INSUBRIA UNIVERSITY

Department of Theoretical and Applied Sciences

XXXV CYCLE OF THE PH.D. PROGRAM IN COMPUTER SCIENCE AND THE MATHEMATICS OF COMPUTATION

A novel and validated agile Ontology Engineering methodology for the development of ontology-based applications

Ph.D. candidate: Daniele Spoladore

Ph.D. candidate ID: 742355

Supervisor: prof. Alberto Trombetta

ACADEMIC YEAR 2021/2022

Abstract

The goal of this Thesis is to investigate the status of Ontology Engineering, underlining the main key issues still characterizing this discipline. Among these issues, the problem of reconciling macro-level methodologies with authoring techniques is pivotal in supporting novice ontology engineers. The latest approach characterizing ontology engineering methodologies leverages the agile paradigm to support collaborative ontology development and deliver efficient ontologies.

However, so far, the investigations in the current support provided by these methodologies and the delivery of efficient ontologies have not been investigated. Thus, this work proposes a novel framework for the investigation of agile methodologies, with the objective of identifying the strong point of each agile methodology and its limitations. Leveraging on the findings of this analysis, the Thesis introduces a novel agile methodology – AgiSCOnt – aimed at tackling some of the key issues characterizing Ontology Engineering and weaknesses identified in existing agile approaches.

The novel methodology is then put to the test as it is adopted for the development of two new domain ontologies in the field of health: the first is dedicated to patients struggling with dysphagia, while the second addresses patients affected by Chronic obstructive pulmonary disease.

Preface

This Thesis is based on the results of two research papers on collaborative and agile Ontology Engineering, which is a topic addressed during this candidate's work at the Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing (STIIMA) of the National Research Council of Italy (CNR). A part of this work constitutes some of the results of the activities conducted by STIIMA-CNR of the research project HUB sPATIALS[3]. The research project, recognizing that nutrition plays a pivotal role in human well-being, especially in people characterized by specific health conditions that could be exacerbated by malnutrition, foresaw STIIMA-CNR to develop ontology-based Decision Support Systems to support patients and clinical personnel in managing different nutrition-related diseases. The candidate was the principal researcher in charge of ontology engineering.

Acknowledgments

I owe my gratitude to my STIIMA-CNR coworkers for their support over these years. Sara Arlati provided me with valuable critical comments and precious suggestions. Vera Colombo supported me with her knowledge and patience during HUB sPATIALS^3's project activities. Elena Pessot shared my passion for Ontology Engineering and its methodologies.

I owe affection and gratitude to Elisabetta Oliveri and Rossella Scaioli, who entrusted me with two classes of students from the Advanced professional Master courses on Industry 4.0 they organized, granting me access to the samples required for my experiments.

To Marco Sacco, Head of Research in STIIMA-CNR, my thanks for trusting in my research and granting me the opportunity to prove its efficacy in many research projects.

Last but not least, I express my gratitude to my supervisor, Prof. Alberto Trombetta, for supporting this Thesis and guiding me over these three years.

Table of contents

Chapter 1 – Introduction

This Chapter provides an Introduction to this Thesis, presenting the main definitions and setting the background for the remaining Chapters. The notions of ontology and Ontology Engineering are illustrated from a historical perspective. At the same time, an overview of the existing ontological languages provides some insights into the main languages adopted in this work. Finally, the Chapter concludes with an illustration of the approach adopted in this Thesis and highlights the main research contributions it aims at delivering.

1.1 Semantic Web and ontology

The Semantic Web was originally proposed by Tim Berners-Lee as a way to tackle and solve the problem of semantic heterogeneity of data in the web. This phenomenon consists in having the same information represented in different ways: different data schemas representing the same domain are modelled by different people whose styles, visions, and needs are likely to diverge [1]. The Semantic Web, acting as an extension of the Web, foresees the information having a well-defined meaning to enable the computer to perform richer and semantically-grounded queries. The Semantic Web requires access to structured collections of information and inference rules (to enable automated reasoning on information). Also, the Semantic Web would require a set of protocols and languages to be fully realized – the Semantic Web Stack, represented in Figure 1.1.

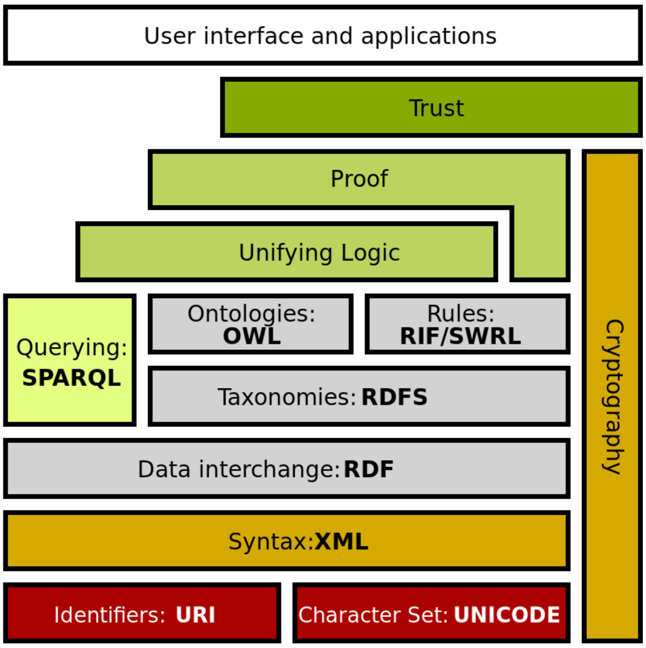

Fig. 1.1. The Semantic Web Stack, a graphical illustration of the architecture of Semantic Web.

Among the technologies represented in the stack, the ontology acts as the collection of information for defining the relationships occurring between the information represented. A more fortunate definition of ontology is given by Gruber in 1993: an ontology is a formal, explicit specification of a conceptualization [2]. The definition hints at the fact that an ontology is expressed in a machine-readable format (formal), which is used to represent the body of knowledge (concepts, objects, relationships, and other entities) composing the abstract representation of a view of the word (conceptualization). To simplify, the ontology is a description, similar to the formal specification of a program, of the concepts and relationships that exist in a domain of knowledge. In 1997, Borst refined the definition: an ontology is a formal specification of a shared conceptualization [3]. This definition underlined the fact that the conceptualization underlying the ontology should be an expression of the view of several parties, a consensus view on a domain rather than an individual perspective. A year later, Studer and colleagues merged the two definitions into a single one: an ontology is a formal, explicit specification of a shared conceptualization [4]. The evolution of the definition of ontology – which was very debated up to the first decade of the 2000s [5] – sets forth a pivotal attribute for the ontology: its underlying conceptualization should be the result of a consensus process aimed at identifying the knowledge, concepts, and relationships holding among them. The definition of ontology provided by Carla Sofia Pereira is emblematic of this shift: ontologies are a form of a priori social agreements made about a conceptualization of a given part of the world [6].

1.2 The "art" and the "craft" of building ontologies

The shift from the early perception of ontology as artifacts to be developed by ontology engineers towards a collaborative effort involving domain experts and ontologists is deepened in Chapter 2 of this work. Nonetheless, the focus on the sharable ontologies, developed in cooperation with different stakeholders, produced an evolution of Ontology Engineering (OE) – i.e., the discipline aimed at studying methods and methodologies for the development of ontologies and aimed at providing ontology evolution through their life cycle.

