NeOn Methodology for Building Ontology Networks: a Scenario-based Methodology

Asunción Gómez-Pérez and Mari Carmen Suárez-Figueroa

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

Abstract. The 1990s and the first years of this new millennium have witnessed the growing interest of many practitioners in methodologies that support the creation of single or isolated ontologies. All these approaches have supposed a step forward since they have transformed the art of constructing single ontologies into an engineering activity. With the goal of speeding up the ontology development process, ontology practitioners are starting to reuse and re-engineer as much as possible knowledge resources (such as ontologies, thesauri, lexicons, and classification schemas), which already have reached some degree of consensus. In this paper, we present the set of nine scenarios identified in the NeOn Methodology framework. Additionally, we present how such scenarios have been followed in different use cases.

Keywords: Methodology, ontology development, reuse, re-engineering.

1 Introduction

There are well known methodological approaches (e.g., METHONTOLOGY [1], On-To-Knowledge [5], and DILIGENT [4]) that have gone a step forward by having transformed the art of constructing single ontologies[1] into an engineering activity. Other methods that use ontology learning techniques [2] propose to (semi)-automatically build ontologies.

However, the development of ontologies in different international and national projects has revealed that there are different ways to build ontologies. Just to name a few of them, in the Esperonto[2] project ontologies were built from scratch using METHONTOLOGY; in Knowledge Web[3] the aligning and versioning of ontologies was treated as well as the use of best practices or patterns; and in the SEEMP[4] project, a good ontology specification document helped to find existing ontologies and consensual knowledge resources that were re-engineered into ontologies.

Up to date, there are no methodologies that help ontology developers to build large ontologies in settings where distributed teams could collaboratively build

---

[1] A *single ontology* is an ontology that has not got any type of relationship (domain dependent or independent) with other ontologies.
[2] http://www.esperonto.net
[3] http://knowledgeweb.semanticweb.org
[4] http://www.seemp.org

ontologies by reusing and possibly re-engineering knowledge resources, using alignments and having in mind the continuous evolution of the ontologies.

It is foreseeable that the Semantic Web of the future will be characterized by using a large number of ontologies embedded in ontology networks[5] built collaboratively by distributed teams. Such networks may include ontologies that already exist, mappings among ontologies, ontologies in continuous evolution, or they could be developed by reusing available ontological and/or non-ontological resources (such as glossaries, lexicons, classification schemes, and thesauri). For this reason, our objective is to create the NeOn Methodology that supports both the collaborative aspects of ontology development and the reuse and dynamic evolution of ontology networks.

In this paper we have identified a set of *nine scenarios for building ontologies and ontology networks in the framework of the NeOn Methodology*, emphasizing the reuse of existing knowledge resources (ontological and non-ontological), generalizing from previous experiences, covering the drawbacks of the existing methodologies, and taking into account the new trends based on collaboration and dynamism. We also present how such scenarios have been used in different use cases.

The rest of the paper is organized as follows: Section 2 presents the state of the art on methodologies for ontology development; Section 3 describes the research methodology followed to create the NeOn Methodology; Section 4 explains the set of scenarios for building ontology networks; and Section 5 shows the use of the scenarios in different project use cases. Finally, Section 6 provides some conclusions.

2 State of the Art on Methodologies for Ontology Development

METHONTOLOGY, On-To-Knowledge, and DILIGENT were up to now the most referred methodologies for building ontologies. These methodologies mainly include guidelines for single ontology construction ranging from ontology requirements specification to ontology implementation and they are mainly targeted to ontology researchers. In this section we compare them according to the following dimensions: collaboration[6] and dynamism[7]. Since reuse can be seen as a kind of collaboration, we have also analyzed the degree of coverage of these methodologies with regard to the reuse of ontological and non-ontological resources.

METHONTOLOGY [1] enables the construction of ontologies at the knowledge level. It includes (a) the identification of the ontology development process; (b) a life cycle based on evolving prototypes; and (c) some techniques to carry out management, development-oriented, and support activities. With respect to the aforementioned dimensions, notions of *collaboration* are not included. Although *dynamic* aspects are mentioned, detailed guidelines about how to manage versions are not provided. Regarding the reuse of knowledge resources, METHONTOLOGY includes the list of activities to be carried out during *ontology reuse and re-engineering processes*, but it does not provide

---

[5] An *ontology network* is a collection of ontologies related together via a variety of different meta-relationships such as mapping, modularization, version, and dependency relationships.
[6] Collaboration refers to consider distributed ontology engineering among heterogeneous and geographically distributed groups of domain experts and ontology practitioners.
[7] Dynamism refers to the evolution and versioning of the ontologies.

detailed guidelines for such processes, nor does it consider different levels of granularity during the reuse of ontological resources (e.g., modules or statements). Moreover, METHONTOLOGY does not consider the *reuse of ontology design patterns (ODPs)*[8] neither the *reuse nor re-engineering of non-ontological resources*.

