

# The NeOn Methodology framework: A scenario-based methodology for ontology development

Mari Carmen Suárez-Figueroa    Asunción Gómez-Pérez    and Mariano Fernández-López

*Ontology Engineering Group, Escuela Técnica Superior de Ingenieros Informáticos, Universidad Politécnica de Madrid, Madrid, Spain*

This paper describes a scenario-based methodology called the NeOn Methodology framework. The aim of this framework is to speed up the construction of ontologies and ontology networks by reusing available knowledge resources (ontologies, non-ontological resources and ontology design patterns). The methodology is founded on four pillars: (1) a glossary of processes and activities; (2) a set of nine scenarios for building ontologies and ontology networks; (3) two ontology life-cycle models; and (4) a set of prescriptive methodological guidelines for performing specific processes and activities. The paper also shows how to apply this methodology framework to develop an ontology network within the multimedia domain. Finally, it presents an empirical evaluation of the NeOn Methodology framework.

## 1. Introduction

Managing information and knowledge in organizations is a crucial issue. The main resources of the systems and applications used by organizations today are vast quantities of data and knowledge in different types of distributed formats. These data are normally structured, but some are very heterogeneous and, hence, not necessarily interoperable.

Nowadays, ontologies are used for making information explicit and allowing it to be shared. A good solution for improving data and knowledge management would be to create an ontology-based application that could properly manage all the organization's knowledge. Such a process would involve modeling this knowledge in an ontology.

The emergence of ontology development methodologies over the last few decades has transformed the art of building ontologies into an engineering activity. Several researchers such as Gómez-Pérez et al. (2003), Stadlhofer et al. (2013) and Iqbal et al. (2013) have provided detailed surveys and comparative

studies of these methodologies. METHONTOLOGY (Fernández-López et al., 1997, 1999) and On-To-Knowledge (Staab et al., 2001) are considered to be the most complete methodologies for building single ontologies from scratch. Worthy of note is the ontology-building method described by Noy and McGuinness (2001), which reports and shows how to avoid some common ontology development errors. Several methodologies for collaborative and distributed ontology development, such as DILIGENT (Pinto et al., 2004), and tools, such as WebProtégé (Tudorache et al., 2013), have recently appeared (Simperl & Luczak-Rösch, 2014). WebProtégé is actually a lightweight ontology editor for the Web that provides collaboration features that are integrated into its implicit ontology development methodology. The above methodologies include high-level guidelines for single ontology construction but do not provide sufficient details about the methods and techniques that they employ. The guidelines normally: (a) cover all issues from specification through to implementation, but only a few provide unspecific guidelines on ontology reuse; and (b) mainly target ontology researchers.

On the other hand, several initiatives in the ontology development field have revealed ad-hoc alternatives for building ontologies, most of which, however, are not included in the above mentioned methodologies. Some examples are the Knowledge Web<sup>1</sup> Network of Excellence, which tackles both the issue of aligning and versioning ontologies and the use of best practices as part of W3C activities; the SEKT<sup>2</sup> project, whose main goal is the argumentative development of ontologies; and the UMLS<sup>3</sup> project, which focuses on reengineering the UMLS Semantic Network into an OWL ontology. Additionally, some projects such as the UK PRODIGY and Drug Ontology Projects (Hodge, 2000) aim to transform tangled hierarchies (e.g., hierarchies derived from ambiguous “broader than/narrower than” thesauri in library science) into formal ontologies; whereas the WonderWeb<sup>4</sup> project aims to build a modular ontology library around DOLCE.<sup>5</sup> This can be seen as an initial attempt to build an ontology network.

These new ways of developing ontologies have led to a very large number of ontologies being embedded in ontology networks and developed by reusing and reengineering knowledge resources. The resulting ontology networks (Haase et al., 2006) are defined as collections of ontologies related by means of a variety of different meta-relationships, such as mapping, modularization, version and dependency.<sup>6</sup>

Despite the wide range of proposed ontology development methods and methodologies, none, as far as we know, supports all the new trends currently emerging in ontology network development. These trends comprehend complex scenarios requiring the reuse and reengineering of knowledge resources (ontologies, non-ontological resources and ontology design patterns) and involving software developers and ontology practitioners. The above methodologies do not provide engineers with guidelines to take them through the construction of a product that is now much more complex than it was several years ago. Even though some contributions (see, e.g., Francesconi et al., 2010; Barresi et al., 2008) take into consideration ontology reuse, transformation and integration, they focus on a particular scenario. However, if engineers opt to follow different methodologies, they may find that such methodologies contain descriptions of similar processes and activities that bear different names. This will lead to a heavier workload and, possibly, misunderstandings.

---

<sup>1</sup><http://knowledgeweb.semanticweb.org>.

<sup>2</sup><http://www.sekt-project.com/>.

<sup>3</sup><http://www.nlm.nih.gov/mesh/umlsforelis.html>.

<sup>4</sup><http://wonderweb.man.ac.uk/>.

<sup>5</sup><http://www.loa.istc.cnr.it/old/DOLCE.html>.

<sup>6</sup>Also noteworthy in relation to the ontology networks addressed in this paper is the notion of ‘hyperontologies’ (heterogeneously structured ontologies, which offer distributed networks of ontologies written in different formalisms) (Kutz et al., 2010).

On this ground, we present the *NeOn Methodology framework*, a scenario-based methodology that provides accurate details for developing ontologies and ontology networks<sup>7</sup> and pays special attention to the reuse and reengineering of knowledge resources. Unlike previous methodological approaches, this framework does not prescribe a rigid procedure, but suggests a variety of scenarios and guidelines for performing different processes and activities. The methodology gives a simple and direct description, which is comprehensible for both ontology practitioners and software developers, of all the details. The nine scenarios proposed in the framework cover commonly occurring situations, such as when the ontologies need to be reengineered, aligned, modularized and localized to support different languages and cultures, and require integration with both ontology design patterns and non-ontological resources such as classification schemes or thesauri.

The framework presented in this paper together with the methods and techniques that it includes (which are merely outlined here for reasons of space) are the result of abstracting our ontology network development experience as part of complex projects mostly developed with partners from several countries.

The rest of the paper is organized as follows. Section 2 describes the NeOn Methodology framework. Section 3 shows how to apply the methodological framework to develop an ontology network of multi-media issues. Section 4 reports the empirical evaluation of the NeOn Methodology framework. Finally, Section 5 discusses the conclusions and future work.

## 2. The NeOn Methodology framework

The NeOn Methodology framework (Suárez-Figueroa, 2010) is a scenario-based methodology that provides accurate details about key aspects of the ontology engineering process, paying special attention to the reuse and reengineering of ontological and non-ontological resources. This framework is founded on four pillars: (1) a glossary of processes and activities; (2) a set of nine scenarios for building ontologies and ontology networks; (3) two modes of organizing ontology developments, called ontology life-cycle models; and (4) a set of precise methodological guidelines for performing specific processes and activities.

### 2.1. *NeOn glossary of processes and activities*

Ontology engineering is an evolving discipline: new ontology building approaches, such as creating ontology networks and reusing any available knowledge resource, appear all the time, and new applications based on semantic issues, such as the Linked Data initiative, are also emerging. Because of this constant evolution, however, the terminology used by ontology practitioners is sometimes fuzzy. For instance, the difference between ontology modification and ontology update is not absolutely clear, which may cause misunderstandings. Whereas other activities have multiple natural language definitions (e.g., ontology merging), which may also obscure their meaning.

As established in Bloom's taxonomy (Bloom, 1956), the lowest level of knowledge in any field is its terminology, and no proper knowledge transmission is possible without it. Therefore, the terminology re-

---

<sup>7</sup>Meta-relationships such as mappings, modularization, version and dependency are central issues in ontology networks. Given that this paper focuses on the methodology framework rather than on the details of its activities, these relations are not described thoroughly here. Interested readers are referred to (Suárez-Figueroa et al., 2012), which contains chapters specifically addressing matching, modularization and evolution, etc.

quires clarification so that the people using the NeOn Methodology to build ontologies can communicate with no trouble.

For this reason, we have created a glossary of ontology engineering terms, called the NeOn glossary, as part of the NeOn Methodology framework. Its main purpose is to unify and specify the ontology engineering terminology.

The terms included in the glossary refer to the processes and activities involved in ontology development, where *activity* is an action to be performed including its required input and output information and *process* is a group or set of ordered activities.

The NeOn glossary<sup>8</sup> (Suárez-Figueroa, 2010; Suárez-Figueroa et al., 2013) provides natural language definitions and explanations in English of the 59 key processes and activities potentially involved in ontology construction. The glossary is ordered alphabetically by type of term (process or activity), and, if necessary, each term is related to other terms. The glossary was built collaboratively by means of consensus decision making. The consensus decision-making process involved a group of about thirty geographically dispersed experts (ontology engineers, ontology editors and users).

Figure 1 shows the whole NeOn glossary of processes and activities. The meanings of the different arrow types are as follows: (1) arrows with dashed lines denote a process that is divided into activities;

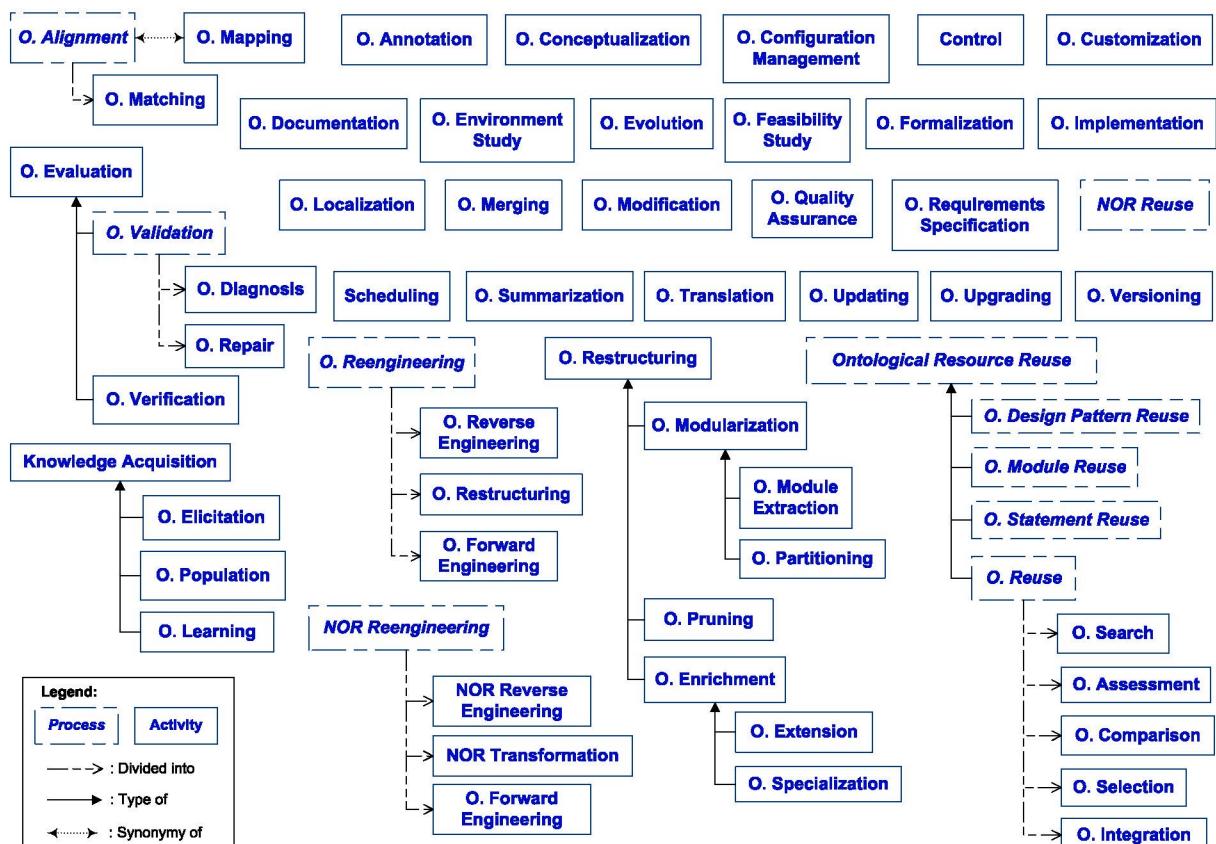


Fig. 1. NeOn glossary of processes and activities. (Colors are visible in the online version of the article; <http://dx.doi.org/10.3233/AO-150145>.)

<sup>8</sup><http://mayor2.dia.fi.upm.es/oeg-upm/files/pdf/NeOnGlossary.pdf>.

(2) arrows with solid lines denote type-of relationships; and (3) arrows with dotted lines denote synonymy. Some examples of terms defined in the glossary are: *ontology enrichment* (which refers to the activity of adding new conceptual structures, e.g., concepts, roles and axioms to an ontology); *ontology evaluation* (which refers to the activity of checking the technical quality of an ontology against a frame of reference); and *ontology partitioning* (which refers to the activity of dividing an ontology into a set (not necessarily disjoint) of modules which together form an ontology, but which can be treated separately).

## 2.2. Scenarios for building ontologies and ontology networks

Based on experience gained from our involvement in different types of projects, we identified three different ontology-building situations: (1) single ontologies; (2) sets of interconnected single ontologies; and (3) ontology networks. The first case refers to ontologies that have no relationship whatsoever (domain dependent or independent) with other ontologies. The second situation denotes a set of ontologies linked by some kind of domain-dependent relationship. The last instance refers to a group of interconnected single ontologies that are linked to each other by a variety of different meta-relationships or domain-independent relationships (based on Haase et al., 2006). Therefore, it can be said that when software developers and ontology practitioners explicitly define meta-relationships such as mapping, modularization, version and dependency between a set of interconnected ontologies and/or between an ontology and its components, they are developing an ontology network.

In this context, we have identified a set of nine flexible scenarios for building ontologies and ontology networks, placing special emphasis on knowledge resource reuse and reengineering. The set of the nine most widespread scenarios identified in Fig. 2 provides information on the different ontology development pathways. Figure 2 shows the directed lines with their associated numbered circles that represent the different scenarios. Scenarios are sets of ordered processes and activities defined in the NeOn glossary (Section 2.1). Processes are represented by rounded boxes with a solid black border, whereas activities are represented by circles with a solid black border. The possible inputs (knowledge resources available for reuse) and the possible outputs (ontologies, ontology networks and alignments) of the execution of some of the scenarios are represented as boxes with a dotted border.

There follows a non-exhaustive summary of the nine scenarios.

- *Scenario 1: From specification to implementation.* This scenario refers to the development of ontologies from scratch. It consists of the core activities to be performed in any ontology development and should be combined with the other scenarios. In this scenario, ontology developers should first specify the requirements to be met by the ontology by means of the *ontology requirements specification activity*.<sup>9</sup> The objective here is to output the ontology requirements specification document (ORSD) (Suárez-Figueroa, 2010). This document includes the purpose, scope and implementation language of the ontology network, its target group and intended uses, and the set of requirements to be met by the ontology network, as well as a glossary of the terms that the ontology should cover. The terms are used to search for possible knowledge resources that will be reused at later stages of the development process. Possible resources are ontologies stored in repositories or non-ontological resources (e.g., thesauri) negotiated by large organizations (e.g., ISO, FAO, WHO) or owned by private corporations. The candidate resources provide clues for identifying the scenarios to be applied during ontology development. Developers should then perform the *scheduling activity* (Suárez-Figueroa, 2010) in order to identify, on the one hand, the set of scenarios to be applied, together with the different activities and processes to be performed during ontology development

---

<sup>9</sup>Note that the processes and activities defined in the NeOn glossary are shown in italics in the description of scenarios.

