Construction de la cartographie de domaine

Sommaire

4.1 Introduction

Une fois les ressources alignées et un ensemble de correspondances n:m établi entre les entités qui les composent, il reste à exploiter le résultat obtenu. On peut bien entendu chercher à fusionner ces ressources pour en construire une nouvelle mais c'est un cas extrême ; le plus souvent on s'appuie sur l'alignement pour isoler une sous-partie intéressante ou pour analyser les différents choix de modélisation proposés. Dans tous les cas, il faut commencer par prendre connaissance des ressources disponibles. C'est à cet effet que nous cherchons à construire des cartographies de domaines qui présentent les ressources disponibles en les articulant les unes aux autres (liens de correspondance) ainsi que par rapport au texte de référence (liens d'annotation) et en faisant ressortir les zones d'intérêt dans l'ensemble de liens de correspondance. La cartographie peut être exploitée dans le cadre de nombreuses applications telles que la construction d'une ressource sémantique, le découpage d'ontologies en blocs cohérents (la modularisation) et la fusion de ressources hétérogènes. Ces applications peuvent influencer la définition de ce qui est considéré comme zone d'intérêt et pourrait être paramétré par l'ingénieur

L'analyse de ces correspondances met en effet en évidence des configurations intéressantes. Certaines sont « anormales » et font apparaître que les ressources alignées reposent sur des choix de modélisation très différents, voire incompatibles. D'autres au contraire sont « remarquables » et font apparaître des points de jonction entre les ressources. Notre objectif est de doter les cartographies de domaine d'un outil d'interrogation qui permette à l'utilisateur d'explorer ces configurations pour l'aider à prendre connaissance des ressources à sa disposition lorsque ces dernières sont de grande taille et donc difficiles à analyser. Ces

configurations sont également utiles pour valider les correspondances fournies par notre algorithme d'alignement, si besoin est. Ce type d'outil d'exploration des alignements est d'autant plus utile qu'on peut envisager à terme d'aligner non pas deux ontologies mais plusieurs ressources sémantiques les unes par rapport aux autres.

D'autres chercheurs se sont intéressés à la révision des sorties d'alignement. [Hanif *et al.*, 2006] proposent par exemple une méthode permettant d'éliminer des correspondances erronées en utilisant deux techniques d'alignement (terminologique et structurelle) et en privilégiant les correspondances obtenues par les deux techniques. C'est naturellement une approche très sélective. [Stuckenschmidt *et al.*, 2005] proposent d'évaluer une sortie d'alignement à partir de ses propriétés de cohérence et de minimalité. [Wang et Xu, 2008] montrent comment repérer les erreurs d'alignement comme par exemple les correspondances redondantes qui sont inutiles à présenter à la fin du processus d'alignement. Notre construction de cartographies de domaine s'inscrit dans le prolongement de ces travaux qui analysent et retraitent les sorties d'alignement.

Nous nous intéressons dans ce chapitre non plus à la cooccurrence des entités dans le texte mais plutôt à leur position dans les ontologies. Notre méthode permet de détecter des anomalies en raisonnant sur la structure formée par chacune des ontologies et l'ensemble des correspondances, l'objectif à terme étant d'assurer la cohérence de l'ensemble (éliminer les correspondances erronées) et de mettre en évidence les points de jonction les plus intéressants. Nous définissons pour cela un premier ensemble de configurations intéressantes, anormales ou remarquables. Cet ensemble peut être enrichi en fonction de la finalité de la cartographie. Nous proposons deux méthodes pour les repérer dans les sorties d'alignement : la première est algorithmique et la seconde repose sur le moteur de recherche sémantique Corese [Corby *et al.*, 2006] et des requêtes SPARQL. Nous proposons enfin une méthode d'élimination automatique des configurations anormales même si celles-ci peuvent aussi être analysées manuellement.

Ce chapitre est organisé autour de deux sections centrales qui portent sur les configurations anormales et leur élimination (section 4.2) et sur les configurations remarquables et leur affichage (section 4.3).

4.2 Détection et élimination des anomalies

Nous avons repéré trois configurations, que nous considérons anormales qui sont relatives à des relations de type équivalence et association. Une configuration dite anormale est détectée en raisonnant sur la structure des deux ontologies alignées. Une configuration anormale est une configuration qui regroupe des correspondances qui génèrent des incohérences (ou inconsistances) dans l'une ou l'autre des ontologies.

4.2.1 Configurations anormales

Trois configurations anormales sont détectées. Ces configurations ne couvrent évidemment pas tous les problèmes possibles. Elles permettent, néanmoins, de caractériser les cas des problèmes avec :

– une inversion de hiérarchie ;

– une entité ambiguë ;
– une ambiguïté de relations.

Configuration avec inversion de hiérarchie (C_{hi})

Définition 1 *Une configuration avec inversion de hiérarchie est une anomalie où les entités, qui font partie des deux relations de correspondance de type équivalence, sont structurées d'une manière hiérarchique inversée dans l'une et l'autre des ontologies (voir figure 4.1). On parle d'une incompatibilité des liens d'équivalence. On note une telle configuration par* $C_{hi}(eq_{ij}, eq_{uv})$ *où* $eq_{ij} < idE_x, e_i^1, e_j^2, score, equiv >$, $eq_{uv} < idE_y, e_u^1, e_v^2, score, equiv >$ *sont deux relations de correspondance de type équivalence incompatibles, idE : l'identifiant de la correspondance de type équivalence et* $x \neq y$. *On dit que* eq_{ij} *et* eq_{uv} *sont incompatibles si et seulement si* $[(e_u^1 \sqsubset e_i^1) \wedge (e_j^2 \sqsubset e_v^2)] \vee [(e_i^1 \sqsubset e_u^1) \wedge (e_v^2 \sqsubset e_j^2)]$.

Ce cas peut provenir de (i) une erreur de calcul dans les correspondances, (ii) une erreur d'identification des entités sémantiques, et (iii) d'une erreur de typage de la correspondance. Même si une telle inversion de hiérarchie peut refléter un choix de modélisation différent, nous considérons que c'est une configuration anormale parce qu'elle reflète des choix de modélisation incompatibles.

