Chapter 1
Introduction: Ontology Engineering in a Networked World

Mari Carmen Suárez-Figueroa, Asunción Gómez-Pérez, Enrico Motta, and Aldo Gangemi

Abstract While ontology engineering is rapidly entering the mainstream, expert ontology engineers are a scarce resource. Hence, there is a need for practical methodologies and technologies, which can assist a variety of user types with ontology development tasks. To address this need, this book presents a scenario-based methodology, the NeOn Methodology, which provides guidance for all main activities in ontology engineering. The context in which we consider these activities is that of a networked world, where reuse of existing resources is commonplace, ontologies are developed collaboratively, and managing relationships between ontologies becomes an essential aspect of the ontological engineering process. The description of both the methodology and the ontology engineering activities is grounded in a comprehensive software environment, the NeOn Toolkit and its plugins, which provides integrated support for all the activities described in the book. Here we provide an introduction for the whole book, while the rest of the content is organized into 4 parts: (1) the NeOn Methodology Framework, (2) the set of ontology engineering activities, (3) the NeOn Toolkit and plugins, and (4) three use cases. Primary goals of this book are (a) to disseminate the results from the NeOn project in a structured and comprehensive form, (b) to make it easier for students and practitioners to adopt ontology engineering methods and tools, and

M.C. Suárez-Figueroa (✉) • A. Gómez-Pérez
Ontology Engineering Group, Facultad de Informática, Universidad Politécnica de Madrid, Campus de Montegancedo sn., 28660 Boadilla del Monte, Madrid, Spain
e-mail: mcsuarez@fi.upm.es; asun@fi.upm.es

E. Motta
Knowledge Media Institute (KMi), The Open University, Walton Hall, Milton Keynes MK7 6AA, UK
e-mail: e.motta@open.ac.uk

A. Gangemi
Semantic Technologies Lab, Institute of Cognitive Sciences and Technologies (National Research Council – CNR), Via Nomentana 56, 00161 Rome, Italy
e-mail: aldo.gangemi@cnr.it

M.C. Suárez-Figueroa et al. (eds.), *Ontology Engineering in a Networked World*,
DOI 10.1007/978-3-642-24794-1_1,

(c) to provide a textbook for undergraduate and postgraduate courses on ontology engineering.

1.1 Introduction

The Semantic Web is characterized by the existence of a very large number of distributed semantic resources, which subscribe to alternative but often overlapping modeling schema (i.e., ontologies). Together these resources define a *network of ontologies* related through a variety of different meta-relationships such as versioning, inclusion, inconsistency, similarity, and others. This emerging scenario is radically different from the relatively narrow contexts in which ontologies have been traditionally developed and applied, and calls for new methods and tools to support effectively the development of a new kind of network-oriented semantic applications.

Hence, ontologies on the Web are not stand-alone artifacts. They relate to each other in ways that might affect their meaning, and are inherently distributed in a network of interlinked semantic resources. More precisely, a *network of ontologies* or an *ontology network* is a collection of ontologies related together via a variety of relationships, such as alignment, modularization, version, and dependency. Accordingly, a *networked ontology* is an ontology included in such a network, sharing relationships with a potentially large number of other ontologies.

Intuitively, this aspect of considering ontologies as included in a network implies that they are defined not only through their content but also in terms of ontology metadata, which provide information about their provenance, purpose, and the relations with other ontologies and semantic resources, among other things.

One of the most common ways for two ontologies to relate is to be dependent on each other. More precisely, it is often the case that in order to define its own model, an ontology refers to the definitions included in another ontology. The OWL language includes a primitive (owl:imports) allowing an ontology developer to declare such a relationship, merging the definitions of the imported ontology with those from the importing one.

Aligning ontologies is a way to put different models in correspondence by declaring which entities in one ontology are the same as those in another ontology, or a generalization or specialization. The main purpose of alignments is to ensure semantic interoperability, making it possible to merge ontologies in a meaningful way by representing information in one ontology in terms of the entities in another.