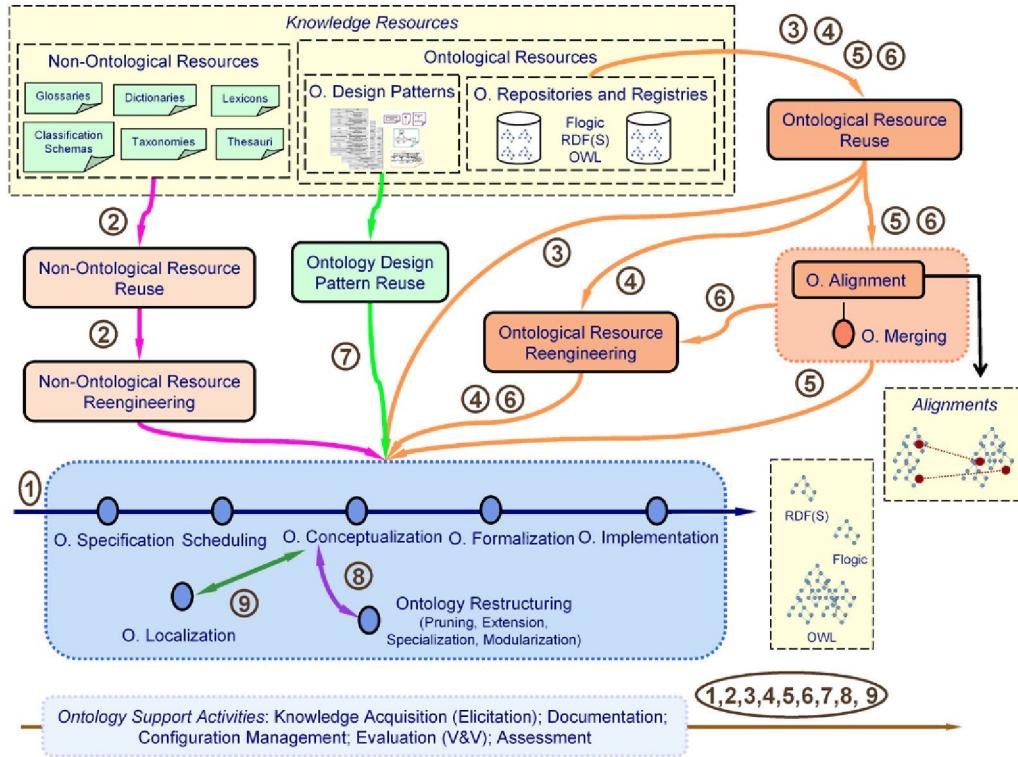


Fig. 2. Ontology network-building scenarios. (Colors are visible in the online version of the article; <http://dx.doi.org/10.3233/AO-150145>.)

and, on the other hand, their arrangement, and the time and resources needed for their completion. This activity outputs a schedule in the form of a Gantt chart. The following activities are: (1) the *ontology conceptualization activity*, in which knowledge is organized and structured into meaningful models at the knowledge level; (2) the *ontology formalization activity*, in which the conceptual model is transformed into a semi-computable model; and (3) the *ontology implementation activity*, in which a computable model (implemented in an ontology language) is generated.

- **Scenario 2: Reusing and reengineering non-ontological resources (NORs).** A non-ontological resource is defined as a knowledge resource (glossary, dictionary, lexicon, classification scheme, taxonomy and thesaurus) whose semantics has not been formalized by an ontology. This scenario is applicable when ontology developers have access to NORs that, to some extent, cover the domain of the ontology network being built and that normally reflect some degree of community consensus; these resources can be used to develop the new ontology. Ontology developers should perform the *non-ontological resource reuse process*. This entails: (1) searching highly reliable Web sites, domain-related sites, and resources within organizations for NORs, using the terms included in the ORSD as input; (2) assessing candidate NORs using three criteria (coverage, precision and consensus); (3) selecting the most suitable NORs to be used to build the ontology network. After this selection, developers must carry out the *non-ontological resource reengineering process* to transform the selected NORs into ontologies. This process is divided into the following activities (Villazón-Terrazas et al., 2010): (1) *non-ontological resource reverse engineering*, whose goal is to analyze the selected NORs and identify their underlying components and then create repre-

sentations of the resources at different levels of abstraction (design, requirements and conceptual); (2) *non-ontological resource transformation*, whose goal is to generate a conceptual model from each selected NOR (a transformation based on the use of the so-called patterns for reengineering non-ontological resources (PR-NOR) (Villazón-Terrazas et al., 2010)); and (3) *ontology forward engineering*, whose goal is to output a new implementation of the ontologies on the basis of the new conceptual models identified in the *NOR transformation activity*. Ontology developers should then use the resultant ontologies as input for Scenario 1. This approach can both speed up ontology development, thereby cutting development costs, and deal with agreed-upon knowledge.

If the ultimate aim is to develop an ontology network, developers should explicitly define meta-relationships, such as *useImports* or *isReusing*, in order to denote that the final ontology is importing or reusing other ontologies (these ontologies are, in this case, the result of a NOR reengineering process).

- *Scenario 3: Reusing ontological resources.* This scenario is applicable when developers have access to ontological resources that cover, to some extent, the domain of the ontology network being built and can be reused to develop the new ontology. If the ultimate aim is to build an ontology network, then developers should explicitly define meta-relationships (e.g., *useImports* or *isReusing*) in order to model such reuse.

There are different ways of reusing ontological resources (Suárez-Figueroa, 2010). For example, ontological resources can be used as a whole, as a part or module,<sup>10</sup> and as ontology statements or triples.<sup>11</sup> In all three cases, the *ontological resource reuse process* entails: (1) searching for ontological resources by means of semantic search engines (such as Swoogle<sup>12</sup> and WATSON<sup>13</sup>) using the terms included in the ORSD as input; (2) assessing the ontological resources in order to find out if such resources satisfy the needs identified in the ORSD; (3) comparing the candidate ontological resources against a set of criteria (e.g., reuse economic cost, code clarity, content quality); (4) selecting the best set of ontological resources for the development; and (5) integrating the selected ontological resources as input for Scenario 1. This approach aims to: (a) reduce ontology development time and costs, (b) handle ontologies that are intended to spread good practices (from well-developed ontologies) and (c) increase the overall quality of ontological models.

- *Scenario 4: Reusing and reengineering ontological resources.* This scenario applies in cases where the outcomes of Scenario 3 are not entirely usable as they are, and the ontological resources therefore have to be reengineered to serve the intended purpose. The processes to be performed in this scenario are as follows: ontology developers should carry out first the *ontological resource reuse process*<sup>14</sup> (as explained in Scenario 3) and then the *ontological resource reengineering process* in order to select the most suitable ontological resources and reengineer the selected ontological resources, respectively. The *ontological resource reengineering process* is composed of three activities: (1) *ontological resource reverse engineering* (the activity of outputting a possible conceptual model on the basis of the code in which the ontology is implemented); (2) *ontological resource restructuring* (the activity of correcting and reorganizing the knowledge contained in an initial conceptual model and detecting missing knowledge); and (3) *ontological resource forward engineering*

---

<sup>10</sup>A module (d'Aquin et al., 2007) is the part of the ontology that defines the relevant set of terms for a particular purpose.

<sup>11</sup>An ontology statement or triple contains the following three components: subject, predicate and object.

<sup>12</sup><http://swoogle.umbc.edu/>.

<sup>13</sup><http://watson.kmi.open.ac.uk/WatsonWUI/>.

<sup>14</sup>In this case, meta-relationships such as *useImports* or *isReusing* should be defined if the objective is to develop an ontology network.

(the activity of outputting a new implementation of the ontology on the basis of the new conceptual model) (Suárez-Figueroa, 2010). Finally, ontology developers should use the resulting ontological resources after the reengineering process as input for Scenario 1.

The *ontological resource reengineering process* proposed here was inspired by the software reengineering process (Byrne, 1992).

- *Scenario 5: Reusing and merging ontological resources.* This scenario applies in cases where:
  - (a) several ontological resources from the same domain can be selected for reuse (Scenario 3); and
  - (b) ontology developers need to create a new ontological resource by *aligning* and *merging* two or more, possibly overlapping, ontological resources. Alternatively, the ontology developer may only need to establish mappings or alignments among the selected ontological resources in order to create the ontology network. Ontology developers should then use the outcomes as input for Scenario 1. If the ultimate aim is to build an ontology network, several types of meta-relationships should be defined in this scenario, including domain-independent relationships, such as *useImports* or *isReusing*, which represent the reuse of ontological resources, and meta-relationships such as *hasMapping*, which represent the alignments among resources.
- *Scenario 6: Reusing, merging and reengineering ontological resources.* This scenario applies in cases where the result of Scenario 5 (a merged ontological resource) is not usable as it is (e.g., we have an ontology in F-logic with a taxonomy built according to a particular criterion A, whereas the new ontology should be in OWL with a taxonomy built according to criterion B). In such a case, the merged ontological resource should be *reengineered* (as explained in Scenario 4) to serve the intended purpose. The resultant ontological resource should be used as input for Scenario 1. As in Scenario 5, when the objective is to build an ontology network, developers should include domain-independent relations such as *useImports* or *isReusing* and *hasMapping* to represent reuse and alignment, respectively.
- *Scenario 7: Reusing ontology design patterns (ODPs).* Ontology design patterns (Gangemi, 2005) represent modeling solutions to some well-known ontology modeling problems. The *ODP reuse process* is divided into the following activities: (1) search for possible ODPs for reuse in a repository or catalogue (examples of repositories are the Ontology Design Pattern Portal,<sup>15</sup> Semantic Web Best Practices and Deployment,<sup>16</sup> and the Ontology Design Patterns Public Catalog<sup>17</sup> (Egaña Aranguren et al., 2008; Egaña et al., 2008)); (2) select the ODPs that best match the modeling problem and (3) adapt selected ODPs. The search activity (1) entails conducting a careful analysis of the different information sections included in the ODP templates in order to find possible ODPs to solve the modeling problem. The selection activity (2) involves comparing the analyzed ontology design pattern templates against the real use case in order to find out the level of overlap. Finally, the adaptation activity (3) entails modifying (e.g., instantiating or extending) the selected ODPs. The ODPs are reused mainly during the *ontology conceptualization* and *implementation* activities. Ontology developers should then integrate the outcomes as input for Scenario 1. Note that, when the ultimate aim is to build an ontology network, developers should explicitly define meta-relationships such as *useImports* or *isReusing* in order to represent pattern reuse, since an ontology design pattern can be considered an ontology.
- *Scenario 8: Restructuring ontological resources.* This scenario applies in cases where the set of ontological resources (output by different scenarios) to be integrated in the new ontology needs to be

<sup>15</sup><http://ontologydesignpatterns.org/>.

<sup>16</sup><http://www.w3.org/2001/sw/BestPractices/>.

<sup>17</sup><http://www.gong.manchester.ac.uk/odp/html/index.html>.

reorganized. *Restructuring* involves activities that can be combined in any manner and order. The activities are as follows: (1) *modularize* the ontology to facilitate reuse of the knowledge included in the network;<sup>18</sup> (2) *prune* the branches of the taxonomy that are considered to be unnecessary to cover the ontology requirements included in the ORSD; (3) *extend* the ontology by adding (breadth-wise) new concepts and relationships;<sup>19</sup> and (4) *specialize* branches that require more granularity by introducing more specialized domain concepts and relationships.<sup>20</sup> Note that the *ontology restructuring activity* can be performed either independently, as explained in this scenario, or as part of the *ontological resource reengineering process*, as described in Scenario 4.

- **Scenario 9: Localizing ontological resources.** This scenario applies in cases where the terms of the ontology network have to be written in different natural languages. Developers should perform the *ontology localization activity* once the ontology is almost finished and ready for use. This activity requires the translation of all the ontology terms, usually in English, into a natural language (Spanish, French and German, among others) other than the language used in the *conceptualization*. One key task in this activity is to select the most appropriate linguistic assets (multilingual thesauri and electronic dictionaries, for example, EuroWordNet<sup>21</sup>) for translating the ontology terms.

Note that these scenarios can be flexibly combined in different ways, and any combination of scenarios should include Scenario 1, which is composed of the core ontology development activities. In fact, as Fig. 2 shows, the results of all the other scenarios should be integrated into the results output by Scenario 1. Support activities,<sup>22</sup> such as knowledge acquisition,<sup>23</sup> documentation, configuration management,<sup>24</sup> evaluation<sup>25</sup> and assessment, are necessary to assure the successful completion of an ontology development project. These activities should be carried out throughout ontology or ontology network development, that is, as part of any scenario used to develop the ontology (network). The intensity of such support activities depends on the particular development phase.

Finally, as already mentioned, scenarios are sets of the ordered processes and activities defined in the NeOn glossary. Table 1 matches the set of scenarios to the processes and activities defined in the glossary. There are two options: (1) a process or an activity can be carried out as part of a particular scenario; or (2) a process or an activity can be carried out as part of more than one scenario.

### 2.3. Two ontology network life-cycle models

Software engineering acknowledges that no one standard life-cycle model is valid for all software development projects and whether or not a life-cycle model is appropriate for a specific project depends on

---

<sup>18</sup>In this case, meta-relationships such as *containsModules* should be explicitly defined.

<sup>19</sup>In this case, meta-relationships such as *isExtension* should be explicitly defined when the aim is to build an ontology network.

<sup>20</sup>In this case, meta-relationships such as *isSpecialization* should be explicitly defined when the aim is to build an ontology network.

<sup>21</sup><http://www.ilc.uva.nl/EuroWordNet/>.

<sup>22</sup>Definitions for the support activities mentioned here are available in the NeOn glossary (available at <http://mayor2.dia.fi.upm.es/oeg-upm/files/pdf/NeOnGlossary.pdf>).

<sup>23</sup>The goal of this activity is to acquire knowledge from experts of a given domain or through some kind of (semi)automatic process, which is called ontology learning. Gómez-Pérez and Manzano-Macho (2004) presented a survey of ontology learning techniques.

<sup>24</sup>This activity is related to ontology evolution. A summary of the guidelines for this activity is presented in the Appendix. Palma et al. (2012) presented detailed guidelines for evolving ontologies.

<sup>25</sup>An excerpt of the guidelines for this activity is presented in the Appendix. Sabou and Fernandez (2012) presented detailed guidelines for evaluating ontologies.