OE as a discipline was born in the same years Gruber and other researchers started to investigate the ontology (early years of the 90s). The discipline dedicated to the building of ontologies, Ontology Engineering, evolved consequently from the standalone approach to participative approaches. The methodologies for ontology development started to take into account collaborative approaches concurrently with the interest researchers showed in the Semantic Web and the adoption of ontologies in the early 2000s. In the same years, due to the spread and availability of novel methodologies, many researchers underlined the fact that OE was more similar to an art rather than engineering: the absence of standardized sets of activities, life cycles, systematic criteria, and techniques pointed out that OE was similar to art, lacking, therefore, the tools to bring the discipline to its maturity [7].

Nonetheless, OE progressed considerably in the last two decades, and the development of many ontologies (in particular in the field of biomedical sciences) attests that the scientific community's interest in these artifacts did not fade. However, OE still presents some issues that prevent it from reaching its full potential: the (apparent) irreconcilability between macro-level methodologies and ontology authoring, the difficulties in acquiring and processing domain knowledge, the necessity of involving large communities in the development process while delivering working prototypes in an adequate amount of time are just some of the open issues in OE. According to some researchers, although significant, the steps forward in OE transformed in status from art to craft [8]. In more recent years, the emergence of an agile paradigm for OE seems promising in tackling some of the traditional issues of OE; however, most of the agile approach's promises still need to be thoroughly investigated.

1.3 Overview of ontological languages

OE and ontology, in general, contributed to igniting the attention towards ontological languages, i.e., languages that are adopted to develop an ontology. Pulido et al. [9] identified the requirements that ontology languages must meet:

- They must have a compact syntax
- They must be highly intuitive for humans
- Their formal semantics must be well-defined
- They must be powerful enough to represent human knowledge
- They must enable reasoning properties
- They must have the potential to build knowledge bases
- They must be linked with existing web standards to ensure interoperability

In other words, ontological languages should be read by both machines and humans while granting the possibility of developing different vocabularies – and letting them evolve, as it happens in the decentralized web. As highlighted in some work (for example, [10]), some methodologies for OE can be used only with specific tools, which require the adoption of specific ontological languages. The languages can be classified as follows:

- Traditional languages, often based on first-order predicate logic, frame-based, Description Logic
- Web-based languages used to facilitate information interchange on the web

Some ontological languages can belong to both classes. In this Section, an overview of the most relevant ontological languages is presented, with a particular focus on World Wide Web Consortium (W3C) recommendations.

1.3.1 Knowledge Interchange Format (KIF)

The Knowledge Interchange Format (KIF) is a formal language developed for information and knowledge interchange among computer programs [11]. It is characterized by declarative semantics, and its expressions can be understood without using an interpreter to manipulate them. However, it is not primarily intended as a language to facilitate human comprehension. Still, it is helpful for describing representation language semantics and assisting human users in knowledge base translation problems.

KIF requires the developer to adopt a conceptualization of the world composed of objects, functions, and relations: it is a declarative language based the extended version of the first-order predicate calculus. KIF enables the representation of arbitrary sentences. Its high expressiveness may pose a threat to the development of fully-conforming ontologies, which may result in heavyweight models – i.e., they are large and might be less efficient than ontologies adopting more restricted languages.

1.3.2 F-Logic

F-Logic [12] combines conceptual modelling with first-order logic predicate calculus with object-oriented and flag-based languages' characteristics. It is characterized by a declarative and compact syntax. This language also encompasses the possibility of using non-monotonic reasoning rules. Contrary to DL-based languages, F-Logic semantics is a closed-world assumption.

This language can potentially represent any aspect of an object-oriented paradigm, thanks to the integration of conceptual modelling constructs into a coherent, logical framework. Among the features that made its fortune, F-Logic lists the possibility of modelling classes and attributes with domain and range definitions, *is-a* hierarchies with set inclusion of subclasses, and logical axioms between elements of an ontology and its instance.

1.3.3 CYC

CYC [13] is a declarative language based on first-order logic with some extensions. It adopts a reasoning engine to perform several kinds of reasoning. CYC is part of a larger Artificial Intelligence project aimed at specifying a large common-sense ontology. Still, it is not a monolithic ontology – it is a set of micro theories for a set of domains. Although the goal of achieving a common-sense ontology still needs to be reached, it provides formal axiomatic theories for many aspects of common-sense knowledge for developing ontologies for a wide variety of specific domain applications.

1.3.4 Resource Description Framework

The Resource Description Framework (RDF) [14] is a W3C standard for the representation of metadata on the web. RDF model uses the triple (a node for the subject, an arc for the attribute, and another node for the object) to compose graphs. Each part of the triple is identified by a URI. The triple-based model is very flexible and can potentially represent complex concepts. RDF's triple structure provides natural semantic units since all objects are independent entities.

RDF Schema (RDFS) [15] extends the semantic representation data model of RDF. This extension allows RDF to step up to the status of ontological language since it provides the relevant constructs to develop vocabularies. RDFS lets developers define a particular vocabulary for RDF data and specify the kind of object to which these attributes may be applied. This mechanism provides a basic type system for RDF models and the interpretation of RDF expressions.

1.3.5 Ontology Interchange Language (or Ontology Inference Layer)

The Ontology Interchange Language (OIL, also known as Ontology Inference Layer) [16] is based on frames, Description Logic, and web standards. It is compatible with RDFS.

OIL was developed to be highly understandable by humans, leveraging the success of frame-based languages and object-oriented paradigms. It is supported by well-defined formal semantics with established reasoning properties. OIL attempts to match these criteria by unifying the three main aspects that characterize ontological languages: rich expressivity typical of frames, formal semantics, and efficient reasoning typical of Description Logic, a standard proposal for syntactical exchange notations (as provided by RDF).

The language is structured as a set of layered sub-languages: each additional layer adds functionality and complexity to the previous one. Standard OIL captures the modelling primitives that provide expressivity, specifying the semantics and enabling inference. Instance OIL allows representing individuals – with a full-fledged database capability. Heavy OIL adds more representational and reasoning capabilities with a more expressive rule language.

The layered architecture has the advantage of making applications work with a language that has the required expressivity – not more. Also, applications able to process a lower (or higher) level of complexity can still use some of the ontology's aspects (or all of the ontology's aspects).

1.3.6 DARPA Agent Markup Language

The DARPA Agent Markup Language (DAML) is an effort endorsed by the Government of the United States of America to foster the next evolution of the web. DAML consists of two portions, the ontology language and a language for expressing constraints and adding inference rules. It also includes mappings to other semantic web languages (including OIL and RDF).

The ontology language is DAML+OIL [17]. It is based on Description Logic (and on RDF and RDFS) and can count on well-defined semantics. It is completed by an interface language DAML-L. The two languages provide a markup language for the semantic web with expressive power and well-defined semantics for reasoning.

1.3.7 Web Ontology Language

Compatible with most of the other ontological languages, Web Ontology Language (OWL) [18] is the standard de facto for the development of ontologies endorsed by the W3C. Derived from DAML+OIL in 2004, OWL offers expressivity to model data and perform automatic reasoning – although not fully decidable in all its profiles. In 2009 the second version of the language (OWL 2) was released.

Contrary to some of the other languages, some of OWL's constructs can be complex for human readers, and it is not immediately easy to use – therefore, there exist several authoring tools adopting OWL. Developing ontologies in OWL requires the developers to be aware of the constructs to build efficient models. The trade-off between reasoning efficiency and expressiveness of the language brought to the identification of three OWL profiles (dated 2004):

- OWL Full: is characterized by the full expressive power of the language but comes at the cost of inefficient reasoning;
- OWL DL: grants efficient reasoning with some restrictions on the constructs of the language. It is less time-complex but loses the backward compatibility with OWL Full;
- OWL Lite: an even more restricted version of OWL Full, supports a classification hierarchy and simple constraint features, like cardinality constraints. It is particularly appreciated for its ease of use.

