Pattern for Re-engineering a Classification Scheme, Which Follows the Adjacency List Data Model, to a Taxonomy

http://ontologydesignpatterns.org/wiki/Submissions:Classification_scheme_-_adjacency_list_model_-_to_Taxonomy

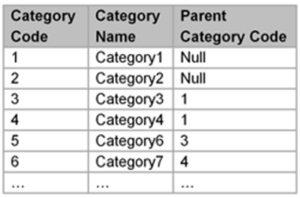
Boris Villazón-Terrazas[1], Mari Carmen Suárez-Figueroa[1], and Asunción Gómez-Pérez[1]

Ontology Engineering Group, Departamento de Inteligencia Artificial, Facultad de Informática, Universidad Politécnica de Madrid, Spain
{bvillazon,mcsuarez,asun}@fi.upm.es,
WWW home page: http://www.oeg-upm.net/

1 Introduction

This pattern for re-engineering non-ontological resources (PR-NOR) fits in the Schema Re-engineering Category proposed by [3]. The pattern defines a procedure that transforms the classification scheme components into ontology representational primitives. This pattern comes from the experience of ontology engineers in developing ontologies using classification schemes in several projects (SEEMP[1], NeOn[2], and Knowledge Web[3]). The pattern is included in a pool of patterns, which is a key element of our method for re-engineering non-ontological resources into ontologies [2]. The patterns generate the ontologies at a conceptualization level, independent of the ontology implementation language.

2 Pattern

<table>
<tr><th colspan="2">Problem</th></tr>
<tr><td colspan="2">Re-engineering a classification scheme, which follows the adjacency list model, to design a taxonomy.</td></tr>
<tr><th colspan="2">Non-Ontological Resource</th></tr>
<tr><td>A non-ontological resource holds a classification scheme which follows the adjacency list model. A classification scheme is a rooted tree of concepts, in which each concept groups entities by some particular degree of similarity.
The semantics of the hierarchical relation between parents and children concepts may vary depending of the context. The adjacency list data model [1] for hierarchical classifications proposes to create an entity which holds a list of items with a linking column associated to their parent items.</td><td></td></tr>
</table>

Category Code	Category Name	Parent Category Code
1	Category1	Null
2	Category2	Null
3	Category3	1
4	Category4	1
5	Category6	3
6	Category7	4
...	...	...

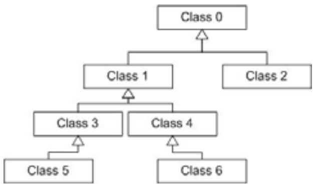
[1] http://www.seemp.org
[2] http://www.neon-project.org
[3] http://knowledgeweb.semanticweb.org

Applicability

The semantics of the relation between parent and children items are *subClassOf*.
There is not multi-inheritance nor cyclic relations.

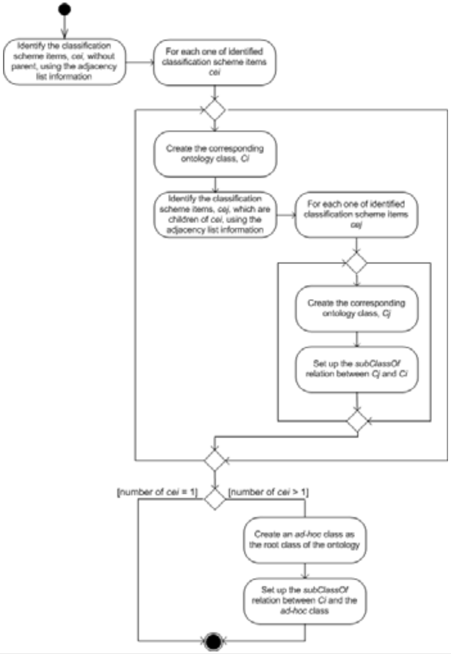
Ontology Generated

The ontology generated will be based on the taxonomy architectural pattern (AP-TX-01) [4].
Each category in the classification scheme is mapped to a class, and the semantics of the relationship between children and parent categories are mapped to *subClassOf* relations.