Table 1  
Processes and activities and their related scenarios

Process	Related scenarios
<i>Ontology Alignment</i>	S5, S6
<i>Ontology Design Pattern Reuse</i>	S7
<i>Ontology Module Reuse</i>	S3, ..., S6
<i>Ontology Reengineering</i>	S4, S6
<i>Non-Ontological Resource Reengineering</i>	S2
<i>Non-Ontological Resource Reuse</i>	S2
<i>Ontological Resource Reuse</i>	S3, ..., S6
<i>Ontology Reuse</i>	S3, ..., S6
<i>Ontology Statement Reuse</i>	S3, ..., S6
<i>Ontology Validation</i>	S1, ..., S9
Activity	Related scenarios
<i>Ontology Annotation</i>	S1, ..., S9
<i>Ontology Assessment</i>	S1, ..., S9
<i>Ontology Comparison</i>	S3, ..., S6
<i>Ontology Conceptualization</i>	S1
<i>Ontology Configuration Management</i>	S1, ..., S9
<i>Control</i>	S1, ..., S9
<i>Ontology Customization</i>	S8
<i>Ontology Diagnosis</i>	S1, ..., S9
<i>Ontology Documentation</i>	S1, ..., S9
<i>Ontology Elicitation</i>	S1, ..., S9
<i>Ontology Enrichment</i>	S4, S6, S8
<i>Ontology Environment Study</i>	S1
<i>Ontology Evaluation</i>	S1, ..., S9
<i>Ontology Evolution</i>	S1
<i>Ontology Extension</i>	S4, S6, S8
<i>Ontology Feasibility Study</i>	S1
<i>Ontology Formalization</i>	S1
<i>Ontology Forward Engineering</i>	S2, S4, S6
<i>Ontology Implementation</i>	S1
<i>Ontology Integration</i>	S2, ..., S6
<i>Knowledge Acquisition for Ontologies</i>	S1, ..., S9
<i>Ontology Learning</i>	S1, ..., S9
<i>Ontology Localization</i>	S9
<i>Ontology Mapping</i>	S5, S6
<i>Ontology Matching</i>	S5, S6
<i>Ontology Merging</i>	S5, S6
<i>Ontology Modification</i>	S1
<i>Ontology Modularization</i>	S4, S6, S8
<i>Ontology Module Extraction</i>	S4, S6, S8
<i>Ontology Partitioning</i>	S4, S6, S8
<i>Ontology Population</i>	S1, ..., S9
<i>Ontology Pruning</i>	S4, S6, S8
<i>Ontology Quality Assurance</i>	S1, ..., S9
<i>Ontology Repair</i>	S1, ..., S9

Table 1  
(Continued)

Activity	Related scenarios
<i>Ontology Requirements Specification</i>	S1
<i>Non-Ontological Resource Reverse Engineering</i>	S2
<i>Non-Ontological Resource Transformation</i>	S2
<i>Ontology Restructuring</i>	S4, S6, S8
<i>Ontology Reverse Engineering</i>	S4, S6
<i>Scheduling</i>	S1
<i>Ontology Search</i>	S3, ..., S6
<i>Ontology Selection</i>	S3, ..., S6
<i>Ontology Specialization</i>	S4, S6, S8
<i>Ontology Summarization</i>	S1, ..., S9
<i>Ontology Translation</i>	S1
<i>Ontology Updating</i>	S1
<i>Ontology Upgrading</i>	S1
<i>Ontology Verification</i>	S1, ..., S9
<i>Ontology Versioning</i>	S1

several features (Taylor, 2008). Bearing in mind the analogy between software engineering and ontology engineering, it is not very realistic to propose a single life-cycle model for all ontology network developments. For this reason, Suárez-Figueroa and Gómez-Pérez (2008) proposed a preliminary collection of ontology network life-cycle models, where an ontology network life-cycle model describes how to organize an ontology network development project. The collection includes the following five models: the waterfall model, the incremental model, the iterative model, the evolutionary prototyping model and the spiral model. All the models were designed based on the life-cycle models described and used in software engineering. The specific features of ontology network development were taken into account, assuming a rather centralized and controlled ontology engineering setting, disregarding approaches such as mining ontologies from tags.

Dzbor et al. (2009) presented preliminary evaluations of the use of the above models. Such evaluations showed that ontology developers had some trouble distinguishing among incremental, iterative, evolving prototyping and spiral ontology network life-cycle models. Therefore, we decided to modify the preliminary collection of ontology network life-cycle models and commit to only two models: the waterfall model and the iterative-incremental model. This decision was based on three factors. First, the preliminary results of the evaluation showed that ontology developers found it hard to differentiate between the different iterative (incremental, iterative, evolutionary prototyping and spiral) life-cycle models. Second, Larman (2002) suggested two different models: waterfall and iterative-incremental life-cycle models in software engineering. Third, thanks to the experience that we gained by developing ontologies as part of different research projects (Esperonto, Knowledge Web, SEEMP, among others), we found that the main models used in practice were based on notions of waterfall and iterative modeling. The NeOn Methodology framework adopts these two models, which have been adapted to the specific features of ontology network development from software engineering models (Sommerville, 2007).

### 2.3.1. Waterfall life-cycle model family

In an ontology development project, this family of models represents the stages as sequential phases, where each stage must be completed before the next stage begins and backtracking is only permitted in

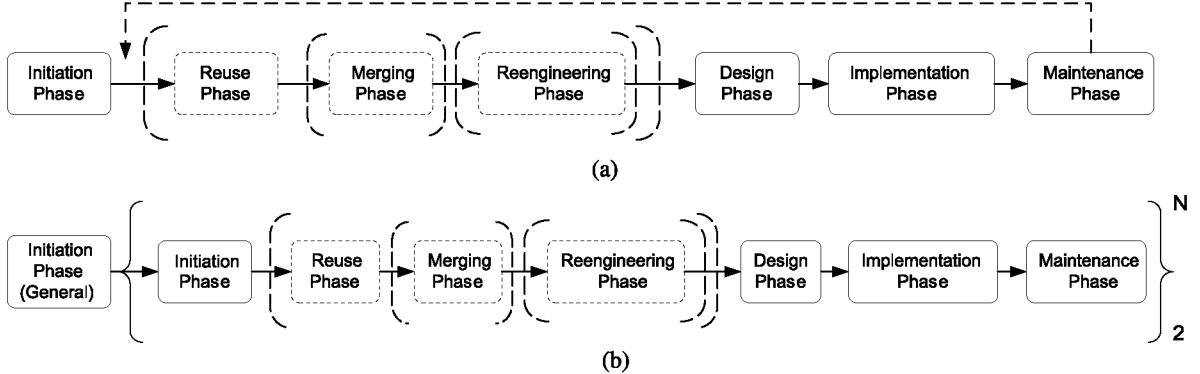


Fig. 3. Ontology network life-cycle models proposed in the NeOn Methodology framework. (a) Waterfall model family. (b) Iterative-incremental model.

the maintenance phase to fix errors or include missing knowledge. This model includes a set of support activities (e.g., knowledge acquisition, content evaluation) that must be performed in all phases.

This model is applicable in the following situations:

- Short ontology projects (e.g., 2 months);
- Ontology projects whose goal is to develop an existing ontology in a different formalism or language;
- Ontology projects with closed requirements, e.g., for implementing an ontology based on an ISO standard or previously agreed-upon resources;
- Ontology projects where ontologies are to cover a small well-understood domain.

Due to the importance of reusing and reengineering knowledge resources and merging ontological resources, we have defined five different versions of the waterfall model as follows: the four-phase waterfall model, the five-phase waterfall model, the five-phase + merging phase waterfall model, the six-phase waterfall model and the six-phase + merging phase waterfall model. Figure 3(a) shows a diagram of these versions. The diagram uses a graphical variation of BNF notation. In Fig. 3(a), the rounded boxes with a solid black border represent the phases of the basic version of the model; the rounded boxes with a dotted black border represent optional phases depending on the model version; the square brackets represent model version-dependent composition options; and, finally, the dashed directed line denotes optional backtracking (in the maintenance phase) in order to fix errors or add missing knowledge.

The five versions are intrinsically related to the scenarios described in Section 2.2, as shown in Table 2.

### 2.3.2. Iterative-incremental ontology network life-cycle model

This model, shown in Fig. 3(b), organizes the ontology development as a set of iterations (or short mini-projects with a fixed duration). The braces in Fig. 3(b) denote repetition, and the numbers (2 and N) indicate the number of iterations performed. Each iteration is scheduled as a single ontology project developed according to any of the versions of the waterfall model presented in Section 2.3.1. In other words, the iterative-incremental model is basically composed of a set of iterations developed according to any version of the waterfall ontology network life-cycle model. Thus, the relationship between scenarios and the iterative-incremental model depends on the different versions of the waterfall model used in the iterative-incremental model. Therefore, the relationships presented in Table 2 are equally valid for this model.

This model is applicable in the following scenarios:

Table 2  
Relationship between scenarios and waterfall life-cycle model versions

	Four-phase model	Five-phase model	Five-phase + merging phase model	Six-phase model	Six-phase + merging phase model
Scenario 1	×				
Scenario 2				×	
Scenario 3		×			
Scenario 4				×	
Scenario 5			×		
Scenario 6					×
Scenario 7		×			
Scenario 8	×				
Scenario 9	×				

- Ontology projects involving large groups of developers with different profiles and roles;
- Ontology projects in which development involves several different poorly understood domains;
- Ontology projects where requirements are partially unknown or can change during ontology development.

The requirements specified in the ORSD can be divided into different subsets, and each subset can be covered in different iterations. The result of any iteration is a functional and partial ontology that satisfies a subset of the requirements. A partial ontology can be used, evaluated and integrated into any other ontology network.

The model is based on adding to and successively improving the ontology network by performing multiple iterations with cyclic feedback and adaptation.

#### 2.4. Prescriptive methodological guidelines for processes and activities

The set of prescriptive methodological guidelines provides information on how to perform a particular process or a particular activity defined in the NeOn glossary. The guidelines include guiding cards, a workflow, and some examples explaining how to use the guidelines.

The *guiding card* is a user-friendly element (similar to a fact sheet) that provides brief, practical, easy-to-understand information on each process and activity of the NeOn Methodology. These cards are reminiscent of the so-called ‘flashcards’, cards with pictures and/or text normally used by teachers in the classroom. Our fact sheets include the following fields: (a) definition, which is taken from the NeOn glossary; (b) goal, which explains the main objective that the process or activity is intended to achieve; (c) input, which includes the resources needed to carry out the process or the activity; (d) output, which includes the results output after performing the process or the activity; (e) performer (who), which identifies the people or teams involved in the process or the activity; and (f) time (when), which explains when the process or the activity should be carried out. Table 3 shows the guiding card for the ontology requirements specification activity.

The graphical *workflow* shows how the process or the activity should be performed. The workflow contains the inputs, outputs, actors and details (methods, techniques and tools) prescribing how to perform the process or activity. Figure 4 illustrates the workflow for the ontology requirements specification activity.

The NeOn Methodology framework includes detailed methodological guidelines for performing both the different processes and activities involved in Scenarios 1, 2, 3, 5, 7, 8 and 9 and the support activities

Table 3  
Guiding card for ontology requirements specification

<b>Ontology Requirements Specification</b>	
<b>Definition</b>	
<p><i>Ontology Requirements Specification</i> refers to the activity of gathering the requirements to be met by the ontology (e.g., reasons for building the ontology, and identification of target groups and intended uses). Such requirements may be reached through a consensus-building process.</p>	
<b>Goal</b>	
<p>State the purpose of the ontology, its intended uses, its end users, and the requirements to be met by the ontology.</p>	
<b>Input</b>	<b>Output</b>
A set of ontological needs	Ontology Requirements Specification Document (ORSD)
<b>Who</b>	
<p>Members of the ontology development team (ODT), i.e., software developers and ontology practitioners, in collaboration with users and domain experts</p>	
<b>When</b>	
<p>At the beginning of the ontology project, in parallel with the knowledge acquisition activity</p>	

such as ontology evaluation and evolution.<sup>26</sup> The processes and activities are shown in Table 4. We have included a summary of the methodological guidelines for 13 processes and activities in the Appendix.

### 3. How to use the NeOn Methodology framework

We now describe how to apply the NeOn Methodology framework to a real use case for building a user-context ontology network (usercONText). This ontology network was developed as part of the mIO!<sup>27</sup> research project and is used to provide context-aware services for both mobile environments, such as social event recommenders in a city (Melero et al., 2010), and for contextualized social life providers (Penela & Ruiz, 2010).

To develop usercONText, the ontology development team (composed of people with a range of skills from the different project partner institutions) first performed the ontology requirements specification activity. This activity was carried out according to the guidelines shown in Fig. 4. As a result, the team output the ORSD, which contains more than 300 functional requirements. The requirements include the following terms: device, camera, interface, sensor, environment, user, source, localization, building, provider, service, network, access, interval, time, preference and role, among others.

<sup>26</sup>During this activity, developers can explicitly include meta-relationships such as *hasPriorVersion* to indicate that a new version of an ontology is being developed. In doing so, developers are building an ontology network.

<sup>27</sup><http://www.cenitmio.es/>.

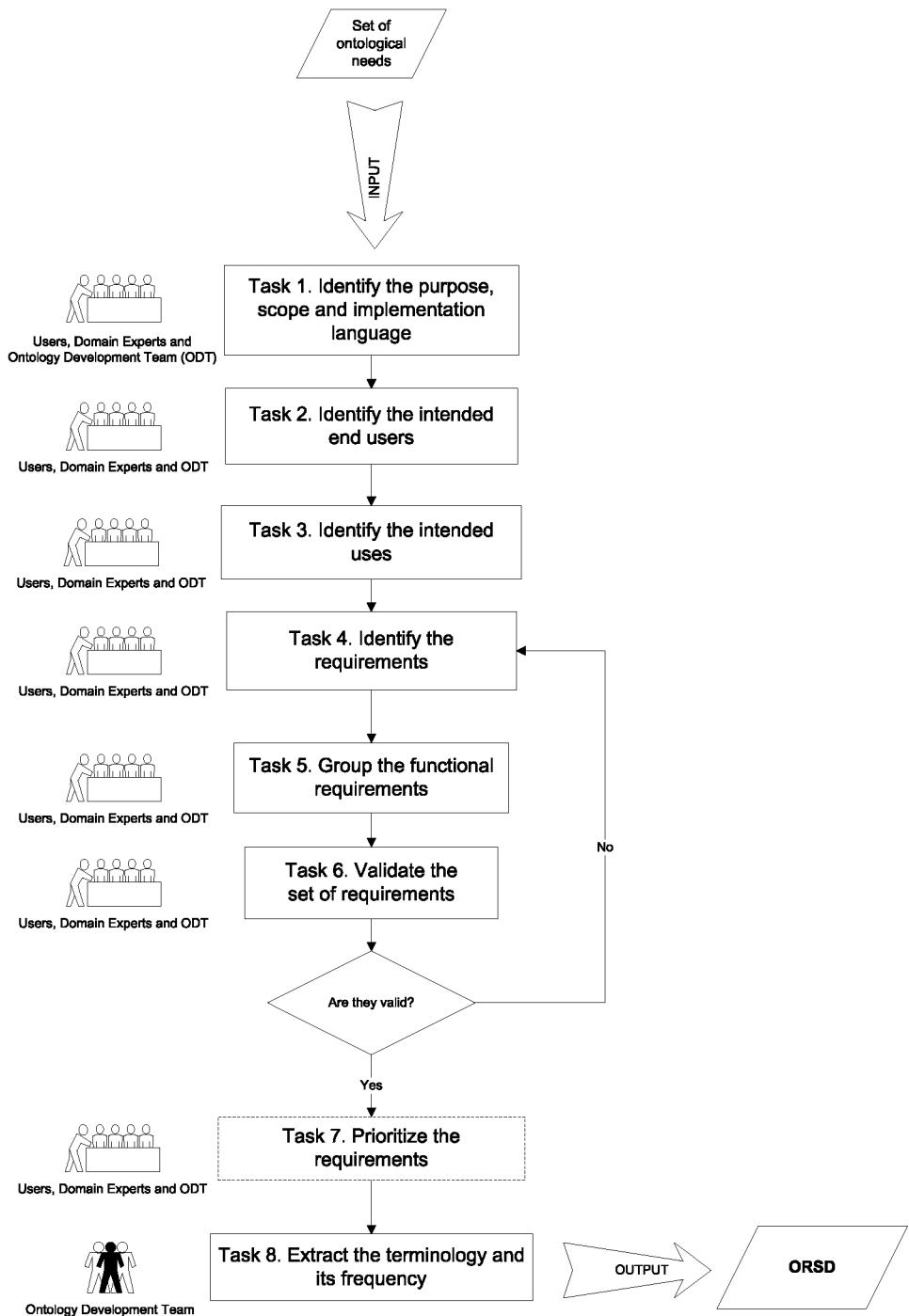


Fig. 4. Workflow for ontology requirements specification.

