NeOn-project.org

NeOn: Lifecycle Support for Networked Ontologies

Integrated Project (IST-2005-027595)

Priority: IST-2004-2.4.7 – "Semantic-based knowledge and content systems"

D5.4.3. Revision and Extension of the NeOn Methodology for Building Contextualized Ontology Networks

---

Deliverable Co-ordinator: **Mari Carmen Suárez-Figueroa**

Deliverable Co-ordinating Institution: **UPM**

Other Authors: **Asunción Gómez-Pérez (UPM), María Poveda (UPM), Jose Angel Ramos (UPM), Jérôme Euzenat (INRIA), Chan Le Duc (INRIA)**

This deliverable presents the third version of the NeOn Methodology for building ontology networks. The version extends the previous deliverables in WP5 and provides methodological guidelines for reusing ontology modules, mapping and merging ontologies, and combining different types of knowledge resources during the ontology network development.

Document Identifier:	NEON/2010/D5.4.3/v1.0	Date due:	January 31st, 2010
Class Deliverable:	NEON EU-IST-2005-027595	Submission date:	January 31st, 2010
Project start date:	March 1, 2006	Version:	v1.0
Project duration:	4 years	State:	Final
		Distribution:	Public

NeOn Consortium

This document is a part of the NeOn research project funded by the IST Programme of the Commission of the European Communities by the grant number IST-2005-027595. The following partners are involved in the project:

Open University (OU) – Coordinator Knowledge Media Institute – KMi Berrill Building, Walton Hall Milton Keynes, MK7 6AA United Kingdom Contact person: Enrico Motta E-mail address: e.motta@open.ac.uk	**Universität Karlsruhe – TH (UKARL)** Institut für Angewandte Informatik und Formale Beschreibungsverfahren – AIFB Englerstrasse 28 D-76128 Karlsruhe, Germany Contact person: Andreas Harth E-mail address: aha@aifb.uni-karlsruhe.de
Universidad Politécnica de Madrid (UPM) Campus de Montegancedo 28660 Boadilla del Monte Spain Contact person: Asunción Gómez Pérez E-mail address: asun@fi.upm.es	**Software AG (SAG)** Uhlandstrasse 12 64297 Darmstadt Germany Contact person: Walter Waterfeld E-mail address: walter.waterfeld@softwareag.com
Intelligent Software Components S.A. (ISOCO) Calle de Pedro de Valdivia 10 28006 Madrid Spain Contact person: Jesús Contreras E-mail address: jcontreras@isoco.com	**Institut 'Jožef Stefan' (JSI)** Jamova 39 SI-1000 Ljubljana Slovenia Contact person: Marko Grobelnik E-mail address: marko.grobelnik@ijs.si
Institut National de Recherche en Informatique et en Automatique (INRIA) ZIRST – 655 avenue de l'Europe Montbonnot Saint Martin 38334 Saint-Ismier France Contact person: Jérôme Euzenat E-mail address: jerome.euzenat@inrialpes.fr	**University of Sheffield (USFD)** Dept. of Computer Science Regent Court 211 Portobello street S14DP Sheffield United Kingdom Contact person: Hamish Cunningham E-mail address: hamish@dcs.shef.ac.uk
Universität Koblenz-Landau (UKO-LD) Universitätsstrasse 1 56070 Koblenz Germany Contact person: Steffen Staab E-mail address: staab@uni-koblenz.de	**Consiglio Nazionale delle Ricerche (CNR)** Institute of cognitive sciences and technologies Via S. Martino della Battaglia, 44 - 00185 Roma-Lazio, Italy Contact person: Aldo Gangemi E-mail address: aldo.gangemi@istc.cnr.it
Ontoprise GmbH. (ONTO) Amalienbadstr. 36 (Raumfabrik 29) 76227 Karlsruhe Germany Contact person: Jürgen Angele E-mail address: angele@ontoprise.de	**Food and Agriculture Organization of the United Nations (FAO)** Viale delle Terme di Caracalla 1 00100 Rome Italy Contact person: Caterina Caracciolo E-mail address: Caterina.Caracciolo@fao.org
Atos Origin S.A. (ATOS) Calle de Albarracín, 25 28037 Madrid Spain Contact person: Tomás Pariente Lobo E-mail address: tomas.parientelobo@atosorigin.com	**Laboratorios KIN, S.A. (KIN)** C/Ciudad de Granada, 123 08018 Barcelona Spain Contact person: Antonio López E-mail address: alopez@kin.es

Work package participants

The following partners have taken an active part in the work leading to the elaboration of this document, even if they might not have directly contributed to the writing of this document or its parts:

UPM, INRIA, FAO

Change Log

Version	Date	Amended by	Changes
0.00	11-09-2009	Asunción Gómez-Pérez	ToC and first version of the introduction
0.10	15-09-2009	Mari Carmen Suárez-Figueroa	Second version of the introduction. First version of chapter 2: the NeOn Methodology
0.20	20-10-2009	María Poveda	First version of chapter 3: ontology module reuse
0.30	19-11-2009	María Poveda	Update of chapter 3:ontology module reuse
0.40	20-12-2009	María Poveda	First version of the chapter on combining different type of knowledge resources
0.45	29-12-2009	María Poveda Mari Carmen Suárez-Figueroa	Revision and update of chapter 3: ontology module reuse
0.50	3-1-2010	María Poveda	Update of the chapter on combining different type of knowledge resources
0.55	5-01-2010	Jose Angel Ramos	Chapter on ontology merging
0.60	7-1-2010	María Poveda Mari Carmen Suárez-Figueroa	Revision and update of the chapter on combining different type of knowledge resources
0.65	10-01-2010	Jose Angel Ramos Mari Carmen Suárez-Figueroa	Revision and update of the chapter on ontology merging
0.75	10-01-2010	Mari Carmen Suárez-Figueroa	Update of the introduction. Update of chapter 2: the NeOn Methodology
0.80	10-01-2010	Mari Carmen Suárez-Figueroa	Conclusions
0.85	15-01-2010	Jose Angel Ramos	Update of the chapter on ontology merging
0.90	15-01-2010	Jerome Euzenat Chan LeDuc	Chapter on ontology mapping
0.95	15-01-2010	Mari Carmen Suárez-Figueroa	Update of the introduction and conclusions
0.96	5-02-2010	Mari Carmen Suárez-Figueroa	Implemented comments from Q.A.
0.97	9-02-2010	Mari Carmen Suárez-Figueroa	Implemented comments from Q.A.
0.98	10-02-2010	Mari Carmen Suárez-Figueroa	Implemented comments from Q.A.
0.99	11-02-2010	Mari Carmen Suárez-Figueroa	General revision
1.00	12-02-2010	Mari Carmen Suárez-Figueroa	General revision

Executive Summary

The main goal of the current deliverable (D5.4.3) is to present the following contributions:

- Methodological guidelines proposed for reusing ontology modules (Chapter 3).
- Methodological guidelines proposed for mapping ontologies (Chapter 4).
- Methodological guidelines proposed for merging ontologies (Chapter 5).
- Methodological guidelines proposed for combining different types of knowledge resources during the ontology network development (Chapter 6).

Table of Contents

NeOn Consortium 2

Change Log 3

Executive Summary..4

Table of Contents..5

List of Tables..5

List of Figures 6

1. Introduction..7
1.1. Deliverable Main Goals and Contributions..8
1.2. Deliverable Structure..9

2. NeOn Methodology for Building Ontology Networks 10
2.1. Overview of the Scenarios for Building Ontology Networks..10

3. Ontology Module Reuse..14
3.1. State of the Art 16
3.2. Guidelines Proposed for Ontology Module Reuse 17
3.3. Example 25

4. Ontology Mapping 33

5. Ontology Merging 36
5.1. State of the Art 36
5.2. Guidelines Proposed for Ontology Merging 37

6. Combining Different Types of Knowledge Resources 42
6.1. ICPS Ontology Network Development Experience..42
6.1.1. Quick search for knowledge resources 43
6.1.2. Identification of scenarios executed 46
6.1.3. Development of the upper-level conceptualization model..47
6.1.4. Overview of the remaining processes and activities of the ICPS ontology development 51
6.2. Guidelines Proposed for Combining Different Types of Knowledge Resources 53
6.2.1. Ontology network development by combining ontological resources, non-ontological resources and ODPs 53
6.2.2. Sequence proposed to obtaing the conceptualization products..56
6.2.3. Preliminary guidelines proposed to select knowledge resources..56

7. Conclusions and Future Work..58

List of Tables

Table 1. Filling card for the process Ontology Module Reuse 18

Table 2. A hypothetical example of a table for the assessment of ontology modules 24

Table 3. A hypothetical example of a table for the selection of ontology modules 25

Table 4. Ontologies selected to be modularized in the mIO! ontology network 27

Table 5. Measurements from the extracted modules.. 28

Table 6. Ontology module assessment table for the Enviroment subdomain 29
Table 7. Ontology module assessment table for the Location subdomain 29
Table 8. Ontology module assessment table for the Service subdomain 30
Table 9. Ontology module assessment table for the Network subdomain 30
Table 10. Selection table for the Environment subdomain 31
Table 11. Selection table for the Network subdomain 31
Table 12. Filling card for the activity Ontology Merging 38
Table 13. Correspondence among knowledge areas identified by the ontology developers and terms of the ICPS framework 48

List of Figures

Figure 1. Scenarios for building ontology networks 11
Figure 2. Different types of ontological resource reuse 15
Figure 3. Activities for reusing ontology modules 19
Figure 4. Cases of Ontology Modularization... 22
Figure 5. Upper-level conceptual model of the mIO! ontology network 32
Figure 6. Tasks for Ontology Mapping 34
Figure 7. Tasks for Ontology Merging... 39
Figure 8. Excerpt of Competency Questions (CQs) about Falls 45
Figure 9. Excerpt of Competency Questions (CQs) about Pressure Ulcer.................................... 45
Figure 10. Upper-level conceptual model of Falls ontology [Montiel-Ponsoda et al., 2009b] 49
Figure 11. Upper-level conceptual model of Pressure Ulcer ontology [Montiel-Ponsoda et al., 2009c] 50
Figure 12. Combination of scenarios in the ICPS ontology development 51
Figure 13. Overview of the steps to develop an ontology network by combining the reuse of several types of knowledge resources 55

1. Introduction

The development of large-scale semantic applications in the near future will be characterized by the use of a very large number of ontologies embedded in ontology networks. These ontologies will be developed collaboratively by distributed teams following a specific methodology.

Thus, in the context of WP5, we are creating the *NeOn Methodology*, a methodology that supports the collaborative aspects of ontology development and the reuse and dynamic evolution of networked ontologies in distributed environments, in which contextual information is introduced by developers (domain experts, ontology practitioners) at different stages of the ontology development process.

The NeOn Methodology for building collaboratively ontology networks includes methodological guidelines and proposes methods, techniques and tools for carrying out different processes and activities, defined in the NeOn Glossary of Processes and Activities, during the ontology network life cycle.

The first version of the NeOn Methodology was presented in D5.4.1 [Suárez-Figueroa et al., 2008a] and included methodological guidelines for the following processes and activities: ontology requirements specification; reusing and re-engineering non-ontological resources; reusing ontological resources (focused on general or common ontologies, domain ontologies as a whole, and ontology statements); and reusing ontology design patterns by inexperienced users.

The second version of the NeOn Methodology, included in D5.4.2 [Suárez-Figueroa et al., 2009], can be seen as a revision and an extension of the first version. This second version presented methodological guidelines for the following processes and activities: reusing and re-engineering non-ontological resources; reusing ontology design patterns; modularizing existing ontologies; evaluating ontology (networks); evolving ontologies; and localizing ontologies.

The third version of the NeOn Methodology (included in this deliverable) is an extension of the previous versions; thus, processes and activities not covered in previous versions are covered in D5.4.3 by means of providing methodological guidelines for their execution.

In this version we still continue with the idea of reusing, as much as possible, knowledge resources (ontologies, ontology modules, ontology statements and ontology design patterns) as well as non-ontological resources (thesauri, lexicons, databases, UML diagrams, and classification schemas built by others that already have some degree of consensus). This reuse allows speeding up the ontology (network) development process, saving time and money, and promoting the application of good practices.

Thus, in this document we aim to complete the guidelines for reusing ontological resources provided in D5.4.1 [Suárez-Figueroa et al., 2008a] by means of proposing methodological guidelines for reusing ontology modules. Additionally, we propose methodological guidelines for mapping ontologies, merging ontologies, and combining different types of knowledge resources (non-ontological resources, ontological resources, and ontology design patterns) in order to build ontology networks.

1.1. Deliverable Main Goals and Contributions

The main goal of this deliverable is to present the third version of the NeOn Methodology for building networks of ontologies.

The principles that guide the construction of such a methodology, as presented in deliverable D5.4.1 [Suárez-Figueroa et al., 2008a] are

a. The methodology should be general enough so that it should help software developers and ontology practitioners to build networks of ontologies with the NeOn Toolkit and with other widely used platforms such as Protégé or Top Braid Composer.

b. The methodology should define each process or activity precisely; state clearly its purpose, inputs and outputs, the actors involved, and when its execution is more convenient. Additionally, it should propose the methods, techniques and tools to be used for executing each process or activity.

c. The methodology should facilitate a prompt assimilation by software developers and ontology practitioners; thus, the methodology we present here is explained in a clear and prescriptive way and includes examples on how to use the methodology in different use cases.

The scope of this deliverable (D5.4.3) is limited to providing methodological guidelines for the following processes and activities:

- *Ontology Module Reuse*. As already mentioned in the NeOn Methodology, the reuse approach is related to reusing knowledge resources (ontological and non-ontological) and, therefore, should be taken into account in the ontology network development.

  Some methodological guidelines (a) for reusing ontological resources at different levels of granularity (general or common ontologies[1] as a whole, domain ontologies as a whole, and ontology statements) and (b) for reusing and re-engineering non-ontological resources have been provided in previous versions of the NeOn Methodology. The missing point at this stage is the creation of methodological guidelines for reusing only one part or *module*[2] of an ontology that is relevant for the ontology development. For example, when building an ontology on lung cancer it is not necessary to reuse an entire ontology about the human body, it suffices to reuse a module describing concepts related to the lung.

- *Ontology Mapping*. Relationships between ontologies are the basis of networked ontologies. Methodologically, it is worthwhile to express relations between ontologies since this facilitates (a) working with small and self sufficient modules rather than with monolithic ontologies and (b) expressing the links between two versions of the same ontology, thus allowing the upgrade of data from one ontology to another; it also makes easier to put back an ontology in the context of an upper-level ontology so as to better handle other ontologies. In networked ontologies, the relationships between ontologies are as important as the ontologies themselves. Yet, very little support exists for the activity of establishing such relationships at the methodological or at the tool level.

- *Ontology Merging*. This activity is needed when several ontological resources in the same domain can be selected for reuse and when the ontology developer wishes to create a new ontological resource from two or more, possibly overlapping, source ontological resources. However, the paradox is that methodological guidelines for merging ontologies do not yet exist; and the methods that can be used for that purpose do not cover all the activity ranging from original ontologies to the final merged ontology. Current methods just solve specific problems based on algorithmic solutions. In this deliverable we present MERGETHODOLOGY, a

---

[1] A general or common ontology represents knowledge reusable in different domains.

[2] We consider a module [d'Aquin et al., 2007] as a part of the domain ontology that defines the relevant set of terms.

methodology developed with the OEG (Ontological Engineering Group at UPM) experience. This methodology describes precisely the merging process and proposes some solutions to solve some specific problems.

- *Combination of different types of knowledge resources*. When developing ontologies by means of reusing existing knowledge resources as proposed in the NeOn Methodology, the ontology development team can find diverse situations that can involve different decisions. One important decision is how to perform the combination among different types of knowledge resources for building the ontology network. Methodological guidelines to support ontology developers in such an action are needed.

1.2. Deliverable Structure

The deliverable is structured as follows:

- Chapter 2 presents an overview of the different scenarios for building ontology networks identified in the NeOn Methodology.
- Chapter 3 describes the methodological guidelines proposed for carrying out the ontology module reuse process, and includes an example on how to carry out this process in the mIO project[3].
- Chapter 4 tackles the methodological guidelines proposed for carrying out the ontology mapping activity.
- Chapter 5 deals with the methodological guidelines proposed for carrying out the ontology merging activity.
- Chapter 6 provides an example from the ICPS project[4] on how to combine different types of knowledge resources within an ontology network development; it also includes the methodological guidelines proposed for selecting different types of knowledge resources to be reused and combined during the ontology network development.
- Chapter 7 presents the conclusions and future work.

[3] mIO! Tecnologías para prestar servicios en movilidad en el futuro universo inteligente (CENIT-2008-1019). http://www.cenitmio.es/

[4] ICPS (International Classification for Patient Safety). http://www.who.int/patientsafety/implementation/taxonomy/en/

2. NeOn Methodology for Building Ontology Networks

The development of ontologies in different national and international projects has revealed that different alternative ways or possibilities are available to build ontologies [Suárez-Figueroa et al., 2008a]. Thus, it is not premature to affirm that a new ontology development paradigm is now starting, whose emphasis is on the reuse and, possibly, subsequent re-engineering of knowledge resources, on the collaborative and argumentative ontology development, and on the building of ontology networks, as opposed to custom-building new ontologies from scratch.

Taking into account such a paradigm, we are creating the *NeOn Methodology for building ontology networks*, a scenario-based methodology. This methodology identifies and defines a set of nine scenarios for building ontologies and ontology networks and provides prescriptive guidelines for carrying out different processes and activities potentially involved in the ontology development.

The first version of the methodology was published in D5.4.1 [Suárez-Figueroa et al., 2008a] and provided the following:

- Guidelines for carrying out the ontology requirements specification activity.
- Guidelines for reusing and re-engineering non-ontological resources.
- Guidelines for reusing ontological resources, specifically for reusing general or common ontologies, domain ontologies as a whole, and ontology statements.
- Guidelines for reusing ontology design patterns by inexperienced users.

Additionally, in D5.3.2 [Suárez-Figueroa et al., 2008b] guidelines for scheduling ontology development projects are also included.

The second version of the NeOn Methodology included in D5.4.2 [Suárez-Figueroa et al., 2009] is an extension of the previous version and presented the following:

- Update of the guidelines for reusing and re-engineering non-ontological resources.
- Guidelines for reusing ontology design patterns in general; and guidelines for reusing content ontology design patterns.
- Guidelines for modularizing existing ontologies.
- Guidelines for the ontology (network) evaluation activity.
- Guidelines for the ontology evolution activity and for supporting ontology engineers and domain experts in exploiting tools that facilitate the evolution of their ontologies.
- Guidelines for localizing ontologies.

The third version of the methodology included in this deliverable presents guidelines for a) reusing ontology modules, b) mapping ontologies, c) merging ontologies, and d) combining non-ontological resources, ontological resources, and ontology design patterns during the ontology network development.

Finally, this chapter briefly includes an overview of the set of 9 scenarios for building ontologies and ontology networks.

2.1. Overview of the Scenarios for Building Ontology Networks

As already mentioned, one of the key elements in the NeOn Methodology framework is the set of 9 scenarios identified for building ontologies and ontology networks. This set of scenarios is shown in Figure 1. The scenarios proposed within the NeOn Methodology are flexible because they can combine among them, unlike the scenarios for building ontologies that are presented in the most well-known ontology engineering methodologies, which are very rigid.