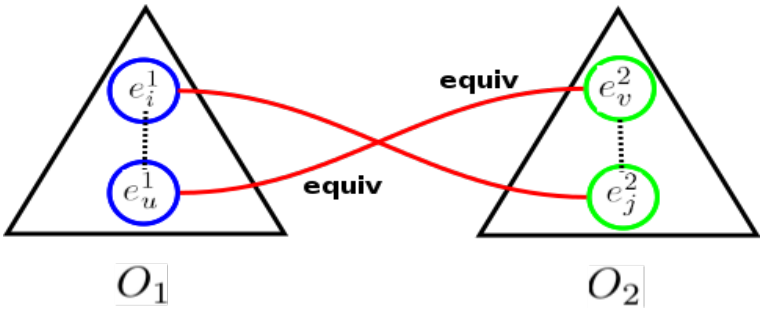


FIGURE 4.1 – Schéma de la hiérarchie inversée $C_{hi}(eq_{ij}, eq_{uv})$

Exemple 1 *La figure 4.2 montre une hiérarchie inversée. Il existe un lien hiérarchique entre les deux entités « pont » et « passerelle » de la première ressource BDTopo et un lien hiérarchique inversé entre les deux entités « chemin » et « sentier » de la deuxième ressource BDCarto.*

Méthodes de détection de l'anomalie Nous proposons deux méthodes pour détecter l'anomalie de la hiérarchie inversée. La première est algorithmique et consiste à tester si les correspondances de type équivalence sont incompatibles. Cette méthode prend en entrée un ensemble de liens d'équivalence $E_{12} = \{< idE, e_x^1, e_y^2, score, equiv > / e_x^1 \in R_1$ et $e_y^2 \in R_2\}$, et donne en sortie une liste de couples de correspondances de type équivalence incompatibles.

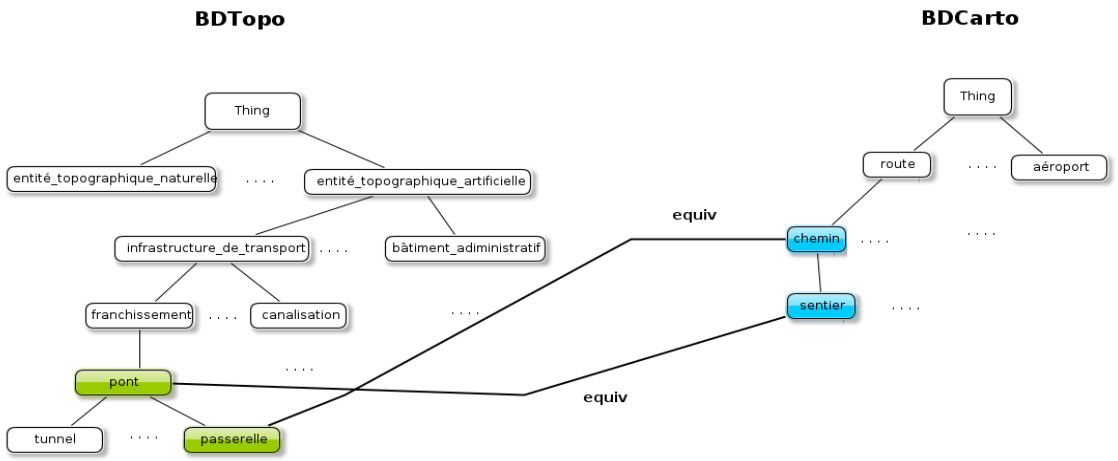


FIGURE 4.2 – Exemple de l'anomalie : hiérarchie inversée

Algorithme 1 *InconsistentLinks*() : liste de couples de correspondances de type équivalence incompatibles

Entrée : $E_{12} = \{< idE, e_x^1, e_y^2, score, equiv >\}$
Sortie : L_{incomp} : liste de couples de correspondances de type équivalence incompatibles
Variable : $incomp$: booléen
Début
pour tout $eq_{ij} \in E_{12}$ *ET* $eq_{uv} \in E_{12}$ **faire**
  pour tout ij allant de 1 à $taille(E_{12})$ **faire**
    pour tout uv allant de $ij + 1$ à $taille(E_{12})$ **faire**
      %% Tester si eq_{ij} et eq_{uv} sont incompatibles
      $incomp = \underline{IncompCheck}(eq_{ij}, eq_{uv})$
      si $incomp = vrai$ **alors**
        inserer$(eq_{ij}, eq_{uv}, L_{incomp})$
      fin si
    fin pour
  fin pour
fin pour
retourne L_{incomp}
Fin

Algorithme 2 *IncompCheck*($eq_{ij} : < idE_x, e_i^1, e_j^2, score, equiv >$, $eq_{uv} < idE_y, e_u^1, e_v^2, score, equiv >$) : booléen

Sortie : *incomp* : booléen
Début
si $[(e_u^1 \sqsubset e_i^1) \wedge (e_j^2 \sqsubset e_v^2)] \vee [(e_i^1 \sqsubset e_u^1) \wedge (e_v^2 \sqsubset e_j^2)]$ **alors**
  $incomp = vrai$
sinon
  $incomp = faux$
fin si
retourne *incomp*
Fin

La deuxième méthode consiste à appliquer une requête formelle structurée (SPARQL) dans un moteur de recherche sémantique, comme Corese [1]. La requête s'appuie sur une base de correspondances sous le format RDF de la campagne d'évaluation OAEI et les deux ontologies lexicalisées sous format OWL. La requête SPARQL qui permet de détecter le problème de la hiérarchie inversée est la suivante :

```
PREFIX align:<http://knowledgeweb.semanticweb.org/heterogeneity/alignment#>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl:<http://www.w3.org/2002/07/owl#>
SELECT ?x1 ?x2 ?y1 ?y2 ?cell1 ?cell2 ?t ?v1 ?v2 WHERE
        {?cell1 rdf:type align:Cell
        ?cell1 align:entity1 ?x1
        ?cell1 align:entity2 ?y1
        ?cell1 align:relation ?t
        ?cell2 rdf:type align:Cell
        ?cell2 align:entity1 ?x2
        ?cell2 align:entity2 ?y2
        ?cell2 align:relation ?t
        ?v1 align:onto1 align:uri1
        ?x1 owl:Class ?v1
        ?y2 owl:Class ?v1
        ?v2 align:onto2 align:uri2
        ?x2 owl:Class ?v2
        ?y1 owl:Class ?v2
        ?y2 rdfs:subClassOf ?x1
        ?y1 rdfs:subClassOf ?x2
        Filter regex(?t, "=")
        }
```

Où : Cell est une balise ($< Cell > ... < /Cell >$) dans le format RDF proposé par la campagne d'évaluation OAEI, qui regroupe les entités des deux ontologies mises en

1. http://www-sop.inria.fr/edelweiss/software/corese/

correspondance par un type de relation, et = : pour la correspondance de type équivalence sémantique.

Configuration avec une entité ambigüe (C_{AmEq})

Définition 2 *Une configuration avec une entité ambigüe est une anomalie où une même entité est associée à deux entités distinctes avec deux relations de correspondance de type équivalence (voir figure 4.3). On parle d'une ambiguïté des deux liens d'équivalence. On note une telle configuration par* $C_{AmEq}(eq_{ij}, eq_{uv})$ *où* $eq_{ij} < idE_x, e_i^1, e_j^2, score, equiv >$ *et* $eq_{uv} < idE_y, e_u^1, e_v^2, score, equiv >$*, sont deux relations de correspondance de type équivalence ambigües, idE : identifiant de la correspondance de type équivalence et* $x \neq y$*. On dit que* eq_{ij} *et* eq_{uv} *sont ambigües si et seulement si il existe une même entité* $(e_i^1 = e_u^1)$ *qui est concernée par ces deux correspondances.*

Ce cas peut provenir d'une erreur d'identification des entités sémantiques aussi si ce type de configuration reflète un choix de modélisations.