Table 4  
Processes and activities with methodological guidelines

Scenarios	Methodological guidelines for processes and activities
Scenario 1: From specification to implementation	<ul style="list-style-type: none"> <li>• Ontology Requirements Specification</li> <li>• Scheduling</li> </ul>
Scenario 2: Reusing and reengineering non-ontological resources (NORs)	<ul style="list-style-type: none"> <li>• NOR Reuse</li> <li>• NOR Reengineering</li> </ul>
Scenario 3: Reusing ontological resources	<ul style="list-style-type: none"> <li>• Domain Ontology Reuse</li> <li>• General Ontology Reuse</li> <li>• Ontology Statements Reuse</li> <li>• Ontology Module Reuse</li> </ul>
Scenario 4: Reusing and reengineering ontological resources	–
Scenario 5: Reusing and merging ontological resources	<ul style="list-style-type: none"> <li>• Ontology Alignment</li> <li>• Ontology Merging</li> </ul>
Scenario 6: Reusing, merging and reengineering ontological resources	–
Scenario 7: Reusing ontology design patterns (ODPs)	<ul style="list-style-type: none"> <li>• Ontology Design Patterns Reuse</li> </ul>
Scenario 8: Restructuring ontological resources	<ul style="list-style-type: none"> <li>• Ontology Modularization</li> </ul>
Scenario 9: Localizing ontological resources	<ul style="list-style-type: none"> <li>• Ontology Localization</li> </ul>
<i>Methodological guidelines for cross-scenario processes and activities</i>	
<i>Guidelines for Ontology Evaluation</i>	
<i>Guidelines for Ontology Evolution</i>	

The ontology development team used the glossary of terms included in the ORSD to run an exploratory search for available knowledge resources that could be reused to develop usercONText. As a result, the team collected a variety of ontologies (i.e., *OWL-Time*,<sup>28</sup> *CoDAMoS*<sup>29</sup> and *DeliveryContext*<sup>30</sup>), ODPs (such as *componency*, *part-whole*, *n-ary relation*, *n-ary participation*) and NORs (such as *ISO 3166*<sup>31</sup>). The exploratory search provided clues for identifying the scenarios applicable during the development of usercONText.

After the search, the team used gOntt to schedule the ontology development project (Suárez-Figueroa, 2010; Suárez-Figueroa et al., 2012). gOntt is a tool mainly used for scheduling ontology development projects. It automatically generates the initial ontology development plan in the form of a Gantt chart. In addition, it also provides ontology developers with support during the execution of the ontology development project. gOntt is founded on the methodological guidelines for scheduling projects shown in Fig. 5. These guidelines include the following tasks:

- (1) Select the ontology network life-cycle model. To help the team decide which the most appropriate life-cycle model is, gOntt asks a simple natural language question, “Are the ontology requirements assumed to be fully known at the beginning of the ontology network development?”. In the particular use case presented here, the requirements were expected to change throughout the project, and, consequently, the team selected an iterative-incremental model with three iterations.
- (2) Select the set of scenarios. To help the team with this task, gOntt asks a set of simple natural language questions, “Do you intend to use any ontological resource in your ontology network

<sup>28</sup><http://www.w3.org/TR/owl-time/>.

<sup>29</sup><http://distrinet.cs.kuleuven.be/projects/CoDAMoS/ontology/>.

<sup>30</sup><http://www.w3.org/TR/deontology/>.

<sup>31</sup>[http://www.iso.org/iso/country\\_codes](http://www.iso.org/iso/country_codes).

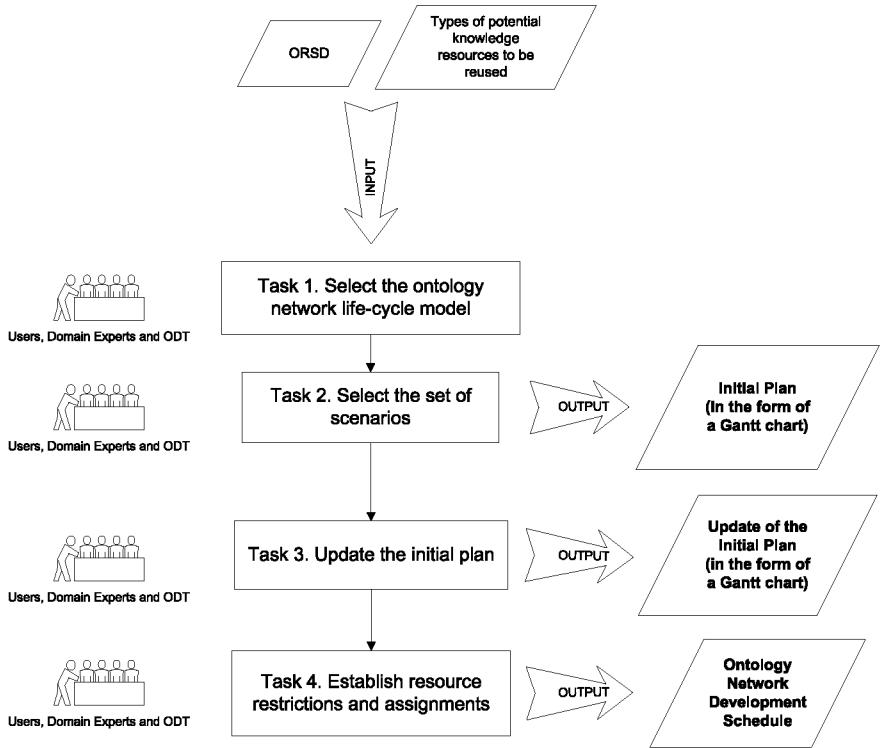


Fig. 5. Workflow for scheduling.

development?" and "Do you intend to develop your ontology network in different ontology languages?". gOntt uses the responses to these questions to output the initial ontology network life cycle, that is, an initial plan for ontology network development, based on the methodological foundations presented by Suárez-Figueroa (2010). In the particular case presented here, this set of questions has to be answered once per iteration, that is, three times. The selected scenarios were as follows:

- In the first iteration, the team applied Scenarios 3 and 7. They reused the ontologies (Scenario 3) in two different ways: (a) by reusing whole ontologies (e.g., SOUPA, RECO, the OWL Time ontology, the FOAF ontology) and (b) by reusing ontology modules from the CoDAMoS and DeliveryContext ontologies. Additionally, they reused the modular and the n-ary relationship patterns as part of Scenario 7.
- In the second iteration, the team applied Scenarios 2, 7 and 8. They reused and reengineered the ISO 3166 NOR into an ontology about countries as part of Scenario 2. Then, as part of Scenario 7, they reused the compency pattern to denote that objects either are proper parts of other objects or have proper parts. Finally, as part of Scenario 8, they restructured the different ontological resources from the previous and the current iterations.
- In the third iteration, the team deployed Scenarios 7, 8 and 9. They reused the price pattern to enrich services with their prices as part of Scenario 7. They also restructured the ontology network as part of Scenario 8. Finally, they transformed all the ontology labels into Spanish as part of Scenario 9.

- (3) Update the initial plan. The goal of this task is to modify (if necessary) the initial plan presented by gOntt. In the case presented here, the ontology development team changed the order of the processes and activities as follows: in the first iteration, the plan was to perform the ontological resource reuse processes and the ODP reuse process in parallel; in the second iteration, the plan was to perform the ODP reuse process in parallel with Scenario 2 processes.
- (4) Establish resource restrictions and assignments. In this case, information about both time scheduling (updating the iteration duration and the time automatically assigned to the planned processes and activities) and human resource assignments were included in the ontology network development plan.

The above three iterations and their respective scenarios were developed according to the methodological guidelines included in the NeOn Methodology framework. As a result, the team output usercONText.<sup>32</sup> This ontology is composed of 13 modular ontologies representing the following sub-domains: device, environment, interface, location, network, provider, role, service, source, time, units of measurement, and user.

Figure 6 shows how the NeOn Methodology was applied to this particular use case.

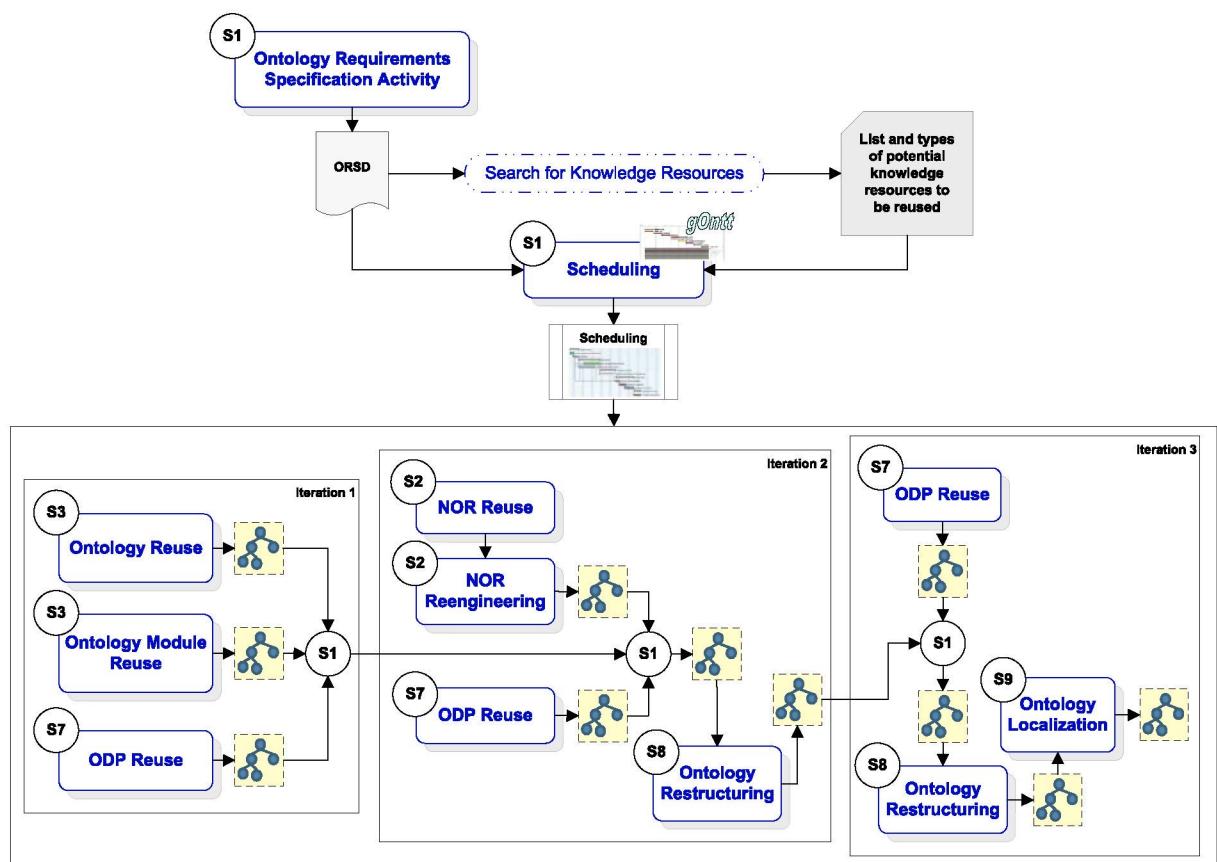


Fig. 6. NeOn Methodology framework application. (Colors are visible in the online version of the article; <http://dx.doi.org/10.3233/AO-150145>.)

<sup>32</sup><http://mayor2.dia.fi.upm.es/oeg-upm/index.php/en/ontologies/82-mio-ontologies>.

## 4. Evaluation of the NeOn Methodology framework: Use cases and developer experiences

As de Hoog (1998) said, “It is extremely difficult to judge the value of a methodology in an objective way. Experimentation is, of course, the proper way to do it, but it is hardly feasible because there are too many conditions that cannot be controlled”. On the one hand, the introduction of an experimental toy problem would violate the basic assumption behind the need for a methodology: the development process is complex. On the other hand, if we apply the argument provided by de Hoog for knowledge-based systems to the ontology engineering field, we find it is unlikely that anyone will be willing to pay twice for building the same complex ontology using different approaches.

In this section, we report how we evaluated the NeOn Methodology framework. Then we provide a summary of different real use cases in which the methodological framework has been applied to develop ontology networks. Finally, we show the experience gained by different ontology developers using the methodology.

### 4.1. The NeOn Methodology framework applied to different use cases

People with different backgrounds (e.g., ontology engineers, domain experts, final users and linguists, among others) have used the NeOn Methodology framework in different ontology development projects. These projects targeted domains as far apart as satellite data processing, seeking program funding, fishery stocks, e-employment and multimedia. Table 5 summarizes six use cases (one of which is presented in Section 3) in which the NeOn Methodology framework was applied. Table 5 includes information on the ontology and ontology development process. The information includes the following items:

- *Ontology name*: name of the ontology, URL (if the ontology is available) and references.
- *Description*: summary describing the knowledge modeled in the ontology. The information also includes the languages (ontological and natural languages) used in the development, and whether the ontology was part of any other project.
- *Applied NeOn scenarios*: list of the scenarios employed during ontology development, and a description explaining why such scenarios were needed.
- *Life-cycle model used*: model used in ontology development, stating, if the iterative-incremental model was used, the number of iterations.
- *Employed technological support*: the ontology editor(s) handled during the development, including information on whether gOntt was used to schedule the ontology development project.

As shown in Table 5, the NeOn Methodology framework is tool-independent since it has been used in a variety of development environments (NeOn Toolkit, Protégé, TopBraid Composer, etc.).

### 4.2. Developers’ experiences using the NeOn Methodology framework

One way of evaluating a methodology is to use surveys (Kitchenham, 1996; Palvia & Nosek, 1990) in which the users who have applied the methodology in past projects are asked to provide information about their experience.

In order to analyze developer experiences of using the NeOn Methodology framework as part of their own ontology network developments, we have designed a questionnaire to elicit their views on the methodology and draw some conclusions regarding the following general issues:

- Did the NeOn Methodology work and/or was it useful and applicable?
- Did the NeOn Methodology help ontology network developers to do their job?
- Are some methodological components more helpful than others within the NeOn Methodology?