In this section we provide a summary of the 9 scenarios identified in the context of the NeOn Methodology for building ontology networks.

Figure 1 presents the set of 9 scenarios for building ontologies and ontology networks. Directed arrows with numbered circles associated represent the different scenarios. Each scenario is decomposed into different processes or activities that are represented with coloured circles or with rounded boxes. The processes and activities are defined in the NeOn Glossary of Processes and Activities [Suárez-Figueroa and Gómez-Pérez, 2008]. Figure 1 also shows (as dotted boxes) the knowledge resources to be reused and possible outputs (implemented ontology networks and alignments) that result from the execution of some of the presented scenarios.

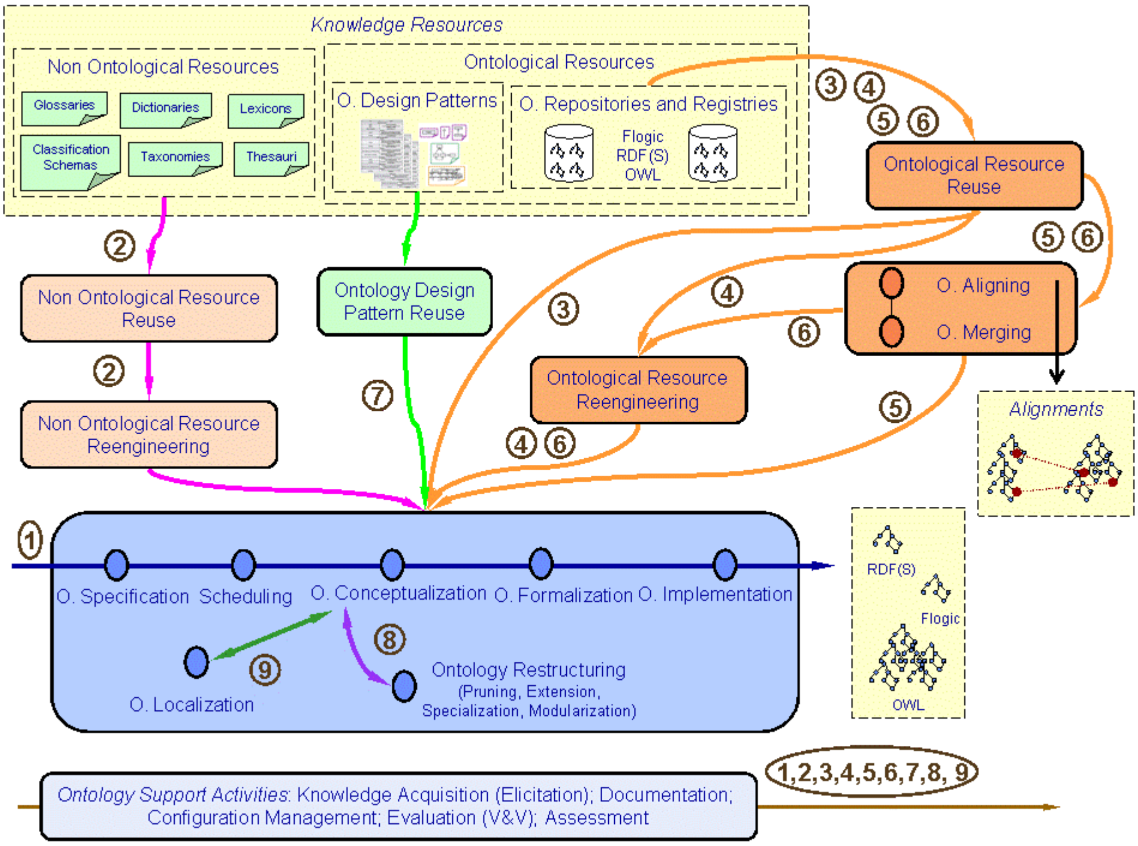

Figure 1. Scenarios for building ontology networks

The most common scenarios[5] that may arise during the ontology development are the following:

- *Scenario 1: From specification to implementation*. The ontology network is developed from scratch, that is, without reusing existing knowledge resources. Ontology developers should specify the requirements that the ontology should fulfil, by means of the ontology requirements specification activity. The objective of this activity is to output the ontology requirements specification document (ORSD). After this activity and using as input the terms appearing in the ORSD, it is advisory to carry out a quick[6] search for the potential knowledge resources to

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

[6] Quick search for knowing which types of resources are available for a possible reuse during the ontology network development; a detailed search for knowledge resources should be carried out later on in the different scenarios that involves knowledge resource reuse.

be reused. The search results allow knowing which types of resources are available for a possible reuse during the ontology network development. Then, the scheduling activity must be carried out, using the ORSD and the results of such a quick search. After the scheduling activity, the ontology developers should follow the plan established in the scheduling activity.

- *Scenario 2: Reusing and re-engineering non-ontological resources (NORs)*. Ontology developers should carry out the non-ontological resource reuse process for deciding, according to the requirements in the ORSD, which NORs can be reused to build the ontology network. Then, the NORs selected should be re-engineered into ontologies.
- *Scenario 3: Reusing ontological resources*. Ontology developers use ontological resources (ontologies as a whole, ontology modules[7], and/or ontology statements[8]).
- *Scenario 4: Reusing and re-engineering ontological resources*. Ontology developers reuse and re-engineer ontological resources.
- *Scenario 5: Reusing and merging ontological resources*. This scenario arises only in those cases where several ontological resources in the same domain are selected for reuse and when ontology developers wish to create a new ontological resource from two or more ontological resources.
- *Scenario 6: Reusing, merging and re-engineering ontological resources*. Ontology developers reuse, merge, and re-engineer ontological resources in the ontology network building. This scenario is similar to Scenario 5; however, here developers decide not to use the set of merged resources such as it is, but to re-engineer it.
- *Scenario 7*: Reusing ontology design patterns (ODPs). Ontology developers access repositories[9] to reuse ODPs.
- *Scenario 8: Restructuring ontological resources*. Ontology developers restructure (modularize, prune, extend, and/or specialize) the ontological resources to be integrated in the ontology network being built.
- *Scenario 9: Localizing ontological resources*. Ontology developers adapt an ontology to other languages and culture communities, obtaining a multilingual ontology.

The activities of knowledge acquisition and elicitation, documentation, configuration management, evaluation and assessment should be performed during the whole ontology development.

From this set of scenarios, we can say that Scenario 1 is the most typical for building ontology networks without reusing existing knowledge resources. However, as already mentioned, more and more ontology developers build ontology networks by reusing knowledge resources (ontological and non-ontological). For this reason, the NeOn Methodology differentiates scenarios that involve the reuse of ontological resources from those that involve the reuse and re-engineering of non-ontological resources.

It is worth mentioning that the aforementioned scenarios can be combined in different ways, and that any combination of scenarios should include Scenario 1 because this scenario is made up of the core activities that have to be performed in any ontology development. In fact, as Figure 1 shows, the results of any other scenario should be integrated in the corresponding activity of Scenario 1.

Although we think this set of scenarios covers the most plausible ways for building ontology networks, it can not be considered exhaustive.

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

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

[9] http://ontologydesignpatterns.org

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

Additionally, in this deliverable we present guidelines for realizing the processes and activities involved in Scenario 3 (ontology module reuse) and in Scenario 5 (ontology mapping and merging).

3. Ontology Module Reuse

Ontology Module Reuse refers to the process of employing available ontology modules for the solution to different problems [Suárez-Figueroa and Gómez-Pérez, 2008].

Ontology Module Reuse belongs to the development scenario called Building Ontology Networks by Reusing Ontological Resources (Scenario 3 in Figure 1), identified in the NeOn Methodology [Suárez-Figueroa et al., 2008a]. In this scenario, software developers and ontology practitioners should analyse whether ontological resources (e.g., ontologies as a whole, ontology modules and/or ontology statements) can be reused to build an ontology network. The underlying principle here is that reusing existing ontological resources reduces time and costs associated to the ontology development.

The reuse of ontological resources is encouraged by a recent increase in the number of online available ontologies, ontology libraries and repositories[10].

The ontological resource reuse process is often influenced by the type of ontology to be reused, as Figure 2 shows. Several types of ontologies exist:

- General [van Heijst et al., 1997] or common ontologies [Mizoguchi et al., 1995] provide conceptualization of generic topics such as time, space, and mereology, and represent knowledge reusable in different domains. They are usually based on well studied theories[11]: mereology, which formalizes parthood relation; topology, which formalizes connection relation; time theories, which formalize terms like *time interval*, *time point*, etc. Given the generality of the topic described, it is common to have several ontologies on the same topic, each of them taking a different standpoint in the conceptual model. For example, in the case of the time topic one ontology can model a particular temporal point, whereas other ontology models a temporal interval. When reusing one of these ontologies, the ontology engineer needs to be aware of the different views and assumptions the ontology relies on. Guidelines for reusing general or common ontologies are provided in [Suárez-Figueroa et al., 2008a].
- Domain ontologies provide knowledge of a specific domain, such as medicine, pharmacy, fisheries, etc. These ontologies can be helpful in cases when domain ontology in the same domain is being built.

The reuse of large ontologies (such as WordNet[12] and the NCI – National Cancer Institute-ontology [Golbeck et al., 2003]) is complex because they contain a big amount of knowledge that may not be needed when developing a particular ontology. Sometimes, the reuse requires retrieving bits of knowledge (e.g., modules, statements) to be integrated in the new ontology being built rather than reusing entire ontologies [d'Aquin et al., 2007b].

For this reason, we also distinguish different levels of granularity in the reuse of ontologies, as shown in Figure 2.

- Ontologies can be reused *as a whole* if they closely meet the expectations and the needs of the ontology engineer. Methodological guidelines for reusing ontologies as a whole are provided in [Suárez-Figueroa et al., 2008a].

[10] See for example a list of novel ontology search engines described at: http://esw.w3.org/topic/TaskForces/CommunityProjects/LinkingOpenData/SemanticWebSearchEngines.

[11] A theory is considered here as a system of definitions, axioms and theorems that can be formal, semi-formal or informally represented.

[12] http://wordnet.princeton.edu/wordnet/documentation/

- In certain cases, only one part or *module*[13] of an ontology is relevant for reuse. For example, when building an ontology about lung cancer, one does not need to reuse an entire ontology about the human body; it suffices to reuse a module describing concepts related to the lung. Guidelines for reusing ontology modules are provided in Section 3.2.
- In other cases, only some knowledge components (the description of a particular entity, the branch in the taxonomic hierarchy in which an entity appears, or entity neighbourhoods in the ontology) from the ontology are relevant for the development needs. In these cases, the reuse of ontological knowledge is performed at the *statement* level, allowing the ontology developer maximum control of the material being reused. Methodological guidelines for reusing ontology statements[14] are provided in [Suárez-Figueroa et al., 2008a].

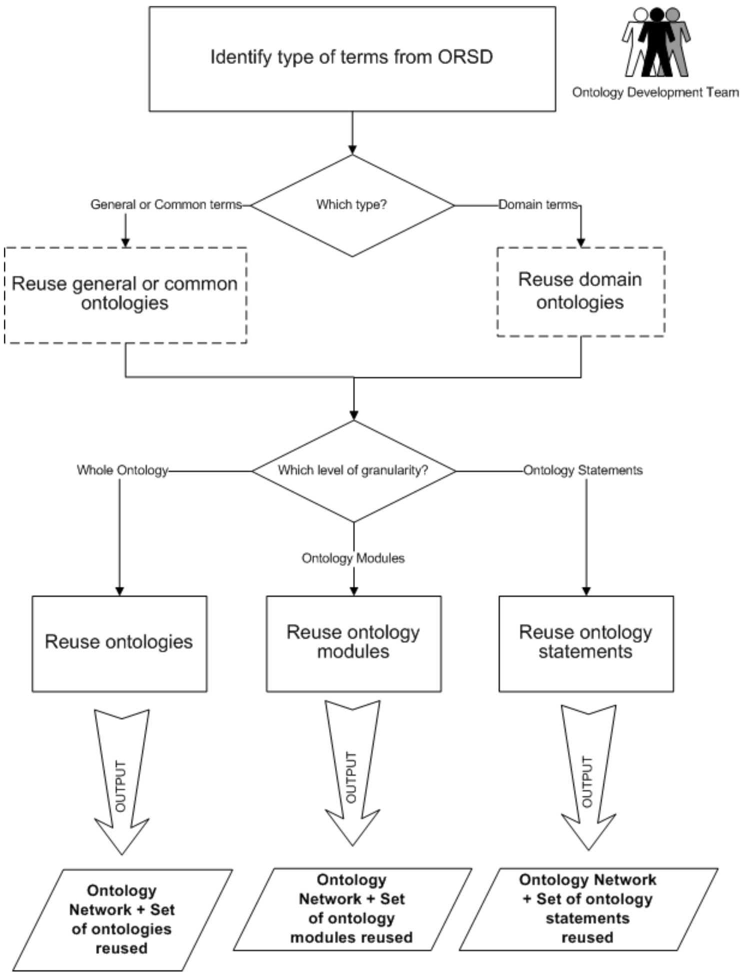

Figure 2. Different types of ontological resource reuse

It is also worth mentioning another type of ontological resource, the Ontology Design Patterns (ODPs), which can be also reused for developing ontology networks. However, the reuse of ODPs is treated in a separate scenario of the NeOn Methodology (Scenario 7 in Figure 1). Guidelines for reusing ODPs are provided in [Suárez-Figueroa et al., 2009].

In this chapter we provide a brief introduction to the existing techniques and tools for ontology module reuse. We also propose the NeOn methodological guidelines for carrying out the reuse of ontology modules. It should be noted that that these guidelines for ontology module reuse only

[13] We consider a module [d'Aquin et al., 2007] as a part of the ontology that defines the relevant set of terms.

[14] An ontology statement (or triple) contains the following three components: *subject*, *predicate*, and *object*.

consider reusing modules that come from domain ontologies; the modules coming from general or common ontologies will be treated in future work. Additionally, in this chapter, we include an example on how to use the guidelines proposed in the context of mIO! Spanish project.

3.1. State of the Art

In [Suárez-Figueroa et al., 2008a] we can learn about a number of available methodological guidelines with which to reuse a) ontologies as a whole, both general or common ontologies and domain ontologies, and b) ontology statements. However, it seems that no methodological guidelines exist that help ontology developers to reuse ontology modules.

In the field of ontology modules, researchers have mainly focused their efforts on (a) creating modularization techniques to extract modules from an ontology [Doran, 2006; Cuenca et al., 2006; Bezerra et al., 2009], or (b) creating techniques to extract modules and to create partitions of an ontology [d'Aquin, et al., 2009; d'Aquin, et al., 2007] instead of developing methodological guidelines for reusing ontology modules. Thus, only few approaches propose a methodology to reuse ontology modules. Here we provide a brief summary of some of them.

In [Jimenez-Ruiz et al., 2008a] the authors propose a working cycle based on two phases to extract modules from ontologies and import them into the ontology being developed. This process ensures the safety and economic reuse of ontologies in a logical fashion as defined in [Jimenez-Ruiz et al., 2008a], but the modules obtained are not checked against the requirements of the ontology being developed. Moreover, a logical-based methodology may not be a user-friendly approach for non-ontology experts.

A more elaborated methodology is proposed in [Doran et al., 2007]. Here, the ontology module reuse is discomposed in the following steps: (1) defining competency of the module, (2) ontology evaluation, (3) ontology selection, (4) ontology translation, (5) extracting the module, (6) checking competency, (7) refining, and (8) integrating. While this work presents a sequence of reasonable steps to reuse an ontology module, it only provides weak criteria to use in these steps, such as ontology evaluation and ontology selection. Furthermore, as in the previous approach, it only considers the ontology extraction within the ontology modularization, instead of considering ontology extraction and ontology partitioning as parts of ontology modularization.

As for the tools to perform the reuse of ontology modules, we can mention here the following ones:

- There is a plug-in for Protégé 4 called ProSé[15] [Jimenez-Ruiz et al., 2008b] that implements the approach presented in [Jimenez-Ruiz et al., 2008a].
- The algorithm presented in [Doran et al., 2007] has been implemented in a standalone tool called ModTool[16] [Doran, 2006] that provides a graphical user interface to generate ontology modules.
- ModOnto is a tool that implements a modularization approach based on object-oriented components [Bezerra et al., 2008].
- There is two NeOn Toolkit plug-in[17] called Ontology Module Extraction and Ontology Partitioning [d'Aquin et al., 2008] that allows respectively extracting a module from an ontology and dividing an ontology into modules.

[15] http://krono.act.uji.es/people/Ernesto/safety-ontology-reuse

[16] http://www.csc.liv.ac.uk/~pdoran/modtool/index.html

[17] http://neon-toolkit.org/wiki/Neon_Plug-ins

3.2. Guidelines Proposed for Ontology Module Reuse

As mentioned before, the goal of ontology module reuse is to find and select one or several ontology modules related to the domain, or to a subdomain, of the ontology being developed in order to be used in such an ontology network development. The output of this reuse process is a set of ontology modules integrated in the ontology network.

Ontology developers need methodological guidelines that help them to reuse small parts of whole ontologies, that is, to reuse ontology modules. For this purpose, within the NeOn Methodology we have devised a filling card for the process of ontology module reuse, and a set of detailed activities for this process in the form of a workflow.

Note that, as opposed to a single, monolithic ontology, an ontology network is essentially a modular ontology, made of components (the individual ontologies) interacting with each other in a particular context. The approach presented here is applied on individual ontologies (possibly networked) to create either networks of ontologies or elements for networks of ontologies.

In the framework of the NeOn Methodology for building ontology networks, we propose the filling card for the process of reusing ontology modules, presented in Table 1, which includes the definition, goal, input, output, actors involved in the process and time when the process should be carried out.

Ontology Module Reuse	
Definition	
Ontology Module Reuse refers to the process of using available ontology modules for the solution to different problems.	
Goal	
The goal of this process is to find and select ontology modules to be integrated in the ontology network being developed.	
Input	*Output*
The Ontology Requirements Specification Document (ORSD).	A set of ontology modules integrated in the ontology network being developed.
Who	
Software developers and ontology practitioners.	
When	
The process of reusing ontology modules should be carried out after the ontology requirements specification activity.	

Table 1. Filling card for the process Ontology Module Reuse

The activities for reusing ontology modules can be seen in Figure 3.

