données relationnelles", "exemple">,
 <"Bases de données relationnelles", "exercice">}

Fonction d'acquisition: {

<"Bases de données relationnelles", "principes", "intermédiaire">,
 <"Bases de données relationnelles", "exemple", "intermédiaire">,
 <"Bases de données relationnelles", "exercice", "intermédiaire">}

Nous allons supposer que l'objet lo_902 est décrit quant-à-lui par les métadonnées sémantiques suivantes :

Pré-requis : { <"Théorie relationnelle", "principes", "intermédiaire">}

Contenu: {

<"Bases de données relationnelles", "principes">,
 <"Bases de données relationnelles", "exemple">
 }

Fonction d'acquisition: {

<"Bases de données relationnelles", "principes", "intermédiaire">,

<"Bases de données relationnelles", "principes", "intermédiaire">}

Nous allons supposer que l'objet lo_006 est décrit quant-à-lui par les métadonnées sémantiques suivantes :

Pré-requis : {<"Bases de données relationnelles", "principes", "intermédiaire">}

Contenu: {<"Bases de données relationnelles", "exercice">}

Fonction d'acquisition : {<"Bases de données relationnelles", "exercice", "débutant">}

Dans cet exemple nous trouvons les deux cas de figures autorisés. En effet, pour le premier objet d'apprentissage lo_902 dans le graphe de composition nous avons un pré-requis qui est satisfait puisque l'apprenant ne peut accéder à cet objet que si son profil satisfait les pré-requis de l'objet composé lo_903. Or les pré-requis de lo_903 incluent le seul pré-requis de lo_902. Donc on est sûr que l'apprenant à un niveau intermédiaire par rapport aux principes de la théorie relationnelle lui permettant d'accéder à l'objet lo_902.

L'objet d'apprentissage lo_006 a un seul pré-requis. Celui-ci ne fait pas parti des pré-requis de l'objet composé lo_903. Donc, à priori l'apprenant qui accède au contenu de l'objet lo_903 n'est pas censé avoir un niveau intermédiaire par rapport aux principes relatifs aux bases de données relationnelles. Toutefois, l'apprenant doit accéder, d'après le graphe de composition abstrait, à l'objet lo_006 qu'une fois qu'il a acquis le contenu de l'objet lo_902. Or ce dernier, d'après sa fonction d'acquisition, permet à l'apprenant justement d'avoir un niveau intermédiaire par rapport aux principes relatifs aux bases de données relationnels et donc de satisfaire les pré-requis de l'objet lo_006.

L'intérêt de cet exemple est double. D'une part il permet de présenter les deux cas qui permettent de garantir la satisfaction des pré-requis. D'autre part il met en évidence la difficulté de la vérification manuelle même dans le cas d'un objet ayant le graphe de composition le plus simple qu'un objet d'apprentissage composé peut avoir. Ainsi, nous proposons un algorithme permettant la vérification automatique de cette règle.

3.2.3. Algorithme de vérification de la satisfaction des pré-requis

Afin d'introduire notre algorithme nous introduisant la relation d'ordre (\prec) entre les triplets de la forme $\langle c, r, l \rangle$ où : c est un concept parmi les concepts du modèle du domaine, r est un rôle éducatif et l : un niveau de maîtrise d'un concept.

Nous supposons que $\langle c_i, r_i, l_i \rangle \prec \langle c_j, r_j, l_j \rangle$ si $c_i \approx c_j$ et $r_i = r_j$ et $l_i \leq l_j$.

Le signe ‘ \approx ’ exprime une proximité sémantique entre deux concepts. Nous considérons que : $c \approx c'$ si les deux concepts c et c' sont en relation via une relation de type spécifique/générique ou une relation rhétorique (*contraste*, *similaire* ou équivalent) dans le modèle du domaine.

L’algorithme suivant décrit le processus de vérification automatique de la satisfaction de pré-requis. Nous utilisons la notation suivante :

G_i est le graphe de composition abstrait de l’objet d’apprentissage composé i . P_i est l’ensemble des pré-requis de l’objet i . A_i est l’ensemble des fonctions d’acquisition de l’objet i . C_i est l’ensemble des contenus de l’objet i . $prev_i$ est l’ensemble des objets d’apprentissages qui doivent être acquis par l’apprenant avant l’objet i d’après le graphe de composition abstrait défini par l’auteur. Nous considérons que clo est l’identifiant de l’objet d’apprentissage composé en cours de vérification.

Algorithme 7 : Vérification de la satisfaction des pré-requis

Booléen `verifierSatisfactionPrerequis (clo : ObjetComposé)`

Début

Pout tout $lo / lo \in G_{clo}$ **Faire**

Pour tout $p \in P_{lo}$ **Faire**

Si $(\exists x / p \prec x, x \in \{P_{clo} \cup A_i\} / i \in prev_{lo})$ **Alors**

Retourner (Faux)

Fin

Fin

Fin

Retourner (Vrai)

Fin

3.3. Cohérence sémantique du contenu

3.3.1. Principe de la cohérence sémantique du contenu

Lors de la composition et du renseignement des métadonnées sémantiques d’un nouvel objet d’apprentissage composé certaines anomalies peuvent apparaître à l’insu de la volonté de l’auteur. Ces anomalies peuvent être répertoriées en deux catégories :

- Métadonnées sémantiques non fidèles au contenu : c'est le cas où les métadonnées de l'objet composé ne décrivent pas fidèlement le contenu tel qu'il est décrit par les métadonnées sémantiques des objets réutilisés.
- Contenu non fidèle aux métadonnées sémantiques : c'est le cas de certains objets réutilisés dont le contenu a une description sémantique non cohérente à celle de l'objet composé.