Table 5  
Selection of use cases in which the NeOn Methodology framework was used

Ontology name	Description	Scenarios	Life cycle model	Technological support
usercONText, <sup>33</sup> a user-context ontology network (see Section 3 for more details)	This ontology, implemented in OWL DL, represents user-context-related knowledge, for example, information on location and time, user information and current or planned activities, as well as nearby devices and available services. This ontology network was developed as part of a research project called mIO!. <sup>34</sup> It is now being used to provide context-aware services for mobile environments such as social event recommenders in a city (Melero et al., 2010) and contextualized social life providers (Penela & Ruiz, 2010).	<ul style="list-style-type: none"> <li>• Scenario 2, where NORs (e.g., ISO 3166) are transformed into ontologies.</li> <li>• Scenario 3, where ontologies available in different sub-domains (e.g., OWL-Time and CoDAMoS) are reused.</li> <li>• Scenario 7, where ODPs (e.g., n-ary relations) are reused.</li> <li>• Scenario 8, where the different ontological resources are restructured.</li> <li>• Scenario 9, where the ontology can be translated into different natural languages.</li> </ul>	Iterative-incremental model with three iterations	NeOn Toolkit
SEEMP ontology network <sup>35</sup> (Villazón-Terrazas et al., 2011)	This ontology, developed as part of the SEEMP project, represents knowledge about CVs and employment service job offers located in different countries. This ontology network consists of the following elements: (1) a reference ontology implemented in WSML, the core component of the system, whose aim is to provide a consensual knowledge model of the employment domain for use by public employment services (PES), more specifically within the information and communication technology (ICT) domain; (2) a set of local ontologies implemented in WSML (each PES uses its own local ontology, which describes the employment market in its own terms); (3) a set of mappings between each local ontology and the reference ontology; and (4) a set of mappings between the PES schema sources and the local ontologies.	<ul style="list-style-type: none"> <li>• Scenario 2, where the local and the reference ontology networks using several NORs (international standards such as NACE and ISCO-88; ES classifications; and international codes such as ISO 3166) are built.</li> <li>• Scenario 4, where the DAML time ontology is reused and reengineered.</li> <li>• Scenario 5, where the alignments between each local ontology and the reference ontology are established.</li> </ul>	Iterative-incremental model with three iterations	WebODE

Table 5  
(Continued)

Ontology name	Description	Scenarios	Life cycle model	Technological support
Invoice ontology network <sup>36</sup> (Gómez-Pérez et al., 2012)	This ontology network, implemented in OWL within the scope of a pharmaceuticals case study within the NeOn project, <sup>37</sup> covers: (a) the most important standards of the electronic invoicing domain, EDIFACT <sup>38</sup> (Electronic Data Interchange for Administration, Commerce and Transport) and UBL <sup>39</sup> (Universal Business Language); (b) all the concepts needed for representing any invoice issued by a company; and (c) concepts for representing the invoicing workflow.	<ul style="list-style-type: none"> <li>• Scenario 2, where different NORs for electronic invoicing (e.g., UBL, EDIFACT) are reused and transformed into ontologies.</li> <li>• Scenario 6, where DOLCE Ultra Lite, the W3C Time ontology, TOVE and SUMO are reused, merged and reengineered.</li> <li>• Scenario 8, where the invoice ontology is specialized for each laboratory concerned.</li> <li>• Scenario 9, where the ontology is localized into Spanish.</li> </ul>	Iterative-incremental model	NeOn Toolkit
Semantic nomenclature ontology network <sup>40</sup> (Pariente-Lobo & Herrero-Cárcel, 2012)	This ontology network, implemented in OWL as part of the pharmaceuticals case study within the NeOn project, <sup>41</sup> represents products of the Spanish pharmaceuticals sector. It represents general aspects of the main terms, as well as objects related to drugs; it classifies these pharmaceutical terms according to the Anatomical Therapeutic Chemical (ATC) <sup>42</sup> classification.	<ul style="list-style-type: none"> <li>• Scenario 2, where different NORs (e.g., the Digitalis and Integra databases sourced from the Spanish Medicines and Healthcare Products Agency (AGEMED)<sup>43</sup> and the publicly available part of the Spanish General Council of Pharmacists' Association's BOTPlus<sup>44</sup> commercial database) are reused and reengineered into ontologies.</li> <li>• Scenario 3, where both the W3C time ontology and some modules of the Galen ontology are reused.</li> <li>• Scenario 6, where the ontologies output by Scenarios 2 and 3 are reused and reengineered.</li> <li>• Scenario 7, where different ontology design patterns (e.g., the part-of pattern) are reused.</li> <li>• Scenario 8, where both the clinical drugs ontology and the corresponding relations are enriched.</li> <li>• Scenario 9, where the ontology is localized into Spanish.</li> </ul>	Iterative-incremental model with two iterations	NeOn Toolkit and gOntt

Table 5  
(Continued)

Ontology name	Description	Scenarios	Life cycle model	Technological support
Digital Multimedia Repositories Ontology (DMRO) <sup>45</sup> (Lawrynowicz & Palma, 2012)	<p>DMRO describes multimedia artifacts in digital multimedia repositories in terms of their usage, reviews and content type, as well as their relationship with related events and agents. This background knowledge is intended for use in semantic data mining tasks. Therefore, unlike the other ontologies available for describing multimedia artifacts that focus mainly on a detailed description of their characteristics, which falls outside DMRO's application requirements, DMRO focuses on the resource usage information, which is key for data mining in multimedia repositories.</p> <p>DMRO consists of a set of six small, interrelated modules that represent DMRO concepts such as multimedia resources, users, events, places, reviews, Web usage mining-related concepts and domain topics (in our case Videolectures.net topic category).</p> <p>DMRO's ontological language is OWL 2 and its natural language is English.</p> <p>DMRO was part of the EU FP7 e-LICO project.<sup>46</sup></p>	<ul style="list-style-type: none"> <li>• Scenario 2, where non-ontological resources in the form of the dataset are reused. The dataset is a snapshot from the Videolectures.net website.</li> <li>• Scenario 3, where relevant ontological resources, like DOLCE Ultralite (DUL) are reused as an upper ontology. In addition, individual ontology statements are also reused from ontologies such as: (i) Dublin Core for modeling general metadata properties of resources; (ii) FOAF for modeling users, authors, participants, their personal data and social aspects; (iii) SWRC ontology for events; (iv) OBO Relation Ontology for standard relations; (v) RDF Review Vocabulary for reviews, comments and feedback; (vi) Viewings &amp; Downloads modules from the myExperiment Ontology for modeling the resource usage; and (vii) Annotations module from myExperiment Ontology for modeling tags.</li> <li>• Scenario 5, where ontological resources are reused and merged.</li> <li>• Scenario 7, where ontology design patterns (ODPs) are reused in order to follow best practices in ontology development and to resolve modeling issues.</li> <li>• Scenario 8, where ontological resources are modularized and restructured.</li> </ul>	Iterative-incremental model with two iterations	NeOn Toolkit and Protégé. The ontology development was scheduled using gOntt

Table 5  
(Continued)

Ontology name	Description	Scenarios	Life cycle model	Technological support
Ontología de Modelado del Estudiante (Clemente, 2011; Clemente et al., 2011)	<p>The <i>Ontología de Modelado del Estudiante</i> (Student Modeling Ontology), developed in OWL, represents the various types of student knowledge considered necessary for proper student modeling in an intelligent tutoring system (ITS) to provide for adaptive learning depending on individual student characteristics. This ontology was designed for use in teaching methods, e.g., in learning activities related to a graphical user interface or to virtual learning environments for procedural training.</p> <p>The ontology was developed as part of the ENVIRA (<i>Entornos Virtuales Inteligentes para el Aprendizaje – Intelligent Virtual Learning Environments</i>) project. It is intended for use in the SEPIA (<i>Modelos Semánticos del Entorno y la Persona para una Interacción Adaptativa en entornos virtuales-Semantic Environment and Person Models for Adaptive Interaction in Virtual Environments</i>) project.</p>	<ul style="list-style-type: none"> <li>• Scenario 2, where non-ontological resources are reused in order to include learning environment knowledge in the ontology.</li> <li>• Scenario 8, where the ontology is modularized.</li> </ul>	Iterative-incremental model with four iterations	Protégé-OWL Editor and TopBraid Composer

<sup>33</sup><http://mayor2.dia.fi.upm.es/oeg-upm/index.php/en/ontologies/82-mio-ontologies>.

<sup>34</sup><http://www.cenitmio.es/>.

<sup>35</sup><http://mayor2.dia.fi.upm.es/oeg-upm/index.php/en/ontologies/99-hrmontology>.

<sup>36</sup><http://www.neon-project.org/nw/Ontologies>.

<sup>37</sup>[http://www.neon-project.org/nw/Welcome\\_to\\_the\\_NeOn\\_Project](http://www.neon-project.org/nw/Welcome_to_the_NeOn_Project).

<sup>38</sup><http://www.unece.org/trade/untidid/welcome.htm>.

<sup>39</sup><http://www.oasis-open.org/committees/UBL>.

<sup>40</sup><http://www.neon-project.org/nw/Ontologies>.

<sup>41</sup>[http://www.neon-project.org/nw/Welcome\\_to\\_the\\_NeOn\\_Project](http://www.neon-project.org/nw/Welcome_to_the_NeOn_Project).

<sup>42</sup>[http://www.whocc.no/atc\\_ddd\\_index/](http://www.whocc.no/atc_ddd_index/).

<sup>43</sup><http://www.aemps.es/>.

<sup>44</sup><https://botplusweb.portalfarma.com/>.

<sup>45</sup><http://129.194.69.119/?q=node/288>. The main ontology file is available at <http://129.194.69.119/public/dmro/DMRO.owl>.

<sup>46</sup><http://129.194.69.119/?q=node/4>.

- Are ontology network developers satisfied with the NeOn Methodology?
- Would the ontology network developers use the NeOn Methodology again and recommend it to other developers?

The questionnaire is composed of two different sections: (1) one related to background knowledge, and (2) another related to the experiences gained by using the NeOn Methodology framework. The first part includes four questions (two open questions and two multiple-choice questions). The second part is composed of 14 questions: five multiple-choice questions and nine open questions asking survey participants about their opinions and comments.

The questionnaire was administered to people participating in different international research projects as part of which ontologies had to be developed. Our questionnaire is available at <http://bit.ly/SLyRZO>. So far, we have received 26 responses from colleagues involved in ontology development projects from all over the world and with different backgrounds.

Regarding background knowledge, it is noteworthy that 65% of survey respondents are specialized in ontology engineering, whereas 35% have some knowledge of the field, although they cannot be considered ontology experts. With respect to the number of ontologies developed, 54% of respondents have built more than five ontologies.

Moving on to the core part of the questionnaire, particularly the responses to the multiple-choice questions, we have found that

- Most survey respondents considered that the NeOn Methodology is useful and well explained, as shown in Fig. 7(a) and (b).
- Ninety six percent of respondents considered that the NeOn Methodology was helpful during ontology development, whereas 4% did not consider the methodology to be of any help whatsoever (see Fig. 7(c)).
- Ninety two percent of the respondents said that they would use the NeOn Methodology again in future ontology developments, and 62% of the respondents would recommend the methodology to other colleagues (see Fig. 7(d) and (e)).

We can characterize the whole population of NeOn Methodology users by taking a random experimental sample of 26 people, the same number of people who responded to our questionnaire, and asking each person the question: (a) “Did you find the NeOn methodology useful?”. Let  $X_a$  be the random variable representing the result of this experiment whose value is *yes* or *no*. This variable fits a binomial model with size 26 and probability  $p$ . Now, let us provisionally assume the hypothesis  $H_0: p = 0.5$ , that is, if we take one person at random, the probability that (s)he will answer *yes* to our question is 0.5. Using R statistical software (R Core Team, 2014), we have calculated  $P(X_a \geq 26)$  in  $B(26, 0.5)$ , the result of which was approximately  $10^{-6}$ . This value is so small that we have to reject our initial hypothesis and assume that most of the people who have used NeOn Methodology would answer *yes* to our question.

If instead of assuming the hypothesis of  $p = 0.5$ , we assume  $p = 0.77$ , then  $P(X_a \geq 26) = 0.04225047$ . Therefore, we can also reject this new hypothesis. In other words,  $p$  has to be greater than 0.77. Then we can claim that, at least, 77% of the people who have used NeOn Methodology would answer *yes* to our question.<sup>47</sup>

Applying a similar reasoning to the other questions, the result is the same for question (b) (where again there were 26 positive answers). In the case of question (c), if we consider together *yes*, *always*,

---

<sup>47</sup>If  $p = 0.78$ , then  $P(X \geq 26) = 0.05349754 > 0.05$ , which is the generally applicable significance level (Box et al., 2005, p. 95). For the other cases, we will omit, for simplicity's sake, the limit value of  $p$  when  $P(X > A) > 0.05$ , where  $A$  is the number of positive responses. This value is rounded to two decimal places, such that  $P(X > A) \leq 0.05$ .

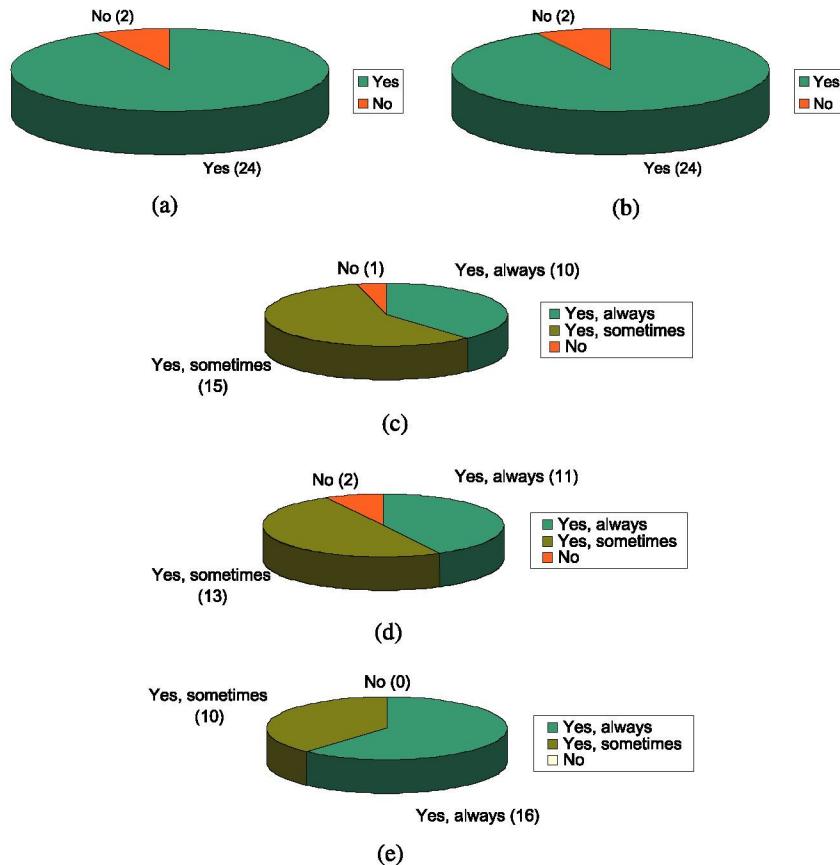


Fig. 7. Results for the multiple-choice questions. (a) Distribution of the responses to the question, “Did you find the NeOn Methodology useful?”. (b) Distribution of the responses to the question, “Do you think the NeOn Methodology is well explained?”. (c) Distribution of the responses to the question, “In general, do you think the NeOn Methodology has helped you during the development?”. (d) Distribution of the responses to the question, “In your future ontology network developments, will you use the NeOn Methodology again?”. (e) Distribution of the responses to the question, “Would you recommend the NeOn Methodology to other colleagues involved in ontology development projects?”. (Colors are visible in the online version of the article; <http://dx.doi.org/10.3233/AO-150145>.)

and *yes, sometimes*, then  $p > 0.83$ . As for (d), the result is  $p > 0.77$ , and the result for (e) is  $p > 0.89$ .