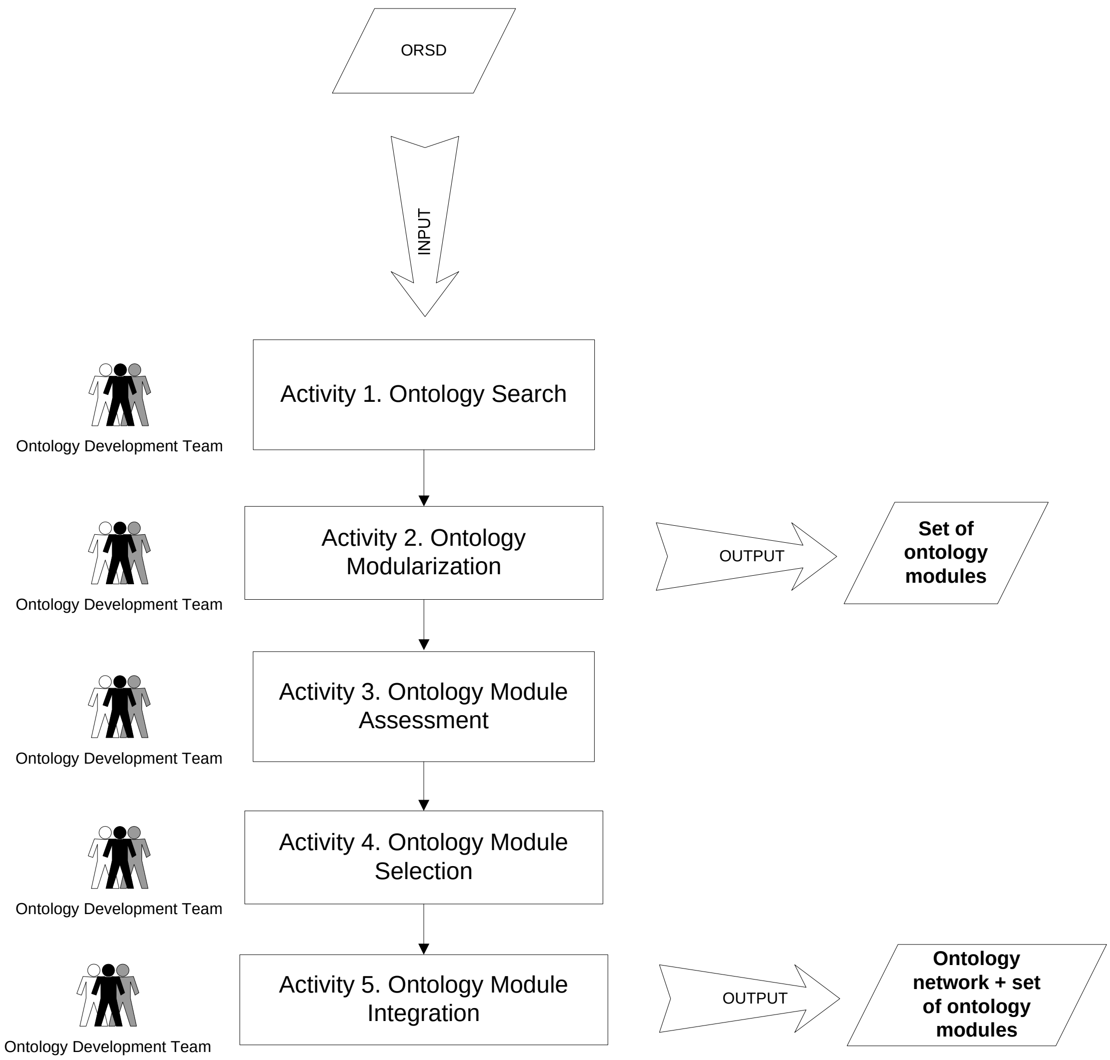

Figure 3. Activities for reusing ontology modules

The activities of the process for reusing ontology modules are explained as follows:

Activity 1. Ontology Search.

As defined in [Suárez-Figueroa et al., 2008a], the objective of this activity is to search in libraries, repositories and registries for candidate domain ontologies that could satisfy the needs of the ontology network being developed. This activity is carried out by the ontology development team, taking as input the Ontology Requirements Specification Document (ORSD) and, specifically, those terms that have a high frequency in the ORSD, using tools as Watson[18], Oyster[19], Swoogle[20], etc.

The activity output is a set of candidate domain ontologies that could be implemented in different languages.

[18] http://watson.kmi.open.ac.uk/WatsonWUI/

[19] http://oyster.ontoware.org/

[20] http://swoogle.umbc.edu/

Because of the increasing number of on-line ontologies available, after an ontology search using the main terms in the ORSD, it is probable to obtain a large amount of candidate ontologies. After obtaining the candidate ontologies, we have the following possibilities:

a. To assess and select the most useful ontologies (following the guidelines included in [Suárez-Figueroa et al., 2008a]), and then to carry out the modularization activity but only with the ontologies selected.

b. To modularize all the candidate ontologies.

Both options involve a time consuming process; additionally, in the second one it might not be very useful to modularize all the ontologies obtained and then to assess and select those modules that are more suitable to be reused.

Bearing this in mind, we propose a hybrid approach that consists in (a) carrying out a quick ontology selection among the ontologies retrieved from the web, and (b) doing a detailed assessment (described in Activity 3) of those ontology modules obtained through the modularization activity (described in Activity 2), and then making a selection (described in Activity 4) of the most appropriate modules.

To carry out a quick selection of the ontologies to be modularized in Activity 2, we should take into account the following criteria:

- The compatibility between the file that contains the ontology and (a) the ontology editor to be used in the ontology development (or the most common ones) and/or (b) the modularization tool to be used in the modularization activity. In case the file and the tools are incompatible, ontology developers can reject the ontology or make some re-engineering activities in the file so as to make them compatible. For example:
  - If the ontology editor informs of which line in the file causes an error, the ontology developers can remove or edit this part of the ontology.
  - If the ontology editor only accepts OWL-DL ontologies, the ontology developers can remove all the OWL-Full sentences that appear in the ontology.
- The scope of the ontology. An ontology can match the ORSD main terms only at the morphological level. For example, if in the ORSD appears the term "Infrastructure", you can obtain an ontology about medicine that has a concept named "Infrastructure_Activities" while you look for an ontology that includes the "Infrastructure" concept and related ones.
- The amount of information suitable to be reused. We can find an ontology, large or small, that matches the domain of the ontology being developed but that contains only little knowledge to be reused. It may not be worth performing an entire ontology module reuse process just to reuse only a few bits of knowledge. In this case, we could optionally carry out an ontology statements reuse process (see methodological guidelines in [Suárez-Figueroa et al., 2008a]). In order to minimize the effort, this criterion should be applied by making a general analysis of the ontology and identifying the main concepts or parts that could be reused. The tasks of checking competency questions (CQs) in the ORSD and perfoming a detail terminological study with the ontology could be carried out in Activity 3 if necessary.
- Ontology usability from a human point of view. In some ontologies, concepts have no names understandable to humans, for example, a concept name can be PATO%3A0000983.

Activity 2. Ontology Modularization.

The goal of ontology modularization is to obtain a module or a set of modules from an ontology that fit the requirements of a particular application or a particular scenario [Suárez-Figueroa et al., 2009]. The ontology development team carries out this activity taking as input the ORSD, particularly those terms that have a high frequency in the ORSD, and the ontologies selected in

Activity 1. Some of the tools that can be used to perform this activity are the Ontology Module Extraction and Ontology Partitioning plug-ins for the NeOn Toolkit, ProSé, etc.

The activity output is a module or a set of modules extracted from the ontologies obtained in Activity 1. It can be added that in practice, ontology modules are themselves ontologies [Suárez-Figueroa et al., 2009].

The tasks to carry out the ontology modularization activity, as well as methodological guidelines for this activity, are included in [Suárez-Figueroa et al., 2009].

In this deliverable we include more guidelines to perform the ontology modularization activity. Specifically, we provide the description of a set of basic situations that can be combined among them. Such situations depend on two aspects: (1) how many ontologies have been obtained in Activity 1 and (2) how many ontology modules are obtained as output of this Activity 2. Ontology developers should choose a case or set of cases among those represented in Figure 4, and combine the cases selected if needed. Each case is described as follows:

- Case (a): There is only one ontology modularization process that takes as input one ontology and obtains as output one ontology module. Ontology developers may choose this case if they are interested in a part of the ontology.
- Case (b): The ontology modularization input is an ontology, and the output is a set of modules. Ontology developers may choose this case if they are interested in using independently several parts of the ontology. Depending on the type of modularization approach [Suárez-Figueroa et al., 2009] one or several modularization process can be carried out.
  - If the modularization approach selected is “module partition” there will be only one ontology modularization process whose output is a set of modules.
  - If the modularization approach selected is “module extraction” there will be an ontology modularization process per module obtained.
- Case (c): The ontology modularization input is a set of ontologies, and the output is a module. The ontology developers may choose this case if they are interested in knowledge that can be formed by aggregating knowledge from the set of ontologies. This case is divided into two more detailed cases.
  - Case (c.1): Here ontology developers carry out the ontology merging activity (see Chapter 5) before the ontology modularization one. In this case, ontology developers obtain first an ontology formed by all the ontologies, and after, a module extracted from this aggregated ontology. This case should be carried out when the ontology developers have some input ontologies whose domains are overlapped and when they want to obtain an ontology module as complete as possible. Thus, the ontology developers aggregate first all the knowledge from the set of input ontologies by means of the merging activity, and then they carry out the modularization activity to get the ontology module.
  - Case (c.2): The ontology developers carry out first N ontology modularization activities, one per input ontology. Then, they merge (see Chapter 5) the N modules obtained so that to obtain a more complete module. This case should be carried out when the ontology developers have large input ontologies with possibly overlapping domains and with several parts that are not interesting for the module to be obtained. In this case, the ontology developers can modularize first the input ontologies to get only the parts most interesting. Then, they can merge the ontology modules obtained to get a complete module.
- Case (d): Ontology developers may choose this case if they are interested in having several bits of knowledge formed by aggregating knowledge from different ontologies. In this situation, the input is a set of ontologies and the output is a set of ontology modules. With

the aim of obtaining some modules formed by knowledge from all the ontologies, it will be necessary to carry out the ontology merging activity (see Chapter 5) before the modularization activity. Like case (b), depending on the type of modularization approach [Suárez-Figueroa et al., 2009] one or several modularization process can be carried out.

- If the modularization approach selected is "module partition", then there will be only one ontology modularization process whose output is a set of modules.
- If the modularization approach selected is "module extraction", then there will be an ontology modularization process per module obtained.

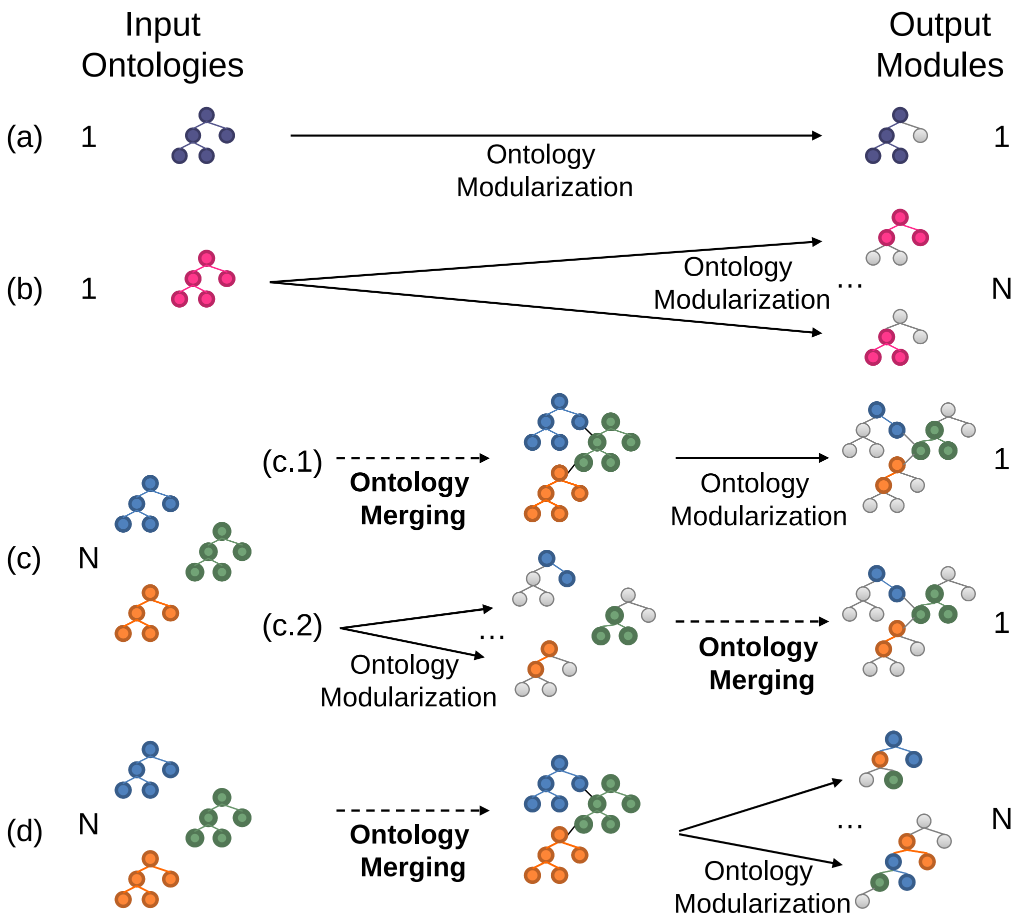

Figure 4. Cases of Ontology Modularization

As commented in Section 3.1, there are some techniques and tools to carry out the ontology modularization activity. Ontology developers can follow manually the techniques described in [Doran, 2006; Jimenez-Ruiz et al., 2008a; Cuenca et al., 2006] to extract modules from an ontology, or they can follow the techniques proposed in [d'Aquin, et al., 2009] to extract modules and to partition an ontology. Additionally, there are some automatic tools to carry out the modularization activity, for example, (a) the two NeOn Toolkit plug-in called Ontology Module Extraction and Ontology Partitioning [d'Aquin et al., 2008]; (b) ProSé [Jimenez-Ruiz et al., 2008b], the plug-in for Protégé 4; (d) ModTool [Doran, 2006]; and (d) ModOnto [Bezerra et al., 2009].

Activity 3. Ontology Module Assessment.

The goal of ontology module assessment is to find out which ontology modules, from the set of candidate ontology modules, are useful for the development of the ontology network. The ontology development team carries out this activity taking as input the set of ontology modules obtained in Activity 2 and using the following criteria (based on the criteria proposed in [Suárez-Figueroa et al., 2008a]) for deciding if a particular ontology module is useful or not.

- Checking non-functional ontology requirements established in the ORSD. Examples of requirements can be the following: terms to be used in the ontology must be taken from standards, multilinguality must be represented in the ontology to be developed, etc.
- Checking the CQs included in the ORSD against the candidate ontology modules, taking into account the terminological level. That is, the ontology development team calculates the precision and recall of the candidate ontology modules with respect to the terminology included in CQs. Precision and recall (or coverage) have been defined in [Suárez-Figueroa et al., 2008a], specifically in Section 6.5.
- Checking other considerable features. This criterion can be customized depending on important ad-hoc characteristics for each ontology development. Some examples of these features are
    - If the candidate ontology module contains some hierarchy susceptible to be reused. These hierarchies can be interesting because of their completeness or because they can be a starting point for the ontology development.
    - If the candidate ontology module contains a group of instances susceptible to be reused. This group of instances can be a set of typical instances that often appear in the domain of the ontology to be developed.
    - If the concepts contained in the ontology module are described and defined through "comment" annotations.
    - If the ontology modules or the ontology the module comes from, contain ontology design patterns.

The activity output is an assessment table that analyses each candidate ontology module following the aforementioned criteria. In the assessment table, useful ontology modules are shadowed. To decide that an ontology module is useful, it is advisable that most of the criteria related to the ontology requirements and CQs be satisfied.

Table 2 shows a hypothetical example of an assessment table with three ontology modules of devices. The example shows that in the OSRD it is established the following: OWL-DL is required as language for implementing the ontology, standard terminology is not needed in the ontology to be developed, and multilinguality is not needed in the ontology. Table 2 shows two columns shadowed (*Ontology Module 2* and *Ontology Module 3*) for the ontology modules considered useful. The other column is not shadowed because *Ontology Module 1* does not satisfy three of the aforementioned criteria (ontology language, useful hierarchies and useful instances).

		Ontology Module 1	Ontology Module 2	Ontology Module 3
Ontology Characteristics		Ontology Module Characteristics		
	From the ORSD			
O. Language	**OWL-DL**	**OWL-Full**	**OWL-DL**	**OWL-DL**
Standards	**Not needed**	**-**	**-**	**-**
Natural Language	**English**	**English**	**English**	**English**
Multilinguality	**No**	**No**	**No**	**No**
Checking CQs				
Terminological Level: Precision		**8'25%**	**23'6%**	**17'30%**
Terminological Level: Recall		**16'57%**	**20'5%**	**15'45%**
Other considerable features (importance: high ***, medium **, low*)[21]				
Useful hierarchies ***		**No**	**No**	**Yes**
Useful instances ***		**No**	**Yes**	**No**

Table 2. A hypothetical example of a table for the assessment of ontology modules

Activity 4. Ontology Module Selection.

The objective of this activity is to find out which ontology modules are the most suitable for the development of the ontology network. The ontology development team carries out this activity taking as input the useful ontology modules from the assessment table obtained in Activity 3, using some criteria for selecting the most suitable ontology modules. The criteria are explained in detail in [Suárez-Figueroa et al., 2008a] (specifically in Section 7.2) and are the following:

- Ontological Resource Understandability: check if the ontology module has good documentation.
- Ontological Resource Integration Effort: check if the estimation effort for integrating the ontology module is low and if the ontology module uses naming conventions.
- Ontological Resource Reliability: check if the ontology module is reused by other ontologies or other ontology-based projects and if the ontology module has been evaluated.

The ontologies that satisfy the larger number of criteria are selected in the selection table and appear in the shadowed columns.

The activity output is a set of domain ontologies selected from the table.

We can see in Table 3 how the two useful ontology modules from Table 2 are analysed following the aforementioned criteria; thus *Ontology Module 2* is selected.

[21] To customize this part of the table, ontology developers should add a file per each characteristic that they want to analyze into the ontology module and should set a level of importance for this characteristic (high ***, medium ** or low*).

	Ontology Module 2	Ontology Module 3
From the **ontology module**		
Ontology Module Integration Effort		
Estimated Integration Effort	**Low**	**Low**
Naming conventions	**Yes**	**No**
From the **ontology** that the module comes from		
Ontology Understandability		
Documentation	**Yes**	**No**
Ontology Reliability		
Reused by Others	**Yes**	**Not available**
Evaluated	**Yes**	**Not available**

Table 3. A hypothetical example of a table for the selection of ontology modules

Activity 5. Ontology Module Integration.

The objective of this activity is to integrate the ontology modules selected in the ontology network being developed. The ontology development team carries out this activity taking as input the set of ontology modules selected in Activity 4. For each ontology module included in the input set, the ontology development team decides one of the following three modes for the integration:

- The ontology module selected is reused as it is. Then, the ontology development team integrates the ontology module in the ontology network being developed.
- The ontology module selected is reused with significant changes (e.g., use the ontology module in a different implementation language). In this case, the ontological resource re-engineering activity should be carried out with the ontology module selected. Then, Scenario 4 (Figure 1) should be followed.
- Several ontology modules in the same domain are merged to obtain a new module. In this case, Scenario 5 or Scenario 6 (Figure 1) should be followed.

Before reusing the ontology modules selected by following any reuse mode, it is convenient to evaluate the resulting ontology through the *ontology evaluation activity* (See methodological guidelines in [Suárez-Figueroa et al., 2009]).

The activity output is an ontology network that includes the set of ontology modules selected.

3.3. Example

In this section we provide a general example of how to use the methodological guidelines proposed for the ontology module reuse. This example represents the use case of mIO![22] ontology network. mIO![23] is a Spanish project whose goal is to create technologies that provide ubiquitous services in smart environments and that can be adapted to each user and his/her context. This is done through the mobile terminal based interaction with services provided both by companies and users that create and share their own micro-services. In the framework of the mIO! project an ontology network about "Context" is being developed. Such ontology network represents

[22] http://www.cenitmio.es/

[23] mIO! Tecnologías para prestar servicios en movilidad en el futuro universo inteligente (CENIT-2008-1019).

knowledge of domains related to context such as Device, Environment, Interface, Time, Network, Service, Provider, Location, Preference/Profile/Role, User, and Source.