The *On-To-Knowledge methodology* [5] proposes to build ontologies taking into account how these are going to be used in knowledge management applications. The processes proposed by this methodology are the following: feasibility study, kickoff, refinement, evaluation, and maintenance. Regarding the aspects analyzed in this paper, such a methodology does not consider *collaboration*. Regarding the *dynamic* evolution of ontologies, it proposes to create a new version after testing possible effects to the application. However, no guidelines about how to manage different versions and when to create them are provided. With respect to the reuse of knowledge resources, in the kickoff process it is mentioned that developers should look for potentially *reusable ontologies*. However, this methodology does not provide detailed guidelines for identifying such ontologies nor for reusing them. Besides, the methodology does not explicitly mention guidelines for the *reuse and re-engineering of non-ontological resources*, nor for the *reuse of ODPs*.

The *DILIGENT methodology* [4] is intended to support domain experts in a distributed setting in order to engineer and evolve ontologies. This methodology is focused on collaborative and distributed ontology engineering. Its ontology development process includes the following five activities: building, local adaptation, analysis, revision, and local update. With respect to the dimensions analyzed here, *collaboration* is the central point in this methodology. Regarding the *dynamic* dimension, DILIGENT proposes the creation of different versions of the ontology, but it does not provide guidelines on how to manage such versions or when to create different versions, nor how changes can affect. With regard to the reuse of knowledge resources, the methodology does not include guidelines for the *reuse and re-engineering of existing knowledge resources*.

Taking into account the current status of methodologies, our aim is to create the *NeOn Methodology for building ontologies and ontology networks* covering the drawbacks presented in the three methodologies analyzed, and benefiting from them.

3 Research Methodology

As *approach to build the NeOn Methodology*, we used a *"divide and conquer"* strategy by decomposing the general problem to be solved in different subproblems. To obtain the solution to the general problem, that is, the development of an ontology network, the solutions to the different subproblems are combined. In our case, the subproblems are the 9 identified scenarios, which are composed of processes and activities. To identify the scenarios, we based on the different studies carried out to revise the state of the art of ontology development, on the experience of building ontologies in different projects, and on the analysis of the NeOn[9] use cases.

It is worth mentioning that based on [3], the *main principles that guided the construction of the NeOn Methodology* for building ontology networks were:

---

[8] ODPs are considered ontology modeling solutions that after being recurrently used for solving similar design problems can be identified as generalized design solutions.
[9] http://www.neon-project.org

1. The methodology should be general enough in the sense that it should not be driven to solve ad-hoc cases.
2. The methodology should be independent of the existing technology.
3. The methodology should define each process or activity precisely; state clearly its purpose, inputs and outputs, the actors involved, when its execution is more convenient, and the set of methods, techniques and tools to be used.
4. The methodology should facilitate a promptly assimilation by software developers and ontology practitioners.

Apart from the aforementioned principles, we also used the following ones, based on ideas presented in [3]:

*   **Completeness**. A methodology must consider all the cases presented and propose solutions to all of them.
*   **Consistency**. A methodology must produce the same set of results for the same problem, independently of who carries it out.
*   **Efficiency**. A methodology must be efficient and able to achieve its objective. This means that the methodology should allow the construction of ontologies with fewer resources (time, money, etc.) and with better quality.
*   **Environment**. Methodologies can be classified into scientific (where ideas are validated) and technological (where artifacts are built). The NeOn Methodology can be considered as a technological one, because the main result after applying it should be a technological product, that is, an ontology network.
*   **Finiteness**. A methodology must be composed of a finite number of the elements and a finite number of activities and processes.
*   **Flexibility**. A methodology must allow the adaptation to concrete needs.
*   **Perspectives**. A methodology must facilitate the use of different approaches.
*   **Transparency**. A methodology must be like a white box, allowing the users to know in every moment the processes or activities that are being performed, who is performing them, etc.
*   **Usability**. The degree of difficulty in using the methodology must be minimal.