Large, monolithic ontologies are hard to manipulate, use, and maintain. Modular ontologies on the contrary divide the ontological model in self-contained, interlinked components, which can be considered independently, while at the same time participate to the definition of a specific aspect of an ontology. Therefore, modules share the relation that they are common components of a larger ontology, and often include dependencies and alignments to other modules.

Finally, versioning relates to the activity of keeping track of the different versions of an ontology. This is of particular importance in a collaborative ontology engineering environment, where the ontology evolution process needs to be carefully monitored and managed. The OWL language includes primitives to declare versioning relations between ontologies, but these do not consider fine-grained changes and are not often used in practice.

In this networked world, ontology practitioners need both methodological and technological support for the development and use of ontology networks. We aim to provide such a support in this book.

1.2 NeOn Methodology Framework

One of the main contributions of this book is the NeOn Methodology framework, which is described in the first part of this book. Although methodological approaches already exist in the literature – e.g., METHONTOLOGY, On-To-Knowledge, and DILIGENT, they do not provide the comprehensive set of methods described in the NeOn Methodology, especially with respect to key activities in a network-centric scenario, such as those related to reusing and managing the dynamics of ontologies.

The *NeOn Methodology* (Chap. 2) uses a scenario-based approach to ontology development and provides a comprehensive set of methods and guidelines for carrying out the variety of activities required when developing ontologies in a networked world.

The NeOn Methodology includes (1) a set of nine scenarios that involve different activities for collaboratively building ontologies and ontology networks, (2) a glossary of processes and activities relevant to ontology development, (3) a collection of ontology life cycle models, and (4) a set of methodological guidelines. The NeOn Methodology defines each process or activity in a precise manner, stating its purpose, inputs and outputs, the actors involved, when its execution is more appropriate, and a set of proposed methods, techniques, and tools to be used. The methodology is presented in a prescriptive way to facilitate its adoption by students and practitioners.

Current methodologies for ontology engineering, such as METHONTOLOGY, On-To-Knowledge, and DILIGENT, mainly include guidelines for single ontology construction, ranging from ontology requirements specification to ontology implementation, and they are mainly targeted to ontology researchers. In contrast to the aforementioned approaches, the NeOn Methodology does not prescribe a rigid workflow but instead suggests pathways and activities for a variety of scenarios. The nine scenarios described in the book cover commonly occurring situations, e.g., when existing ontologies need to be reengineered, aligned, modularised, localized to support different languages and cultures, or integrated with non-ontological resources (NORs), such as folksonomies or thesauri.

Another important aspect of the NeOn Methodology is the pattern-based design approach described in Chap. 3. In this chapter, different types of ontology design patterns (ODPs) are presented as well as an associated method (named eXtreme Design) to assist in ontology development. Ontology design patterns provide modeling solutions which can be applied to solve recurrent ontology design problems. The availability of a library of ontology design patterns is an important step toward achieving the ultimate goal of turning ontology design into a structured and reproducible engineering process. The pattern library also includes patterns for reengineering non-ontological resources (such as thesauri, classification schemas, etc.) into ontologies.

In addition, as part of the methodological framework, three models are proposed to represent information about ontology networks. They play a critical role, as they allow keeping track of the provenance, purpose, and design of ontologies, as well as covering multilinguality issues. The three models are:

- The *Ontology Metadata Vocabulary (OMV)*. An ontology that defines classes and relations to describe authoring aspects, ontology type, purpose, etc.
- The *Collaborative Ontology Design Ontology (C-ODO)*. An ontology network that enables designers to describe design entities (ontologies, modules, ontology elements, requirements, activities, tools, reusable knowledge, teams, people, etc.).
- The *Linguistic Information Repository (LIR)*. An ontology that defines a set of linguistic classes, whose nature accounts for the localization of ontology terms in a particular language.