In this section we present the process for reusing ontology modules carried out in the mIO! project following the guidelines proposed of the NeOn Methodology.

It is worth mentioning that the process for reusing ontology modules was carried out for the following subdomains, whose development is part of the UPM work: Environment, Interface, Network, Service, Provider and Location. The rest of subdomains have not implicated any process for reusing ontology modules because the Time subdomain was developed by an ontology reuse process; the Preference/Profile/Role and User subdomains are competence of *Fundación CTIC*[24]; the Device subdomain is competence of Telvent[25]; and the Source subdomain was identified after the process for reusing ontology modules was carried out.

Activity 1. Ontology Search.

Having Watson and Swoogle as semantic search engines, we obtained several ontologies in different domains of the mIO! ontology network, concretely we had 13 ontologies for the Environment subdomain, 6 ontologies for Interface, 19 ontologies for Location, 7 ontologies for Provider, 9 ontologies for Network, and 12 ontologies for Service.

These ontologies can be modularized to be reused. After a quick overview of these ontologies, we selected those that best fit the field of knowledge to be represented for each subdomain. Most ontologies were discarded because their scopes were quite different from that of the ontology being developed, they only had a morphological match in some annotations or concepts names. Others were discarded because they had little information suitable to be reused at a glance. Some others were discarded because they had non-human understable concepts names. Moreover, a re-engineering process was carried out over “extension”, “sensor-jpa”, “ontosem” and “Profile” ontologies from Table 4 because the files that contain theses ontologies were incompatible with the ontology editor that supports the plug-in selected for the modularization activity. A small quantity of ontologies was discarded because their files could not be imported in ontology editors even after a re-engineering process. Finally, we obtained one ontology for the Environment subdomain, two ontologies for Location, two ontologies for Network and two ontologies for Service, as can be seen in Table 4.

[24] http://www.fundacionctic.org/

[25] www.telvent.com

	Subdomain			
Ontology	Environment	Location	Network	Service
CoDAMoS[26]	x			
extension[27]		x		
sensor-jpa[28]		x		
dcs[29]			x	
deliveryContext[30]			x	
ontosem[31]				x
Profile[32]				x

Table 4. Ontologies selected to be modularized in the mIO! ontology network

Activity 2. Ontology Modularization.

In the use case of mIO!, the modularization activity was aimed at facilitating the reuse of parts of some ontologies by extracting modules of the ontologies to be reused. Specifically, we had seven ontologies to be modularized and four subdomains to be developed. Bearing in mind the cases presented in Figure 4, we selected the case (a) for the seven input ontologies, and therefore, for each input ontology we obtained an ontology module.

The tool selected to carry out the modularization activity was the NeOn Toolkit plug-in for ontology modularization.

The modularization approach selected was module extraction. The modularization criterion here is mainly that the module extracted should contain ontological elements relevant to this particular topic. The modularization criterion was the same for every subdomain, although it could vary from one to another.

Considering the criteria above, we decided to apply the NeOn Toolkit plug-in for ontology module extraction [d'Aquin et al., 2008].

Throughout the modularization process, the most common and significant terms from the ORSD[33] [Cadenas, 2009] have been used as the most relevant elements in the module that we want to obtain. In this way, we have extracted modules that contain the main terms in the ORSD and their most related terms. The main terms from the ORSD per subdomain are

- **Environment**: Activity, environmental conditions, environment, humidity, infrastructure, light, noise, temperature
- **Location**: AGPS, height, town hall, bank, street, city, CoO, coordinate, building, error, static, outdoor, GPS, indoor, latitude, location, located, length, measurement, museum, country, park, sports complex, precision, restaurant, location

[26] http://www2.cs.kuleuven.be/~distrinet/projects/CoDAMoS/ontology/context.owl

[27] http://www.aktors.org/ontology/extension

[28] http://www.mindswap.org/~evren/services/sensor-jpa.daml

[29] http://ontology.ist-spice.org/mobile-ontology/0/10/dcs/0/dcs.owl

[30] http://www.w3.org/2007/uwa/ontologies/DeliveryContext.owl

[31] http://morpheus.cs.umbc.edu/aks1/ontosem.owl

[32] http://www.daml.org/services/owl-s/1.1/Profile.owl

[33] It is worth mentioning that the mIO! ORSD is still in evolution, therefore [Cadenas, 2009] may contain a non-update version of mIO! requeriments.

- **Network**: Access, scope, architecture, banda, basic, bluetooth, wired, quality, training, capacity, coverage, handover, wireless, interface jitter, mobility, nodes, operator priority, QoS, network, delay, noise, signal rate, Terminal, type, topology, Wi-Fi, ZigBee
- **Service**: Capacity, client-server, collaborative, component, context, implementation, functionality, local template, private, public, service

Finally, we have obtained the modules presented in Table 5.

	Measurement			
Extracted module file	**classes**	**relationships**	**attributes**	**instances**
CoDAMoS_Environment Module	16	9	0	0
dcs_Network Module	45	27	12	5
DeliveryContext_Network Module	58	37	16	73
extension_Location Module	8	1	2	0
ontosem_Service Module	117	0	0	0
Profile_Service Module	31	20	9	2
sensor-jpa_Location Module	4	3	1	0

Table 5. Measurements from the extracted modules

Activity 3. Ontology Module Assesment.

Once the modules have been extracted, the next activity is to find out if the set of candidate ontology modules are useful for the development of the ontology network. In the mIO! ontology module reuse process, several assessment tables have been built. Specifically, one table for each subdomain that has at least one candidate ontology module to be reused.

As can be seen in Table 6, the unique candidate ontology module to be reused when developing the Environment subdomain has been rated as useful, although it does not match all the criteria though it has the same ontology language and natural language, and a high recall with the ontology to be developed. In the case of the Location (Table 7) and Service (Table 8) subdomains, none of the candidate ontology modules has been marked as useful since all of them have a poor precision and recall, and they have not useful hierarchies or instances. Finally, in the case of the Network subdomain (Table 9), the module from the DeliveryContext Ontology has been selected as useful for the network ontology development because its useful instances. Although the module that comes from dcs ontology might seem better because of its recall, none of modules has enough high precision or recall. However, the DeliveryContext Ontology module has a set of typical instances in the domain, even though the terms do not appear in the ORSD, and therefore they do not increase the recall of the module. For the Location and Service domains, we have found that none ontology modules is useful to be reused. In this case we can consider the reuse of ontology statements [Suárez-Figueroa et al., 2008a] from the ontologies whose modules have been discarded or, if there are not other ontological or non-ontological resources suitable for reuse, to develop the ontologies from scratch.

Subodomain: environment		CoDAMoS _Environment Module
Ontology requirements		Module characteristics
	From ORSD	
O. Language	**OWL-DL**	**OWL-DL**
Standards	**Not needed**	**-**
Natural Language	**English**	**English**
Multilinguality	**No**	**No**
Checking CQs		
Terminological Level: Precision		**28%**
Terminological Level: Recall		**77'78%**
Other considerable features (importance: high ***, medium **, low*)		
Useful hierarchies ***		**No**
Useful instances ***		**No**

Table 6. Ontology module assessment table for the Environment subdomain

Subodomain: location		extension_Location Module	sensor-jpa_Location Module
Ontology requirements		Module characteristics	
	From ORSD		
O. Language	**OWL-DL**	**OWL-DL**	**OWL-DL**
Standards	**Not needed**	**-**	**-**
Natural Language	**English**	**English**	**English**
Multilinguality	**No**	**No**	**No**
Checking CQs			
Terminological Level: Precision		**36'36%**	**25%**
Terminological Level: Recall		**15'38%**	**7'69%**
Other considerable features (importance: high ***, medium **, low*)			
Useful hierarchies ***		**No**	**No**
Useful instances ***		**No**	**No**

Table 7. Ontology module assessment table for the Location subdomain

Subodomain: servicio		ontosem_Service.owl	Profile_Service.owl
Ontology requirements		Module characteristics	
	From ORSD		
Syntactic Level: O. Language	**OWL-DL**	**OWL-DL**	**OWL-DL**
Standards	**Not needed**	**-**	**-**
Natural Language	**English**	**English**	**English**
Multilinguality	**No**	**No**	**No**
Checking CQs			
Terminological Level: Precision		**3’41%**	**3’22%**
Terminological Level: Recall		**33’33%**	**16’67%**
Other considerable features (importance: high ***, medium **, low*)			
Useful hierarchies ***		**No**	**No**
Useful instances ***		**No**	**No**

Table 8. Ontology module assessment table for the Service subdomain

Subodomain: network		dcs_Network Module	DeliveryContext_Network Module
Ontology requirements		Module characteristics	
	From ORSD		
Syntactic Level: O. Language	**OWL-DL**	**OWL-Full**	**OWL-Full**
Standards	**Not needed**	**-**	**-**
Natural Language	**English**	**English**	**English**
Multilinguality	**No**	**No**	**No**
Checking CQs			
Terminological Level: Precision		**12’35%**	**3’26%**
Terminological Level: Recall		**36’67%**	**20%**
Other considerable features (importance: high ***, medium **, low*)			
Useful hierarchies ***		**No**	**No**
Useful instances ***		**No**	**Yes**

Table 9. Ontology module assessment table for the Network subdomain

Activity 4. Ontology Module Selection.

The aim of this activity is to find out which ontology modules are the most suitable for the development of the ontology network. The useful ontology modules of the assessment tables obtained in Activity 3 are taken as input.

In the mIO! ontology network case, we have rated as useful two ontology modules, one for the Environment subdomain and other for the Network subdomain, as explained in the previous activity. Although, it makes no sense to select one ontology for each subdomain from a set of one ontology for each subdomain, we performed this activity by filling in the corresponding tables.

According to the resulting tables we can have two possibilities: (a) to select the only useful ontology module for each subdomain; or (b) not to select any ontology module. In the later case, we could consider reusing statements [Suárez-Figueroa et al., 2008a] from the ontologies whose modules have not been selected or, if we do not have other ontological or non-ontological resources suitable for the reuse, to develop the ontologies from scratch.

In the case of mIO! ontology network development, once selection tables were completed (Table 10 and Table 11), we selected modules from CoDAMoS Ontology and DeliveryContext Ontology to be reused in the Environment and Network ontology development, due especially to their low estimated integration effort and to the documentation available about the ontologies from which it is derived.

Subodomain: Environment	CoDAMoS_Environment Module
From the **ontology module**	
Ontology Module Integration Effort	
Estimated Integration Effort	**Low**
Naming conventions	**Yes**
From the **ontology** that the module comes from	
Ontology Understandability	
Documentation	**Yes**
Ontology Reliability	
Reused by Others	**No** [34]
Evaluated	**Not available**

Table 10. Selection table for the Environment subdomain

Subodomain: Network	DeliveryContext_Network Module
From the **ontology module**	
Ontology Module Integration Effort	
Estimated Integration Effort	**Low**
Naming conventions	**No (Partially)**
From the **ontology** that the module comes from	
Ontology Understandability	
Documentation	**Yes**
Ontology Reliability	
Reused by Others	**Yes**
Evaluated	**Yes**

Table 11. Selection table for the Network subdomain

Activity 5. Ontology Module Integration.

In this activity the objective is to integrate the ontology modules selected from Activity 4 in the ontology network being developed.

In the case of the ontology module reused in the Environment ontology, the integration activity has been carried out by means of ontology pruning [Suárez-Figueroa et al., 2007a]. Through this

[34]https://doc.novay.nl/dsweb/Get/Document-62334/Techniques%20for%20describing%20and%20manipulating%20context%20information

activity the concepts that did not fit the subdomain requirements were removed from the ontology module, although they could fit the general ontology network requirements. For example, the concepts “Mood” and “User” were removed from the CoDAMoS_Environment_Module. Once the ontology module is pruned, it is imported in the Environment subdomain ontology to continue with its development.

In the case of the ontology module reused in the Network ontology, the integration activity has also been carried out by a re-engineering approach. In this case, it might have been necessary to carry out a rename activity, since the module only covers this requirement partially. However, after the pruning activity, the ontology module only contains concepts following the naming conventions[35] of mIO!. Once the ontology module is pruned, it is imported in the Network subdomain ontology to continue with its development.

As result of the process of reusing ontology modules, we obtained an ontology network formed by ten ontologies, two of which have an ontology module as component. Figure 5 shows the mIO! ontology network conceptual model. In that figure, the white triangles with a blue line represent the ontologies that are part of the ontology network and the green triangles represent the modules reused in the ontologies development.

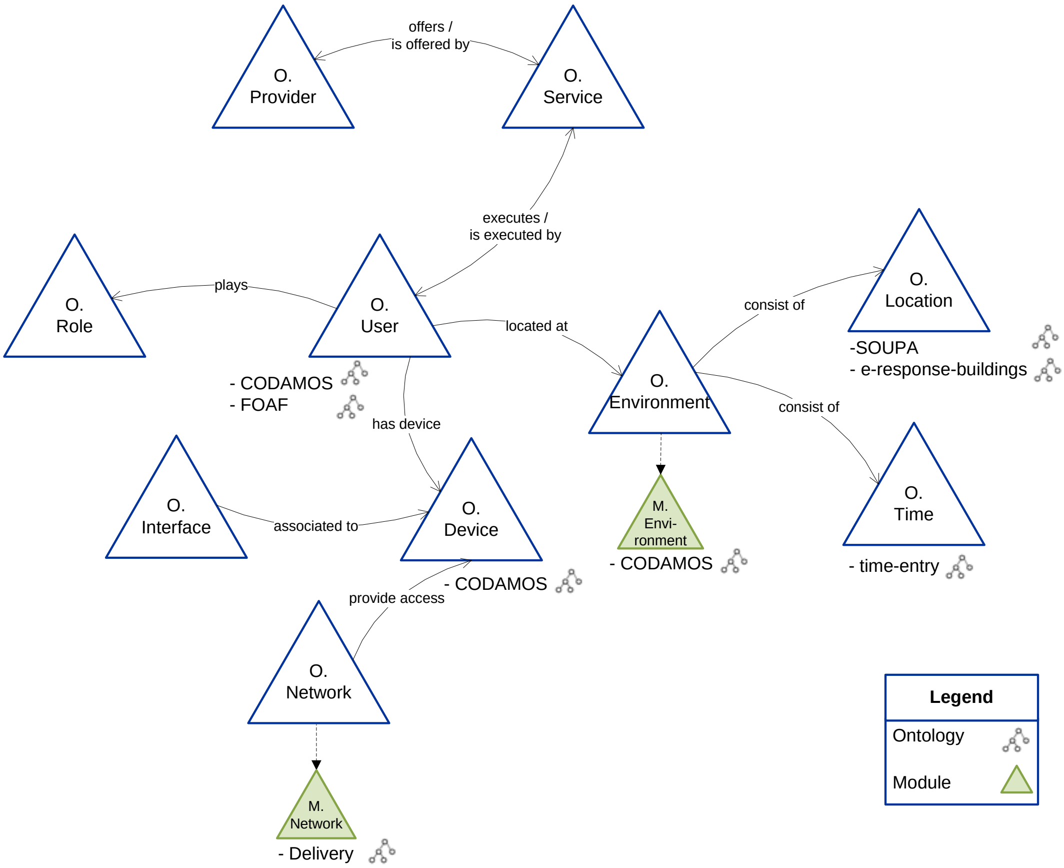

Figure 5. Upper-level conceptual model of the mIO! ontology network

[35] The naming conventions followed in mIO! are the following: (a) the first letter of each word capitalized and the rest in lower case for class names; and (b) the first word in lower case and the following words with first letter capitalized and the rest in lower case for relationships, attributes and instances.

4. Ontology Mapping

Ontology Mapping [Suárez-Figueroa and Gómez-Pérez, 2008] refers to the activity of finding the correspondences between two or more ontologies and storing/exploiting them. A synonym for this activity is Ontology Aligning.

Ontology alignments [Euzenat et al., 2008] are sets of relationships between ontology entities called correspondences. Such relations may tell that a “district” in one ontology is the same as a “kreis” in other ontology, or that “fishery” in an ontology is a subclass of “company” in another. An alignment can be used to link an ontology with its background (set it in a more general context), which is typically achieved by providing an alignment with an upper level ontology. An alignment can also be used to link an ontology with its previous versions or alternative ontologies in other applications.

Alignments can be expressed in various languages. For instance, the two relations mentioned above can be expressed in OWL through `owl:equivalentClass` and `rdfs:subClassOf`, but they can also be expressed in SKOS through `skos: exactMatch` and `skos:broaderMatch`. However, in order to avoid early committing to a particular type of usage, it is preferred to keep the alignments in a declarative language, such as the Alignment format used in the NeOn Toolkit alignment plug-in. This language allows generating the required representation (OWL, SKOS and others) when necessary.

Indeed, when the alignment is expressed in OWL, its only possible use is to “merge” two OWL ontologies, thus it cannot be used to import data from one ontology to another or to export owl:sameAs links between instances. Using a neutral and declarative representation provides the opportunity to distribute and share alignments among applications.

The activity of establishing alignments between ontologies is called ontology matching[36]. Ontology matching [Euzenat and Shvaiko, 2007] has been the focus of a great deal of attention in recent years. However, little work has been carried out on the methodological support for finding alignments. Here we provide the outline for such a methodology.

The tasks for the activity of ontology mapping can be seen in Figure 6.

[36] Ontology Matching [Suárez-Figueroa and Gómez-Pérez, 2008] refers to the activity of finding or discovering relationships or correspondences between entities of different ontologies or ontology modules. Ontology Matching can be seen as the first stage of Ontology Aligning.

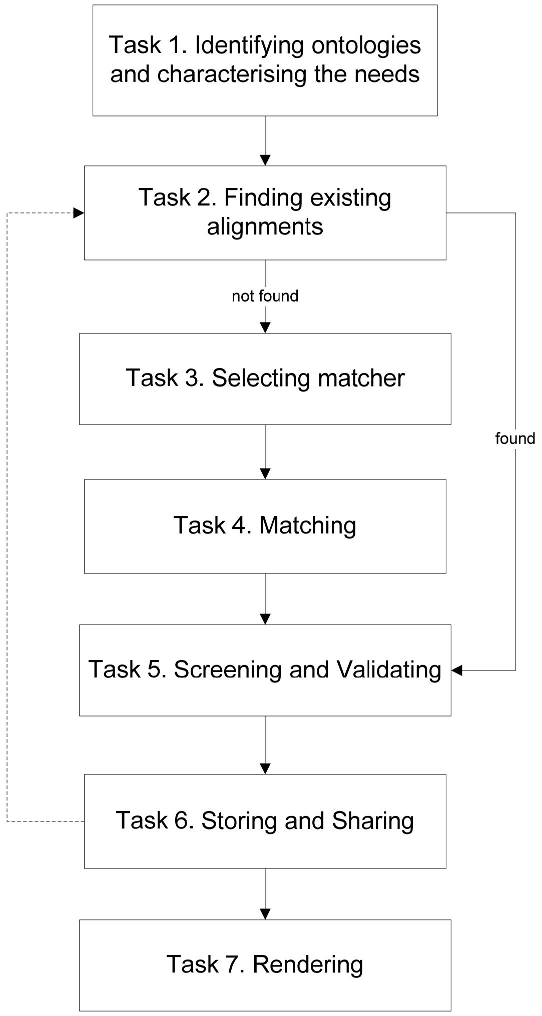

Figure 6. Tasks for Ontology Mapping

The tasks of the activity of ontology mapping are explained as follows:

The first task when searching alignments is to identify the ontologies to be matched and to characterise the alignment needed. Indeed, the type of alignment required will be different if the goal is to merge two ontologies in a knowledge-based system or to add another data source to a query mediator. In the former case, the alignment will have to be absolutely correct; otherwise the system may draw incorrect inferences. In the latter case, the lack of completeness is not a problem since other sources may return the missing answers, but relations other than equivalence are not straightforwardly used in query mediation. It is also useful to characterise the kind of ontologies by answering the following questions: Are they labelled in the same natural language? What is their expressiveness? Are individuals related to the ontologies available?