4 NeOn Scenarios for Building Ontology Networks

The NeOn Methodology for developing ontology networks takes into account the existence of multiple ontologies in ontology networks, the collaborative ontology development, the dynamic dimension, and the reuse and re-engineering of knowledge resources. One of the key elements in this methodology is the set of 9 scenarios identified for building ontologies and ontology networks, shown in Fig. 1. Directed arrows with numbered circles associated represent the different scenarios. Each scenario is decomposed into different processes or activities that are represented with colored circles or with rounded boxes. Such processes and activities are defined in the NeOn Glossary of Processes and Activities [6]. Fig. 1 also shows (as dotted boxes) existing knowledge resources

to be reused and possible outputs (ontology networks and alignments) that result from the execution of some of the scenarios.
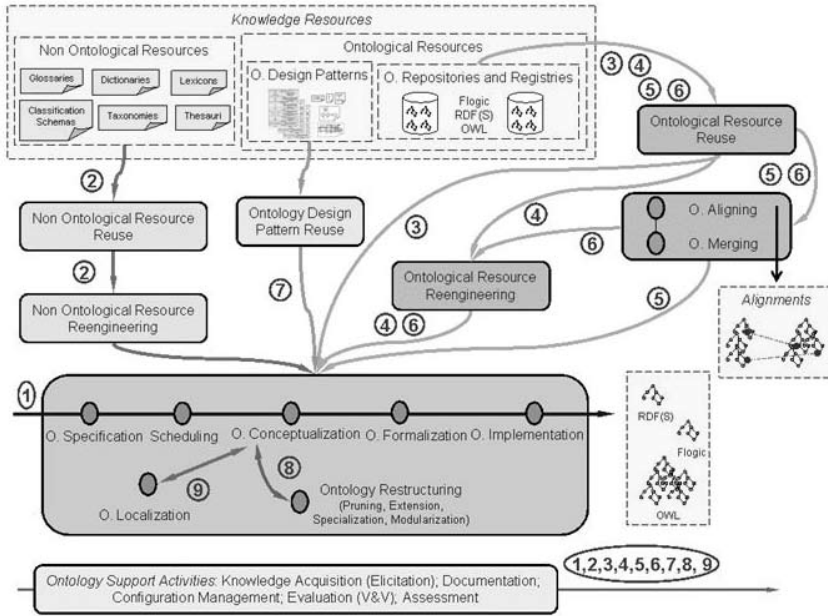

Fig. 1. Scenarios for Building Ontology Networks.

The most common scenarios 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.** Ontology developers develop the ontology network from scratch, that is, without reusing existing knowledge resources. The ontology development should start with the *knowledge acquisition activity*, which should be carried out during the whole development. Simultaneously with the knowledge acquisition, 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) that includes the purpose, the scope and the implementation language of the ontology network, target group and intended uses of the ontology network, and the set of requirements the ontology network should fulfill, mainly in the form of competency questions (CQs). After such an activity, it is advisory to carry out a quick search for knowledge resources using as input the terms appearing in the ORSD. The search results allow knowing which types of resources are available for a possible reuse during the development. Then, the *scheduling activity* must be carried out, using the ORSD and the results of such a quick search. After the scheduling, the ontology developers should carry out the rest of the activities (*conceptualization*, *formalization*, and *implementation*) using METHONTOLOGY or On-To-Knowledge.

***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. Then, the *non-ontological resource re-engineering process* should be performed to transform the selected NORs into ontologies.

Scenario 3: Reusing ontological resources. Ontology developers use existing ontological resources for building ontology networks. There are different ways of reusing ontological resources: (a) ontologies can be reused *as a whole*; (b) only one part or *module*[10] can be reused; and (c) *ontology statements*[11] can be reused. Terms that appear in the ORSD are used by ontology developers in the search for existing ontological resources to be reused. For integrating the ontological resources to be reused, ontology developers can decide to reuse them such as they are in the ontology network being developed following the activities of Scenario 1. If the resources are needed, for example, in another implementation language, ontology developers can perform a re-engineering process following Scenario 4. Another possibility is to have several ontological resources on the same domain; then, the ontological resources could be merged to obtain a new ontological resource following Scenarios 5 or 6.

Scenario 4: Reusing and re-engineering ontological resources. Ontology developers reuse existing ontological resources and re-engineer them before their integration in the ontology network. The *ontological resource re-engineering process* is composed of the following activities [6]: *ontological resource reverse engineering, ontological resource restructuring, and ontological resource forward engineering*. These activities might be carried out at four different levels, depending on the needs of each particular case: at the specification level, at the conceptualization level, at the formalization level, and at the implementation level. After carrying out the ontological resource re-engineering process, ontology developers should integrate its result into the corresponding activity of Scenario 1.

Scenario 5: Reusing and merging ontological resources. Ontology developers reuse and merge existing ontological resources in the development of the ontology network. 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, possibly overlapping, source ontological resources. It is also possible that ontology developers wish only to establish alignments among the selected resources in order to create the network.