With OWL 2, a different set of profiles was trimmed from OWL 2 Full, according to the developing necessities [19, 20]:

- OWL 2 Full: the most expressive of the profiles, allows the use of any construct of the language, but reasoning may be inefficient or undecidable;
- OWL 2 DL: a subset of OWL 2 Full with favorable computational properties. Grants efficient reasoning at the cost of some expressiveness. Compared to OWL (1) DL, it presents more modelling features.
- OWL 2 EL (existential logic): a subset of OWL 2 DL, it is very useful for ontologies characterized by a large number of classes and properties linked together by relationships. Dedicated reasoning algorithms for this profile are available and have been demonstrated to be implementable in a highly scalable way. The EL acronym reflects the profile's basis in the EL family of description logics [EL++], logics that provide only Existential quantification.
- OWL 2 QL (query logic): a subset dedicated to efficiently processing a very large amount of instance data, and query answering is the most important reasoning task to be performed. It contains the necessary constructs to represent the main features of a conceptual model, such as UML class diagrams and ER diagrams. The QL acronym reflects the fact that query answering in this profile can be implemented by rewriting queries into a standard relational Query Language.
- OWL 2 RL (rule logic): a subset dedicated to efficiently processing business rules. The ontology consistency, class expression satisfiability, class expression subsumption, instance checking, and conjunctive query answering problems can be solved in time that is polynomial with respect to the size of the ontology. The RL acronym reflects the fact that reasoning in this profile can be implemented using a standard Rule Language.

Figure 1.2 represents the relationships occurring among the different profiles of OWL, OWL 2, and RDFS.

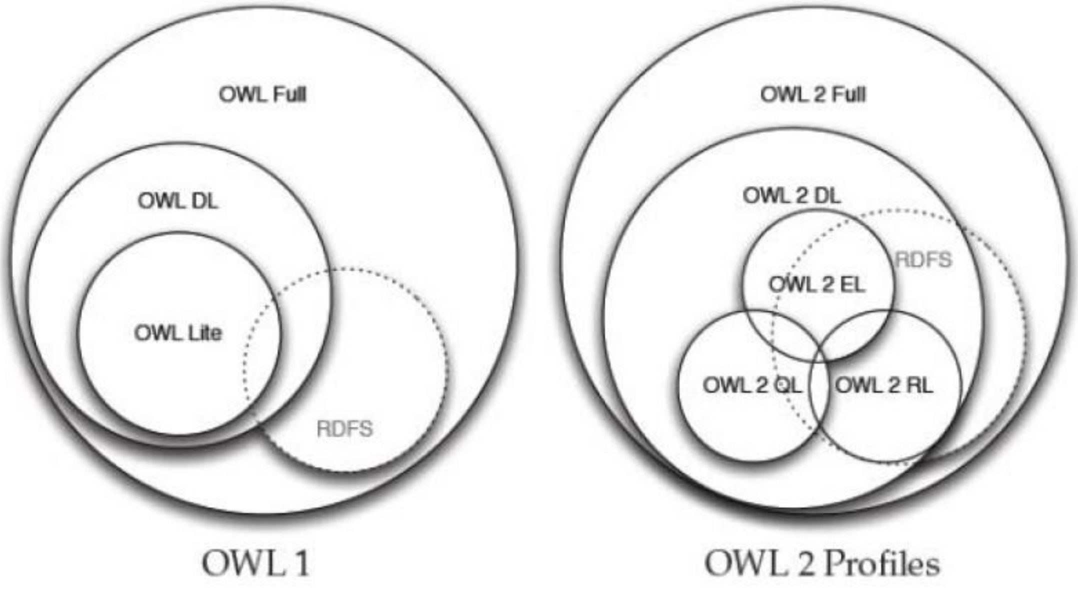

Fig. 1.2: OWL(1) and OWL 2 sublanguages (soruce: [21]*).*

1.3.8 Distributed Ontology, Model and Specification Language

The Distributed Ontology, Modeling and Specification Language (DOL) is maybe the most recent ontological language (2018). It is designed to achieve integration and interoperability among different ontologies. The reasons behind the development of DOL lie in the necessity to make ontologies developed with different ontological languages (such as OWL, F-Logic, etc.) interoperable by providing a unified metalanguage to manage diversity.

DOL provides constructs for the “as-is” use of ontologies, models, and specifications formulated in a specific ontology (or model or specification), whether they are formalized in any language or logic profile. Its use is expected to be in the field of ontology alignment, as it provides the means to represent the meta-knowledge between two (or more) existing ontologies.

1.4 The approach presented in this work

In the previous Sections, the “evolution” of the idea of ontology from standalone development to a cooperative effort was introduced, underlining researchers’ attention toward the creation of methodologies. The Ontology Engineering Methodologies (OEMs) have evolved over the last two decades significantly (as described in Chapter 2), but the discipline is still characterized by a lack of a universally accepted methodology. OEMs are various and different, and their later iteration takes into account agile programming paradigms. This paradigm was introduced as a promising way to avoid some of the well-known limitations of OEMs while focusing on collaborative development.

However, very few OEMs (agile or not) have been investigated from a user perspective – i.e., trying to understand if they really help novel ontologists in developing ontologies. Therefore, it is essential to analyze agile OEMs trying to understand if they can support ontology engineers throughout the OE process. The approach adopted in this work makes use of an experiment involving a sample of ontology engineers who approached the OE for the first time.

Leveraging on the findings of this investigation, a new agile OEM is presented and tested, comparing its results to those gained from other agile OEMs. A second experiment involving novice ontology engineers allows for gathering data on the efficacy of this new agile OEM and comparing it to the existing approaches already investigated.

Finally, the work is concluded by the presentation of two novel domain ontologies developed through 2021 and 2022 using the new agile OEM.

1.4.1 Research contributions and organization of this Thesis

The aim of this work is to trace the evolution of OE and OEMs, underlining the key issues still characterizing the discipline and the widespread adoption of (one or more) OEMs. Focusing on agile paradigm-based OEMs, this Thesis provides a novel framework for OEM’s evaluation, then it introduces a novel agile methodology and tests it.

Chapter 2 of this work delves into OE as a discipline and OEMs evolution. Differently from other existing works in literature, this Thesis adopts a meta-review approach, identifying the features that different researchers deemed relevant for OEMs. This approach allows for capturing some of the open research questions on OE and the adoption of OEMs. In this investigation, the methodologies based on the agile paradigm are also considered, as they constitute the latest type of methodologies.

Chapter 3 faces the issue of OEM’s analysis. In the last two decades, the study of OEMs has always been conducted without taking into account the end-users of methodologies, i.e., novice ontology engineers. These are developers in charge of building one or more ontologies who can benefit from the instructions and suggestions provided by OEMs. However, to the best of this author’s knowledge, there are no studies investigating the methodologies from a user perspective, while the analysis of the adequateness of ontologies developed with specific OEMs is scarce.

Therefore, this Thesis proposes a framework for the analysis of agile OEMs – the latest approach gathering researchers’ attention in OE – taking into account ontologists’ perspectives and the features of the ontologies they developed, with the aim of assessing the OEMs. An experiment involving novice ontology engineers is then set up. The results gathered through the experiment allow to sketch the differences, strong points, and weaknesses of each agile OEM.