Finding existing alignments which satisfy the need of the application is the second step. Alignments may be published directly on the web or on specialised alignments servers. They ideally should come with annotations characterising their level of trustworthiness, the purpose for which they have been built, and the type of relations they use. If apparently suitable alignments are available, the user can directly go to the validation step.

Otherwise, it will be necessary to build a new alignment from the ontologies. For that purpose, a suitable matcher has to be found. Several studies have tackled how to choose a matcher depending on the characteristics of the ontologies and those of the expected alignments. They are worth taking into account. However, what should be considered is the result of the various matcher evaluation campaigns that have been run. Of course, the chosen matcher should be available, but it should above all be adapted to the current task.

The next step consists in running the matcher against the ontologies and collecting the resulting alignment. The user should not hesitate to run the matcher several times or to run several matchers, trying different sets of parameters and different thresholds. It may be useful to test the results with consistency checking tools. It is also useful to process the matching incrementally by refining the returned alignment and feeding it again to the matcher for improving it.

Once a satisfying alignment has been obtained, it is necessary to perform a final screening and validation. Ideally, this should be done by asking an independent expert to assess the quality of the alignment and perform some manual editing. This step can also be applied on the alignments found

An extra step is to save the alignment obtained in a declarative format so that it can be shared and to give it proper annotations to record its provenance and purpose. This will help others to reuse it.

Finally, the alignment can be “rendered” in the format that best corresponds to its expected use.

NeOn supports ontology alignments in both the NeOn Toolkit and the Cupboard ontology server.

- *The NeOn Toolkit alignment plug-in* works in two modes: offline and online. In the former, the user can work locally on the alignments. Users can run the matchers that are embedded in a particular toolkit against ontologies in the NeOn Toolkit and manipulate the alignments that are in their local environment. The online mode connects the NeOn Toolkit to an alignment server that permits sharing ontologies and applying the same operations on the alignments stored on the server. Of course, the alignments can move back and forth from the server to the local environment.

  Both online and offline modes provide the functions of the Alignment API: retrieving alignments, matching ontologies, trimming alignments under various thresholds, storing them in permanent stores, and rendering them in numerous output formats. These operations support the whole alignment lifecycle.

- *The Cupboard ontology server* [d’Aquin et al., 2009b] allows indexing the alignments available from the alignment servers. Hence, these alignments can be available to each Cupboard user so that they can be stored and, just like the ontologies, rated and annotated.

  The Cupboard provides direct access to alignments as well as indirect access to the Alignment server to generate new alignments when they are missing.

5. Ontology Merging

As Figure 1 shows (by arrows with the number 5), one of the scenarios identified in the NeOn Methodology is Scenario 5. In this scenario it is supposed that ontology developers reuse and merge ontological resources in the development of the ontology network.

Scenario 5 arises in those cases where several ontological resources in the same domain can be selected for reusing and when ontology developers wish to create a new ontological resource from two or more, possibly overlapping, source ontological resources. It could also happen that ontology developers wish only to establish alignments among the ontological resources selected in order to create the ontology network.

To apply Scenario 5, ontology developers should perform first the *ontological resource reuse process* to select the most suitable ontological resources that are to be used for building the ontology network. Specifically, ontology developers should carry out the activities presented in Scenario 3 as part of the ontological resource reuse process. After the ontology selection activity, ontology developers should decide how they will reuse the ontological resources selected. In Scenario 5, ontology developers decide to perform the ontology aligning and ontology merging activities, because the resources selected are valid as they are but not completely for the concrete case to be modelled if such resources are isolated. The activities to be performed are the following:

- **Activity 1. Ontology Mapping or Aligning**. Ontology developers carry out this activity with the aim of obtaining a set of alignments among the ontological resources selected. Guidelines are included in Chapter 4.
- **Activity 2. Ontology Merging**[37]. Ontology developers can merge the ontological resources selected using the alignments (output of Activity 1) to obtain a new ontological resource from the overlapping selected ones.

As already mentioned, the ontology developers have here two different possibilities: (1) to establish the mappings among the resources selected; or (2) to establish the mappings and merge the resources selected. If the ontology developers choose the second possibility, they should use the resultant ontological resource merged as input of some of the activities included in Scenario 1, as shown in Figure 1.

As stated in Chapter 1, some methodological guidelines for merging ontologies are needed. For this reason, we present here the methodological guidelines called MERGETHODOLOGY, a methodology developed by OEG (Ontological Engineering Group at the UPM), with a wide experience in the merging activity. These guidelines propose some solutions to solve some specific problems. Additionally, to help ontology engineers, we reference some of the techniques and methods available in the ontology merging area.

5.1. State of the Art

In this section we briefly present one method and two approaches to perform the ontology merging activity: FCA-Merge, PROMPT and OEGMerge.

FCA-Merge [Stumme and Maedche, 2001] is a method for ontology merging based on formal concept analysis. This method is very different from the two other approaches presented in this section. FCA-Merge takes as input the two ontologies to be merged plus a set of documents on the domains of the ontologies. The merging is performed by extracting, from the documents, instances that belong to concepts of both ontologies. Thus, if the concept C_1 of the ontology O_1 has instances in the same documents as the concept C_2 of the ontology O_2, then C_1 and C_2 are candidates to be

[37] Ontology Merging [Suárez-Figueroa and Gómez-Pérez, 2008] refers to the activity of creating a new ontology or an ontology module from two or more, possibly overlapping, source ontologies or ontology modules.

considered the same concept. To establish this relation between concepts and documents, we have created a table for each ontology. Each table relates each concept *C* of the associated ontology to the documents where instances of *C* appear. A lattice structure is generated from the tables and, finally, the merged taxonomy is obtained from the structure. Unfortunately, this method only works for lightweight ontologies.

The *PROMPT method* [Noy and Musen, 2000] is an approach to ontology merging and has a matching algorithm, and there is a related tool focused on ontology versioning (PROMPTDiff). The main assumption of PROMPT is that the ontologies to be merged are formalized with a common knowledge model based on frames. A plug-in of Protégé-2000 merges ontologies according to this method. PROMPT first proposes to elaborate a list of the operations to be performed when merging the two ontologies (e.g., merging two classes, two slots, etc.). This activity is carried out automatically by the PROMPT plug-in. Then, a cyclic process starts. In each cycle, the ontology engineering selects an operation from the list and executes it. Then a list of conflicts resulting from the execution of the operation is generated, and the list of possible operations for the following iterations is updated. Some of the new operations are included in the list because they are useful to solve the conflicts.

OEGMerge [Fernández-López et al., 2006] is an approach that models ontologies and mappings and defines merging using case-base reasoning. This approach specifies and formalizes the cases found in different projects. One of the most relevant characteristic of this approach is that it is not an algorithm but a rule-based system. In fact, the characteristics of the problems found when merging ontologies (use of incomplete knowledge, heuristic, ontological engineering knowledge, etc.) make us think that a rule-based model is more suitable than a procedural algorithm, when the ontologies and the intermapping establish the knowledge base. WebODE has a plug-in called ODEMerge [Ramos, 2001] that supports this method.

5.2. Guidelines Proposed for Ontology Merging

The goal of the ontology merging is to unify the knowledge represented by, at least, two ontologies, into a single ontology.

The NeOn Methodology proposes the filling card, presented in Table 12, for the ontology merging activity; the card includes the definition, goal, inputs and outputs, actors involved in the activity and time for the activity to be carried out. These features correspond to *Mergethodology* methodology, developed by OEG[38].

[38] http://www.oeg-upm.net/

Ontology Merging	
Definition *Ontology Merging* refers to the activity of creating a new ontology or an ontology module from two or more, possibly overlapping, source ontologies or ontology modules.	
Goal The merging activity unifies the knowledge represented by, at least, two ontologies into a single ontology using a set of mappings among such ontologies.	
Input A set of ontologies (at least two) and a set of mappings among these ontologies.	*Output* An ontology.
Who Ontology engineers in collaboration with domain experts.	
When This activity must be carried after the ontology aligning activity.	

Table 12. Filling card for the activity Ontology Merging

The tasks for carrying out the ontology merging activity can be seen in Figure 7.

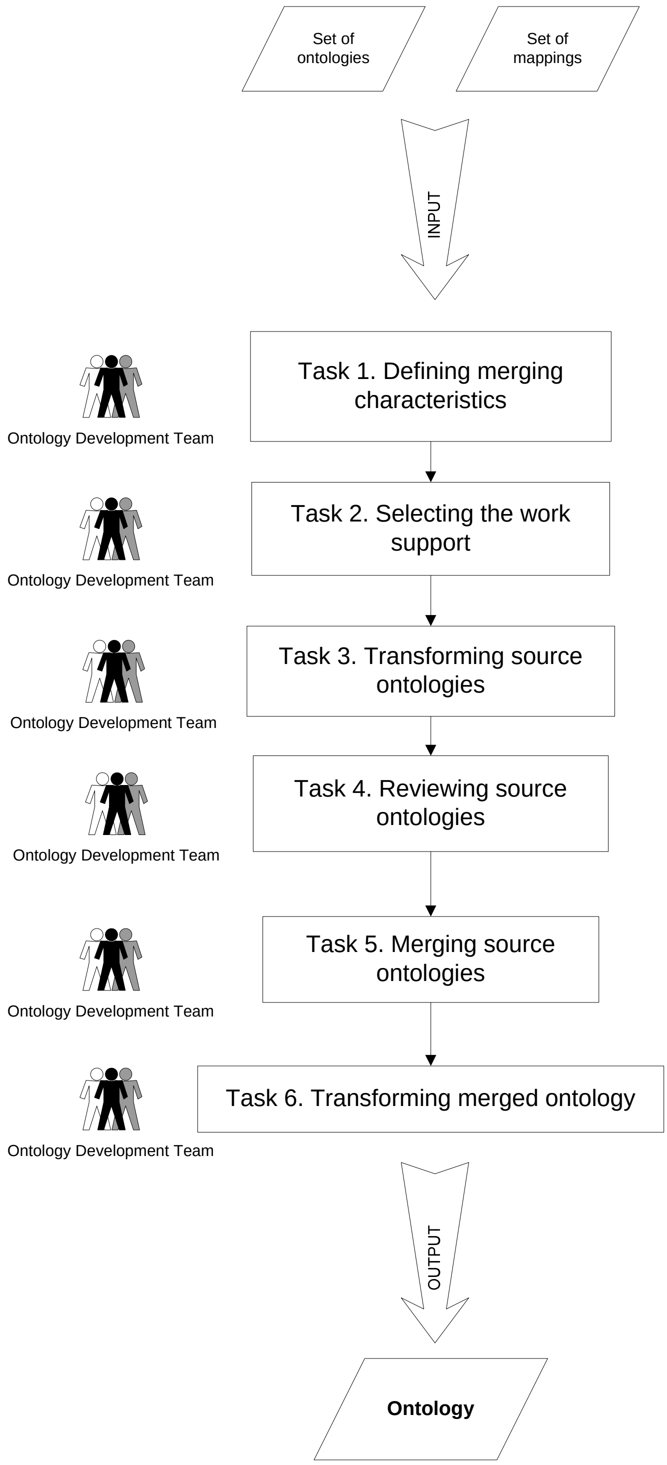

Figure 7. Tasks for Ontology Merging

The tasks for carrying out the ontology merging activity are explained in more detail in the following:

Task 1. Defining merging characteristics.

The objective of this task is to identify the merging characteristics. Examples of these characteristics can be input ontology languages, input model expressiveness, output model expressiveness, available merging tools/platforms, merging tool expressiveness, and so on. These parameters or characteristics have to be instantiated for each merging case, and will be used later on to select tools, platforms and methods in the whole merging activity.

The ontology development team, which should include knowledge engineers, carries out this task taking as input sources (ontologies, mappings and other resources) and targets features and available tools/platforms to obtain the merging characteristics. The development team should compile all the information about sources and targets to define the merging characteristics.

Task 2. Selecting the work support.

The objective of this task is to select the platform/tool that will support the revision and merging tasks. This selection is crucial because the success of the merging process depends on it. There are some merging platform/tool that can be evaluated in order to be used in the merging activity. See HCONE [Kotis and Vouros, 2004], Chimaera [McGuiness et al., 2000], OntoMerge[39] [Dou et al., 2003], FCA-Merge [Stumme and Maedche, 2001], PROMPT [Noy and Musen, 2000] (that is a plug-in of Protégé tool), and ODEMerge [Ramos, 2001] (that supports OEGMerge method [Fernández-López et al., 2006] into WebODE Platform).

The ontology development team, which should include knowledge engineers, carries out this task taking as input ontology tool features, source ontology features, and final ontology requirements to obtain the selected platform/tool.

The development team should compare the different platforms/tools at its disposal under the following features: all information about sources and targets to define the merging characteristics, knowledge model expressiveness, transformation facilities, and merging support or edition easiness.

We advise to select a tool or platform that supports at least an automatic part of the merging.

Task 3. Transforming source ontologies.

The purpose of this task is to adapt the source ontology format to the format or knowledge model supported by the tool selected in Task 2. This task should be carried out only if the source ontologies are not in the same format than the one required by the tool selected.

It is worth mentioning that during this transformation task, some information may be missed; therefore, we have to report this event (e.g., in a document, software repository, or other available support).

The ontology development team, which should include knowledge engineers and tool experts, carries out this task taking as input source ontologies to obtain these source ontologies (in the format supported by the selected platform/tool) and also the transformation report.

During the transformation, the ontology development team can make use of automatic wrappers or translators (e.g., ontology importers). If this is not possible, the development team should study some source and destination formats and develop an *ad hoc* automatic translator for this transformation.

Task 4. Reviewing source ontologies.

The objective of this task is to identify and correct all the possible mistakes of the source ontologies. This task is crucial, since errors in the source ontologies would be propagated through the merging activity if they are not corrected.

The ontology development team, which should include ontology engineers, carries out this task taking as input ontologies obtained in Task 3 for obtaining a set of revised ontologies. The development team should refer to the mistakes identified in [Gómez-Pérez, 2001] and to the design criteria appeared in [Vilches-Blázquez et al., 2009]. It is worth mentioning that this task may involve a joint revision of sources.

Because sources can change in the transformation performed in Task 3, it is advisable to carry out their revision after the sources are transformed into the merging model.

[39] http://cs-www.cs.yale.edu/homes/dvm/daml/ontology-translation.html

Task 5. Merging source ontologies.

The objective of this task is to merge the set of input ontologies. The ontology development team, which should include ontology engineers, carries out this task taking as input the revised ontologies obtained in Task 4 for obtaining a merged ontology. To carry out this task, the development team can use one of the techniques/methods for merging ontologies, or develop a new *ad hoc* technique/method. A compilation of methods and techniques to merge ontologies can be found in [Fernández-López et al., 2006].

Task 6. Transforming merged ontology.

The objective of this task is to transform the merged ontology in a final format different from the one used during the merge activity. It should be note that this task is optional.

The ontology development team, which should include knowledge engineers and tool experts, carries out this task taking as input the merged ontology obtained in Task 5 for obtaining the final merged ontology and a transformation report. If the merged ontology obtained in Task 5 has to be transformed, the ontology development team should carry out Task 3 taking as source ontology the ontology merged. As already mentioned in Task 3, during the transformation, the ontology development team can make use of automatic wrappers or translators (e.g., ontology importers). If this is not possible, the development team should study the source and destination formats and develop an *ad hoc* automatic translator for this transformation.

6. Combining Different Types of Knowledge Resources

When developing ontologies by reusing existing knowledge resources as the NeOn Methodology proposes, the ontology development team can find itself in situations in which may have to take different decisions. For example, they can have at their disposal an ontological resource that models similar knowledge as a non-ontological resource, an ontological resource that can be extended with a non-ontological resource, an ontological resource that can be extended with an ODP, and so on. Thus, the ontology development team can make itself different questions, such as: Which option is better to reuse an ontology, or to reuse a non-ontological resource?, For which purpose is better to combine an ontology and a non-ontological resource?, and so on.

In general, the ontology development team should have to face questions similar to the following ones:

- How to select among different resources modelling similar knowledge?
- How to extend a particular knowledge resource?
- How to combine different knowledge resources?
- How to add a new ontological resource to an ontology network?

In this chapter we provide methodological guidelines to combine different types of knowledge resources during the ontology network development. Other questions are out of the scope of this deliverable.

First, we describe here the experience we had during the development of an ontology network in the ICPS project (Section 6.1). Second, we extract some general conclusions from such an experience that are presented in Section 6.2 as methodological guidelines.

6.1. ICPS Ontology Network Development Experience

In this section we show how we have tackled an ontology network development by combining ontological resources, non-ontological resources, and ODPs. This development process is being carried out within a project called ICPS[40] (International Classification for Patient Safety) funded by the World Alliance for Patient Safety of the World Health Organization[41] (WHO). The main objective of the ICPS project is to create a categorization or classification of the concepts related to *patient safety* that will enable an efficient monitoring, analysis and interpretation of the information on that subject. The final aim of the project is to improve patient care and policy plans all over the world.

In this framework, the ICPS is being developed for optimum semantic interoperability. Classification, which is the main aim of this project, will be supported by an extensive and robust knowledge representation, in the form of an ontology network, and this will make such a classification easily processable by machines. The ontology network development at this stage is focused on two different sub-domains within the "patient safety" domain:

- **Falls** as a type of incident that occurs to patients while receiving hospital care and that can result, or indeed resulted, in unnecessary harm to the patient.
- **Pressure Ulcer** as a type of injury or damage to a tissue caused by an agent or a circumstance.

[40] http://www.who.int/patientsafety/implementation/taxonomy/en/

[41] http://www.who.int/en/

To start the ontology network development, we carried out the *ontology requirements specification activity* following the NeOn Methodology guidelines for such an activity [Suárez-Figueroa, et al., 2009b]. As output of the activity we obtained the Ontology Requirements Specification Document (ORSD) that is available in [Montiel-Ponsoda et al., 2009a]. Since this ORSD is not a public document, in this section we will include excerpts from it when necessary to clarify some points.

This section is organized as follows: Section 6.1.1 describes the quick search carried out in order to obtain different types of knowledge resources to be reused in the ICPS ontology network development, Section 6.1.2, identifies the scenarios from NeOn Methodology (Chapter 2) that will be executed into the ontology network development. In Section 6.1.3 presents how we have created the upper-level conceptual models for Falls and Pressure Ulcer subdomains. Finally, Section 6.1.4 shows an overview of the remaining ICPS ontology network development.

