Scenarios for Building Ontology Networks within the NeOn Methodology

Asunción Gómez-Pérez

Ontology Engineering Group, Departamento de Inteligencia Artificial. Facultad de Informática.
Universidad Politécnica de Madrid.
Boadilla del Monte, Madrid, Spain
asun@fi.upm.es

Mari Carmen Suárez-Figueroa

Ontology Engineering Group, Departamento de Inteligencia Artificial. Facultad de Informática.
Universidad Politécnica de Madrid.
Boadilla del Monte, Madrid, Spain
mcsuarez@fi.upm.es

ABSTRACT

In this poster, we present a set of nine scenarios, identified in the NeOn Methodology, for building ontology networks.

Categories and Subject Descriptors

I.2. Artificial Intelligence; I.2.4. Knowledge Representation Formalisms and Methods

General Terms

Algorithms, Management, Documentation, Design, Experimentation, Theory

Keywords

Methodology, ontology development, reuse, re-engineering.

INTRODUCTION

The mid 1990s and the first years of this new millennium have witnessed the growing interest of many practitioners in approaches that support the development of single ontologies. There are well known methodological approaches (e.g., [1], [3], and [2]) that have gone a step forward by having transformed the art of constructing single ontologies into an engineering activity.

However, the development of ontologies in different projects has revealed that there are different ways to build ontologies. Just to name a few of them, in the Esperonto project ontologies were built from scratch using METHONTOLOGY; in Knowledge Web the aligning and versioning of ontologies was treated as well as the use of patterns; and in the SEEMP project, a good requirements specification helped to find consensual knowledge-aware resources that were re-engineered into ontologies.

Up to date, there are no methodologies that help ontology developers to build large ontologies in such different ways, by reusing and possibly re-engineering knowledge-aware resources, using alignments, having in mind the continuous

evolution of the ontologies, and using ontologies embedded in ontology networks[1] built collaboratively by teams.

To cover the aforementioned limitation, we have created the *NeOn Methodology* for building ontology networks, which is a scenario-based methodology. Thus, in this poster we present the set of *nine scenarios for building ontologies and ontology networks,* emphasizing the reuse of existing ontological and non-ontological resources, generalizing from previous experiences, covering the drawbacks of the existing methodologies, and taking into account the new trends based on collaboration and dynamism.

NeOn SCENARIOS FOR BUILDING ONTOLOGY NETWORKS

One of the key elements in the NeOn Methodology framework is the set of 9 scenarios identified for building ontologies and ontology networks. This set of scenarios is shown in Figure 1.
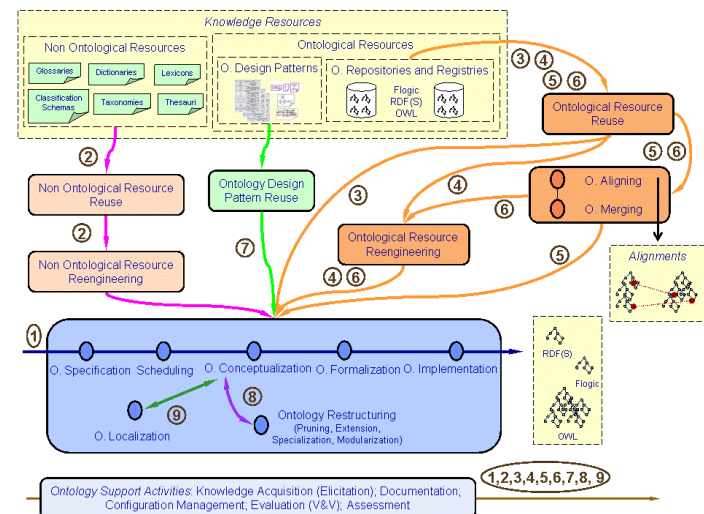

Figure 1. Scenarios for Building Ontology Networks

Directed arrows with numbered circles associated represent the different scenarios in Figure 1. Each scenario is decomposed into different processes or activities that are repre-

---

[1] Collections of ontologies related together through a variety of different relationships such as mapping, modularization, and version.

sented with colored circles or with rounded boxes. Such processes and activities are defined in the NeOn Glossary [4]. Figure 1 also shows (as dotted boxes) existing knowledge resources to be reused and possible outputs that result from the execution of some of the scenarios.