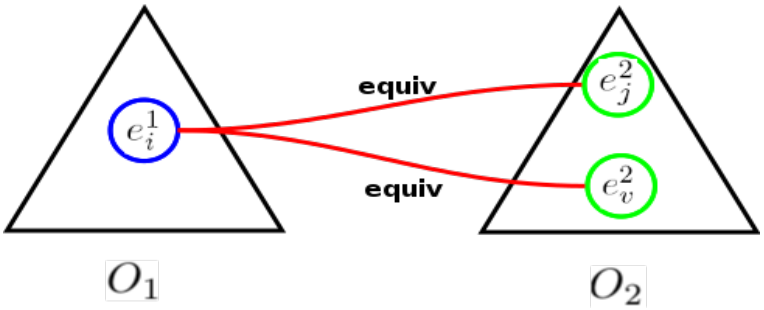


FIGURE 4.3 – Schéma du problème d'ambiguïté avec une entité sémantique

Exemple 2 *La figure 4.4 montre une ambiguïté avec l'entité sémantique « river ». Cette dernière (« river ») de la première ressource OntoBiotope est mise en relation d'équivalence avec deux entités « sludge » et « soil » de la deuxième ressource EnvO.*

Méthodes de détection de l'anomalie On peut détecter une anomalie avec une entité ambigüe de deux façons. La méthode algorithmique teste si les correspondances de type équivalence sont ambigües. Elle prend en entrée un ensemble de liens d'équivalence $E_{12} = \{< idE, e_x^1, e_y^2, score, equiv > / e_x^1 \in R_1 \text{ et } e_y^2 \in R_2\}$, et donne en sortie une liste de couples de correspondances de type équivalence ambigües.

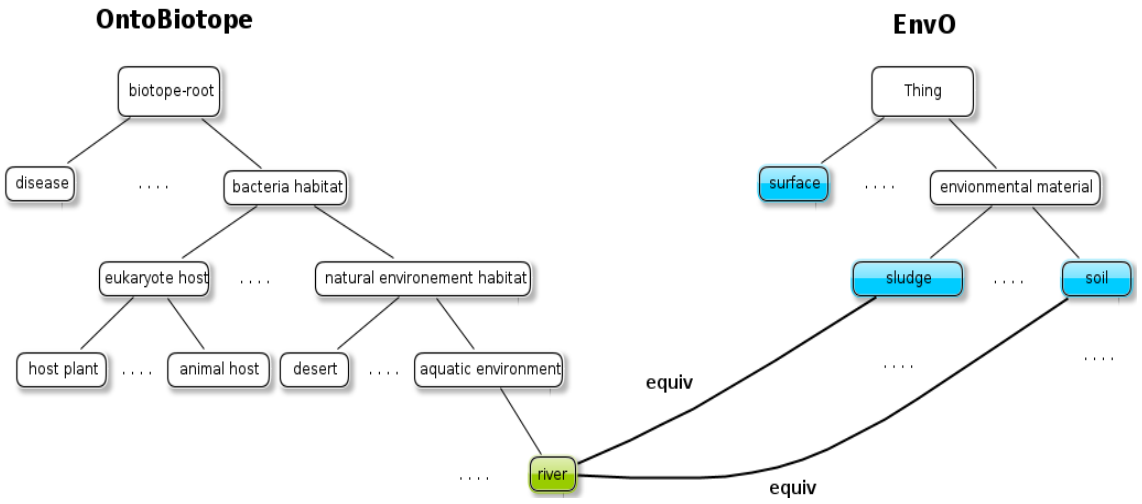


FIGURE 4.4 – Exemple de la configuration avec une entité ambigûe

Algorithme 3 *AmbiguousLinks*() : liste de couples de correspondances de type équivalence ambigüs

Entrée : $E_{12} = \{< idE, e_x^1, e_y^2, score, equiv >\}$
Sortie : L_{amb} : liste de couples de correspondances de type équivalence ambigüs
Variables : amb : booléen
Début
pour tout $eq_{ij} \in E_{12}$ *ET* $eq_{uv} \in E_{12}$ **faire**
  pour tout ij allant de 1 à $taille(E_{12})$ **faire**
    pour tout uv allant de $ij + 1$ à $taille(E_{12})$ **faire**
      %% Tester si eq_{ij} et eq_{uv} sont ambigus
      $amb = \underline{AmbiguityCheck}(eq_{ij}, eq_{uv})$
      si $amb = vrai$ **alors**
        inserer$(eq_{ij}, eq_{uv}, L_{amb})$
      fin si
    fin pour
  fin pour
fin pour
retourne L_{amb}
Fin

Algorithme 4 *AmbiguityCheck*(eq_{ij} : < $idE_x, e_i^1, e_j^2, score, equiv$ >, eq_{uv} < $idE_y, e_u^1, e_v^2, score, equiv$ >) : booléen

1: **Sortie :** *amb* : booléen
2: **Début**
3: **si** ($e_i^1 == e_u^1$) $\wedge$ (e_j^2 et e_v^2) sont distinctes) **alors**
4: $amb = vrai$
5: **sinon**
6: $amb = faux$
7: **fin si**
8: retourne *amb*
9: **Fin**

La requête formelle sur la base de connaissances des correspondances de type équivalence qui permet de détecter ces correspondances est la suivante :

```
PREFIX align:<http://knowledgeweb.semanticweb.org/heterogeneity/alignment#>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl:<http://www.w3.org/2002/07/owl#>
SELECT ?x1 ?y1 ?y2 ?cell1 ?t ?v1 ?v2 WHERE
        {?cell1 rdf:type align:Cell
        ?cell1 align:entity1 ?x1
        ?cell1 align:entity2 ?y1
        ?cell1 align:entity2 ?y2
        ?cell1 align:relation ?t
        ?v1 align:onto1 align:uri1
        ?x1 owl:Class ?v1
        ?v2 align:onto2 align:uri2
        ?y1 owl:Class ?v2
        ?y2 owl:Class ?v2
        Filter regex(?t, "=")
        }
```

Configuration avec une ambiguïté de relations ($C_{AmEqAss}$)

Définition 3 *Une configuration avec une ambiguïté de relations est une anomalie où une même entité correspond à une autre entité avec deux relations de correspondance de type équivalence et association (voir figure 4.5). On parle d'une ambiguïté des deux liens d'équivalence et d'association.*

On note une telle configuration par $C_{AmEqAss}(eq_{ij}, ass_{uv})$ *où* eq_{ij} < $idE_x, e_i^1, e_j^2, score, equiv$ > *et* ass_{uv} < $idA_y, e_u^1, e_v^2, score, assoc$ >, *sont deux relations de correspondance de type équivalence et association ambigües, idE : identifiant de la correspondance de type équivalence et idA : identifiant de la correspondance de type association. On dit que* eq_{ij} *et* ass_{uv} *sont ambigües si et seulement si* $e_i^1 = e_u^1$ *et* $e_j^2 = e_v^2$.