Afin de pouvoir découvrir ces éventuelles anomalies nous devons utiliser les informations sémantiques disponibles à propos de l'objet composé et des objets réutilisés au niveau de leurs métadonnées. La vérification doit donc porter sur l'adéquation entre les métadonnées de chaque objet réutilisé et ceux de l'objet composé.

3.3.2. Exemple

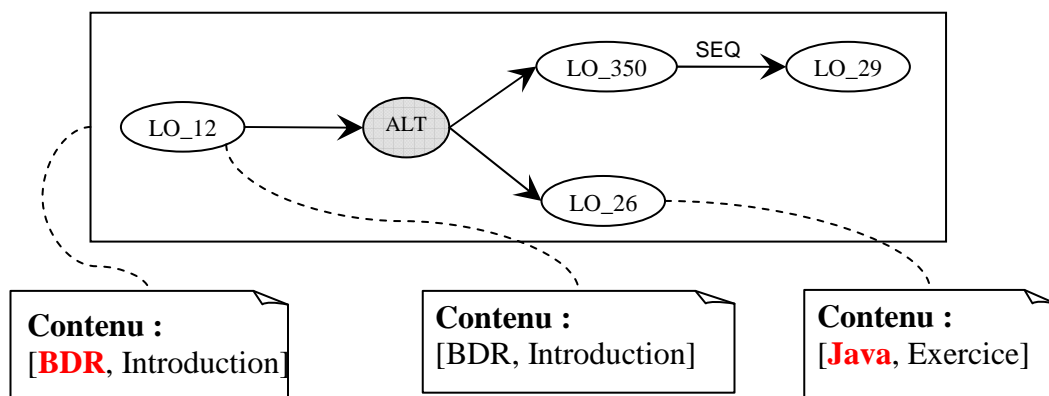


Figure 27 : Graphe de composition abstrait de l'objet d'apprentissage composé lo_904

Cet exemple illustratif présente un objet d'apprentissage composé lo_904. Le contenu de l'objet est censé être une introduction aux bases de données relationnelles. Si on examine les objets réutilisés dans le graphe de composition abstrait, on peut tomber sur l'un des deux cas suivants. Le premier cas est celui de l'objet lo_12 qui concerne le concept « bases de données relationnelles ». Ceci doit être considéré comme acceptable puisque ce concept est en proximité sémantique avec le concept traité par l'objet composé.

Par contre, le contenu de l'objet lo_26 concerne le concept « Java ». Supposons que le modèle de domaine indique que ce concept n'est pas en proximité sémantique avec le concept « bases de données relationnelles ». A ce moment nous devons être capable d'informer l'auteur sur ce risque d'incohérence dû à la présence d'un objet dont le contenu concerne un concept qui n'est pas en proximité sémantique avec aucun concept parmi ceux qui décrivent le contenu de l'objet d'apprentissage composé.

3.3.3. Algorithme de vérification de la cohérence sémantique du contenu

Afin d'introduire notre algorithme nous utilisons l'opérateur ' \approx ' qui exprime une proximité sémantique entre deux concepts. Nous considérons que $c \approx c'$ si les deux concepts c et c' sont en relation via une relation de type spécifique/générique ou une relation rhétorique (*contraste*, *similaire* ou équivalent) dans le modèle du domaine.

L'algorithme suivant décrit le processus de vérification automatique de cohérence sémantique du contenu. Nous utilisons la notation suivante :

G_i est le graphe de composition abstrait de l'objet d'apprentissage composé i . C_i est l'ensemble qui décrit le contenu de l'objet i . Nous considérons que clo est l'identifiant de l'objet d'apprentissage composé en cours de vérification.

Algorithme 8 : Vérification de la cohérence sémantique du contenu

Booléen `verifierCohérenceContenu` (clo : `ObjetComposé`)

Début

Pout tout $lo / lo \in G_{clo}$ **Faire**

Pour tout $\langle c, r \rangle \in C_{lo}$ **Faire**

Pour tout $\langle c', r' \rangle \in C_{clo}$ **Faire**

Si $c' \neq c$ **Alors**

Retourner (Faux)

Fin

Fin

Fin

Fin

Retourner (Vrai)

Fin

4. Conclusion

Dans ce chapitre nous avons présenté en détails la phase de vérification de la cohérence d'un objet d'apprentissage composé. Les vérifications touchent à la fois aux aspects structurels et aux aspects sémantiques.

Ces vérifications doivent permettre de dévoiler à l'auteur les différentes anomalies ou incohérences possibles afin de pouvoir y remédier assez tôt. Nous allons détailler dans le chapitre suivant la phase d'analyse.