Process - Solution

1. Identify the classification scheme items which do not have a parent key value, i.e. classification scheme items without parents.
2. For each one of the above identified classification scheme items ce_i:
   - 2.1. Create the corresponding ontology class, C_i class.
   - 2.2. Identify the classification scheme items, ce_j, which are children of ce_i, by using the parent key values.
   - 2.3. For each one of the above identified classification scheme items ce_j:
     - 2.3.1. Create the corresponding ontology class, C_j class.
     - 2.3.2. Set up the *subClassOf* relation between C_j and C_i.
     - 2.3.3. Repeat from step 2.2 for ce_j as a new ce_i.
3. If there are more than one classification scheme items without parent ce_i
   - 3.1. Create an *ad-hoc* class as the root class of the ontology.
   - 3.2. Set up the *subClassOf* relation between C_i class and the root class.

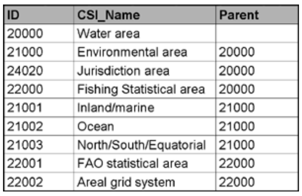

Identify the classification scheme items, cei, without parent, using the adjacency list information
For each one of identified classification scheme items cei
Create the corresponding ontology class, Ci
Identify the classification scheme items, cej, which are children of cei, using the adjacency list information
For each one of identified classification scheme items cej
Create the corresponding ontology class, Cj
Set up the subClassOf relation between Cj and Ci
[number of cei = 1]
[number of cei > 1]
Create an ad-hoc class as the root class of the ontology
Set up the subClassOf relation between Ci and the ad-hoc class

Example

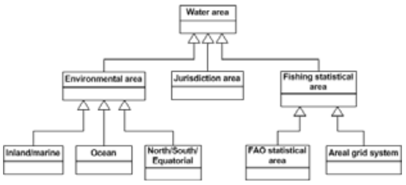
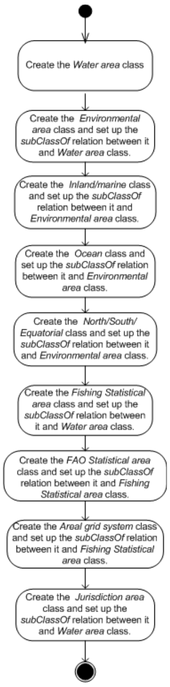
Suppose that someone wants to build an ontology based on the water areas classification published by FAO. This classification scheme follows the adjacency list data model.

Non-Ontological Resource

The FAO classification for water areas groups them according to some different criteria as environment, statistics, and jurisdiction, among others. This classification scheme is available at http://www.fao.org/figis/servlet/RefServlet



ID	CSI_Name	Parent
20000	Water area	
21000	Environmental area	20000
24020	Jurisdiction area	20000
22000	Fishing Statistical area	20000
21001	Inland/marine	21000
21002	Ocean	21000
21003	North/South/Equatorial	21000
22001	FAO statistical area	22000
22002	Areal grid system	22000