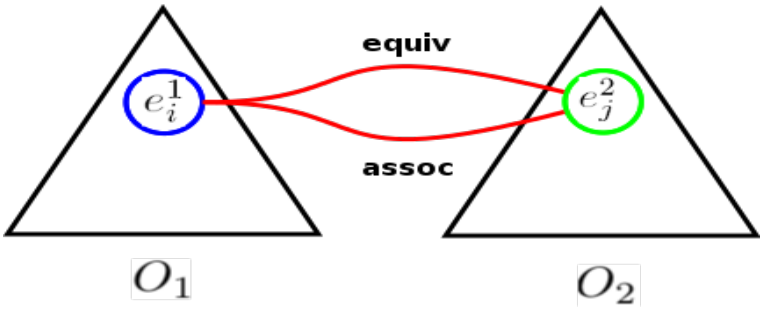


FIGURE 4.5 – Schéma de la configuration avec une ambiguïté de relations

Exemple 3 *La figure 4.6 montre une ambiguïté entre deux relations d'équivalence et d'association entre les deux entités « China » et « city ». L'entité « China » de la première ressource OntoBiotope est mise en correspondance avec l'entité « city » de la deuxième ressource EnvO, avec deux types de liens différents.*

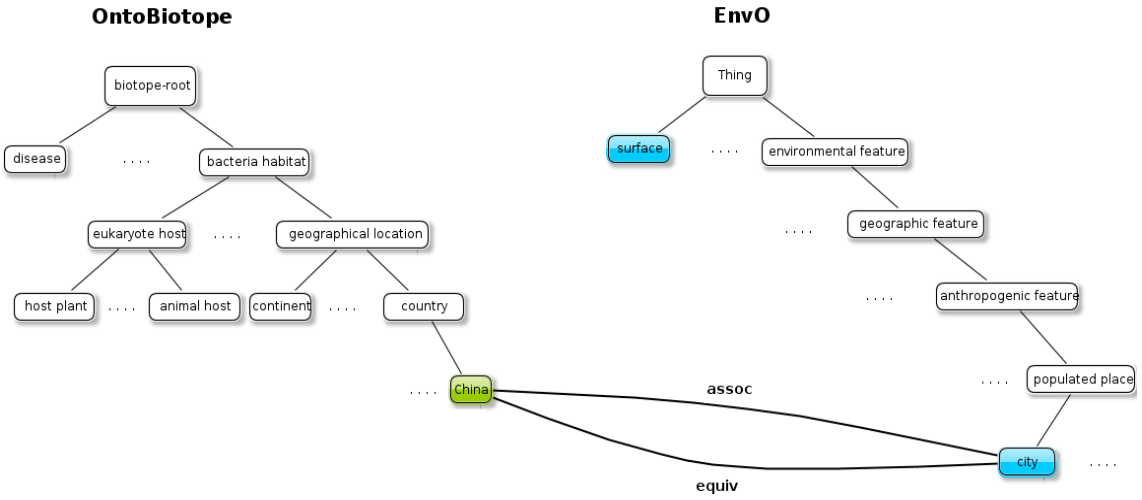


FIGURE 4.6 – Exemple d'une ambiguïté de relations

Méthodes de détection de l'anomalie La méthode algorithmique prend en entrée un ensemble de liens d'équivalence et d'association $E_{12} = \{< idE, e_x^1, e_y^2, score, equiv > \cup < idA, e_z^1, e_t^2, score, assoc > / e_x^1, e_z^1 \in R_1 \text{ et } e_y^2, e_t^2 \in R_2\}$, et donne en sortie une liste de couples de correspondances de type équivalence et association ambigus.

Algorithme 5 *AmbiguousLinksEqAss*() : liste de couples de correspondances de type équivalence et association ambigus

Entrée : $E_{12} = \{< idE, e_x^1, e_y^2, score, equiv > \cup < idA, e_z^1, e_t^2, score, assoc >\}$
Sortie : $L_{ambEqAss}$: liste de couples de correspondances de type équivalence et association ambigus
Variables : $ambEqAss$: booléen
Début
pour tout $(eq_{ij} \in E_{12}) \wedge (ass_{uv} \in EAR_{12})$ **faire**
  pour tout ij allant de 1 à $taille(E_{12})$ **faire**
    pour tout uv allant de $ij + 1$ à $taille(E_{12})$ **faire**
      %% Tester si eq_{ij} et ass_{uv} sont ambigus
      $ambEqAss = \underline{AmbiguityCheckEqAss}(eq_{ij}, ass_{uv})$
      si $ambEqAss = vrai$ **alors**
        inserer$(eq_{ij}, ass_{uv}, L_{ambEqAss})$
      fin si
    fin pour
  fin pour
fin pour
retourne $L_{ambEqAss}$
Fin

Algorithme 6 *AmbiguityCheckEqAss*$(eq_{ij} : < idE_x, e_i^1, e_j^2, score, equiv >, ass_{uv} < idA_y, e_u^1, e_v^2, score, assoc >)$: booléen

Sortie : $ambEqAss$: booléen
Début
si $(e_i^1 == e_u^1) \wedge (e_j^2 == e_v^2)$ **alors**
  $ambEqAss = vrai$
sinon
  $ambEqAss = faux$
fin si
retourne $ambEqAss$
Fin

La requête SPARQL sur la base de connaissances des correspondances de type association et équivalence est la suivante :

```
PREFIX align:<http://knowledgeweb.semanticweb.org/heterogeneity/alignment#>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl:<http://www.w3.org/2002/07/owl#>
SELECT ?x1 ?y1 ?cell1 ?t1 ?t2 ?v1 ?v2 WHERE
        {?cell1 rdf:type align:Cell
        ?cell1 align:entity1 ?x1
        ?cell1 align:entity2 ?y1
        ?cell1  align:relation ?t1
```

```
?cell1  align:relation  ?t2
?v1 align:onto1 align:uri1
?x1 owl:Class ?v1
?v2 align:onto2 align:uri2
?y1 owl:Class ?v2
Filter regex(?t1, "=") && regex(?t2, "<>")
}
```

Où : <> : pour la correspondance de type association sémantique.

4.2.2 Élimination des anomalies

L'élimination des anomalies consiste à éliminer des correspondances qui sont : (i) incompatibles, et (ii) ambigües. Cette étape peut être interactive si l'ingénieur de la connaissance veut avoir la main et s'assurer qu'on ne supprime pas des correspondances qui ont été jugées anormales alors qu'elles proviennent de choix de modélisation différents. En effet, notre calcul d'incohérence repose sur une vision unifiée des deux ontologies et suppose un raisonnement uniforme. Pour cela, nous pouvons d'abord proposer d'afficher ces configurations à l'instar des configurations remarquables (cf. section 4.3) pour ensuite activer ou non l'élimination automatique.