Chapter 4 introduces a novel agile OEM. Leveraging on the finding of the experiment, this work introduced a novel agile OEM – AgiSCOnt. The main characteristics of this methodology consist in trying to tackle the issue of reconciling macro-level methodologies' instructions with ontology authoring while fostering the reuse of existing (bits of) knowledge; also, the novel agile methodology adopts existing techniques to involve domain experts and ontologists in a collaborative effort to reach an efficient knowledge elicitation (usually one of the bottlenecks of OE). AgiSCOnt is also tested, following the same framework adopted for other agile OEMs, and compared to the existing agile approaches.

Chapter 5 puts AgiSCOnt to the test. The new agile methodology is adopted for the development of two novel domain ontologies in the clinical research field: taking advantage of a research project, the two ontologies tackle nutrition-related issues for two different populations (patients affected by dysphagia and patients affected by Chronic obstructive pulmonary disease), with the aim of serving as the backbone for two decision support systems. The ontologies are presented following AgiSCOnt's structure, and results pertaining to querying the ontologies are also illustrated.

Chapter 2 – Ontology Engineering and its Methodologies

This Chapter is based on the first Sections of this candidate's work published in Spoladore, Daniele, and Elena, Pessot. "Collaborative ontology engineering methodologies for the development of decision support systems: case studies in the healthcare domain." Electronics 10.9 (2021): 1060 (DOI: 10.3390/electronics10091060).

This Chapter describes the main features and evolution of Ontology Engineering as a discipline, focusing on the aspects that, from the early 1990s, contributed to the debate on methodologies and techniques to model ontologies. The Chapter adopts a meta-review approach to identify the common focuses that researchers adopted throughout the years and delves into authoring techniques. Finally, it concludes with general remarks on the status of the discipline. The Chapter is completed with an Appendix Section, illustrating the result tables of the investigations conducted.

2.1 Macro-level development and ontology authoring

Ontology Engineering (OE) is a discipline aimed at studying methods and methodologies for the development of ontologies and aimed at providing ontology evolution through their life cycle. Ontology Engineering Methodologies (OEMs) can be defined as a set of techniques and methods that guide the development of an ontology. The necessity of guidance in the process of ontology development is evident since from Gruber's definition of ontology – a formal and explicit specification of a shared conceptualization – which underlines different dimensions upon which OEMs can provide support: an ontology engineer may need assistance in selecting the entities to represent the conceptualization, of may require guidance in the cooperative aspects of the engineering process. OE as a discipline can be traced back to the early 1990s when ontologies were proposed as a way to foster information interoperability in a seamless way: considering the two-decade-long history of OE and the different dimensions it may provide support for, researchers have dedicated significant efforts in developing different OEMs. As a consequence, OEMs have evolved over time, together with the discipline, and several methodologies have been published to date: some methodologies stress the formalization aspect, while others originated specifically for supporting cooperative development of ontologies; some others focus on the knowledge elicitation and conceptualization phases. Among this variety of OEMs, no one emerged as a "standard" methodology: on the contrary, different methodologies stress different aspects of the development process while partially neglecting others.

Nonetheless, two fundamental perspectives emerge in OEMs: i) the provision of guidance, suggestion, and methods to move from an informal set of information to an ontology by guiding the ontology engineering throughout the whole OE process. ii) The provision of methodological activities to support the ontology engineer in the formalization process. While the first perspective, which is generally named "macro-level engineering methodologies", is more concerned with the process underlying the OE and the ontology application, the second, named "micro-level engineering methodologies" or "ontology authoring", focuses on the ontology design choices, axiomatization and selecting entities to formalize concepts. Macro-level development OEMs help ontologists answer questions like "which steps need to be performed to get to an ontology?", while ontology authoring OEMs revolves around questions like "how is it possible to model this information?".

Macro- and micro-development are complementary in OE, as each OEM needs to foresee at a certain point some activities dedicated to knowledge formalization using an ontological language. In fact, many eminent researchers underline that authoring per se may be a part of macro-level development OEMs. For instance, Mizoguchi [22], in detailing some practical guidelines for OE, stresses that OE is a process that encompasses authoring activities and is composed of three layers:

- Top-layer, which is the coarsest and more general layer, addresses the whole OE process.
- Middle-layer, which specifies the generic constraints and guidelines used to specify ordered steps in which OE occurs.
- Bottom-layer, dedicated to concept identification and authoring.

The complementarity of macro- and micro-development OEMs is yet to be fully achieved, as most of the OEMs that can be traced in literature are macro-level development ones. Also, ontology authoring has been primarily investigated in relation to the domains for which authoring methods were required – and, therefore, not all micro-development methods can be generalized and applied to all domains of knowledge. A significant exception is represented by Ontology Design Patterns (ODPs), further discussed in Sect. 2.5.

In general, OEMs evolution over time concerned the approaches to OE. In particular, macro-level development methodologies can be characterized by two dimensions: the degree of collaboration and the type of development.

2.2 Macro-level OEMs

Although there exist several macro-level development OEMs, the activities composing the engineering process can be mainly summarized in three stages [23]:

1. Ontology management: this phase addresses all the activities involved in the preparation preceding the actual development, such as feasibility studies, cost-benefit analysis, preliminary identification of the type of ontology to be developed, etc.
2. Ontology development and support: this phase collects the core development activities, including knowledge elicitation and formalization, development (including authoring), and documentation of the ontology and its engineering process. In this phase, the problem domain is analyzed (also adopting Competency Questions (CQs) [24] and motivating scenarios)
3. Ontology use: this phase groups those activities dedicated to maintaining and updating the developed ontology, as well as supporting users in adopting the ontology in applications.

These three phases – represented in Figure 2.1 – encompass each set of activities and are expected to be conducted in an orderly fashion.

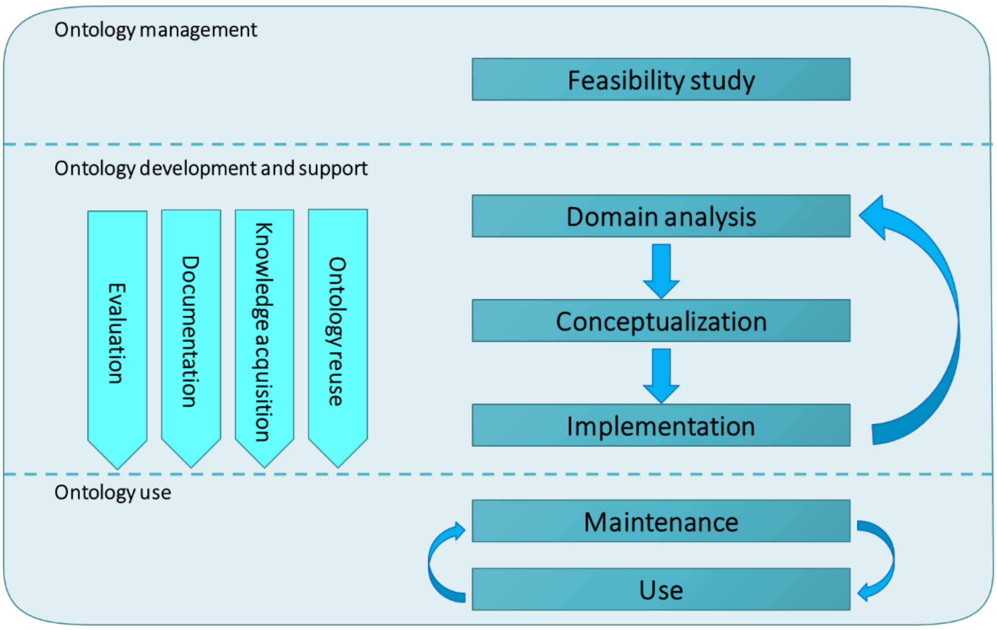

Fig. 2.1. Three phases of OE and their activities (adapted from [23]*).*

In general, the list of activities concerning OE in a macro-level development OEM can be summarized as:

1. Feasibility study: considering the problem for which the ontology is considered to be developed, this task investigates whether the use of a domain ontology or the adoption of an ontology-based application can contribute effectively in solving the issue.
2. Domain analysis: investigating the domain(s) of knowledge involved in the problem.
3. Conceptualization: identifying the main concepts composing the domains of knowledge and the relations occurring among them.
4. Implementation: using formal representation languages to provide an actual implementation of the domain. This activity includes the selection of the ontological languages to be adopted and requires the ontologists to commit themselves to some design choices. The result of Implementation consists in the prototypical ontology.
5. Maintenance: once the prototype is developed, the ontology can still evolve through adaptations to new requirements: in fact, new requirements may come from stakeholders (in the case of collaborative engineering) or because of the emergence of new needs that the ontology is required to cover.
6. Use: after the development, these activities foresee the use of the ontology in applications. Moreover, in these activities, the possible alignment and mapping with other existing ontologies are carried out (i.e., the activity of identifying corresponding concepts between different ontologies).

During the tasks of domain analysis, conceptualization, and implementation, some OEMs stress the importance of knowledge acquisition (often considered one of the bottlenecks of OE [25]), reuse of existing ontologies, evaluation of the ontology, and documentation of the development process. In particular, ontology reuse is deemed a fundamental activity in the Semantic Web [26], and it can occur in different ways.

Starting from these three general phases, researchers have investigated different perspectives and aspects of macro-level development OEMs, taking advantage of the adoption of ontologies in several fields. As a result, there exist many macro-level development OEMs – some of them share many similarities, while others differ greatly. In addition, OEMs' structures and processes did not significantly change even when Semantic Web technologies were deemed as one of the most promising approaches to model data in IoT contexts: some researchers pointed out that the wider adoption of ontology did not lead to the development of a unified OEM [23]. In fact, each OEM stresses different aspects of the ontology engineering process: some of them are mostly focused on domain analysis, while others further investigate the development phase.

OEMs also differ according to the approach underlying the engineering process. Keet [27] proposes a three-sided classification of macro-level development OEMs, adopting the general approach as a taxonomical criterion:

- Waterfall approach: these OEMs foresee an ordered sequence of steps that must be followed to achieve the development of the ontology. The steps are organized according to the list of activities presented in Figure 2.1. Thus, the engineering process starts with the identification of prerequisites and ends with the release of the developed ontology and its adoption. Waterfall OEMs have the ontology engineer play a pivotal role by leading the whole process and controlling the interaction with domain experts – who have a passive role in this case. In general, all waterfall OEMs identify a set of preliminary activities to de conducted before the development stage (a feasibility study and/or the definition of a management framework), which illustrate why the ontology is necessary to solve the problem at hand and identify the main actors (ontologists and domain experts) involved in the development process. These preliminary activities are followed by five main activities: (1) Specification: the domain is investigated and relevant knowledge elicited with domain experts; (2) Conceptualization: the knowledge composing a domain is retrieved and represented in an informal way to obtain a "conceptual model" of the domain(s); (3) Formalization: the conceptual model of the domain obtained in the previous activity is further elaborated into a formal or semi-computable model; (4) Implementation: the output of the previous point is represented by means of the selected ontology languages; (5) Maintenance: after the development of the ontology in its prototypical or final form, the model can be updated, amended or enhanced. Requirements for the modification of the ontology can come from the modifications occurring to the domain of knowledge or from ontology users and stakeholders. As highlighted in early waterfall methodologies, the feasibility study (in general, the preliminary activities) may be excluded, but the "five steps" of specification of the domain, conceptualization, formalization implementation, and use & maintenance are always present. Waterfall OEMs were the first methodologies to be discussed, and the collaborative dimension is not particularly stressed, as domain experts may have a passive role in the engineering process [28].

- Lifecycle approach: this approach thinks of an ontology as an evolving product passing through a lifecycle – thus, an ontology may or may not undergo certain activities. Lifecycle OEMs is usually coupled with an iterative engineering process in which the ontology is expected to go through each group of activities, but not necessarily in an orderly fashion [29]. The lifecycles approach splits the engineering activities into different phases: Requirements development, Ontological analysis, Ontology design, System design, Ontology development and reuse, System development and integration, Deployment, Operation and maintenance. Each phase groups activities around specific inputs, outputs and goals. In this context, some phases are dependent on others: for example, the identification of requirements impacts the development phase, but the development phase may vary considerably between two ontologies. In the lifecycle approach, the ontology is expected to go through each phase more than once during its life, and some of the phases may occur in sequence or in parallel. As a consequence, in this scenario, the ontology is constantly evolving. The ontology is seen as part of a complex system in which human operators, different processes, and technologies take part. The development process is guided by a list of questions (concerning the expected outputs of each phase), which guides the ontologists in each group of activities: the answers to be provided cover all the activities from the identification of the requirements to the deployment of the ontology in systems.
- Agile approach: OEMs in this category stress the need to support rapid prototyping and the development of models in a collaborative dimension. The list of steps envisaged by these methodologies is usually limited – if not completely absent – and steps are connected in an iterative cycle that evolves a prototypical ontology into the final model. Rather than presenting a sequence of activities, these OEMs present some fundamental intervention areas (Pre-development, which may include the identification of the problem, goals, requirements, and knowledge elicitation; Development, which can encompass conceptualization of the domain, and its subsequent formalization with ontological languages; Post-development, an intervention area dedicated to maintenance, update, and evaluation of the ontology). These areas can be reiterated until the final version of the ontology is reached. Agile OEMs focus on the role of domain experts and stakeholders as co-participants in the engineering process, reducing the role of ontology engineers to the final area (the one dedicated to formalization with ontological languages). These OEMs recognize that building an ontology is a time-consuming process that requires skilled human resources, while agile approaches best fit the needs of supporting rapidly-changing requirements and models' evolution. Agile OEMs are the most recent type of methodologies and are inspired by agile software engineering methodologies. These OEMs are often suggested in contexts where there is the need to involve a community of stakeholders in development activities, and in some cases, agile OEMs are expected to support non-expert stakeholders in participating in the ontology engineering process [30].

Beyond the classification based on the engineering approach, macro-level OEMs can focus on cooperation – a crucial aspect in different domains that reflects the possibility of involving domain experts and stakeholders in the OE process to some extent. Leveraging on the classification provided by Kotis et al. [31], OEMs can be divided into three groups:

- Non-collaborative OEMs: this group encompasses those OEMs that provide phases in a systematic and formal way, tasks' descriptions and lists, and workflows necessary to develop an ontology. These methodologies do not stress the cooperation among stakeholders, but they are instead focused on describing activities that need to be undertaken to conduct OE.
- Collaborative OEMs: the methodologies belonging to this category also provide a systematic and formal definition of the various steps (including phases, tasks, and workflows) necessary to develop the ontology. In addition, they emphasize the involvement of ontology experts, knowledge engineers, and domain experts throughout the activities in a collaborative context, also adopting different tools to enhance cooperation. The collaborative aspect aimed at getting to a commonly agreed knowledge and its formalization is therefore deemed as the most important factor for the success of these OEMs.