Regarding the results for the open questions, it is noteworthy that:

- According to the survey respondents, the advantages of the NeOn Methodology over other existing methodologies, such as METHONTOLOGY, are that: (a) this methodology is easy to understand in different scenarios, partly thanks to the guiding cards for each process and activity; (b) it is based on a scenario approach; and (c) it includes detailed and helpful documentation.
- On the other hand, the disadvantages are that: (a) the methodology does not explicitly state all the steps to be performed; and (b) the methodology is time-consuming to apply. However, this methodology is generally regarded as being more robust and helpful than the others are. In this respect, one of the respondents commented literally: *“I like that NeOn is oriented to collaborative development of network of ontologies and gives rather good guidance for choices. I consider METHONTOLOGY too complex and hard to apply to real ontology development process” [sic]*.

The respondents replied that the most useful NeOn Methodology framework components were the methodological guidelines and the set of scenarios. Thus, some of the comments were as follows: “*The guidelines are useful to not forget any important step*” [sic]; “*I also found useful the guiding cards, which were short*” [sic]; “*Scenarios are a really good starting point, especially at the beginning*”; “*Scenarios consider different cases and provide guidelines for each one*”; and “*Detecting where you are, what to do, and how to proceed according to the different information sources available is very important in ontology development projects*”.

Additionally, we should add that five respondents considered the guidelines for the ontology requirements specification activity to be a very useful and beneficial component because: (a) they helped them to gather the necessary information and (b) the ORSD template and the different tasks made it easier to gather ontology requirements.

It is worth mentioning that both the glossary and the set of life-cycle models were considered useful by approximately 50% of the respondents and useless by the other 50%.

Figure 8 summarizes the responses with regard to the usefulness of the methodological components.

We will proceed as above in order to characterize the whole population who used the NeOn Methodology, albeit now focusing on the usefulness of the framework components. For each of the three components, the random variable that represents the number of positive responses follows a binomial. Applying the same reasoning as above, we can conclude that at least 73% of the population would answer yes to the question of whether the methodological guidelines are useful. We cannot reject the hypothesis  $p = 0.5$  for either life-cycle model and glossary usefulness. Consequently, our controlled experiment does not provide any convincing evidence regarding the usefulness of these components. Concerning scenario usefulness,  $p > 0.77$ , that is, more than 77% of the population would answer yes to the question of whether the scenarios are useful.

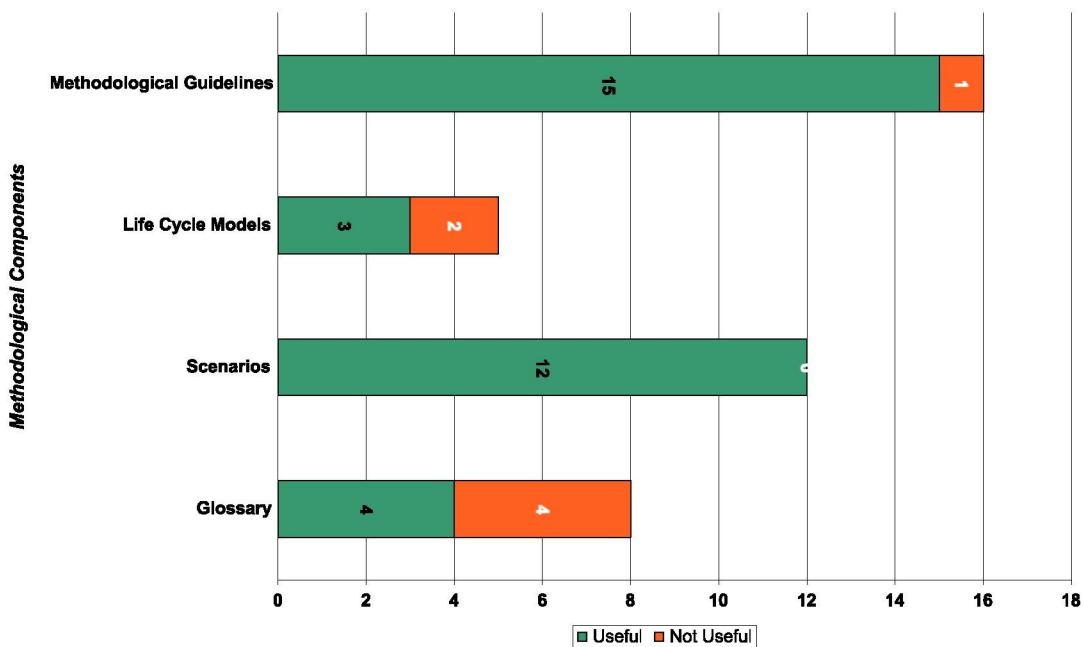


Fig. 8. Statistics on NeOn Methodology framework component usefulness. (Colors are visible in the online version of the article; <http://dx.doi.org/10.3233/AO-150145>.)

Despite the findings regarding the glossary and life-cycle models, our experience shows that both components are necessary. As we see it, the most plausible explanation for this mismatch is that methodology users already have an implicit conceptualization of these components. Therefore, we think that an interesting line of research would be to study the dependence of component usefulness on ontology engineer experience and skill.

Finally, the respondents provided some suggestions that may help improve the NeOn Methodology framework, such as:

- Making the methodology more lightweight and agile because “*like RUP for software engineering NeOn Methodology is good but heavy*”;
- Preparing a light and adapted version of the methodology for linked data practitioners;
- Building NeOn Methodology framework specializations in specific domains (such as BioNeOn, EduNeOn, MedNeOn, etc.);
- Providing more realistic examples of how to carry out the different processes and activities;
- Improving the ‘human factor’, that is, the involvement of domain experts in the knowledge acquisition process;
- Translating the methodology into different natural languages;
- Including the linked data dimension in the set of scenarios;
- Providing more guidelines for addressing the problem of dividing ontology development between several teams, where each team develops one or more parts of the ontology that then have to be integrated;
- Producing specific guidelines for ontology conceptualization that include ontology-building criteria similar to those presented by Rector et al. (2004);
- Enhancing the guidelines and criteria for reusing ontologies in order to reduce the time taken to complete the required information;
- Finally, developing Protégé plug-ins to support the methodology in order to improve the technological dimension.

## 5. Conclusions and future work

There has been a growing trend in recent years towards the reuse of knowledge resources in ontology development in the context of different development initiatives. Thus, the NeOn Methodology framework proposes the reuse and reengineering of available ontologies, ODPs and NORs, all agreed upon by consensus, to construct modular ontologies interconnected as networks. The NeOn Methodology framework is based on four pillars: (1) a glossary of processes and activities; (2) a set of nine flexible ontology-building scenarios; (3) two ontology life-cycle models that organize ontology development; and (4) a set of methodological guidelines for performing specific processes and activities. This framework is a methodology that defines each process or activity in a precise manner by clearly stating its purpose, inputs and outputs, actors involved, execution time, and the proposed set of methods, techniques and tools for use.

The NeOn Methodology framework is a step forward in the evolution of the ontology engineering field, which requires practical and precise methodologies. This methodology solves the problems of: (a) the need for a methodology targeting ontology practitioners (ontology engineers, domain experts, final users, linguists, etc.) rather than just ontology researchers; and (b) the need for a methodology explaining the ontology-building process in the style and with the granularity of software development

methodologies. The NeOn Methodology is presented prescriptively by means of simple descriptions and diagrams so that it is easily understandable for people with different backgrounds (ontology engineers, domain experts, final users, linguists, etc.). The NeOn Methodology adopts a software engineering approach with different levels of complexity so that it can be easily taken on board. It is adaptable to specific needs and users; it can also adopt new processes or activities involved in ontology network development.

The NeOn Methodology framework has already been used flexibly in ontology development projects within a diversity of domains and areas and within the context of different European and national projects (e.g., e-employment, invoices, pharmaceutical products, user-context in tourism (Lamsfus et al., 2009), education (Clemente et al., 2011)). In addition, as already reported in the paper, people with different profiles and backgrounds have applied the methodology successfully.

We have demonstrated that the NeOn Methodology framework is independent of technology since it has been used with several development environments (NeOn Toolkit, Protégé, TopBraid Composer, etc.). The methodology is closely related to the NeOn Toolkit and its plug-ins, but it can also be used with other tools such as Protégé, TopBraid Composer and others.

According to the responses to our questionnaire on the NeOn Methodology framework, we can state that the majority of the respondents consider the methodology to be useful, helpful and well explained. Most said that they would apply it again in future ontology developments, and 62% of respondents added that they would recommend the NeOn Methodology to other colleagues. Generally, the methodology is perceived as being more robust and helpful than others, like METHONTOLOGY, for example. By contrast, the main weaknesses of the methodology are: (a) it does not explicitly state the whole sequence of steps to be performed, although this sequence is generated during the scheduling activity that can be performed using gOntt; and (b) methodology application is time consuming.

Regarding the usefulness of the components included in the NeOn Methodology framework, the most useful are the methodological guidelines and the set of scenarios; on the other hand, the glossary and the set of life-cycle models were considered useful and useless by almost the same number of respondents.

Finally, we would like to mention that we are currently working on a more lightweight and agile version of the NeOn Methodology, as well as on the inclusion of the linked data dimension in the methodology. We also plan to: (a) provide specific guidelines for ontology modeling; (b) enhance the guidelines and criteria for reusing ontologies in order to reduce the time taken to complete the required information; (c) provide technological support for ontology reuse; and (d) provide some guidelines for documenting ontologies that include the different knowledge resources that were reused.

## Acknowledgements

This work was supported by the European NeOn project (FP6-027595) and the Spanish mIO! project (*Tecnologías para prestar servicios en movilidad en el futuro universo inteligente – Technologies for mobility service provision in the future intelligent universe, CENIT-2008-1019*) and is partially funded by the CDTI (*Centro para el Desarrollo Tecnológico Industrial – Industrial Technology Development Center*) as part of the INGENIO 2010 Program. We would like to thank María Poveda-Villalón for her work on the development of the user-context ontology network. We are very grateful to Rosario Plaza for her revisions and comments. We would also like to thank all the people who responded to the questionnaire, as well as the developers and practitioners (Boris Villazón-Terrazas, Agnieszka Lawrynowicz, Raúl Palma, Julia Clemente, Carlos Bobed, Germán Herrero-Carcel and Jose Manuel Gómez-Pérez) who provided us with detailed information about their ontology developments.

## Appendix. Excerpt of methodological guidelines

Scenarios		Excerpt of methodological guidelines for processes and activities
Scenario 1: From specification to implementation	<p><i>Ontology Requirements Specification activity</i> (Suárez-Figueroa, 2010; Suárez-Figueroa et al., 2012). Its goal is to explain why the ontology is being built, which its intended uses are, who the end-users are, and what requirements the ontology should fulfill. The result of this activity is the Ontology Requirements Specification Document (ORSD) that should be written following the ORSD template.</p>	<p><i>Task 1. Identify the purpose, scope and implementation language.</i> The objective is to determine the main goal of the ontology, its coverage and foreseeable granularity, and its implementation language. A set of interviews with users and domain experts are conducted, taking as input a set of ontological needs.</p> <p><i>Task 2. Identify the intended end users.</i> Its goal is to establish who the intended main end-users of the ontology will be. A set of interviews with users and domain experts are conducted, taking as input a set of ontological needs.</p> <p><i>Task 3. Identify the intended uses.</i> Its goal is to obtain the intended uses and use scenarios of the ontology. A set of interviews with users and domain experts are performed, taking as input a set of ontological needs. The scenarios should be described in natural language; they can be expressed as use cases in UML.</p> <p><i>Task 4. Identify requirements.</i> Its goal is to acquire the set of requirements (functional and non-functional) that the ontology should satisfy. It is recommended to use the competency question (CQ) technique (Grüniger &amp; Fox, 1995) to specify functional ontology requirements. This initial set of ontology requirements is obtained by means of a set of interviews with users and domain experts. Other knowledge elicitation techniques can be also applied for the creation of CQs, such as 20-questions, triad analysis, card sort (Rao et al., 2008), and the use of a goal modelling approach (Barbosa et al., 2011), in particular a methodology called Tropos. Mind map tools and Excel can be used for gathering the requirements.</p> <p><i>Task 5. Group functional requirements.</i> Its goal is to group into several categories the list of functional ontology requirements in the form of CQs and their associated answers obtained in Task 4. Usually this task is carried out in parallel to Task 4.</p> <p><i>Task 6. Validate the set of requirements.</i> The aim here is to identify possible conflicts between ontology requirements (functional and non-functional), missing ontology requirements and contradictions between them. The criteria that can be used in this validation task are the following: correctness, completeness, consistency, verifiability, understandability, no ambiguity, conciseness, realism, modifiability and traceability.</p> <p><i>Task 7. Prioritize requirements.</i> Its goal is to provide different levels of priority to the functional and non-functional ontology requirements identified. In the case of the functional requirements, priority should be given to the different groups of CQs, and, within each group, to the different CQs; additionally, priority could be given to each CQs independently of the groups. Priority will be used by the ontology development team for planning and scheduling the ontology development and for deciding which parts of the ontology will be developed first. This task is optional but recommended.</p> <p><i>Task 8. Extract terminology and its frequency.</i> Its goal is to extract a pre-glossary of terms with their frequencies from the list of CQs and their answers, which are identified in Task 4. Terminology extraction techniques and tools supporting such techniques can be used. This pre-glossary of terms is divided in three different parts: terms from the CQs, terms from the CQs' answers, and terms identified as named entities. The set of terms appearing with high frequency will be used later on for searching knowledge resources that could be potentially reused in the ontology development. As heuristic, we can mention that normally the set of terms appearing more frequently is that which requires more effort during the ontology development; for this reason, frequency is important because it takes us towards those resources that allow saving more effort.</p>

Scenarios	Excerpt of methodological guidelines for processes and activities
<p><i>Scheduling activity</i> (Suárez-Figueroa, 2010; Suárez-Figueroa et al., 2012). Its goal is to organize the different processes and activities in time, that is, to state a concrete planning or schedule that guides the ontology network development, including processes and activities, their order and time, as well as human resources restrictions.</p>	<p><i>Task 1. Select the ontology network life cycle model.</i> Its goal is to obtain the most appropriate ontology network life cycle model for the ontology network to be developed. The ontology requirement specification document (ORSD) and the types of potential knowledge resources to be reused during the development are taken as input.</p> <p><i>Task 2. Select the set of scenarios.</i> Its goal is to select the set of scenarios to be followed during the ontology network development. The ORSD and the set of potential knowledge resources to be used during the development are taken as input.</p>
<p><i>NOR Reuse process</i> (Villazón-Terrazas, 2012). Its goal is to choose the most suitable non-ontological resources for building ontologies. The output is a set of non-ontological resources that, to some extent, covers the expected domain.</p>	<p><i>Task 3. Update initial plan.</i> Its goal is to modify (if necessary) the initial plan presented by gOntt. The ORSD and the set of potential knowledge resources to be used during the development are taken as input. Ontology developers can modify the initial plan in the following ways: (a) by including or deleting processes, activities and model phases, and (b) by changing order and dependencies between processes and activities.</p> <p><i>Task 4. Establish resource restrictions and assignments.</i> Its goal is to include information about temporal scheduling and human resource assignments in the life cycle obtained in Task 3. The ORSD and the set of potential knowledge resources to be used during the development are taken as input.</p>
<p><i>NOR Reengineering process</i> (Villazón-Terrazas, 2012). Its goal is to transform a non-ontological resource into an ontology. The output of the process is an ontology.</p>	<p><i>Activity 1. Search non-ontological resources.</i> The goal of the activity is to search NORs from highly reliable web sites, domain-related sites and resources within organizations. The terms appearing most frequently in the ORSD are used to search for the candidate NORs that cover the terminology desired.</p> <p><i>Activity 2. Assess the set of candidate non-ontological resources.</i> The goal of this activity is to assess the set of candidate NORs. This activity is divided into the following 6 tasks: (1) extracting lexical entries of the NORs; (2) calculating precision; (3) calculating coverage; (4) evaluating the consensus; (5) evaluating the quality; and (6) building an assessment table.</p> <p><i>Activity 3. Select the most appropriate non-ontological resources.</i> Its goal is to select the most appropriate NORs that are to be transformed into an ontology. The selection is performed manually and it is recommended to look for resources with consensus, quality, high value of coverage and high value of precision. The activity output is a ranked list of NORs that, to some extent, covers the expected domain. These resources are ready for the reengineering process.</p> <p><i>Activity 1. Non-ontological resource reverse engineering.</i> Its goal is to analyze a NOR, to identify its underlying terms, and to create representations of the resource at different levels of abstraction (design, requirements and conceptual). The activity is divided into 3 tasks (data gathering, conceptual abstraction and information exploration).</p> <p><i>Activity 2. Non-ontological resource transformation.</i> Its goal is to generate a conceptual model from the NOR. It is recommended the use of patterns for reengineering non-ontological resources (PR-NOR) in order to guide the transformation process. Thus, the activity is divided into (1) the search for a suitable pattern for reengineering NORs, (2) the use of reengineering patterns to guide the transformation or perform an ad-hoc transformation and (3) the manual refinement.</p> <p><i>Activity 3. Ontology forward engineering.</i> Its goal is to generate the ontology. The conceptual model obtained in the second task of Activity 2 is transformed into a formalized model, according to a knowledge representation paradigm, such as description logics, first order logic or F-logic. Then, the formalized model is implemented in an ontology language.</p>