Notre intuition, dans cette étape, est qu'une correspondance peut poser à la fois des problèmes d'incompatibilité et d'ambiguïté. Le fait de supprimer cette correspondance diminue plusieurs problèmes simultanément. A titre d'exemple, une relation de correspondance de type équivalence peut faire partie de la configuration avec inversion de hiérarchie et de la configuration avec une entité ambigüe. Quand on élimine cette relation, on résout deux anomalies en même temps au lieu d'agir deux fois pour les résoudre.

Nous proposons donc de raisonner globalement sur le nombre d'incompatibilités et d'ambiguïtés par relation. Si le nombre d'incompatibilités et d'ambiguïtés entre deux relations de correspondance ayant le même type de relation (équivalence ou association) est le même, nous raisonnons localement et par ordre de fiabilité des relations. Autrement dit, nous retenons la correspondance ayant le score le plus élevé. Dans le cas où le nombre d'incompatibilités et d'ambiguïtés est le même pour deux types différents (équivalence et association) de relations de correspondance, nous retenons la relation de correspondance de type équivalence. Le choix peut également être donné à l'ingénieur de la connaissance.

L'objectif du raisonnement global est de supprimer les liens, fournis par l'alignement guidé par le texte, qui posent des problèmes. Nous prenons pour cela tous les liens d'équivalence et d'association et nous supprimons le(s) lien(s) qui posent le plus de problèmes. Nous avons trois listes de couples de liens qui correspondent aux différentes configurations :

- $L_{incomp} = \{(eq_{ij}, eq_{i'j'})\}$: liste de couples de correspondances de type « équivalence incompatible » ;
- $L_{amb} = \{(eq_{ij}, eq_{i'j'})\}$: liste de couples de correspondances de type « équivalence ambiguë » ;
- $L_{ambEqAss} = \{(eq_{ij}, ass_{i'j'})\}$: liste de couples de correspondances de type « équivalence et association ambiguës ».

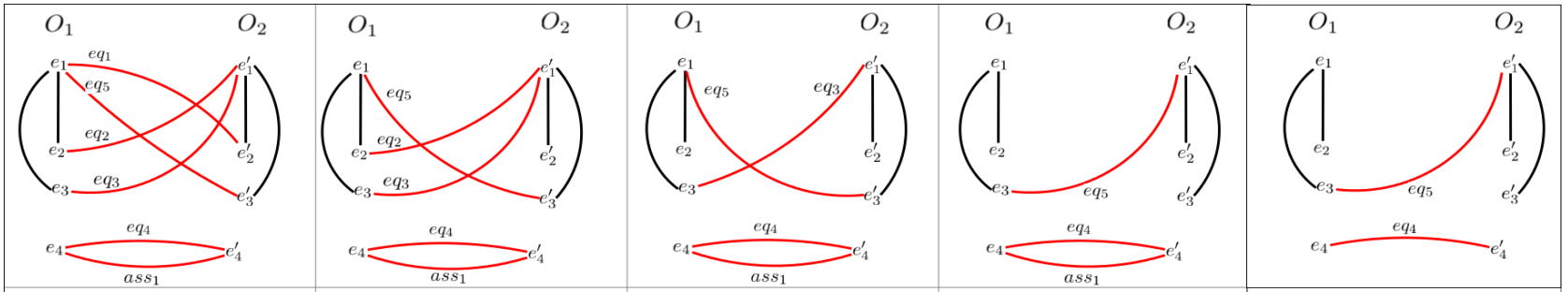
Dans le but de résoudre ces anomalies, nous procédons globalement comme suit :

1) calculer le nombre d'incompatibilités et d'ambiguïtés par type de relation (équivalence et association) ;

Nous répétons le traitement ci-dessous jusqu'à ce que $L_{incomp} = \varnothing$, $L_{amb} = \varnothing$ et $L_{ambEqAss} = \varnothing$

2) détecter le lien qui génère le plus d'incompatibilités et d'ambiguïtés (extraction du maximum) ; si les relations ayant le même type (équivalence ou association) possèdent le même nombre élevé d'incompatibilités et d'ambiguïtés alors on retient la relation de correspondance ayant le score le plus élevé et on rejette l'autre. Si les relations ayant deux types différents (équivalence et association) possèdent le même nombre élevé d'incompatibilités et d'ambiguïtés alors nous retenons la relation d'équivalence ou nous proposons à l'ingénieur de la connaissance d'intervenir pour choisir l'une des deux.
3) mise à jour de L_{incomp}, L_{amb} et $L_{ambEqAss}$ (suppression des couples de relations qui sont dépendants de la relation supprimée) ;
4) mise à jour du nombre d'incompatibilités et d'ambiguïtés des liens qui sont dépendants de la relation supprimée.

L'exemple 4.7 montre l'application de notre algorithme. Dans cet exemple, nous disposons de 4 entités dans O_1 (e_1, e_2, e_3 et e_4) et 4 entités dans O_2 (e'_1, e'_2, e'_3 et e'_4). L'alignement de ces ontologies a permis d'obtenir 6 relations qui posent problème (eq_1, eq_2, eq_3, eq_4, eq_5 et ass_1). Dans l'itération 1, nous obtenons quatre relations de même type ayant le même nombre d'incompatibilités et d'ambiguïtés (3). Nous avons supposé que eq_1 a un score le plus faible pour supprimer toutes les correspondances qui sont dépendantes à cette eq_1. Le même raisonnement dans les autres itérations (ex. dans l'itération 2, eq_2 est supprimée car nous avons supposé qu'elle possède un score faible).



Itération 1	Itération 2	Itération 3	Itération 4	Itération 5
$L_{incomp} = \{(eq_1, eq_2); (eq_1, eq_3); (eq_2, eq_5); (eq_3, eq_5)\}$ $L_{amb} = \{(eq_2, eq_3); (eq_1, eq_5)\}$ $L_{ambEqAss} = \{(eq_4, ass_1)\}$	$L_{incomp} = \{(eq_2, eq_5); (eq_3, eq_5)\}$ $L_{amb} = \{(eq_2, eq_3)\}$ $L_{ambEqAss} = \{(eq_4, ass_1)\}$	$L_{incomp} = \{(eq_3, eq_5)\}$ $L_{amb} = \varnothing$ $L_{ambEqAss} = \{(eq_4, ass_1)\}$	$L_{incomp} = \varnothing$ $L_{amb} = \varnothing$ $L_{ambEqAss} = \{(eq_4, ass_1)\}$	$L_{incomp} = \varnothing$ $L_{amb} = \varnothing$ $L_{ambEqAss} = \varnothing$
relations | Nb. incomp.& ambig. eq_1 | 3 eq_2 | 3 eq_3 | 3 eq_4 | 1 eq_5 | 3 ass_1 | 1	relations | Nb. incomp.& ambig. eq_2 | 2 eq_3 | 2 eq_4 | 1 eq_5 | 2 ass_1 | 1	relations | Nb. incomp.& ambig. eq_3 | 1 eq_4 | 1 eq_5 | 1 ass_1 | 1	relations | Nb. incomp.& ambig. eq_4 | 1 eq_5 | 0 ass_1 | 1	relations | Nb. incomp.& ambig. eq_4 | 0 eq_5 | 0

FIGURE 4.7 – Phase d'alignement : exemple de l'élimination des configurations anormales

Dans la section suivante, nous décrivons les correspondances remarquables qui permettent de marquer les entités intéressantes à exploiter dans les différentes ontologies.