- Custom OEMs: more recently, there emerged some OEMs defined "custom", which do not necessarily define engineering activities in a formal way, although stressing the importance of the involvement of stakeholders and communities of practice. These OEMs stress that ontology development may occur in decentralized contexts in which collaboration is pivotal. The definition of steps and tasks is limited and mostly concentrated on some novel aspects of engineering, which usually is a variation of the activities depicted in Figure 1. It is worth mentioning that Custom OEMs are not necessarily described throughout the whole engineering process, but instead, they highlight some different relevant approaches in one or more tasks in relation to the waterfall, lifecycle, or agile methodologies.

This classification allows highlighting the two fundamental elements for a macro-level OEM: an explicit and formal definition of the tasks required for developing ontologies and the possibility of involving different professionals (domain experts, knowledge engineers, ontology experts, etc.) in the development process and activities.

The first macro-level OEMs can be traced back to the early 1990s, and they were archetypal for the waterfall approach, with no or scarce reference to the collaborative dimension. With the beginning of the 2000s, more effort was dedicated to integrating the role of stakeholders and domain experts in the OE process, thus shifting researchers' attention towards more collaborative OEMs. With the definition of the agile programming paradigm, some researchers proposed agile OEMs, stressing how the underlying paradigm was able to foster dynamic cooperation among stakeholders, ontologists, and domain experts.

2.2.1 A survey of macro-level development OEMs

Table 1 summarizes the features of the main macro-level development OEMs. For each methodology, the main development approach is reported (waterfall, lifecycle, agile), and the possibility of using it as a collaborative OE is also stated. Leveraging on the three broad categories for OEMs' activities identified by Simperl and Tempich [23], the table provides a summarized view of the main features of each OEM, also highlighting their peculiarities.

The table reports those methodologies that are described in dedicated papers, neglecting custom approaches to OE. Custom approaches consist in an engineering process that does not fall under any of the approach labels (waterfall, lifecycle, or agile) and can foresee a certain amount of collaboration. On top of that, studies exploiting custom approaches often leverage automatic or semi-automatic extraction of knowledge from corpora or existing ontologies to build ontologies. Some examples of custom approaches can be traced in some recent works, in which the possibility of relying on collaborative and decentralized settings is often underlined. For instance, Rebele et al. [32] leverage data extraction from Wikipedia, WordNet, and GeoNames Knowledge Base to develop the YAGO open-source knowledge base, adopting a knowledge extraction and taxonomy construction approach in a decentralized setting; in this context, parts of an agile methodology to validate the extracted knowledge is used in decentralized and collaborative ways. A similar data-driven approach is also adopted in the development of MedRed [33], an ontology for the description of metadata of clinical trials and studies; the design principles leveraged existing vocabularies and fostered collaboration in some development activities via GitHub. The dimension of distributed cooperation is pivotal in Narula et al. [34], in which the interoperability of social data is represented using semantic data; discussions based on communities, mailing lists, forums, and W3C working groups enabled a distributed cooperation among several actors, and the outputs gathered were used to engineer the ontology. Also, for ontologies generated in an automatic way, feedback inclusion and knowledge updates are essential: in this regard, the works from Salatino et al. [35] and Tommasini et al. [36] – respectively, in the development of a Computer Science Ontology and a Vocabulary for Cataloging and Linking Streams – explicitly refer to the role of collaborative and decentralized tools in different communities as a means to improve the ontologies (also in the updating). Agile principles can also foster the reuse of datasets in Semantic Web, as described by Arndt et al. [37]: adopting Git to enable distributed cooperation among different stakeholders, the authors were able to develop a collaborative environment capable of versioning semantic datasets, providing support in several operations.

Although interesting, these engineering approaches do not rely on existing OEMs. Instead, they apply some general collaborative principles to the automatic development of ontologies to prune, refine or redefine the ontological models. Moreover, these approaches do not specify steps or activities to conduct to move from an informal to a formal representation of knowledge.

2.3 Analysis of macro-level OEMs

Macro-level OEMs share similar characteristics, although each is focused on specific aspects or proposes specialized approaches in conducting some of the engineering activities. This type of OEM gathered interest from researchers as a promising way to involve practitioners in the development and use of ontologies. The more macro-level OEMs are proposed by researchers and knowledge workers, the more the necessity of understanding commonalities and differences among various approaches rises. In the past ten years, several literature reviews analysed the status of macro-level OEMs. In this section, a meta-review approach identifies the main criteria researchers take into account when comparing different methodologies, and a survey of papers dedicated to the evaluation of OEMs is provided.

2.3.1 Criteria for the comparison of macro-level OEMs: a meta-review approach

There exist several contributions in scientific literature dedicated to analysing and comparing different macro-level OEMs [23, 31, 38–43]. These studies evaluate whether or not existing OEMs include the practical aspects to be considered in the ontology engineering process in terms of both activities and recommendations on specific features of ontologies. By considering the three stages of OE depicted in [23], the main criteria adopted in reviews can be grouped according to the activities they foresee. Also, other relevant criteria were cited consistently in most of the surveys, and they relate mainly to cooperative aspects – therefore, they were grouped under the "Collaborative features of OEMs" label. Table 2 summarizes and groups the main criteria adopted in these contributions.

The findings from different reviews and surveys allow for drawing some relevant and general considerations regarding the OEMs, which are discussed in the following subsection.

2.3.2 Considerations on macro-level OEMs

The analysis of the main OEMs conducted by the above-mentioned reviews and surveys allows drawing some interesting considerations regarding the state of the art of OEMs.

Stage 1: Ontology management. It is worth noticing that no review directly addresses the feasibility study as part of the pre-development process. There is a general lack of details also regarding the way feasibility study is performed or supported by specific tools, and no specifications are detailed regarding actors' (domain experts, stakeholders, ontologists) involvement. Nonetheless, feasibility study (and, more in general, management of OE activities) is particularly important – as ontology engineers often underestimate the efforts to be undertaken in the following phases and/or do not take into account the economic costs related to OE [23]. While competencies, requirements of the ontology, and scenarios in which the ontology should be used are generally specified in all OEMs as part of the domain specification activities, the feasibility study is only cited by Fernández-López and Gómez-Pérez [40] (and also in a methodology proposed by Sensuse et al. [44]). Some authors refer to the existence of general "preliminary tasks" (also known as "pre-development activities") as the definition of a strategy for building the ontology. This strategy encompasses an assessment of the level of expertise of available domain experts and the type of ontology that is expected to be developed. Nonetheless, preliminary tasks are not addressed specifically and lack details. Nonetheless, it is worth mentioning that some attempts at assessing the costs related to OE process were investigated in early 2000s – see e.g. ONTOCOM [45].

The few methodologies referring to a Feasibility study often do not offer any particular perspective. CommonKADS [46] – a structured methodology for knowledge bases projects – proposes a different definition of Feasibility study: in this OEM, the activity is strictly connected to organization analysis. Ontologists adopt a two-part study to identify challenges and opportunities related to the problem at hand, putting them in the context of the organization performing the OE activities. In addition, the study includes an evaluation of economic and technical analysis. The goal of the study is to estimate and understand the impact of the ontology-based system in an organization. A very similar approach is adopted in the more recent POEM methodology [47], in which technical and economic feasibility are analysed to identify costs, benefits, organizational needs, and involved processes. This perspective underlines that a Feasibility study should help ontology engineers integrate the ontology-based system into the organization(s) that are developing it [46].