6.1.1. Quick search for knowledge resources

After the ontology requirements specification activity, the NeOn Methodology advises to carry out a *quick search* for the knowledge resources existing in the domain. To perform such a quick search, the main terms identified in the ORSD [Montiel-Ponsoda et al., 2009a], specially in the CQs, should be used. The main objective of this search is to obtain an overview of candidate resources that could be eventually reused during the ontology development process.

In the case of the ICPS project, there are two reasons for carrying out a knowledge resource reuse process. One is that some resources have to be reused in the ontology network development (as it has been established in the ORSD) whereas other resources, developed within the ICPS project environment, are suggested to be reused. This reuse process is mainly focused on aligning the ontology being developed to standard and international classifications. The other reason is that we carried out a resource search in the web in order to reduce the time and effort in the ontology development, as proposed the NeOn Methodology, by means of reusing such resources.

Here we present the obligatory resources to be reused (as stated in the ORSD), the suggested resources developed by other partners in the ICPS project, and the resources found during the quick search in the Web and in ontology repositories.

Knowledge resources to be reused as established in the ORSD

Some classification schemas in the Falls and Pressure Ulcer domains that had to be made interoperable with the ICPS ontologies should be reused as stated in the ORSD [Montiel-Ponsoda et al., 2009a] in the non-functional ontology requirements. These resources are listed below:

- ICD[42] (International Classification of Diseases), a WHO classification
- ICF[43] (International Classification of Functioning, Disability and Health)
- The clinical terminology SNOMED-CT[44] (Systematized Nomenclature of Medicine-Clinical Terms)

Knowledge resources developed by the ICPS project partners

Other sources that we also had to take into account were those that other partners in the ICPS project were producing in parallel, namely:

- The *ICPS Fall Extension* and *ICPS Pressure Ulcer Extension* classifications from the Australian Patient Safety Foundation[45].

[42] http://www.who.int/classifications/icd/en/

[43] http://www.who.int/classifications/icf/en/

[44] http://www.ihtsdo.org/snomed-ct/

[45] http://icps.apsf.org.au/

- The *ICPS.owl* and *ATC.owl* preliminary ontologies also created by the Australian Patient Safety Foundation.
- The Patient_safety.owl[46] ontology developed at the *Université de Saint Etiennes*.

Knowledge resources obtained from the Web and repositories

Additionally, we also looked for available ontologies on the Web that could be of any help for conceptualizing the Falls and Pressure Ulcer domains by means of Semantic Web Search Engines such as *Watson*[47] and *Swoogle*[48]. The search was made using the main terms from the ORSD, namely, fall, injury, patient safety, pressure ulcer, etc. Figure 8 and Figure 9 show an excerpt from the excel file that contains the Fall and Pressure Ulcer CQs. CQs were grouped (as it is proposed in the ontology specification requirements activity [Suárez-Figueroa, et al., 2009b]) by the following knowledge areas: Activity; Anatomy; Assessment; Assistance; Furniture; Incident; Injury; Medication; Person; Place; Stage; Strategy; and Time.

It should be observed that from this quick search we obtained some resources that can be reused, though the set could not be completed. Therefore, during the ontology development, an exhaustive search should be carried out to obtain more knowledge resource candidates. The *candidate ontologies* after a quick search are listed below:

- Owl-Time[49]: an ontology of time proposed by the W3C that could be reused with the aim of modelling aspects such as the time when a certain pressure ulcer is detected.
- OBO (Open Biomedical Ontologies)[50]: In the OBO portal, there is a catalogue of ontologies from which we have selected the following two as candidates for the ICPS ontology development:
  - MeSH (Medical Subject Headings)[51], about medical domain.
  - CARO (Common Anatomy Reference Ontology)[52], about anatomical parts of the organism.

[46] The ICPS.owl, ATC.owl, and Patiente_safety.owl ontologies are only available as internal resources of the ICPS project.

[47] http://watson.kmi.open.ac.uk/WatsonWUI/

[48] http://swoogle.umbc.edu/

[49] http://www.w3.org/TR/owl-time/

[50] http://www.obofoundry.org/

[51] http://www.berkeleybop.org/ontologies/obo-all/mesh/mesh.owl

[52] http://www.berkeleybop.org/ontologies/obo-all/caro/caro.owl

Number	Competency Questions (CQs) -Fall	Answers
CQ1	What type of incident happened?/ Which are the types of incidents?	Clinical Administration; ClinicalProcess/Proce
CQ2	What type of fall caused the incident?/Which are the mechanisms of the fall?	Trip/Stumble; Slip; Collapse; Loss of Balance
CQ3	Which elements were involved in the fall?	Cot; Bed; Chair; Stretcher; Toilet; Therapeutic
CQ4	Which elements caused the fall?	Cot; Bed; Chair; Stretcher; Toilet; Therapeutic
CQ5	Where did the patient fall from?	Bed; Cot; Stretcher/Trolley/Gurney; Chair; Toil
CQ6	Which types of floor surface exist?	Tiles; floor boards; carpet; vinyl; cork; slate; ru
CQ7	Which building structures can be involved in a fall?	Balcony; bridge; flag-pole; floor; railing; roof; tc
CQ8	When did the fall happen?	?
CQ9	Where did the fall happen?	?
CQ10	From which height did the patient/subject fall?	From a height or from the same level.
CQ11	Who observed the fall?	Staff; visitor; family; another patient; unknown;
CQ12	Who assisted the fall?	Staff; visitor; family; another patient; unknown;
CQ13	Who assessed the patient to be at risk for a fall?	Staff; visitor; family; another patient; unknown;
CQ14	Which are the types of assistance that a patient could receive?/Which types of assistanc	Stand-by assistance by a person; physical as
CQ15	What was the patient doing or trying to do at the time of the fall?/What was the patient's a	Ambulating; moving in bed (rolling out from low
CQ16	Which types of mobility aids exist?	Walking stick; walking frame; crutches; wheel
CQ17	What type of injury did the patient sustain as a result of the fall?	Dislocation; fracture; intracranial injury; lacera
CQ18	Which fall risks can be cause by medications?	Dizziness; drowsiness; postural blood pressur
CQ19	What was the patient's medication after the fall?	?
CQ20	What was the patient's treatment at the time of the fall?/On which medication was the pat	?

Figure 8. Excerpt[53] of Competency Questions (CQs) about Falls

Number	Competency Questions (CQs) -Pressure Ulcer	Answers
CQ1	What is a pressure ulcer?	Type of incident
CQ2	Where was the pressure ulcer detected?	Community Care Facility; Hospital
CQ3	Which are the types of facilities?	Community Care Facility; Hospital
CQ4	When was the pressure ulcer detected in the Community Care Facility?	Before commencing care (in a Community Care Facility); after cor
CQ5	When was the pressure ulcer detected?	on admisssion to hospital; during stay in hospital
CQ6	Where is the pressure ulcer detected?	head; neck; trunk; limbs
CQ7	In which parts is the head divided?	Scalp; ear; face
CQ8	Which are the parts of the face?	eye; nose; lip; mouth; chin
CQ9	Which are the parts of the trunk?	shoulder area; chest; scapula; breast; abdominal area; back (uppe
CQ10	Which are the types of limbs?	arms and legs
CQ11	Which are the parts of the arm?	upper arm; elbow; forearm; wrist; hand
CQ12	In which parts is the hand divided?	digits; back of the hand; palm
CQ13	Which are the parts of the leg?	thigh; knee; lower leg
CQ14	Which are the parts of the thigh?	upper thigh and lower thigh
CQ15	Which are the parts of the knee?	back of the knee; front of the knee; lateral inner aspect; lateral out
CQ16	In which parts is the lower leg divided?	ankle; calf; shin
CQ17	Which are the parts of the ankle?	lateral aspect; medial aspect
CQ18	Which are the parts of the foot?	heel; sole; toe and top
CQ19	Which are the stages a pressure ulcer can go through?	stage 1; stage 2; stage 3; stage 4; unstageable; Suspected Deep
CQ20	How many stages did the pressure ulcer advanced during the patients stay?	

Figure 9. Excerpt[54] of Competency Questions (CQs) about Pressure Ulcer

It is worth mentioning that none of the matches for pressure ulcer provided by Watson corresponds to the exact concept “pressure ulcer”, but the words “pressure” or “ulcer” are contained in the Label slot of ontology concepts. However, this search has allowed us to quickly find ontologies in the domain that could be considered knowledge resource candidates to be reused in the ontology development process of ICPS ontologies.

Regarding the reuse of *Ontology Design Patterns (ODPs)*, at a first sight, we have already identified some patterns that could contribute to the development of the Fall and Pressure Ulcer ontologies not only by speeding up the development process in general but also by guaranteeing that we are reusing consensual solutions considered best practices in the Ontology Engineering field.

53 Image captured from [Montiel-Ponsoda et al., 2009a].

54 Imaged captured from [Montiel-Ponsoda et al., 2009a].

We looked for ODPs through catalogues included in *NeOn Deliverable 5.1.1. NeOn Modelling Components* in [Suárez-Figueroa et al., 2007b] and *NeOn Deliverable 2.5.1. A Library of Ontology Design Patterns: reusable solutions for collaboratively design of networked ontologies* in [Presutti et al., 2008] that contains descriptions of the ODPs but not their associated code, and catalogues in the Ontoloy Design Patterns Portal[55], which is a on-line library containing both the descriptions and the associated code for the patterns.

From [Suárez-Figueroa et al., 2007b], we recognized the following patterns as candidates to be reused, accompanied by their identifier in the NeOn repository:

- Architectural Pattern for modelling a Modular architecture (AP-MD-01). This pattern could be reused to design the ontology network by means of a modular approach. Concretely, the ontology is divided into several knowledge areas, matching some of them with the areas from the ICPS framework.
- Logical Pattern for modelling Specified Values for Individuals (LP-SV-01). This pattern could be used to represent the possible values that can have a risk assessment.
- Logical Pattern for Modelling N-ary Relation: Using Lists for Arguments in the Relation (LP-NR-02). This pattern could help us to model the relation between a pressure ulcer and the different stages it can go through.
- Content Pattern for modelling a Part-Whole Relation (CP-PW-01). This pattern could be reused to represent the parts of a hospital.
- Content Pattern for Modelling a Part-Whole Class Hierarchy (CP-PW-02). This pattern could be reused to model the different anatomical parts that can be involved in a Pressure Ulcer injury.

From [Presutti et al., 2008], we identified the following pattern as a candidate reuse component:

- Content Pattern for modelling N-ary Participation (CP-NPAR-01). This pattern could be reused to represent situations that can be composed of various entities, for example, when a patient suffers a fall in which other persons can be involved; when a patient suffers a fall in which furniture or clothing can be involved; etc.

Finally, we have found a *non-ontological resource* related to the classification proposed to reuse in the ORSD. This resource is the ICECI[56] (International Classification of External Causes of Injury) that is defined as a system of classifications that enables a description of how injuries occur. The ICECI is related to the External Causes chapter of the ICD, and can also be freely downloaded as a doc or pdf document.

6.1.2. Identification of scenarios executed

After searching for knowledge resources to be reused during the ICPS ontology network development, the scheduling activity should be carried out as the NeOn Methodology proposed. It should be noted that the ICPS project requirements fix both the times and general stages for the ontology development. In summary, such an established schedule states that the ORSD should be ready by the end of September 2009 and ontologies should be ready by the end of January 2010 (including in the development at least the conceptualization and the implementation activities). However, this vague schedule is not enough to develop ICPS ontologies. For this reason, we carried out the *scheduling* activity in the following way:

- We identified the scenarios to be followed in the ontology development, as proposed the NeOn Methodology.

[55] www.ontologydesignpattern.org

[56] http://www.rivm.nl/who-fic/ICECIeng.htm

- We maintained the timing and deadlines established in ICPS project requirements, fitting in that schedule the processes and activities obtained from the identified scenarios.

We identified the scenarios needed to develop the ICPS ontology network from the set of scenarios in the NeOn Methodology (Chapter 2), which are the following:

- *Scenario 1: From specification to implementation*. This scenario is the basic one, and can be combined with the rest of scenarios. Since the scenario contains the fundamental activities for developing an ontology, it is compulsory to select it in each ontology development.
- *Scenario 2: Reusing and re-engineering non-ontological resources*. We have selected this scenario for two reasons: (1) first, because we have to reuse some non-ontological resources as stated in the ICPS ORSD [Montiel-Ponsoda et al., 2009a], and (2) second, because we have found some suitable non-ontological resources to be reused through a quick search, as explained in Section 6.1.1.
- *Scenario 3: Reusing ontological resources*. We have selected this scenario first because we have some ontological resources developed by other partners in the ICPS project that were suggested to be reused and second, because through a quick search on the web we have found some ontologies suitable to reuse (as explained in Section 6.1.1).
- *Scenario 4: Reusing and re-engineering ontological resources*. We have selected this scenario because we will need to transform some ontological resources, from those reused in Scenario 3, in order to integrate them in the ontology being developed and avoid some of the pitfalls [Poveda et al., 2009] we have observed in the ontological resources.
- *Scenario 7: Reusing ontology design patterns (ODPs*). We have selected this scenario because, as explained in Section 6.1.1, we have planned to reuse ontology design patterns during the ontology network development.
- *Scenario 8: Restructuring ontological resources*. We have selected this scenario because we have planned to restructure some resources and the ontology network itself in order to create a modular ontology network and to avoid some of the pitfalls described in [Poveda et al., 2009].

6.1.3. Development of the upper-level conceptualization model

After having identified the scenarios involved in the ICPS ontology network development, we carried out the *conceptualization* activity. The first task of this activity is the development of the upper-level conceptualization model. We created this model for Fall and Pressure Ulcer ontology networks in a modular fashion. Thus we reused the *Architectural Pattern for modelling a Modular architecture (AP-MD-01)* to provide a modular architecture to the ontologies. It is worth mentioning that the modular architecture was performed first based on the knowledge areas in which the CQs from ORSD were grouped (as explained in Section 6.1.1). Once the knowledge areas that come from CQs were identified we realized that almost all those areas matched the terms identified in the ICPS framework, as shown in Table 13.

Knowledge Areas identified by the ontology developers	Corresponding terms from the ICPS framework
Activity	Contributing Factors / Hazards
Furniture	
Medication	
Anatomy	-
Assistance	Mitigating Factors
Incident	Patient Safety Incident
Injury	Patient Outcomes
Stage	
Person	Patient Characteristic
Place	Incident Characteristic
Time	
Strategy	Detection
Assessment	

Table 13. Correspondence among knowledge areas identified by the ontology developers and terms of the ICPS framework

Accordingly, we adopted the names of the knowledge areas from the ICPS framework term names. This adoption can be seen as a good way of evaluate for the CQs groups because they match the domain experts' view in a natural way. Finally, we obtained the upper-level conceptual model for the Fall domain (Figure 10) and for the Pressure Ulcer Domain (Figure 11). In Figure 10 and Figure 11 the knowledge areas that match the ICPS framework are represented by red dotted triangles, the blue triangles represent additional ontologies that are needed to develop the ICPS ontology network. These figures also contain the knowledge resources (ontologies, classifications, ODPs, and external sources) listed in Section 6.1.1. Each knowledge resource is presented under the ontology in which the resource could be probably reused. To identify the ontology in which a particular resource (classifications, external resources, and ontologies) could be reused, we analysed the resource content to determine the subdomains in which the resource could fit. In the case of ODPs, we identified those possible patterns by analysing the CQs in the ORSD (for example, if a CQ mentions something about a list of values, we could infer that the LP-SV-01 [Suárez-Figueroa et al., 2007b] could be reused).

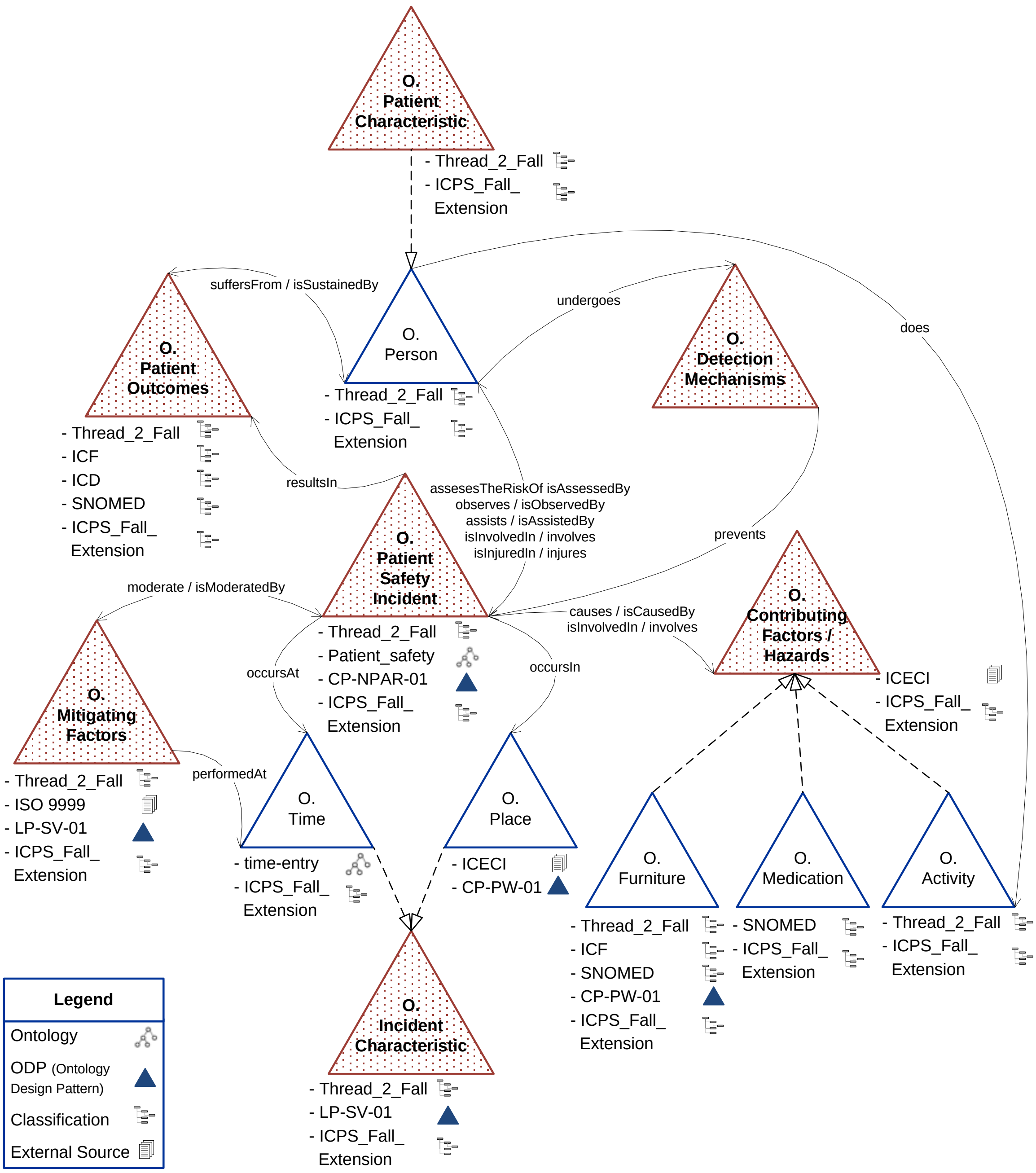

Figure 10. Upper-level conceptual model of Falls ontology [Montiel-Ponsoda et al., 2009b]