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 has the same sequence of activities as Scenario 5; however, here ontology developers can decide not to use the set of merged ontological resource such as it is, but to re-engineer it. Once the set of merged ontological resources is re-engineered, the result of such process should be integrated in the corresponding activity of Scenario 1.

Scenario 7: Reusing ontology design patterns. Ontology developers access ODPs repositories[12] to reuse ODPs for different purposes: to reduce modeling difficulties, to speed up the modeling process, or to check the adequacy of modeling decisions.

---

[10] A module is a part of the ontology that defines a relevant set of terms.
[11] An ontology statement contains three components: *subject*, *predicate*, and *object*.
[12] http://ontologydesignpatterns.org

Scenario 8: **Restructuring ontological resources.** Ontology developers restructure ontological resources to be integrated in the ontology network being built. Such an *ontology restructuring activity* can be performed in the following ways: (1) modularizing the ontology in different ontology modules; (2) pruning the branches of the taxonomy not considered necessary; (3) extending the ontology including (in width) new concepts and relations; and (4) specializing those branches that require more granularity and including more specialized domain concepts and relations.

Scenario 9: **Localizing ontological resources.** Ontology developers adapt an existing ontology to one or various languages and culture communities, obtaining as a result a multilingual ontology. Once the ontology has been conceptualized, its adaptation to a particular natural language different from the language used in the conceptualization can be required. Such an adaptation requires the translation of all ontology labels into one or several natural languages, being these languages other than the original language of the conceptualization.

In this section we have included a brief description of the 9 scenarios identified in the NeOn Methodology. These scenarios can be combined in different ways, and 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.

It is worth also mentioning that in the framework of the NeOn Methodology there are prescriptive methodological guidelines[13] for carrying out processes and activities involved in Scenario 1 (ontology requirements specification and scheduling), Scenario 2, Scenario 3, Scenario 7, Scenario 8 (ontology modularization), and Scenario 9; and also for ontology evaluation and ontology evolution.

5 Applying the NeOn Scenarios

In this section we briefly present 2 different use cases in which the NeOn Scenarios has been applied: the NeOn Invoicing Management case and the SEEMP case.

The *ontology network for the NeOn invoicing management use case*[14], which contains all the concepts related to the invoice management in the pharmaceutical industry, was built following Scenarios 1, 2, 3, 8, and 9. Ontology developers reused different non-ontological resources for electronic invoicing (e.g., Universal Business Language (UBL), EDIFACT, xCBL) as part of Scenario 2. A generic description of concepts from SUMO, DOLCE, and W3C Time ontologies were reused as part of Scenario 3. The general invoice ontology was specialized for each laboratory involved in the use case as part of Scenario 8. Regarding Scenario 9, ontology users belong to different regions in Spain in which different languages are used.

The *ontology network for the SEEMP use case* should represent knowledge about Curricula Vitae and job offers among Employment Services (ES) placed in different countries. Such a network is formed by (1) a set of local ontology networks, representing each of them the local and particular view that each

---

[13] Deliverables D5.4.1, D5.3.2, and D5.4.2 (http://www.neon-project.org/)
[14] Deliverable D8.3.1 (http://www.neon-project.org/)

ES has of the employment market, and (2) a reference ontology network representing a standardized and agreed view of the employment market at the European level. The SEEMP ontology is a network of ontology networks that was built following Scenarios 1, 2, 4 and 5. To build the local and the reference ontology networks, several NORs (international standards such as NACE and ISCO-88; ES classifications; and international codes such as ISO 3166) were reused and re-engineered as part of Scenario 2. The DAML time ontology was reused and re-engineered as part of Scenario 4. Finally, as part of Scenario 5, a set of alignments were defined between each local ontology and the reference ontology in order to create the SEEMP network.

6 Conclusion

In this paper, we have presented the *NeOn Methodology for building ontology networks*, a scenario-based methodology, covering the drawbacks presented in the three aforementioned methodologies and benefiting from their advantages. This paper 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, almost the contrary to the rigid scenario encountered for building ontologies from scratch and presented in METHONTOLOGY, On-To-Knowledge and DILIGENT. Additionally, this paper shows how such scenarios have been followed in different use cases within European projects.

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):37–46. 1999.
2. Gómez-Pérez, A., Manzano-Macho, D. An overview of methods and tools for ontology learning from texts, Knowl. Eng. Rev. 19 (3) (2004). 187–212.
3. Paradela, L.: PhD Thesis: Una Metodología para la Gestión del Conocimiento. Spain. Universidad Politécnica de Madrid, 2001.
4. 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.
5. Staab, S., Schnurr, H.P., Studer, R., Sure, Y.: Knowledge Processes and Ontologies. IEEE Intelligent Systems 16(1):26–34. (2001).
6. 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).