The meta-review elicits that it is generally acknowledged that in the early stages, the aim and scope of the target ontology should be carefully specified, but it is suggested to have a not-too-specific focus to facilitate the exploitation of the ontology from a wider community [31]. The importance of a methodological framework, as recognized in the early years [23], is pivotal to helping ontology engineers in structuring the OE activities: this is particularly helpful for those projects in which OE is one of the tasks and not the main goal. Similarly to the Feasibility study, the Management activity is mentioned in some OEMs but not specifically addressed: in the majority of the macro-level OEMs investigated in this work, the Feasibility study and Management activities are often interchangeable, and their definition is mostly blurred.

Some reviews also include the comparison with the IEEE standard for software development, as its processes include several aspects and activities that are useful also in ontology engineering, but their results show that very few OEMs include management aspects (e.g., [40]). Also, other aspects that usually take place in the management phase are rarely mentioned: a relevant activity, especially in complex projects and from an organizational perspective, is the estimation of economic costs connected to the OE process. This activity requires also estimating the human resources (in terms of their efforts and costs), including the participation of domain experts, ontology engineers, and API developers that are needed to accomplish the engineering task [42].

Stage 2: Ontology development and support. Reviews and surveys argue that most OEMs focus on the description of the development process to be followed in this stage [43]. This phase concentrates on various activities, ranging from domain analysis and specification (including knowledge acquisition, conceptualization, and implementation) to ontology evaluation. The majority of these activities constitute the core of any macro-level development OEM, and therefore they are presented and described in the majority of OEMs. As underlined in the early years, development activities are primarily performed in close collaboration with domain experts [23]. However, several sources stress that the domain specification knowledge acquisition should be supported by training seminars: this is relevant for limiting the impact of the "knowledge elicitation bottleneck" and also for helping the actors involved in the OE process understand the terminology – which is fundamental for those stakeholders who are unfamiliar with ontology [42]. Several sources underline the role of ontology design, also stressing the production of documentation to support the understanding of design choices [40, 41, 44]. These activities are fundamental to ensure that the ontology can be sharable and to document the decisions underlying some design choices. The meta-review highlights the existence of different documentation frameworks that can support ontologists in this activity. Finally, reviews show that the stage of evaluation is also well-supported by means of reference frameworks or formal logics [31]. In addition, sources agree that ontology evaluation should not be considered as the ending of the OE process: the evaluation should serve as a kick-off for a continuous process eliciting feedback from ontology users and stakeholders – ideally, the feedback fuels the evolution of the ontology [44]. All these insights further highlight the importance of developing ontologies that should be primarily aimed to be kept alive and to adopt OEMs that increase their reuse potential, as is the case of some collaborative OEMs reviewed in Kotis et al. [31].

Stage 3: Ontology use. Recent reviews show that several OEMs do not offer suitable support for maintenance, changes in documentation, integration, and interoperability [42]. In fact, ontology needs to evolve over time, and therefore model maintenance (including ontology modifications) is mandatory. However, the support for conducting maintenance (also in a collaborative way) is observed to be often neglected – a fact not limited to early years [41] but also assessed in more recent reviews also in less recent surveys (e.g., [42]).

Collaborative features of OEMs. Criteria dedicated to analysing whether OEMs take into account the needs and roles of stakeholders (including domain experts, knowledge workers, and communities) and whether they endorse a collaborative engineering process have been mainly considered in the more recent reviews or surveys [31, 41–44]. For instance, Kotis et al. [31] argue that a key trend in OEMs is the adoption of collaborative methodologies supported by already available and well-known tools. Collaboration is not relegated only to the management and development phases, but its importance is increasing: [31] and [42] noted that the increasing complexity of the requirements from the real world, together with the need to keep ontologies live and foster their reuse, further led to the need for collaborative OEMs. Nevertheless, Sattar et al. [42] underline that there is a lack of support in providing tools and methodologies to foster cooperation in OEMs.

The choice of an effective macro-level OEM for the development of ontologies is argued to be a difficult task [38]. On the one side, OEMs are not unified, and each group of developers applies its own approach, which may include a combination of different approaches – as seen in the previous subsection and in [39]. On the other side, real-world scenarios require customizable methods rather than pre-defined workflows, as proposed by the majority of methodologies [23]. In general, reviews agree on the fact that no OEM is shown to satisfy all the criteria completely, neither provide details on the description of ontology development sessions, activities and, employed methods, and techniques (authoring) [28, 41, 42]. Some important activities and techniques are also missing in all methodologies, despite their level of maturity [40].

Thus, it is important to understand the features guiding the choices of OEMs that are aligned with the requirements elicited by the stakeholders in terms of feasibility, roles and expertise, and possible scenarios of application and reusability. Indeed, OEMs should be selected in order to support all involved stakeholders, including domain experts, knowledge workers, and ontology engineers, during all phases of the development. Finally, the insights gained from comprehensive findings of case studies and domain-specific guidelines for decision-making would help the OEM research field in obtaining a more elaborated and precise description of the criteria and features to be considered [30].

2.4 Micro-level OEMs

Ontology authoring methodologies emphasize the formalization of a domain. Their steps and methodological activities deal with the generation of a formal model starting from an informal representation of the domain and try to answer fundamental questions like “how is it possible to formalize this relation?” or “how can an existing ontology be reused?”. There is no homogeneity in how micro-level OEMs approach domain formalization: some are focused on identifying and expanding a terminology, others are based on the reuse of modelling patterns, and some others were developed in close relation to a specific domain of knowledge.

Although some authors attempted to provide general steps for ontology authoring OEMs (e.g., Keet [27]), authoring OEMs are hardly comparable for what concerns the fundamental activities (requirements analysis, how to formalize a domain, selecting the representation language). In general, this type of OEM foresees the identification of ontology requirements, its architecture, the language to be adopted – which is dependent on the requirements –, formalization steps (which can also happen through intermediate and non-logic-based representations), and deployment (including maintenance).

With regard to language selection for domain knowledge formalization, it is underlined how language’s features should be in-line with the purposes of the model. If automated reasoning is necessary, the ontology engineers should pay attention to relevant aspects such as the treatability of rules, decidability, and expressivity. On the contrary, if the ontology has the purpose of annotating data or text, lighter languages may be suitable. Figure 2.2 proposes a decisional diagram for the selection of the language.