4.3 Détection et affichage de correspondances remarquables

Notre idée est née du fait qu'on veut présenter à l'ingénieur de la connaissance les résultats d'alignement de manière à l'assister lors du processus de construction d'une ressource sémantique. Après la correction des anomalies, nous avons différentes informations qui facilitent le travail de restitution à l'ingénieur de la connaissance : (1) le nombre de liens qui partent de la même entité, et (2) la distance des entités mises en relation par rapport aux feuilles.

Notre but est donc d'attirer l'attention de l'ingénieur de la connaissance sur des correspondances entre les ontologies qui semblent intéressantes.

Nous proposons pour cela de détecter des configurations dites remarquables. Une configuration remarquable est une configuration qui regroupe des correspondances qui présentent des caractéristiques spécifiques qui mettent en valeur les entités sémantiques mises en relation. Ces caractéristiques portent sur la position des entités par rapport à la racine et aux feuilles ainsi que sur le nombre de liens partagés entre entités.

Nous présentons donc dans la cartographie, des correspondances avec des indications sur des entités pour montrer à l'ingénieur de la connaissance que ces liens sont remarquables.

Cette section est organisée comme suit : nous détaillons dans un premier temps deux configurations remarquables qui présentent : (1) une différence de granularité sémantique, (2) plusieurs liens d'association. Nous présentons ensuite l'affichage de ces configurations.

4.3.1 Configurations remarquables

Dans cette section, nous définissons deux types de configurations remarquables :

1. configuration avec une différence de niveau de généralité : deux entités sont mises en correspondance soit par un lien d'équivalence soit par une association, mais ne sont pas classées au même niveau hiérarchique.
2. configuration avec plusieurs liens d'association : une entité d'une première ontologie est en relation d'association avec plusieurs entités distinctes d'une deuxième ontologie.

Configuration avec une différence de granularité sémantique C_{gs}

Définition 4 *Une configuration avec une différence de niveau de généralité est une configuration où deux entités mises en correspondance, avec un lien de type équivalence ou association, possèdent un niveau de généralité différent dans la description sémantique des deux ontologies (voir figure 4.8). Le niveau de généralité d'une entité d'ontologie est exprimé par sa hauteur par rapport aux feuilles (représentées par le plus lointain descendant).*

On note une telle configuration $C_{gs}(eqAss)$ où $eqAss$ peut être un lien d'équivalence ou un lien d'association avec une différence de niveau de généralité.

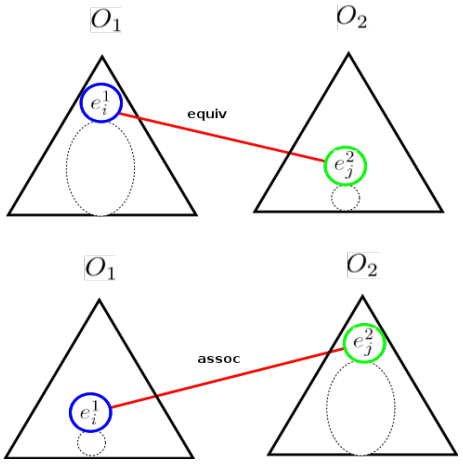


FIGURE 4.8 – Schéma de la configuration avec une différence de granularité sémantique

Exemple 4 *La figure 4.9 montre une différence de granularité sémantique (avec une relation d'équivalence) des deux entités « soil » de OntoBiotope et « soil » de EnvO. La première entité de l'ontologie OntoBiotope possède un niveau de généralité de 5 alors que la même entité dans EnvO est à un niveau de 3.*

Méthode de détection de la configuration remarquable La méthode ci-dessous teste si un ensemble de correspondances de type équivalence ou association se combine avec une différence de granularité sémantique. Cette méthode prend en entrée un ensemble de liens d'équivalence ou d'association $EQG_{12} = \{< idE, e_x^1, e_y^2, score, equiv > / e_x^1 \in R_1$ et $e_y^2 \in R_2\}$, ou $EQG_{12} = \{< idA, e_x^1, e_y^2, score, assoc > / e_x^1 \in R_1$ et $e_y^2 \in R_2\}$ et donne en sortie une liste de couples d'entités mis en correspondance avec leur granularité sémantique. Nous présentons l'algorithme pour les relations d'équivalence.

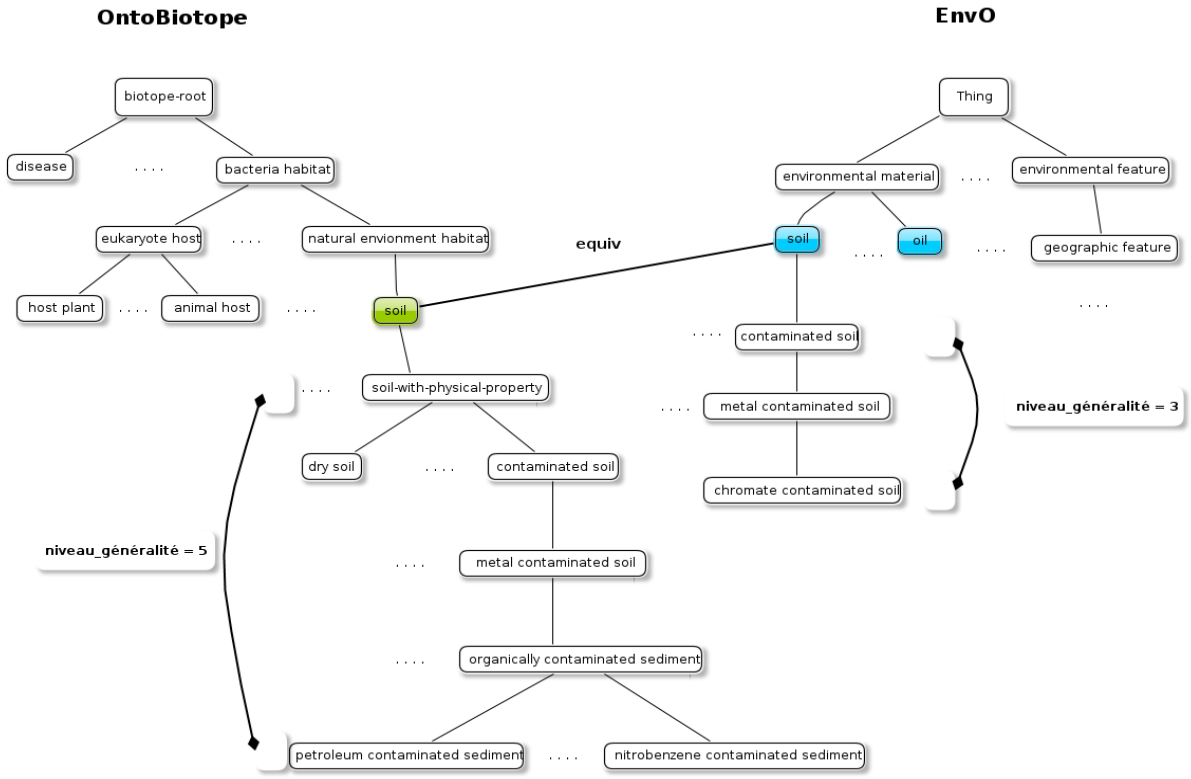


FIGURE 4.9 – Exemple de la configuration avec une différence de granularité sémantique