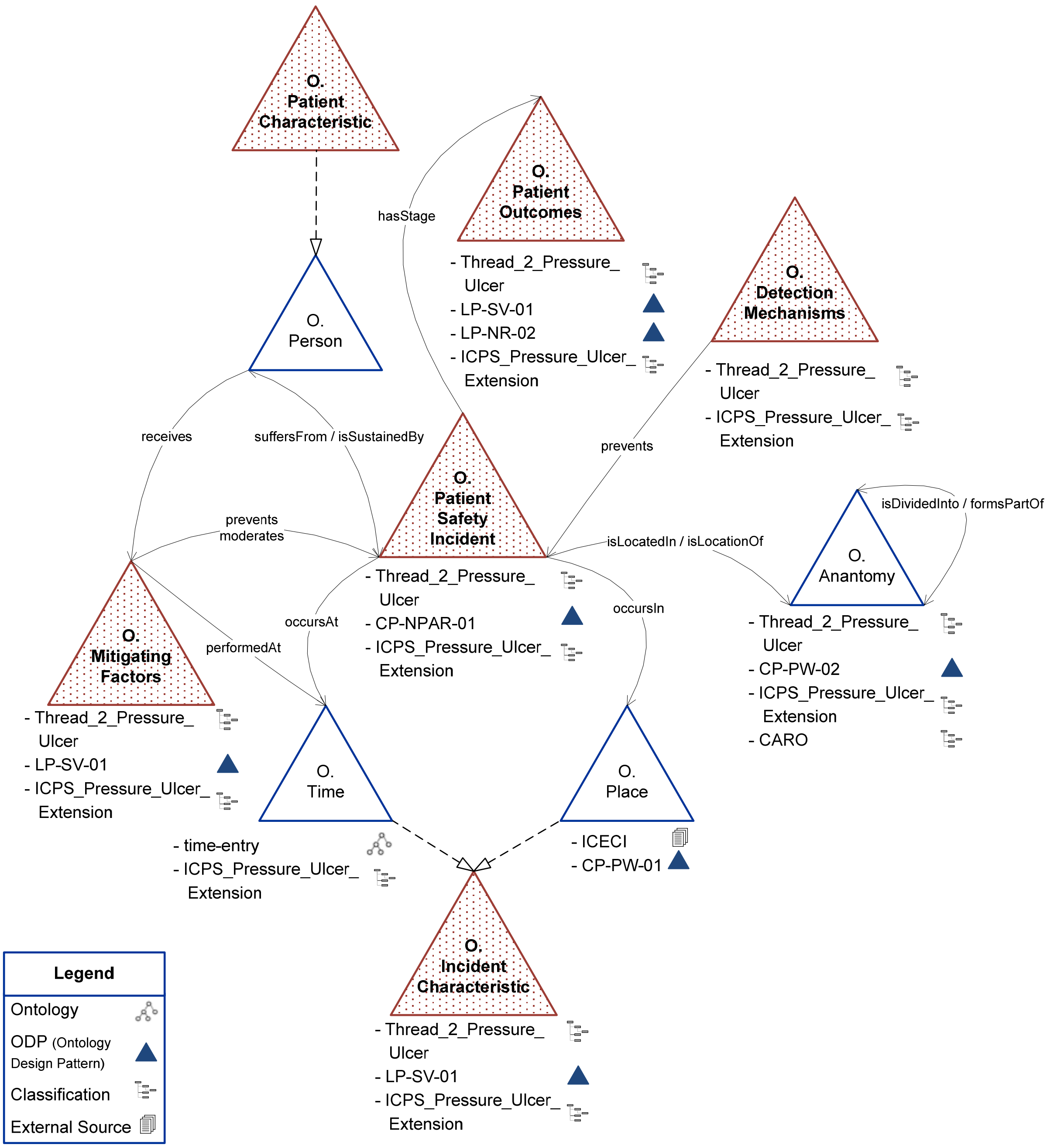

Figure 11. Upper-level conceptual model of Pressure Ulcer ontology [Montiel-Ponsoda et al., 2009c]

After having the upper-level conceptual model, we thought of a sketch about how to combine the different scenarios and resources involved in the ICPS ontology network development. Figure 12 aims to make easier its comprehension to the reader by means of a graphical sketch that represents the combination of scenarios and its development through the time. This figure has been created regarding the different types of knowledge resources and how they could be reused. After that, we have to think about when we can carry out each scenario according to the rest of scenarios and, above all, according to Scenario 1. At the bottom of Figure 12, there is a time line that represents when the ontology development starts and finishes. On the vertical axis, we represent which scenarios can be developed at the same time. In this way, we can see that the scenarios selected for the ICPS ontology development (in this case the scenarios 2, 3, 4, 7 and 8) are developed in parallel to scenario 1, which is compulsory and is the fundamental scenario for any ontology development, as we already mentioned. The arrows represent the transitions between scenarios and/or activities. The numbers in white circles represent the scenario number. The circles with a letter represent an ontological product whose level of formality is indicated by the letter (see Figure 12).

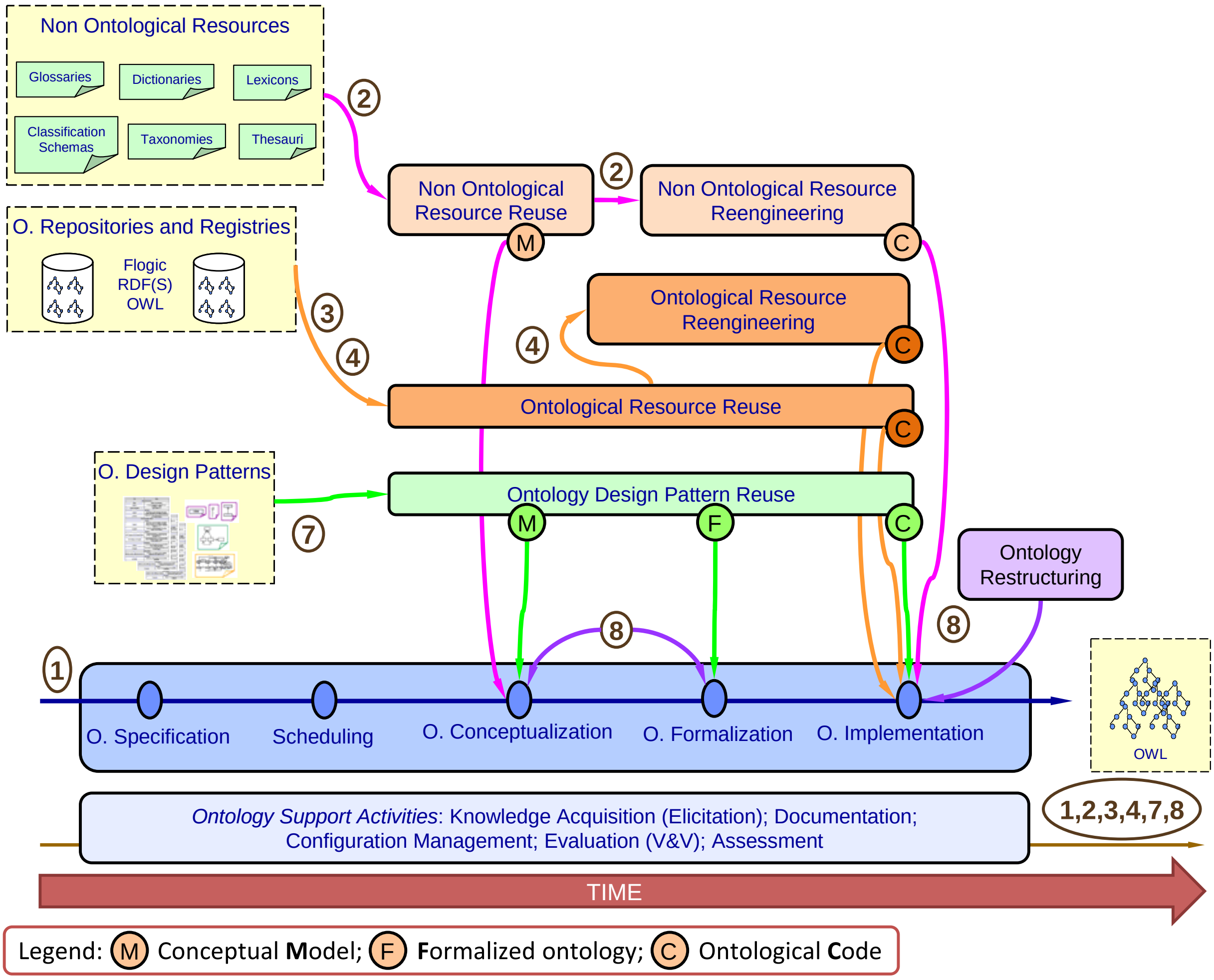

Figure 12. Combination of scenarios in the ICPS ontology development

6.1.4. Overview of the remaining processes and activities of the ICPS ontology development

After having the upper-level conceptual model and the combination of the different scenarios to be followed during the development, the ontology development team might decide whether to reuse any knowledge resource as the starting point of the development or, on the contrary, to start the development in a new ontology and to utilize the candidate knowledge resources to be reused only for completing the new ontology. To take this decision, ontology developers should take into account the following issues:

- Which is the resource that better fits the ontology network requirements?
- Which is the most complete resource?
- Which is the most reliable resource?
- Is there really one (or a set of) resource suitable to be the starting point or are all the resources suitable for complete a new ontology network development from scratch?
- Which format, and the transformation effort if necessary, of the best candidate(s) to be reused?

- Are there any constraints or requirements to start the development from a specific resource?

In the ICPS ontology network development, we have selected the *ICPS.owl* ontology as the starting point for several reasons, which are listed below:

- It fits quite well the ontology network requirements.
- Although it is not the most complete knowledge resource, because these are the *ICPS Fall Extension* and *ICPS Pressure Ulcer Extension* classifications from the Australian Patient Safety Foundation, it is close to these resources with respect to completeness; besides, it is the most complete ontological resource.
- It does not need a transformation effort because it is developed in OWL.
- Since it has been developed within the ICPS project, it contains the terms and definitions approved by the patient safety community.

Since the ICPS ontology network development is still in progress, here we just want to outline how we approach the remaining development process.

Regarding the ontological resource reuse, we can distinguish three resources:

- The *ICPS.owl* ontology will be reused as a starting point to the ICPS ontology network development.
- The *ATC.owl, Patient_safety.owl* and *Mesh* ontologies could be reused to complete little bits of knowledge if necessary, maybe through an ontology statement reuse process.
- The CARO ontology could be reused to extract human anatomy concepts and definitions.

After the ontology reuse process, the ontology re-engineering process could take place.

Since we are starting with a complete ontology, we will focus on the reuse of other resources, especially for complete little parts of the ontology network as we describe further. For example, we propose the *ICPS Fall Extension* and *ICPS Pressure Ulcer Extension* classifications from the Australian Patient Safety Foundation to complete, update and, above all, to add natural language definitions to the ICPS ontology network concepts. We will perform a non-ontological resource reuse process that involves a non-ontological resource re-engineering process. Moreover, the ICD (International Classification of Diseases) and the ICF (International Classification of Functioning, Disability and Health) classifications and SNOMED-CT (Systematized Nomenclature of Medicine-Clinical Terms) will be used to align the ICPS ontology network concepts with their own.

As can be seen in Figure 12, the output ontologies from ontological and non-ontological reuse process could be restructured by an ontology restructuring process and integrate them into the ontology network being developed. Also, the ontology network being developed can be restructured. This ontology restructuring process can take place in the conceptualization, formalization, and implementation activities.

Finally we tackle the ODPs reuse process. From our point of view, the ODPs can take different roles in ontology developments, for example:

- As we have already mentioned, we can reused ODPs to define the architecture of the ontology network in the conceptualization activity through the *Architectural Pattern for modelling a Modular architecture (AP-MD-01)*.
- We can reuse ODPs, both in the formalization and implementation activities, to complete the knowledge represented; for example, we can enrich a mereological relationship through the *Content Pattern for modelling a Part-Whole Relation (CP-PW-01)* and/or the *Content Pattern for Modelling a Part-Whole Class Hierarchy (CP-PW-02)*.

- We can reuse ODPs, both in the formalization and implementation activities, to represent logical structures that are not supported by the ontology language, for example, *Logical Pattern for Modelling N-ary Relation: Using Lists for Arguments in the Relation (LP-NR -02)*.
- We can reuse ODPs, both in the formalization and implementation activities, to join concepts from different ontologies. In the ICPS ontology network development, the *Content Pattern for modelling N-ary Participation (CP-NPAR-01)* will probably be reused for join concepts from different ontologies; also it is usual to reuse the *Logical Pattern for Modelling N-ary Relation: Introducing a New Class for the Relation (LP-NR -01)* for this purpose and this could be reused within the ICPS ontology network developed if necessary.

6.2. Guidelines Proposed for Combining Different Types of Knowledge Resources

Based on the experience obtained from the ICPS use case, in this section we include a proposal on how to tackle, in a general way, the ontology network development by means of combining ontological resources, non-ontological resources and ODPs. For this purpose, in Section 6.2.1 we describe the possible situations that can occur when dealing with several types of knowledge resources. We have also collected the aforementioned situations in a flowchart to illustrate the possibilities. Also, in Section 6.2.2, we propose a sequence to generate the different graphics shown in Section 6.1 and to use them to face the ontology network development. Additionally, in Section 6.2.3, we propose some preliminary questions that could be used to determine how useful are the knowledge resources available to be reused in an ontology development process.

6.2.1. Ontology network development by combining ontological resources, non-ontological resources and ODPs

When trying to reuse several types of knowledge resources to develop an ontology network, we probably have to combine reuse processes both with each other and with manual development processes. According to the degree of reuse to be carried out, we can deal with four main situations (ordered from no reuse process to be carried out to only reuse process to be carried out):

1. Develop the ontology network from scratch, from specification to implementation: In this case none of the knowledge resources available is suitable either as starting point for the development or as complement for the ontology network. Therefore, the entire development will be carried out manually. It is worth mentioning that this situation should be avoided.
2. Develop a part of the ontology network from scratch and complete it by reusing different types of knowledge resources. In this case, none of the knowledge resources available is suitable as starting point, but there are suitable knowledge resources to complete (improve, extend, etc.) the ontology network. Therefore, the ontology network developed is mainly focused on a manually approach accompanied by a reuse process.
3. Develop the ontology network reusing different types of knowledge resources as starting point. In this case, there is one or a set of knowledge resources that can be reused as starting point to develop a part of the ontology network that covers a subset of the ontology network requirements. The rest of the ontology network requirements could be covered either by reusing other knowledge resources or in a manual way. Therefore, the ontology network developed is mainly focused on the reuse of knowledge resources and can be accompanied by a manual development.
4. Develop the ontology network only by reusing different types of knowledge resources. In this case, there is one or a set of knowledge resources that can be reused as starting point to develop the ontology network covering all the ontology network requirements. Therefore, the entire development will be carried out by reusing knowledge resources.

In Figure 13 we present a preliminary guide to ontology network development based on the types of knowledge resources to be reused; following the figure we can learn how to develop the ontology network following each of the aforementioned situations. It is worth mentioning that this figure does not take all scenarios proposed in NeOn Methodology (Chapter 2), just those that involve knowledge resources reuse and others that could be necessary (such as Scenario 8). In Figure 13, the start and finish points are represented by a red and green circle respectively; the rhombus represents questions whose possible answers are represented by labelled arrows; the grey rectangles represent actions to be taken; the coloured rectangles with continuous line represent scenarios from NeOn Methodology (Chapter 2) to be carried out, and finally, the coloured rectangles with discontinuous line represent scenarios from NeOn Methodology (Chapter 2) that should be carried out if necessary.

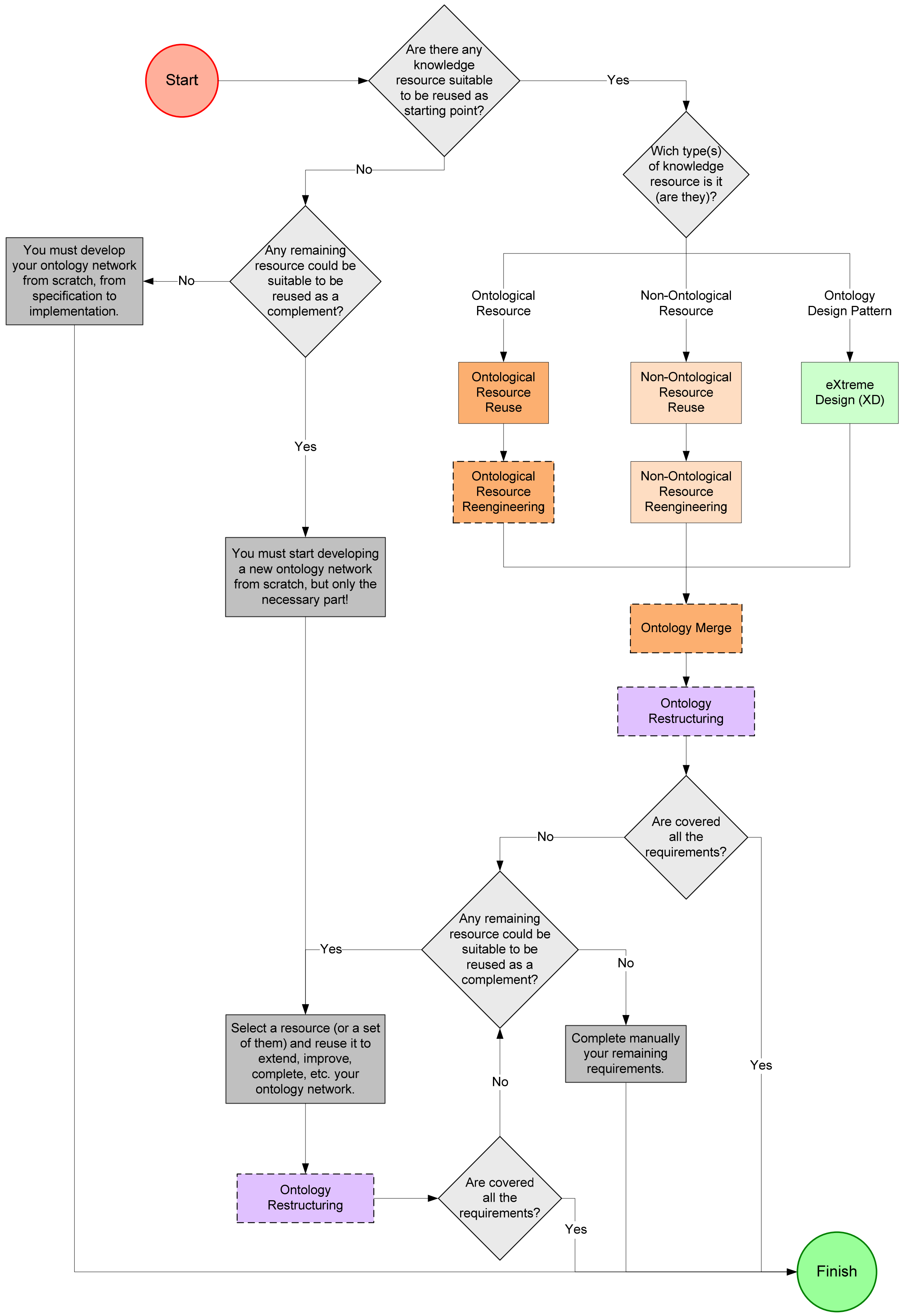

Figure 13. Overview of the steps to develop an ontology network by combining the reuse of several types of knowledge resources

6.2.2. Sequence proposed to obtain the conceptualization products

During the ICPS use case, we have realized that it has not been clearly stated which is the most appropriate order to generate the useful graphics obtained in the conceptualization activity (the upper-level conceptual model shown in Figure 10 and the combination of scenarios shown in Figure 11) to make them as useful as possible during the ontology network development. Moreover, we think that it could be interesting to use the flowchart shown in Figure 13 together with the aforementioned graphics to guide the rest of the development process.