Ontology Generated	
The ontology generated will be based on the taxonomy architectural pattern (AP-TX-01) [4]. Each category in the classification scheme is mapped to a class, and the semantics of the relationship between children and parent categories are mapped to *subClassOf* relations.	 Water area Environmental area Jurisdiction area Fishing statistical area Inland/marine Ocean North/South/ Equatorial FAO statistical area Areal grid system
Process - Solution	
1. Create the `Water area` class. 2. Create the `Environmental area` class, and set up the *subClassOf* relation between the `Environmental` area class and the `Water area` class. 2.1. Create the `Inland/marine` class, and set up the *subClassOf* relation between the `Inland/marine` class and the `Environmental area` class. 2.2. Create the `Ocean` class, and set up the *subClassOf* relation between the `Ocean` class and the `Environmental area` class. 2.3. Create the `North/South/Equatorial` class, and set up the *subClassOf* relation between the `North/South/Equatorial` class and the `Environmental area` class. 3. Create the `Fishing Statistical area` class, and set up the *subClassOf* relation between the `Fishing Statistical area` class and the `Water area` class. 3.1. Create the `FAO statistical area` class, and set up the *subClassOf* relation between the `FAO statistical area` class and the `Fishing Statistical area` class. 3.2. Create the `Areal grid system` class, and set up the *subClassOf* relation between the `Areal grid system` class and the `Fishing Statistical area` class. 4. Create the `Jurisdiction area` class, and set up the *subClassOf* relation between the `Jurisdiction area` class and the `Water area` class.	 Create the *Water area* class Create the *Environmental area* class and set up the *subClassOf* relation between it and *Water area* class. Create the *Inland/marine* class and set up the *subClassOf* relation between it and *Environmental area* class. Create the *Ocean* class and set up the *subClassOf* relation between it and *Environmental area* class. Create the *North/South/ Equatorial* class and set up the *subClassOf* relation between it and *Environmental area* class. Create the *Fishing Statistical area* class and set up the *subClassOf* relation between it and *Water area* class. Create the *FAO Statistical area* class and set up the *subClassOf* relation between it and *Fishing Statistical area* class. Create the *Areal grid system* class and set up the *subClassOf* relation between it and *Fishing Statistical area* class. Create the *Jurisdiction area* class and set up the *subClassOf* relation between it and *Water area* class.
Related Resources	
This pattern is related to the architectural pattern TX-AP-01 [4] for modelling a taxonomy.	

3 Pattern Usage

This pattern was applied to re-engineer the ISTAT[4], geography italian standard, into a Geography Ontology[5], within the context of the SEEMP project. This standard is a classification scheme which consists of 4 divisions, 20 regions and 106 provinces. ISTAT is modelled following the adjacency list data model. Because of the number of divisions, regions and provinces of the ISTAT standard, it was not practical to create the ontology manually. Therefore, we created an

[4] http://www.istat.it/

[5] The ontology is available at http://droz.dia.fi.upm.es/hrmontology/

ad-hoc wrapper, implemented in Java, that reads the data from the resource implementation and automatically creates the corresponding elements of the new ontology following the suggestion given by the pattern.

4 Summary and Future Work

We have presented a pattern for transforming a classification scheme, which is modelled following the adjacency list data model, into a taxonomy. The pattern is included in a pool of patterns, which is a key element of our method for re-engineering non-ontological resources into ontologies [2].

We plan to develop software libraries within a framework that implement the transformation process suggested by the pattern. Moreover, we will include external resources to improve the quality of the resultant ontologies. Finally, we need to calculate how much effort do we save re-engineering classification schemes using patterns compared with re-engineering classification schemes without them.

Acknowledgments. This work has been partially supported by the European Comission projects NeOn(FP6-027595) and SEEMP(FP6-027347), as well as by an R+D grant from the UPM.

References

1. D. Brandon. Recursive database structures. *Journal of Computing Sciences in Colleges*, 2005.
2. A. García, A. Gómez-Pérez, M. C. Suárez-Figueroa, and B. Villazón-Terrazas. A Pattern Based Approach for Re-engineering Non-Ontological Resources into Ontologies. In *Proceedings of the 3rd Asian Semantic Web Conference (ASWC2008)*. Springer-Verlag, 2008.
3. V. Presutti, A. Gangemi, S. David, G. Aguado de Cea, M. C. Surez-Figueroa, E. Montiel-Ponsoda, and M. Poveda. NeOn Deliverable D2.5.1. A Library of Ontology Design Patterns: reusable solutions for collaborative design of networked ontologies. In *NeOn Project. http://www.neon-project.org*, 2008.
4. M. C. Suárez-Figueroa, S. Brockmans, A. Gangemi, A. Gómez-Pérez, J. Lehmann, H. Lewen, V. Presutti, and M. Sabou. Neon modelling components. Technical report, NeOn project deliverable D5.1.1, 2007.