Algorithme 7 *GranularityEntitityEqAss*() : liste de couples d'entités mis en correspondance avec leur granularité sémantique

Entrée : $EQG_{12} = \{< idE, e_x^1, e_y^2, score, equiv >\}$
Sortie : $L_{GranEntityEq}$: liste de couples d'entités mises en correspondance avec l'équivalence associées à leur granularité sémantique.
Variables : $niveau - generalite(e_i)$, $niveau - generalite(e_j)$: entiers
Début
pour tout $(eq_{ij} \in EQG_{12})$ **faire**
  pour tout ij allant de 1 à $taille(EQG_{12})$ **faire**
    %% Tester si $niveau - generalite(e_i) > niveau - generalite(e_j)$
    $niveau - generalite(e_i) = nbrArcs - entre(e_i, feuille(e_i))$
    $niveau - generalite(e_j) = nbrArcs - entre(e_j, feuille(e_j))$
    si $(niveau - generalite(e_i) > niveau - generalite(e_j))$ **alors**
      inserer$(e_i, e_j, niveau - generalite(e_i), niveau - generalite(e_j), L_{GranEntityEq})$
    fin si
  fin pour
fin pour
retourne $L_{GranEntityEq}$
Fin

Configuration avec plusieurs liens d'association C_{plAss}

Définition 5 *Une configuration avec plusieurs liens d'association est une configuration où une entité est reliée par au minimum deux relations de correspondance de type association, avec d'autres entités. On parle de la centralité d'une entité d'ontologie qui est une valeur d'intérêt portée à cette entité. Cette centralité est décrite par le nombre de liens de type association qui relient une entité à d'autres entités. Plus une entité possède des relations de correspondance de type association vers d'autres entités, plus cette configuration est considérée comme remarquable.*

On note une telle configuration $C_{plAss}(ass_{ij}, ass_{uv})$ *où* $ass_{ij} < idA_x, e_i^1, e_j^2, score, assoc >$ *et* $ass_{uv} < idA_y, e_u^1, e_v^2, score, assoc >$ *et* $x \neq y$, *sont deux liens d'association contenant des entités centrales (voir figure 4.10).*

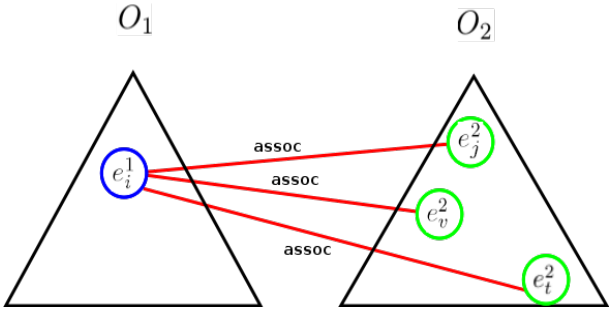


FIGURE 4.10 – Schéma de la configuration avec plusieurs liens d'association

Exemple 5 *La figure 4.11 montre une configuration remarquable où une entité sémantique est reliée à différentes autres entités avec des relations de type association. L'entité « China » est rapprochée de trois entités différentes « petroleum », « cut » et « city ».*

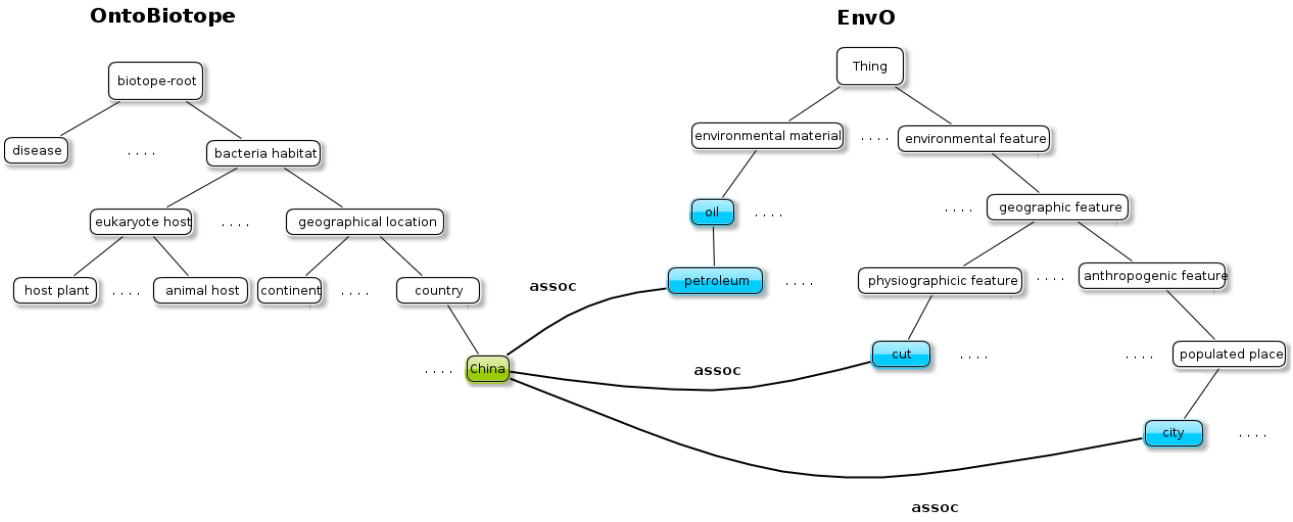


FIGURE 4.11 – Exemple de la configuration avec plusieurs liens d'association

Méthode de détection de la configuration remarquable La méthode ci-dessous teste si les correspondances de type association sont remarquables. Cette méthode prend en entrée un ensemble de liens d'association $EplAss_{12} = \{< idA, e_x^1, e_y^2, score, assoc > / e_x^1 \in R_1 \text{ et } e_y^2 \in R_2\}$, et donne en sortie une liste d'entités mises en relation avec le nombre de liens de type association qu'elles partagent avec les autres entités sémantiques.