From our point of view, after a quick search to find possible knowledge resources to be reused, ontology developers should follow this sequence within the conceptualization activity:

1. To create the upper-level conceptual model: In this model, ontology developers must indicate which are the knowledge resources suitable to be reused and in which part of the ontology network they could be useful to fulfil the requirement.
2. To create the combination of scenario figure: To carry out this point, it is useful to bear in mind the output from the quick search and the upper-level conceptual model created in the previous step. With this information and the guidelines proposed in [Suárez-Figueroa et al., 2008b] ontology developers could create a graphic that represents the interaction and sequence among scenarios in the ontology network development. At this step the output from the quick search should be analyzed enough to select the scenarios as precise as possible. For example, we can analyse the knowledge resources appearing in the upper-level conceptual model to select or reject different knowledge resources reuse process (ontological reuse, non-ontological reuse, ODP reuse). Moreover, we can decide whether it will be necessary to carry out a re-engineering process on a given ontological resource.
3. To follow the flowchart shown in Figure 13: According to the information of the upper-level conceptual model (step 1) and the figure about scenarios combination (step 2), ontology developers should follow the flowchart proposed in Figure 13 in order to get an idea of how to face the ontology network implementation from a knowledge resource point of view. At this step ontology developers should have decided which knowledge resources will be reused; to take this decision they could bear in mind the entire output from the quick search, but above all, the knowledge resources appearing in the upper-level conceptual model. Ontology developers should base on the figure that contains the scenario combination to select or reject the optional activities or scenarios represented in Figure 13 with rectangles with discontinuous line; for example, if the Ontological Resource Re-engineering scenario does not appear in the figure that contain the scenarios combination it should not be necessary to pass by the “Ontological Resource Re-engineering” box in Figure 13.

6.2.3. Preliminary guidelines proposed to select knowledge resources

The flowchart shown in Figure 13 divides the useful knowledge resources (in the case of having any suitable resource to be reused) into two types: (1) those that are suitable for using as the ontology network development starting point, and (2) those that are suitable for completing the ontology network development. To determine how to reuse the knowledge resources suitable to be reused is an important decision that ontology developers must take during the ontology network development. Such a decision depends on several factors, thus, to decide to which group a given resource belongs we could consider the following questions:

- How much does the knowledge resource fit the ontology network requirements?. That is, does the resource cover a large or a small part of the ontology network requirements? This question can be tackled in different ways depending on the nature of the knowledge resource, for instance:

- In the case of ontological and non-ontological resources, we can analyse the resource content in order to determine how much concepts fit the ontology requirements.
- In the case of ODPs, we identified those possible patterns by analysing the CQs in the ORSD; therefore we know how many CQs could be covered by reusing ODPs. We can look for ODPs that match the ontology network requirements with the use case[57] or CQ field[58] of the ODP. We can also identify quickly a possible ODP while reading the CQs, for example:
  - If a CQ mentions something about a list of values, we could infer that the LP-SV-01 [Suárez-Figueroa et al., 2007b] could be reused.
  - If a CQ is about types and subtypes of a given concept, we could infer that the LP-SC-01 or the AP-TX-01 [Suárez-Figueroa et al., 2007b] could be reused.
  - If a CQ contains more than two concepts related, we could infer that the LP-NR-01 [Suárez-Figueroa et al., 2007b] or the CP-NPAR-01 [Presutti et al., 2008] could be reused.
  - If a CQ is about parts of something, we could infer that the CP-PW-01 or the CP-PW-02 [Suárez-Figueroa et al., 2007b] could be reused.

- The level of consensus of the knowledge resource and its reliability. The quality of a resource is usually greater if their content is consensuated in the environment to which belongs.
- The knowledge resource usability from a human point of view.
- The type of knowledge resource (ontological resource, non-ontological resource, and ODP). This is a complex parameter because it is no clear which is the best knowledge resource type to be reused. For example, depending on the ontology development team profile, we can have the following cases:
  - If ontology developers are not programming experts, it could be extremely difficult to create an ad-hoc tool to transform automatically a non-ontological resource into an ontology, or they could waste too much time transforming the resource manually.
  - If ontology developers do not know how to manage an ontology or an ontology development tool, it could be easier to use a non-ontological resource, like a database or an excel file and convert it into an ontology.

[57] This field contains a description in natural language of the general problem addressed by the ODP [Suárez-Figueroa et al., 2007b].

[58] This field contains the CQ that the ODP from http://ontologydesignpatterns.org/wiki/Main_Page is able to fullfil.

7. Conclusions and Future Work

As mentioned in D5.4.1 [Suárez-Figueroa et al., 2008a], our aim within the NeOn project is to create the *NeOn Methodology for building ontology networks,* covering the drawbacks presented in three well-known methodologies (METHONTOLOGY, On-To-Knowledge, and DILIGENT), and benefiting from the advantages included in such methodologies.

Therefore, the first version of the NeOn Methodology for building ontology networks [Suárez-Figueroa et al., 2008a] presents the following contributions:

- Analysis of how argumentation and collaboration aspects are related to the different nine identified scenarios for building collaboratively network of ontologies.
- Prescriptive methodological guidelines for carrying out the ontology specification activity, including three examples of how to apply the methodological guidelines proposed.
- Methodological guidelines for reusing and re-engineering non-ontological resources.
- Prescriptive methodological guidelines for reusing ontological resources, focused on general or common ontologies, domain ontologies as a whole, and ontology statements.
- Methodological guidelines for reusing ontology design patterns by inexpert users.

Additionally, in deliverable D5.3.2 [Suárez-Figueroa et al., 2008b], some methodological guidelines for scheduling ontology development projects are included.

The second version of the NeOn Methodology for building ontology networks is included in D5.4.2 [Suárez-Figueroa et al., 2009] and incorporates the following contributions:

- Overview of the scenarios for building ontology networks identified in the NeOn Methodology.
- Summary of how to develop ontology networks.
- Explanation of the difference between single ontologies and ontology networks.
- Summary and update of the proposed methodological guidelines for non-ontological resource reuse and re-engineering processes, presented in D5.4.1 [Suárez-Figueroa et al., 2008a].
- Methodological guidelines for carrying out the ontology design pattern reuse.
- Methodological guidelines for the ontology modularization activity.
- Methodological guidelines for the ontology (network) evaluation.
- Methodological guidelines for the ontology evolution.
- Methodological guidelines for carrying out the ontology localization activity.

The third version of the NeOn methodology is included in this deliverable (D5.4.3) and is focused on presenting:

- Methodological guidelines for carrying out the ontology module reused process;
- Methodological guidelines for carrying out the ontology mapping activity;
- Methodological guidelines for carrying out the ontology merging activity; and
- Methodological guidelines for selecting and combining non-ontological resources, ontological resources, and ontology design patterns for building ontology networks.

Furthermore, the current and future methodological work (methods, techniques and tools) is focused on:

- Improving the methodological guidelines for reusing ontology modules by considering also modules that come from general or common ontologies.
- Improving the methodological guidelines for merging ontologies;
- Improving the methodological guidelines for selecting, comparing and combining non-ontological resources, ontological resources, and ontology design patterns for building ontology networks;
- Presenting guidelines for deciding which implementation language is better for each type of ontologies;
- Proposing guidelines for deciding which knowledge should be represented as concepts, as relationships, etc.; and
- Defining naming conventions to be used in the ontology network development.

References

C. Bezerra, F. Freitas, J. Euzenat, A. Zimmermann. (2009). *An Approach for Ontology Modularization*. In Philippe Olivier Alexandre Navaux (Ed.): *Proceedings of Colibri – Colloquium of Computation: Brazil / INRIA, Cooperations, Advances and Challenges*. July 2009.

C. Bezerra, F. Freitas, J. Euzenat, A. Zimmermann. (2008). *ModOnto: A tool for modularizing ontologies.* In Fred Freitas, Heiner Stuckenschmidt, Sofia Pinto, Andreia Malucelli, Oscar Corcho (Eds.): *Proceedings of the 3rd Workshop on Ontologies and their Applications*, WONTO 2008.

A. Cadenas. (2008). *E1.1.1 Especificación de requisitos de las ontologías para la representación de la interacción usuario-móvil-servicios/entorno*. Proyecto mIO!. Diciembre 2008.

B. Cuenca Grau, B. Parsia, E. Sirin, A. Kalyanpur. (2006). *Modularity and web ontologies*. In Proceedings of the Tenth International Conference on Principles of Knowledge Representation and Reasoning (KR-2006), Lake District of the United Kingdom, pp. 198–209. AAAI Press. June 2006.

M. d'Aquin, A. Schlicht, H. Stuckenschmidt, M. Sabou. (2009). *Criteria and Evaluation for Ontology Modularization Techniques*. "Ontology Modularization", Christine Parent, Stefano Spaccapietra, Heiner Stuckenschmidt (editors). Springer, (In press) 2009.

M. d'Aquin, J. Euzenat, C. Le Duc, H. Lewen. (2009b). *Sharing and Reusing Aligned Ontologies with Cupboard*. K-CAP 2009. September 1–4, 2009, Redondo Beach, California, USA.

M. d'Aquin, P. Haase, C. Le Duc, A. Zimmermann. (2008). *NeOn D1.1.4 NeOn Formalism for Modularization: Implementation and Evaluation*. NeOn project. http://www.neon-project.org. 2008.

M. d'Aquin, A. Schlicht, H. Stuckenschmidt, M. Sabou. (2007). *Ontology modularization for knowledge selection: Experiments and evaluations*. In Roland Wagner, Norman Revell, and Gunther Pernul, editors, Database and Expert Systems Applications, 18th International Conference, DEXA 2007, Regensburg, Germany, September 3-7, 2007, Proceedings, volume 4653 of Lecture Notes in Computer Science, pages 874–883. Springer, 2007.

M. d'Aquin, M. Sabou, M. Dzbor, C. Baldassarre, L. Gridinoc, S. Angeletou, E. Motta. (2007b). *Watson: A Gateway for the Semantic Web*. Poster session of the European Semantic Web Conference, ESWC 2007.

P. Doran , V. Tamma , L. Iannone. (2007). *Ontology module extraction for ontology reuse: an ontology engineering perspective.* Proceedings of the sixteenth ACM conference on Conference on information and knowledge management (CIKM), November 06-10, 2007, Lisbon, Portugal.

P. Doran. (2006). *Ontology Reuse via Ontology Modularisation*. In Proceedings of KnowledgeWeb PhD Symposium 2006 (KWEPSY2006). Budva, Montenegro. 17th June 2006.

D. Dou, D. Mcdermott, P. Qi. (2003). *Ontology Translation on the Semantic Web*. In Proceedings of International Conference on Ontologies, Databases and Applications of Semantics (ODBASE 2003). LNCS 2888. Springer-Verlag. Catania (Sicily), Italy. 2003. Pgs. 952-969.

J. Euzenat, A. Mocan, F. Scharffe. (2008). *Ontology alignments: an ontology management perspective*. In: Martin Hepp, Pieter De Leenheer, Aldo De Moor, York Sure (eds), Ontology management: semantic web, semantic web services, and business applications, Springer, New-York (NY US), 2008, pp177-206.

J. Euzenat, P.Shvaiko. (2007). *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2007.

M. Fernández-López, A. Gómez-Pérez, J.A. Ramos. (2006). *Estudio y formalización del proceso de mezcla de ontologías*. Revista Iberoamericana de Inteligencia Artificial. Vol 10 – 31, Pages: 81-89. 2006.

J. Golbeck, G. Fragoso, F. Hartel, J. Hendler, B. Parsia, J. Oberthaler. (2003). *The national cancer institute's thesaurus and ontology.* Journal of Web Semantics, 1(1), 2003.

A. Gómez-Pérez. (2001). *Evaluation of Ontologies*. International Journal of Intelligent Systems, 16 (3). 2001.

E. Jimenez-Ruiz, B. Cuenca Grau, U. Sattler, T. Schneider, R. Berlanga. (2008a). *Safe and Economic Re-Use of Ontologies: A Logic-Based Methodology and Tool Support.* European Semantic Web Conference (ESWC), pp. 185- 199, LNCS Proceedings vol 5021, 2008. http://dx.doi.org/10.1007/978-3-540-68234-9 16.

E. Jimenez-Ruiz, B. Cuenca-Grau, U. Sattler, T. Schneider, R. Berlanga. (2008b). *ProSÉ: A Protégé Plug-in for Reusing Ontologies: Safe and Économique*. Demo paper in the Spanish Conference on Databases and Software Engineering, JISBD 2008.

K. Kotis, G. Vouros. (2004). *HCONE approach to Ontology Merging. The Semantic Web: Research and Applications*, First European Semantic Web Symposium, ESWS 2004, Heraklion, Crete, Greece, May 10-12, 2004, Proceedings, Series: Lecture Notes in Computer Science, Vol. 3053, Davies, J.; Fensel, D.; Bussler, C.; Studer, R. (Eds.), Springer-Verlag. 2004.

DL. McGuinness, R. Fikes, J. Rice, S. Wilder. (2000). *An Environment for Merging and Testing Large Ontologies*. Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR2000). Breckenridge. Colorado, 2000.

R. Mizoguchi, J. Vanwelkenhuysen, M. Ikeda. (1995). *Task Ontology for reuse of problem solving knowledge*. In: Mars N (ed.) Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing (KBKS 1995). University of Twente, Enschede, The Netherlands. IOS Press, Amsterdam, The Netherlands, pp 46–57.

E. Montiel-Ponsoda, M. Poveda, M. C. Suárez-Figueroa, O. Corcho. (2009b). *Conceptualization Document for ICPS Ontologies: Falls Ontology*. ICPS: International Classification for Patiente Safety Project. November 2009.

E. Montiel-Ponsoda, M. Poveda, M. C. Suárez-Figueroa, O. Corcho. (2009c). *Conceptualization Document for ICPS Ontologies: Pressure Ulcer Ontology*. ICPS: International Classification for Patiente Safety Project. November 2009.

E. Montiel-Ponsoda, M. Poveda, M. C. Suárez-Figueroa, O. Corcho. (2009a). *Ontology Requirements Specification Document (for the domains of Falls and Pressure Ulcer)*. ICPS: International Classification for Patiente Safety Project. September 2009 (updated in November 2009).

NF. Noy, MA. Musen. (2000). *PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment*. Proceedings of the Seventeenth National Conference on Artifical Intelligence (AAAI-2000). Austin, Texas. 2000.

M. Poveda, M.C. Suárez-Figueroa, A. Gómez-Pérez. (2009). *Common Pitfalls in Ontology Development*. XIII Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA 2009). Sevilla (Spain). November 2009.

V. Presutti, A. Gangemi, S. David, G. Aguado de Cea, M.C. Suárez-Figueroa, E. Montiel-Ponsoda, M. Poveda. (2008). *NeOn D.2.5.1. A Library of Ontology Design Patterns: reusable solutions for collaboratively design of networked ontologies*. http://www.neon-project.org. February 2008.

J.A. Ramos. (2001). *Mezcla automática de ontologías y catálogos electrónicos*. Grade Project. Facultad de Informática, Universidad Politécnica de Madrid. Boadilla del Monte, Madrid, Spain. 2001.

G. Stumme, A. Maedche. (2001). *FCA-Merge: Bottom-Up Merging of Ontologies*. Proceedings of the 17th International Joint Conference on Artificial Intelligent (IJCAI 2001). Seattle, USA, August 2001.

M. C. Suárez-Figueroa, A. Gómez-Pérez, B. Villazón-Terrazas. (2009b). *How to Write and Use the Ontology Requirements Specification Document*. On the Move to Meaningful Internet Systems: OTM 2009, Confederated International Conferences, CoopIS, DOA, IS, and ODBASE 2009. Vilamoura, Portugal, November 1-6, 2009, Proceedings, Part II. Pages: 966-982.

M. C. Suárez-Figueroa, E. Blomqvist, M. D'Aquin, M. Espinoza, A. Gómez-Pérez, H. Lewen, I. Mozetic, R. Palma, M. Poveda, M. Sini, B. Villazon-Terrazas, F. Zablith, M. Dzbor. (2009). *NeOn D5.4.2. Revision and Extension of the NeOn Methodology for Building Contextualized Ontology Networks.* NeOn project. http://www.neon-project.org. February 2009.

M. C. Suárez-Figueroa, A. Gómez-Pérez. (2008). *First Attempt towards a Standard Glossary of Ontology Engineering Terminology*. The 8th International Conference on Terminology and Knowledge Engineering (TKE 2008). Managing ontologies and lexical resources. ISBN: 87-91242-50-9. Pps: 1-15. Copenhagen, Denmark. August 18-August 21, 2008.

M. C. Suárez-Figueroa, M. Fernández-López, A. Gómez-Pérez, K. Dellschaft, H. Lewen, M. Dzbor. (2008b). *NeOn D5.3.2. Revision and Extension of the NeOn Development Process and Ontology Life Cycle*. NeOn project. http://www.neon-project.org. November 2008.

M. C. Suárez-Figueroa, G. Aguado de Cea, C. Buil, K. Dellschaft, M. Fernández-López, A. García, A. Gómez-Pérez, G. Herrero, E. Montiel-Ponsoda, M. Sabou, B. Villazon-Terrazas, Z. Yufei. (2008a). *NeOn D5.4.1. NeOn Methodology for Building Contextualized Ontology Networks*. NeOn project. http://www.neon-project.org. February 2008.

M. C. Suárez-Figueroa, G. Aguado de Cea, C. Buil, C. Caracciolo, M. Dzbor, A. Gómez-Pérez, G. Herrero, H. Lewen, E. Montiel-Ponsoda, V. Presutti. (2007a). *NeOn Deliverable D5.3.1. NeOn Development Process and Ontology Life Cycle*. NeOn Project. http://www.neon-project.org. August 2007.

M.C. Suárez-Figueroa, S. Brockmans, A. Gangemi, A. Gómez-Pérez, J. Lehmann, H. Lewen, V. Presutti, M. Sabou. (2007b). *NeOn Deliverable 5.1.1. NeOn Modelling Components*. At www.neon-project.org. http://www.neon-project.org. March 2007.

G. van Heijst, A.T. Schreiber, B.J. Wielinga. (1997). *Using explicit ontologies in KBS development*. International Journal of Human-Computer Studies 45:183–292. 1997.

L.M. Vilches-Blázquez, J.A. Ramos, F.J. López-Pellicer, O. Corcho, J. Nogueras-Iso. (2009). *An approach to comparing different ontologies in the context of hydrographical information*. Information Fusion and Geographical Information Systems (IF&GIS'09). Popovich et al., (eds.) Lecture Notes in Geoinformation and Cartography (LNG&C) Springer-Verlag. ISBN: 978-3-642-00303-5. San Petersburgo, Rusia. May, 2009. Pp: 193-207.