1.3 Ontology Engineering Activities

The second part of the book provides the reader with a description of the key activities relevant to the ontology engineering life cycle in a networked world. For each activity, a general introduction, methodological guidelines, practical examples (where possible), and the technological support within the NeOn Toolkit (if available) are provided. Methodological guidelines are explained using a common structure, which includes process or activity definition, goal, input and output, actors involved, and a graphical workflow, which describes how the process or activity should be carried out. This structured way of explaining the guidelines maximizes the pedagogical value of the book.

The starting point to develop an ontology network is the gathering of the requirements the ontology should fulfill. This activity is called ontology requirements specification and is described in Chap. 5.

Once requirements are collected, ontology practitioners are encouraged to follow a reuse approach in the ontology building process, which allows speeding up the ontology network development process, saving time and money, and promoting the application of good practices. In this context, both non-ontological resources (Chap. 6) and ontological resources (Chap. 7) can be reused.

An important aspect in a networked world, which involves different natural languages and cultures, is the localization of the ontologies. This activity is described in Chap. 8.

Another key aspect in the ontology network development is the ontology evaluation activity, which is performed at different levels and according to different criteria, as explained in Chap. 9.

Additionally, modularization also needs to be taken into account in the ontology network development according to three different aspects: (1) designing modular ontologies, (2) modularizing existing ontologies, and (3) reusing ontology modules. Methodological guidelines for modularizing existing ontologies are presented in Chap. 10.

Ontology networks need to be kept up to date in order to reflect changes and updates. To this purpose, methodological guidelines for ontology evolution are provided in Chap. 11.

Finally, finding alignments between ontologies is an important task for ontology engineering in a networked world, and is covered in Chap. 12, which provides methodological guidelines for this activity.

1.4 The NeOn Toolkit

The third part of the book presents an overview of the NeOn Toolkit (Chap. 13), focusing in particular on the user interaction side and a detailed description of the plugins, which are most critical to the ontology development process.

Proper management of ontology engineering projects in a networked world requires careful planning, and to this purpose it is recommended that an ontology project plan and schedule is defined. To support this activity, a NeOn Toolkit plugin, called gOntt, has been developed, which is described in Chap. 14.

The tasks of locating, selecting, and accessing NeOn Toolkit plugins are supported by the Kali-ma plugin (Chap. 15). This plugin, which exploits the versatility of the C-ODO Light model, assists ontology engineers and project managers in carrying out such tasks through a unified, shared interaction mode.

Visualizing and navigating ontology networks is a key issue for ontology engineering. In this sense, the NeOn Toolkit provides a novel plugin called KC-Viz (Chap. 16), which exploits an innovative ontology summarization method to support a "middle-out ontology browsing" approach, where it becomes possible to navigate ontologies starting from the most information-rich nodes (*key concepts*).

Finally, reasoning with ontology networks is another key activity in ontology engineering. Chapter 17 presents (a) the NeOn Toolkit query plugin, which allows users to query ontologies in the NeOn Toolkit via the RDF query language SPARQL, (b) the NeOn Toolkit reasoning plugin, which allows for standard reasoning tasks, such as materializing inferences and checking consistency in ontologies, and (c) the RaDON plugin, which supports users in diagnosing and resolving inconsistencies in networked ontologies.

1.5 Case Studies

The fourth and last part of the book describes how the NeOn methods and tools have been applied in three real-world case studies in the fishery and pharmaceutical domains.

- Knowledge management at FAO (Chap. 18). This case study is centered on fisheries data[1] and aims to build a system to enable fisheries experts to have a unified view of the distributed data relevant to fisheries stocks. The result is a prototype of a Fisheries Stock Depletion Assessment System (FSDAS), which illustrates the advantages derived from enriching data with explicit semantics.
- Electronic invoice management in the pharmaceutical sector: the PharmaInnova case (Chap. 19). This chapter deals with the development of an ontology network for automating the exchange of electronic invoices in the pharmaceutical sector.
- Integrating product information in the pharmaceutical sector (Chap. 20). This case study focuses on the development of a network of interconnected pharmaceutical ontologies to provide an integrated view of different drug terminologies.

[1] http://www.fao.org/fishery/en