Scenarios	Excerpt of methodological guidelines for processes and activities
Scenario 3: Reusing ontological resources	<p><i>Domain Ontology Reuse process</i> (Suárez-Figueroa, 2010, 2012). Its goal is to find and select one or several domain ontologies that can be reused in the ontology being developed.</p> <p><i>Activity 1. Domain ontology search.</i> Its goal is to search in libraries, repositories and registries for candidate domain ontologies that could satisfy the needs of the ontology network being developed. Those terms appearing in the pre-glossary of the ORSD are used as input.</p> <p><i>Activity 2. Domain ontology assessment.</i> The objective of this activity is to find out if the set of candidate domain ontologies are useful for the development of the ontology network. The proposal considers analyzing the domain coverage and deciding whether a particular domain ontology is useful or not. This means analyzing whether the domain ontology covers (totally or partially) the requirements identified in the ORSD of the ontology network being developed.</p>
<i>General Ontology Reuse process</i> (Fernández-López et al., 2013). Its goal is to find and select general ontologies and integrate them into the ontology network being developed.	<p><i>Activity 3. Domain ontology select.</i> Its goal is to find out which domain ontologies are the most suitable for the development of the ontology network. As a result of this activity, the ontology developers should select the candidate domain ontology that scored the highest results.</p> <p><i>Activity 4. Domain ontology integration.</i> Its goal is to integrate the domain ontologies selected in Activity 3 into the ontology network being developed.</p> <p><i>Activity 1. Conduct a comparative study of reusable general ontologies.</i> Its goal is to find and compare general ontologies. We propose to build a table to represent the comparative study. From the point of view of the formal concept analysis (FCA) (Stumme, 2009), this table is essentially a formal context. The contextual objects are the ontologies to be compared, and the attributes are both the definitions and the axioms that these ontologies include or features of the models that they describe (e.g., the definition of time points, the assumption of different granularities).</p> <p><i>Activity 2. Make a decision on which general ontology is going to be reused.</i> Its goal is to select the most appropriate general ontology to be reused in the ontology network being developed. The table obtained in the comparative study is used to select the general ontology that best fits the requirements of the ontology to be developed. FCA is used to make this decision by structuring the different options (candidate general ontologies) in lattices. Such lattices are obtained and used during this activity of decision making.</p> <p><i>Activity 3. Assemble the whole ontology.</i> Its goal is to customize and integrate the most appropriate general ontology into the ontology to be developed.</p> <p><i>Note:</i> Let's note that if there is a previous comparative study, then Activity 1 does not have to be carried out, and thus the reuse process should start directly on Activity 2.</p>

Scenarios	Excerpt of methodological guidelines for processes and activities
<p><i>Ontology Statements Reuse process</i> (Suárez-Figueroa, 2010, 2012). Its goal is to make use of ontology statements in the ontology network being developed. The output of this process is a set of ontology statements that will be used.</p>	<p><i>Activity 1. Ontology statement search.</i> Its goal is to search the Web for ontology statements candidates that could satisfy the ontological needs of a particular case. The ORSD is taken as input, specifically those terms that have a high frequency. Some gateways to the Semantic Web such as Watson on-line or the Watson plug-in in the NeOn Toolkit can be used for the search.</p> <p><i>Activity 2. Ontology statement assessment.</i> Its goal is to decide whether a concrete ontology statement is useful for the ontology network being developed. The content and granularity of ontology statements should be inspected to assess whether they satisfy the ontological needs. A set of criteria for assessing each ontology statement is provided.</p> <p><i>Activity 3. Ontology statement selection.</i> Its goal is to decide which of the useful ontology statements provided are the best or most convenient for the ontology network being developed. The criterion of minimum effort needed for integrating the ontology statement should be used when comparing ontology statements and selecting the most convenient ones.</p> <p><i>Activity 4. Ontology statement integration.</i> Its goal is to integrate the ontology statements selected in Activity 3 into the ontology network being developed. After integrating an ontology statement, the following tasks will probably have to be performed: (a) changing names (concepts, properties) to adapt them to the naming conventions used in the ontology network being developed; (b) adding range to properties and changing cardinalities; and (c) adding restrictions.</p>
<p>Scenario 5: Reusing and merging ontological resources</p>	<p><i>Ontology Matching activity</i> (Euzenat &amp; Le Duc, 2012). Its goal is to find or discover relationships or correspondences between entities of different ontologies or ontology modules. This activity can be seen as the first stage of Ontology Alignment, which refers to the activity of finding the correspondences between two or more ontologies and storing/exploiting them.</p> <p><i>Task 1. Identify ontologies and characterise needs.</i> Its goal is to identify the ontologies to be matched and to characterise the needs.</p> <p><i>Task 2. Find existing alignments.</i> Its goal is to find existing alignments that satisfy the needs. Alignments may be found on the web or through specialised directories.</p> <p><i>Task 3. Select a matcher.</i> Its goal is to find a suitable matcher to build a new alignment.</p> <p><i>Task 4. Match ontologies.</i> Its goal is to run the selected matcher against the ontologies and to collect the resulting alignment.</p> <p><i>Task 5. Evaluate alignments.</i> Its goal is to perform a final screening and validation.</p> <p><i>Task 6. Enhance alignments.</i> Its goal is to enhance the alignments either through a manual modification of the alignment or through the application of refinement procedures.</p> <p><i>Task 7. Store and share alignments.</i> Its goal is to save and share the alignment obtained into a declarative format and to give this alignment proper annotation in order to record its provenance and purpose.</p> <p><i>Task 8. Render alignments.</i> Its goal is to transform the alignment into another form or to interpret the alignment so that it can perform actions such as mediation or merging.</p>

Scenarios		Excerpt of methodological guidelines for processes and activities
Scenario 7: Reusing ontology design patterns (ODPs)	<p><i>Ontology Design Patterns Reuse process</i>            (Suárez-Figueroa et al., 2009; Presutti et al., 2012). Its goal is to use available ontology design patterns in order to solve different modelling problems during the development of ontologies.</p>	<p><i>Activity 1. Identify requirements to be addressed.</i> The objective of this activity is to identify which requirement(s), from the ORSD, can be addressed by ontology design pattern reuse. The selection of requirements may be based on the following (non-exhaustive) list of criteria: which requirements are not yet addressed within other activities; which requirements do not provide trivial solutions; and which requirements can be associated with existing pattern types.</p> <p><i>Activity 2. Identify available patterns.</i> Its goal is to identify as many patterns, or rather pattern catalogues, as possible that could help solve the modelling issues proposed by the requirements selected in Activity 1.</p> <p><i>Activity 3. Divide and transform the problem, select a partial problem.</i> Its goal is to prepare the set of modeling problems posed by the requirements for matching the set of available patterns. In order to match the problem with such small partial solutions and to reuse them, the problem needs to be divided into manageable pieces. When the problem has been transformed and divided, one such “manageable piece” is selected as a starting point.</p> <p><i>Activity 4. Match selected partial problem to patterns.</i> Its goal is to identify which patterns can solve which parts of the selected sub-problem, if any. Two general suggestions are provided for performing this activity in order to assist the matching: (a) to identify the type of patterns that may be suitable for solving the problem and to match the problem primarily to these kinds of patterns and (b) depending on the type of pattern, to use the description of the intent of the pattern to match with the problem (for content patterns, this would mean to match the competency questions of the patterns with the competency questions of the problem).</p> <p><i>Activity 5. Select patterns.</i> Its goal is to select a set of patterns for reuse based on the results of the matching process carried out in Activity 4.</p> <p><i>Activity 6. Apply (reuse) selected patterns and compose them.</i> Its goal is to reuse the selected patterns and compose them.</p> <p><i>Activity 7. Evaluate and revise with respect to partial problem.</i> Its goal is to test the solution on the selected partial problem at hand and to ensure that it really solves the problem correctly.</p> <p><i>Activity 8. Integrate partial solutions.</i> Its goal is to integrate the solutions into the partial problems solved during the different iterations or during the solution of different sub-problems by the parallel teams.</p>

Scenarios	Excerpt of methodological guidelines for processes and activities
Scenario 8: Restructuring ontological resources	<p><i>Ontology Modularization activity</i> (d'Aquin, 2012). Its goal is to obtain a module or a set of modules from an ontology which fits the requirements of a particular application or a particular scenario.</p> <p><i>Task 1. Identify the purpose of modularization.</i> Its goal is to identify both the reasons for modularizing the ontologies and the benefits expected from guiding the rest of the process.</p> <p><i>Task 2. Select a modularization approach.</i> Its goal is to choose the best modularization approach between ontology partitioning (the activity of dividing an ontology into a set of (not necessarily disjoint) modules that together form an ontology, but which can be treated separately) and ontology module extraction (the activity of obtaining from an ontology some concrete modules that could be used for a particular purpose). Whenever the purpose relates to the entire ontology (i.e. when improving maintenance and, in some cases, performance), a partitioning approach should be considered. Whenever the purpose relates to extracting specific parts of an ontology (e.g. to customize or reuse it partially), a module extraction should be considered.</p> <p><i>Task 3. Define modularization criteria.</i> Its goal is to define the modularization criteria. These criteria determine the basic characteristics that the resulting modules should have, that is, what should go into a module.</p> <p><i>Task 4. Select a base modularization technique.</i> Its goal is to select the most appropriate technique, depending on the criteria to apply.</p> <p><i>Task 5. Parametrize the technique and applying it.</i> Its goal is to select the different parameters required to obtain interesting and useful results when the technique is applied. Most of the techniques would, in principle, be applied in the same way, namely, taking the original ontology as input and creating modules in the form of smaller ontologies, thus allowing processing the resulting modules iteratively, in the same way as the original ontology.</p> <p><i>Task 6. Combine results.</i> Its goal is to refine and combine the results obtained with various parameters, techniques and approaches in an iterative process.</p> <p><i>Task 7. Evaluate modularization.</i> Its goal is to evaluate the result of the modularization. This evaluation is crucial to decide whether a new iteration is necessary (in such a case, a new set of criteria and a new technique is applied), or whether the current (set of) modules are satisfactory, considering the application scenario. The modularization could be evaluated in two different ways: (a) by checking the criteria and (b) by testing against the purpose of modularization.</p> <p><i>Task 8. Finalize modularization.</i> Its goal (if necessary) is to deploy and exploit the modularization results in an application.</p>

Scenarios		Excerpt of methodological guidelines for processes and activities
Scenario 9: Localizing ontological resources	<i>Ontology Localization activity</i> (Espinoza Mejía et al., 2012). Its goal is to adapt an ontology to a particular language and culture.	<p><i>Task 1. Select the most appropriate linguistic assets.</i> Its goal is to select the most appropriate linguistic assets that help in the localization activity. The task output is a set of linguistic assets that can help reduce the cost, improve the quality, and increase the consistency of the localization activity. The choice of a specific resource is performed manually, taking into account that the linguistic assets comply with the following characteristics: (a) consensus, (b) broad coverage and (c) high precision.</p> <p><i>Task 2. Select ontology label(s) to be localized.</i> Its goal is to select the ontology label(s) to be localized. The task output is a set of ontology labels and their context. The context describes the meaning of a specific label in the ontology and consists of a small excerpt of ontology labels around the ontology label itself (e.g., direct hypernym labels, hyponym labels).</p> <p><i>Task 3. Obtain ontology label translation(s).</i> Its goal is to obtain the most appropriate translation in the target language for each ontology label. Different machine translation (MT) techniques can be used to perform this task automatically.</p> <p><i>Task 4. Evaluate label translation(s).</i> Its goal is to evaluate label translations in the target language. Different linguistic criteria can be used for evaluating label translations. Two levels of evaluation criteria (semantic fidelity evaluation and stylistic evaluation) are proposed.</p> <p><i>Task 5. Ontology update.</i> Its goal is to update the ontology with the label translations obtained for each localized label. The task output is an ontology enriched with labels associated to each localised term in the target language.</p>
Across scenarios	<i>Ontology Evaluation activity</i> (Sabou & Fernandez, 2012). Its goal is to check the technical quality of an ontology against a frame of reference.	<p><i>Task 1. Select individual components of the ontology network.</i> Its goal is to identify the elements of the ontology network that need to be evaluated, including individual ontologies, ontology statements and ontology relations, among others. This decision should be based on two criteria: (a) which ontology network elements are critical for the overall network; and (b) which of these elements can actually be evaluated.</p> <p><i>Task 2. Select an evaluation goal and approach.</i> Its goal is to decide the aim of the evaluation and to select an appropriate evaluation approach. The following evaluation goals can be distinguished: (a) domain coverage, (b) quality of the modelling, (c) suitability for an application/task and (d) adoption and use.</p> <p><i>Task 3. Identify a frame of reference and evaluation metric.</i> Its goal is to select the concrete ingredients of the evaluation that consist of a frame of reference and a set of evaluation metric(s).</p> <p><i>Task 4. Apply the selected evaluation approach.</i> Its goal is to apply the selected evaluation approach. This task requires, on the one hand, a proper setup for the evaluation experiments and the implementation of software tools to compute the evaluation metrics, and on the other, the engagement of the human experts in stimulating sessions to collect their evaluations.</p> <p><i>Task 5. Combine and present individual evaluation results.</i> Its goal is to combine the evaluation results derived from individual components in order to reach a global understanding of the network's quality. Finally, the results of the evaluation should be presented in an appropriate form for possible repair (corrections, additions) and improvements, and for the future evolution of the ontology network.</p>