The most common scenarios[2] that may arise during the ontology development are the following, though such a set of scenarios can not be considered exhaustive.

Scenario 1: From specification to implementation. The ontology network is developed from scratch, that is, without reusing existing knowledge-aware resources. Ontology developers should specify the requirements that the ontology should fulfill, by means of the ontology requirements specification activity. The objective of this activity is to output the ontology requirements specification document (ORSD). After such an activity and using as input the terms appearing in the ORSD, it is advisory to carry out a quick search for potential knowledge-aware resources to be reused. Then, the scheduling activity must be carried out, using the ORSD and the results of such a quick search.

Scenario 2: Reusing and re-engineering non-ontological resources (NORs). Ontology developers should carry out the non-ontological resource reuse process for deciding, according to the requirements in the ORSD, which existing NORs can be reused to build the ontology network. Then, the selected NORs should be re-engineered into ontologies.

Scenario 3: Reusing ontological resources. Ontology developers use existing ontological resources (ontologies as a whole, ontology modules[3], and/or ontology statements[4]).

Scenario 4: Reusing and re-engineering ontological resources. Ontology developers reuse and re-engineer existing ontological resources.

Scenario 5: Reusing and merging ontological resources. This scenario arises only in those cases where several ontological resources in the same domain are selected for reuse and when ontology developers wish to create a new ontological resource from two or more ontological resources.

Scenario 6: Reusing, merging and re-engineering ontological resources. Ontology developers reuse, merge, and re-engineer existing ontological resources in the ontology network building. This scenario is similar to Scenario 5; however, here developers decide not to use the set of merged resources such as it is, but to re-engineer it.

Scenario 7: Reusing ontology design patterns (ODPs). Ontology developers access repositories[5] to reuse ODPs.

Scenario 8: Restructuring ontological resources. Ontology developers restructure (modularizing, pruning, extending, and/or specializing) ontological resources to be integrated in the ontology network being built.

Scenario 9: Localizing ontological resources. Ontology developers adapt an existing ontology to other languages and culture communities, obtaining a multilingual ontology.

It is worth mentioning that the aforementioned scenarios can be combined in different ways, and that any combination of scenarios should include Scenario 1 because this scenario is made up of the core activities that have to be performed in any ontology development.

Currently, in the framework of the NeOn Methodology there are prescriptive methodological guidelines[6] for carrying out processes and activities involved in Scenario 1, Scenario 2, Scenario 3, Scenario 7, Scenario 8, and Scenario 9; and also for ontology evaluation and ontology evolution.

Furthermore, several project use cases and educational courses[7] are using the NeOn Methodology for building ontology networks.

CONCLUSION

We have presented the *NeOn Methodology for building ontology networks*. This poster supposes a step forward since it identifies a set of 9 flexible scenarios for building ontologies and ontology networks. The scenarios proposed are flexible because they can be combined among them.

ACKNOWLEDGMENTS

REFERENCES

[1] Fernández-López, M., Gómez-Pérez, A., Pazos A., Pazos J. *Building a Chemical Ontology Using Methontology and the Ontology Design Environment*. IEEE Intelligent Systems & their applications 4(1). 1999.

[2] Pinto, H. S., Tempich, C., Staab, S.: *DILIGENT: Towards a fine-grained methodology for DIstributed, Loosely-controlled and evolvInG Engineering of oNTologies*. 16th European Conference on Artificial Intelligence (ECAI 2004), pp. 393--397.

[3] Staab, S., Schnurr, H.P., Studer, R., Sure, Y.: *Knowledge Processes and Ontologies*. IEEE Intelligent Systems 16(1):26–34. (2001).

[4] Suárez-Figueroa, M. C., Gómez-Pérez, A.: *Towards a Glossary of Activities in the Ontology Engineering Field*. 6th Language Resources and Evaluation Conference (LREC 2008). Marrakech (Morocco).

---

[2] Scenarios are valid for both building ontologies and ontology networks.

[3] A module is a part of the ontology that defines a relevant set of terms.

[4] An ontology statement contains: subject, predicate, and object.

[5] http://ontologydesignpatterns.org

[6] Deliverables D5.4.1, D5.3.2, and D5.4.2 (http://www.neon-project.org/)

[7] Deliverables D5.6.2, D5.7.1, and D8.3.1 (http://www.neon-project.org/)