Algorithme 8 $MultipleLinksAss()$: liste d'entités mises en relation avec le nombre de liens de type association

Entrée : $EplAss = \{< idA, e_x^1, e_y^2, score, assoc >\}$
Sortie : L_{plAss} : liste d'entités mises en relation de type association avec le nombre de liens
Variables : $plAss$: entier
Début
pour tout $ass_{ij} \in EplAss_{12}$ *ET* $ass_{uv} \in EplAss_{12}$ **faire**
  pour tout ij allant de 1 à $taille(EplAss_{12})$ **faire**
    pour tout uv allant de $ij + 1$ à $taille(EplAss_{12})$ **faire**
      %% Tester si ass_{ij} et ass_{uv} comportent des entités centrales
      $plAss = \underline{ComputeLinks}(ass_{ij}, ass_{uv})$
      si $plAss >= 2$ **alors**
        inserer$(e_i^1, plAss, L_{plAss})$
      fin si
    fin pour
  fin pour
fin pour
retourne L_{plAss}
Fin

Algorithme 9 $ComputeLinks(ass_{ij}, ass_{uv})$

Sortie : $plAss$: entier
Début
$plAss = 0$
si $(e_i^1 == e_u^1) \wedge (e_j^2$ et $e_v^2)$ sont distinctes **alors**
  $plAss = plAss + 1$
fin si
retourne $plAss$
Fin

4.3.2 Affichage de correspondances remarquables

L'affichage des correspondances remarquables consiste à présenter à l'ingénieur de la connaissance les configurations obtenues des types ci-dessus. Nous proposons donc pour la configuration avec une différence de granularité sémantique et la configuration avec plusieurs liens d'association de présenter des informations complémentaires sur les entités mises en correspondance à savoir : (i) leur position par rapport aux feuilles $(< e_i^1, degre -$

$detail(e_i^1), e_j^2, degre - detail(e_j^2) >$ où $e_i^1 \in O_1$ et $e_j^2 \in O_2$), et (ii) le nombre de liens de type association qu'une entité partage avec les autres entités ($< e_i^x, assoc, nbrLienAss >$ où : $e_i^x \in O_1 \bigvee O_2$ et $nbrLienAss$: nombre de liens que e_i^x partage avec les autres entités).

En terme d'implémentation, le fichier RDF ou textuel qui a été obtenu dans la phase d'alignement va être nettoyé au niveau de l'étape de détection et de l'élimination des anomalies puis utilisé pour présenter les configurations remarquables.

4.4 Conclusion

Dans ce chapitre, nous avons présenté un ensemble de configurations qui révèlent des anomalies et ou des zones remarquables dans les résultats d'alignement. En raisonnant sur la structure des ontologies, nous avons détecté trois configurations liées à l'incompatibilité et l'ambiguïté des liens et nous avons proposé une méthode pour les résoudre. Nous avons également proposé deux configurations remarquables qui font ressortir les entités qui semblent particulièrement intéressantes. Les algorithmes de détection des anomalies et des configurations remarquables ont été implémentés en Java et en utilisant la librairie Jena.

La cartographie de domaine obtenue est donc un ensemble de liens de correspondance établis entre deux ontologies alignées entre elles et ancrées dans un texte choisi comme référence, assorti d'un outil d'exploration qui permet de repérer facilement les anomalies et les configurations remarquables, de les résoudre au besoin et de les afficher. La figure 4.12 montre sous la forme d'une maquette comment une telle cartographie peut être affichée.

A ce stade, nous avons mis l'accent sur l'alignement de deux ontologies lexicalisées mais il faudrait à terme étendre l'approche à plus de ressources et d'autres types de ressources.

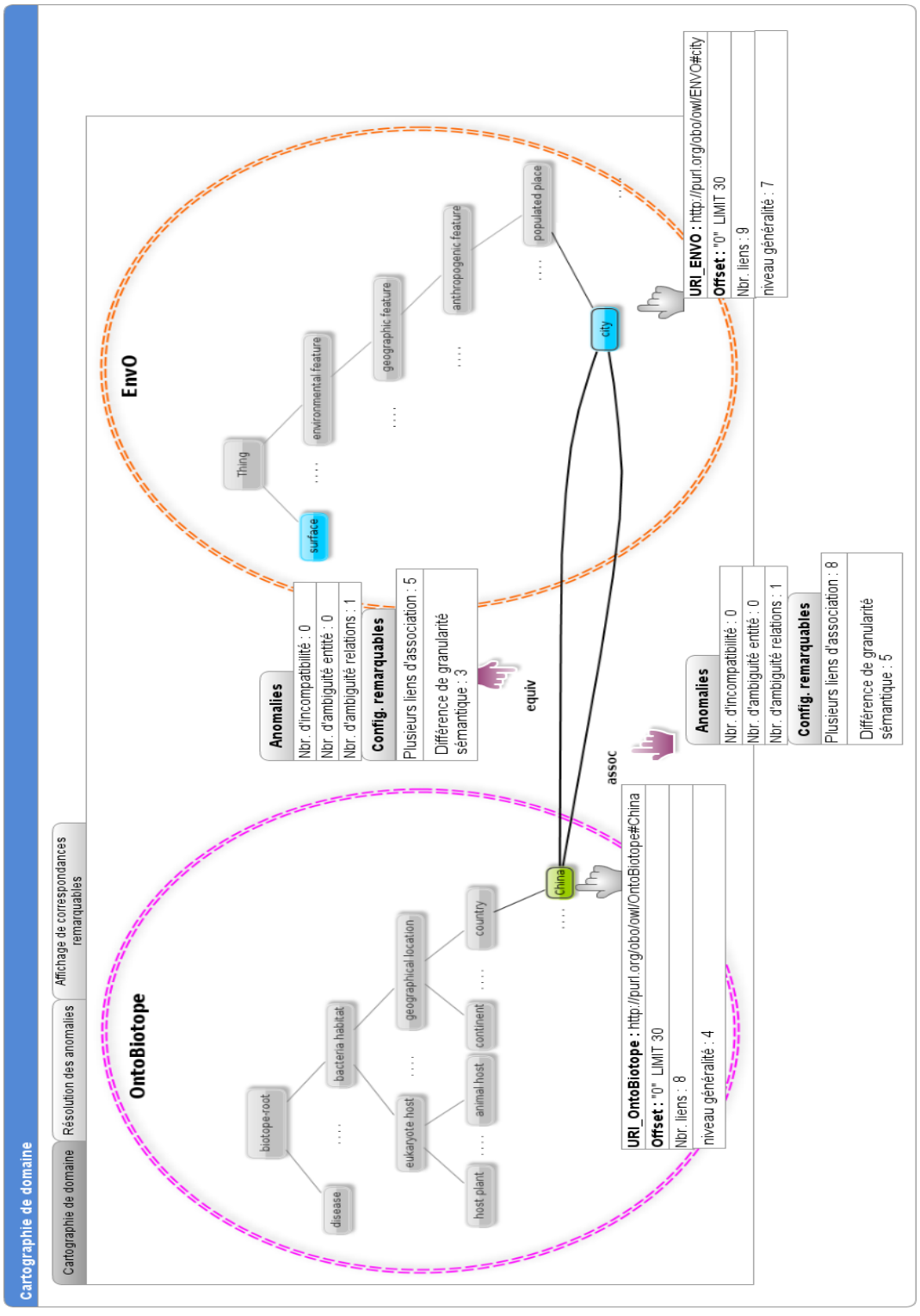


FIGURE 4.12 – Maquette de l'interface de la cartographie de domaine