Scenarios	Excerpt of methodological guidelines for processes and activities
<p><i>Ontology Evolution activity</i> (Palma et al., 2012). Its goal is to facilitate the modification of an ontology by preserving its consistency.</p>	<p><i>Task 1. Request a change.</i> Its goal is to initiate the change process. Changes can either be requested from users or developers, who feel that the ontology is not adequate in its current form; changes can also be discovered.</p> <p><i>Task 2. Plan the change.</i> Its goal is to analyze the change requested and to determine both why the change needs to be made and which part of the ontology is affected by such a change. As a result of the analysis performed during this task, the ontology engineer may decide to implement the change, and if this has many side effects or if the cost of implementation is too high, he may defer the change request to a later time or not implement it at all.</p> <p><i>Task 3. Implement the change.</i> Its goal is to implement the changes. Changes can be as easy as adding or removing a subclass; however, some changes can require complex operations as well as the restructuration of the ontology. Change logging allows tracking which changes have been made; it also allows undoing them easily, in case something goes wrong.</p> <p><i>Task 4. Verification and validation.</i> Its goal is to verify and validate the ontology after this is considered completely evolved. During this assessment, the ontology modified originally is commonly verified in isolation; this activity can include the verification of other artefacts related to the ontology (as mentioned above) to ensure that they were not changed in a wrong way or that they have an unexpected behaviour.</p>

## References

- Barbosa, P., Guizzardi, R.S. & Guizardi, G. (2011). Using goal modeling to capture competency questions in ontology-based systems. *Journal of Information and Data Management (JIDM)*, 2(3), 527–540.
- Barresi, S., Rezgui, Y., Lima, C. & Meziane, F. (2008). *Semantic Resources Integration and Interoperability in the Construction Domain*. Enterprise Information Systems (pp. 336–347). Berlin, Heidelberg: Springer.
- Bloom, B. (Ed.) (1956). *Taxonomy of Educational Objectives*. Handbook I: Cognitive Domain. Addison-Wesley.
- Box, G.E.P., Hunter, J.S. & Hunter, W.G. (2005). *Statistics for Experimenters* (2nd edn.). Wiley & Sons.
- Byrne, E.J. (1992). A conceptual foundation for software re-engineering. In *Proceedings of the International Conference on Software Maintenance and Reengineering* (pp. 226–235). IEEE Computer Society.
- Clemente, J. (2011). Una Propuesta de Modelado del Estudiante Basada en Ontologías y Diagnóstico Pedagógico-Cognitivo no Monótono. PhD thesis, Facultad de Informática, Universidad Politécnica de Madrid.
- Clemente, J., Ramírez, J. & de Antonio, A. (2011). A proposal for student modeling based on ontologies and diagnosis rules. *Expert Systems with Applications*, 38(7), 8066–8078.
- d'Aquin, M. (2012). Modularizing ontologies. In M.C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta and A. Gangemi (Eds.), *Ontology Engineering in a Networked World* (pp. 213–233). Springer. ISBN 978-3-642-24793-4.
- d'Aquin, M., Schlicht, A., Stuckenschmidt, H. & Sabou, M. (2007). Ontology modularization for knowledge selection: Experiments and evaluations. In *18th International Conference on Database and Expert Systems Applications (DEXA 2007)*, Regensburg, Germany.
- de Hoog, R. (1998). Methodologies for building knowledge based systems: Achievements and prospects. In J. Liebowitz (Ed.), *Handbook of Expert Systems*, Boca Raton, FL: CRC Press. Chapter 1.
- Dzbor, M., Suárez-Figueroa, M.C., Blomqvist, E., Lewen, H., Espinoza, M., Gómez-Pérez, A. & Palma, R. (2009). D5.6.2 experimentation and evaluation of the NeOn Methodology. NeOn project. Available at: <http://www.neon-project.org>.
- Egaña Aranguren, M., Antezana, E., Kuiper, M. & Stevens, R. (2008). Ontology design patterns for bio-ontologies: A case study on the cell cycle ontology. *BMC Bioinformatics*, 9(Suppl. 5), S1. doi:[10.1186/1471-2105-9-S5-S1](https://doi.org/10.1186/1471-2105-9-S5-S1). Online ISSN: 1471-2105.
- Egaña, M., Rector, A., Stevens, R. & Antezana, E. (2008). Applying ontology design patterns in bio-ontologies. In *Knowledge Engineering: Practice and Patterns, 16th International Conference, EKAW 2008*. Lecture Notes in Computer Science Acitrezza, Italy, 29 September–2 October 2008 (Vol. 5268, pp. 7–16).
- Espinoza Mejía, M., Montiel-Ponsoda, E., Aguado de Cea, G. & Gómez-Pérez, A. (2012). Ontology localization. In M.C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta and A. Gangemi (Eds.), *Ontology Engineering in a Networked World* (pp. 171–192). Springer. ISBN 978-3-642-24793-4.
- Euzenat, J. & Le Duc, C. (2012). Methodological guidelines for matching ontologies. In M.C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta and A. Gangemi (Eds.), *Ontology Engineering in a Networked World* (pp. 257–278). Springer. ISBN 978-3-642-24793-4.
- Fernández-López, M., Gómez-Pérez, A. & Juristo, N. (1997). METHONTOLOGY: From ontological art towards ontological engineering. In *Spring Symposium on Ontological Engineering of AAAI* (pp. 33–40). Stanford, CA: Stanford Univ.
- Fernández-López, M., Gómez-Pérez, A., Pazos-Sierra, J. & Pazos-Sierra, A. (1999). Building a chemical ontology using methontology and the ontology design environment. *IEEE Intelligent Systems*, 14(1), 37–46.
- Fernández-López, M., Gómez-Pérez, A. & Suárez-Figueroa, M.C. (2013). Methodological guidelines for reusing general ontologies. *Data & Knowledge Engineering*, 86, 242–275.
- Francesconi, E., Montemagni, S., Peters, W. & Tiscornia, D. (2010). Integrating a bottom-up and top-down methodology for building semantic resources for the multilingual legal domain. In *Semantic Processing of Legal Texts* (pp. 95–121). Berlin, Heidelberg: Springer. ISBN 978-3-642-12836-3.
- Gangemi, A. (2005). Ontology design patterns for semantic web content. In Musen et al. (Eds.), *Proceedings of the Fourth International Semantic Web Conference*, Galway, Ireland. Springer.
- Gómez-Pérez, A., Fernández-López, M. & Corcho, O. (2003). *Ontological Engineering*. Advanced Information and Knowledge Processing Series. Springer. ISBN 1-85233-551-3.
- Gómez-Pérez, A. & Manzano-Macho, D. (2004). An overview of methods and tools for ontology learning from texts. *Knowledge Engineering Review (KER)*, 19(3), 187–212.
- Gómez-Pérez, J.M., Méndez, V., Candini, J. & Muñoz, J.C. (2012). Electronic invoice management in the pharmaceutical sector: The PharmaInnova case. In M.C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta and A. Gangemi (Eds.), *Ontology Engineering in a Networked World* (pp. 407–422). Springer. ISBN 978-3-642-24793-4.
- Grüninger, M. & Fox, M.S. (1995). Methodology for the design and evaluation of ontologies. In D. Skuce (Ed.), *IJCAI'95 Workshop on Basic Ontological Issues in Knowledge Sharing*.
- Haase, P., Rudolph, S., Wang, Y., Brockmans, S., Palma, R., Euzenat, J. & d'Aquin, M. (2006). NeOn deliverable D1.1.1. networked ontology model. NeOn project. Available at: <http://www.neon-project.org/>.

- Hodge, G. (2000). Systems of knowledge organization for digital libraries: Beyond traditional authority files. Available at: <http://www.clir.org/pubs/reports/pub91/contents.html>.
- Iqbal, R., Azrifah, M., Murad, A., Mustapha, A. & Sharef, N.M. (2013). An analysis of ontology engineering methodologies: A literature review. *Research Journal of Applied Sciences, Engineering and Technology*, 6(16), 2993–3000.
- Kitchenham, B. (1996). DESMET: A method for evaluating Software Engineering methods and tools. Technical Report TR96-09. ISSN: 1353-7776.
- Kutz, O., Mossakowski, T. & Lücke, D. (2010). Carnap, Goguen, and the hyperontologies: Logical pluralism and heterogeneous structuring in ontology design. *Logica Universalis*, 4(2), 255–333. Print ISSN: 1661-8297. Online ISSN: 1661-8300.
- Lamsfus, C., Alzua-Sorzarabal, A., Martin, D., Salvador, Z. & Usandizaga, A. (2009). Human-centric ontology-based context modelling in tourism. In *Proceedings of KEOD 2009* (pp. 424–434).
- Larman, C. (2002). *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process* (2nd edn.). Pearson Education, Inc. Publishing as Prentice Hall PTR. ISBN 0-13-092569-1.
- Lawrynowicz, A. & Palma, R. (2012). Applications of ontology design patterns in the transformation of multimedia repositories. In *Proceedings of the 3rd Workshop on Ontology Patterns*. CEUR Workshop Proceedings (Vol. 929). Boston, USA, 12 November 2012. Available at: <http://ceur-ws.org/Vol-929/paper7.pdf>.
- Melero, R., Penela, V. & Ruiz, C. (2010). Yes we lunch! In *Demo Paper at 7th Extended Semantic Web Conference 2010 (ESWC2010)*.
- Noy, N.F. & McGuinness, D.L. (2001). Ontology development 101: A guide to creating your first ontology. Technical Report KSL-01-05, Stanford University Knowledge Systems Laboratory.
- Palma, R., Zablith, F., Haase, P. & Corcho, O. (2012). Ontology evolution. In M.C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta and A. Gangemi (Eds.), *Ontology Engineering in a Networked World* (pp. 235–255). Springer. ISBN 978-3-642-24793-4.
- Palvia, P. & Nosek, J.T. (1990). An empirical evaluation of system development methodologies. *Information Resources Management Journal*, 3(3), 23–32.
- Pariente-Lobo, T. & Herrero-Cárcel, G. (2012). Integrating product information in the pharmaceutical sector. In M.C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta and A. Gangemi (Eds.), *Ontology Engineering in a Networked World* (pp. 423–434). Springer. ISBN 978-3-642-24793-4.
- Penela, V. & Ruiz, C. (2010). Context Twitter: Living a social life. In *Demo Paper at 7th Extended Semantic Web Conference 2010 (ESWC2010)*.
- Pinto, H.S., Tempich, C. & Staab, S. (2004). DILIGENT: Towards a fine-grained methodology for DIstributed, Loosely-controlled and evolvInG Engineering of oNTOlogies. In R. López de Mantaras and L. Saitta (Eds.), *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004)*, 22nd–27th August 2004 (pp. 393–397). Valencia, Spain: IOS Press. ISSN: 0922-6389. ISBN 1-58603-452-9.
- Presutti, V., Blomqvist, E., Daga, E. & Gangemi, A. (2012). Pattern-based ontology design. In M.C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta and A. Gangemi (Eds.), *Ontology Engineering in a Networked World* (pp. 35–64). Springer. ISBN 978-3-642-24793-4.
- R Core Team (2014). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. Available at: <http://www.R-project.org/>.
- Rao, L., Reichgelt, H. & Osei-Bryson, K. (2008). Knowledge elicitation techniques for deriving competency questions for ontologies. In *Proceedings of the Tenth International Conference on Enterprise Information Systems (ICEIS 2008)*, Barcelona, Spain, 12–16 June 2008 (Vol. ISAS-2, pp. 105–110). ISBN 978-989-8111-38-8.
- Rector, A., Drummond, N., Horridge, M., Rogers, J., Knublauch, H., Stevens, R., Wang, H. & Wroe, C. (2004). OWL pizzas: Practical experience of teaching OWL-DL: Common errors and common patterns. In *Proceedings of EKAW 2004* (pp. 63–81). Springer.
- Sabou, M. & Fernandez, M. (2012). Ontology (network) evaluation. In M.C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta and A. Gangemi (Eds.), *Ontology Engineering in a Networked World* (pp. 193–212). Springer. ISBN 978-3-642-24793-4.
- Simperl, E. & Luczak-Rösch, M. (2014). Collaborative ontology engineering: A survey. *The Knowledge Engineering Review*, 29, 101–131.
- Sommerville, I. (2007). *Software Engineering* (8th edn.). Addison-Wesley. ISBN 0-321-31379-8.
- Staab, S., Schnurr, H.P., Studer, R. & Sure, Y. (2001). Knowledge processes and ontologies. *IEEE Intelligent Systems*, 16(1), 26–34.
- Stadlhofer, B., Salhofer, P. & Durlacher, A. (2013). An overview of ontology engineering methodologies in the context of public administration. In *The Seventh International Conference on Advances in Semantic Processing, SEMAPRO 2013*, Porto, Portugal (pp. 36–42).
- Stumme, G. (2009). Formal concept analysis. In *Handbook on Ontologies* (pp. 177–199). Berlin, Heidelberg: Springer.
- Suárez-Figueroa, M.C. (2010). NeOn Methodology for building ontology networks: Specification, scheduling and reuse. PhD thesis, Universidad Politécnica de Madrid, Spain. Available at: <http://oa.upm.es/3879/>.
- Suárez-Figueroa, M.C. (2012). *NeOn Methodology for Building Ontology Networks: Specification, Scheduling and Reuse*. Dissertations in Artificial Intelligence (Vol. 338). IOS Press. ISBN 978-1-61499-115-1.

- Suárez-Figueroa, M.C., Aguado-de-Cea, G. & Gómez-Pérez, A. (2013). Lights and shadows in creating a glossary about ontology engineering. *Terminology*, 19(2), 202–236.
- Suárez-Figueroa, M.C., Blomqvist, E., d'Aquin, M., Espinoza, M., Gómez-Pérez, A., Lewen, H., Mozetic, I., Palma, R., Poveda, M., Sini, M., Villazón-Terrazas, B., Zablith, F. & Dzbor, M. (2009). NeOn deliverable D5.4.2. revision and extension of the NeOn Methodology for building contextualized ontology networks. NeOn project. Available at: <http://www.neon-project.org>.
- Suárez-Figueroa, M.C. & Gómez-Pérez, A. (2008). Building ontology networks: How to obtain a particular ontology network life cycle? In *Proceedings of the International Conference on Semantic Systems (ISEMANTICS 2008)* (pp. 142–149).
- Suárez-Figueroa, M.C., Gómez-Pérez, A., Motta, E. & Gangemi, A. (Eds.) (2012). *Ontology Engineering in a Networked World*. Springer. ISBN 978-3-642-24793-4.
- Taylor, J.C. (2008). *Project Scheduling and Cost Control: Planning, Monitoring and Controlling the Baseline*. J. Ross Publishing. ISBN 9781932159110.
- Tudorache, T., Nyulas, C., Noy, N.F. & Musen, M.A. (2013). WebProtégé: A collaborative ontology editor and knowledge acquisition tool for the web. *Semantic Web*, 4(1), 89–99.
- Villazón-Terrazas, B. (2012). *A Method for Reusing and Re-engineering Non-ontological Resources for Building Ontologies. Studies on the Semantic Web* (Vol. 12). AKA Verlag–IOS Press. ISBN 978-3-89838-666-1.
- Villazón-Terrazas, B., Ramírez, J., Suárez-Figueroa, M.C. & Gómez-Pérez, A. (2011). A network of ontology networks for building e-employment advanced systems. *Expert Systems with Applications*, 38(11), 13612–13624.
- Villazón-Terrazas, B., Suárez-Figueroa, M.C. & Gómez-Pérez, A. (2010). A pattern-based method for re-engineering non-ontological resources into ontologies. *International Journal on Semantic Web and Information Systems*, 6(4), 27–63.