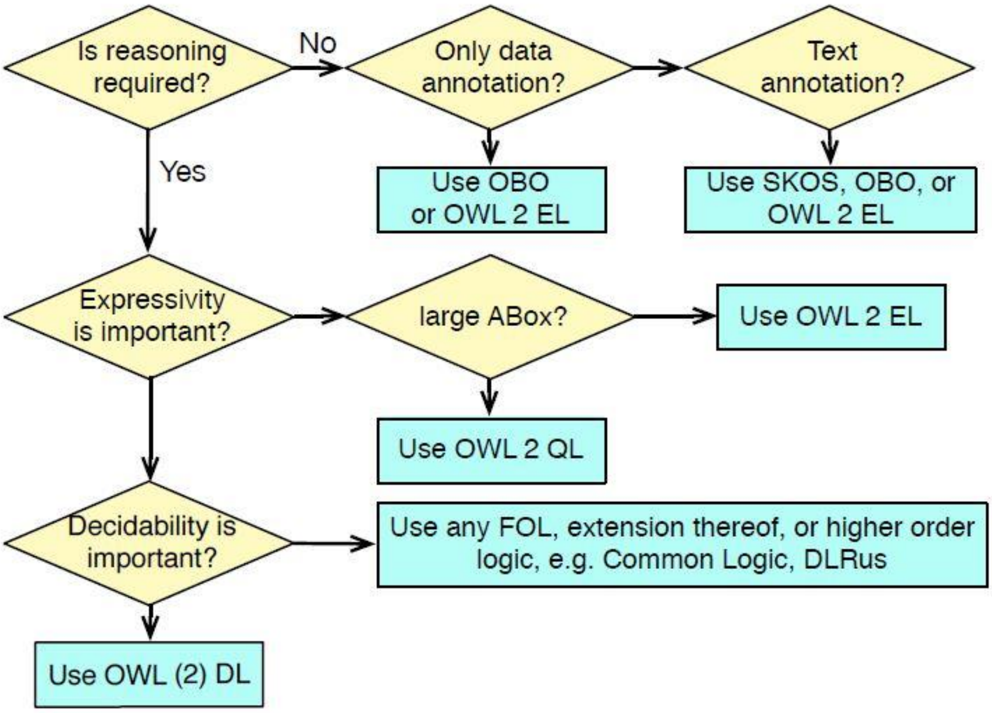

Fig. 2.2. A decision diagram for the selection of a suitable ontology language according to the purposes of the prospective ontology (source: [48]).

2.4.1. A survey of authoring OEMs

This subsection proposes a survey of ontology authoring methodologies, focusing on those OEMs which are described and detailed in scientific literature.

Ontology Development 101 (OD101) [49] is one of the first examples of authoring methodologies, even though it shares many commonalities with process macro-level methodologies. Nonetheless, the focus of the example proposed in OD101 is mostly on how to model some informal knowledge into a logic-based representation.

The OntoSpec methodology [50] helps the ontology engineer to identify and progressively model concepts and relationships within a modelling framework. The methodology takes as an input a set of terms and, for each of them, provides definitions and comments to generate a semi-formal ontology. The following steps leverage the OntoClean methodology (i.e., a methodology for ontology validation under a philosophical perspective) [51] to identify and attribute meta-properties.

An authoring methodology focused on Ontology Design Patterns (ODPs), eXtreme Design [52] is based on a set of principles that aims at involving the customer (by means of CQs, representative customer stories, collaboration) and leverages the reuse of ODPs: customer's stories are translated into CQs, and each question is matched with ODPs – which are reused and integrated. With eXtreme Design, the modeler operates on "parts of the ontology" (the ODPs) instead of focusing on language-oriented operations (e.g., instantiate a property, instantiate a subclass and define its subclasses, etc.).

DiDOn [48] was developed as a method to formalize semi-structured biological diagrams in a logical language. This micro-level OEM enables the formal representation of biological diagrams with the aim of extracting explicit and implicit knowledge and starts with the analysis of the ontology requirements and the selection of a modelling language. DiDOn also provides how-to guidelines in modelling several graphical elements from the diagrams and also taking into account specific modelling choices – such as assessing the functional dependencies of relationships, the possibility to reuse foundational ontologies, the opportunity of importing (parts of) existing ontologies or remodeling them. The methodology is based on two phases: the first concerning the identification of concepts and relationships presented in a biological diagram – thus generating a seed ontology –and the second concerning the population of the seed ontology by leveraging on the formalization algorithm provided. Although explicitly focused on the biological domain, DiDOn provides the means to preserve domain semantics while making clear, traceable, and explicit formalization decisions.

The Guided ENtity reuse and class Expression geneRATOR (GENERATOR) [53] method offers a guide to support ontology engineers in reusing existing ontological resources, guiding the modeler toward the best options among the possible axioms available. GENERATOR consists of three main steps (selection of the classes to be aligned, automatic or semi-automatic alignment of the class to the target ontology, identification of object properties) that can be carried out manually or in an automatic way, using a reasoner. This method is instantiated in the Foundational Ontology and Reasoner-enhanced axiomatiZAtion (FORZA) [54], which leverages different tools to provide enhanced features to link an ontology to target foundational ontologies and to computer part-whole relationships.

In general, data representation in a formal language is a non-trivial and time-consuming activity that may be a significant deterrent for non-expert users: for this reason, it is worth mentioning that in the first decade of 2000s research in ontology authoring attempted to ease this task by combining a Controlled Language approach with ontology engineering (the result of this approach gave birth to experimental tools, such as RoundTrip [55] and Rabbit to OWL Ontology Authoring [56, 57]). Finally, a Test-Driven Development (TDD) tool [58] for ontology authoring makes sure that what is added to an ontology is consistent with its intended meaning: TDD can be used in different scenarios and specifies what need to be added both at TBox and ABox levels.

2.5 Ontology Design Patterns as “building blocks”

One of the methodologies for ontology authoring mentioned in the previous paragraph, eXtreme Design, leverages on ontology design patterns (ODPs) to help ontologists in developing a domain. ODPs can be defined as small ontologies useful to represent recurrent problems in OE [59, 60]. “Design patterns” are no novelty in computer science: Christopher Alexander [61] argued that design patterns are archetypical solutions to design problems in a specific context, and their role can be compared to a set of shortcuts or suggestions to solve context-related problems. Similarly, an ODP is a modelling solution to solve a recurrent ontology engineering problem.

There exist several types of ODPs (as illustrated in Figure 2.3), which can be grouped into six families.

1. **Structural ODPs** include the subcategories of *logical ODPs* and *Architectural ODPs*. The first subcategory presents patterns able to solve design problems where the primitives of the representation language do not support certain logical constructs. Therefore, *logical ODPs* are independent of a specific domain of interest, but they depend on the expressivity of the formalization language adopted. The second subcategory, *Architectural ODPs*, deals with the design choices that affect the whole ontology: some design choices are motivated by specific needs (e.g., computational complexity constraints, selection of an OWL species, the type of DL, …).

2. **Correspondence ODPs** are useful for modifying or mapping an ontology. This family comprises the subcategories of *Reengineering ODPs* and *Mapping ODPs*. The first provides ontologists with solutions and ideas to move from an informal representation of a domain (such as a conceptual map, a database, etc.) into an ontology. These patterns rely on transformation rules to create a target ontology starting from elements of the non-ontological source. Also, rules dedicated to supporting the *refactoring* of an existing ontology are provided. *Mapping ODPs*, instead, are focused on helping ontology engineers in finding semantic relations between two ontologies’ mappable elements, using concepts such as *equivalence*, *containment,* and *overlap*.

3. **Content ODPs**, conceptual design patterns that, to some extent, are complementary to *Logical ODPs*. In fact, if *Logical ODPs* are context-independent, Content ODPs propose patterns for solving design problems for the domain at hand, addressing classes, properties, and individuals populating a specific

domain of knowledge. Content ODPs are not dependent on any specific language; therefore, it was argued that they could serve as "building blocks" of an ontology. This family of ODPs provides solutions to domain modelling problem and affect only the (part of the) ontology dealing with such domain modelling problems. These ODPs can be reused by applying specialization, extension, and composition to them.

4. **Reasoning ODPs**, are defined as applications of *Logical ODPs* oriented to obtain specific results from automated reasoning processes based on the reasoning engine adopted.

5. **Presentation ODPs**, divided into *Naming ODPs* and *Annotation ODPs*, contain good practices for naming conventions (for ontologies, files, and ontology elements in general) and for the use of annotation properties to enhance the understandability of an ontology and its elements.

6. **Lexico-syntactic ODPs**, are linguistic structures associated with some *Logical ODPs* or Content ODPs to help users generalize and understand with the natural language their meaning.

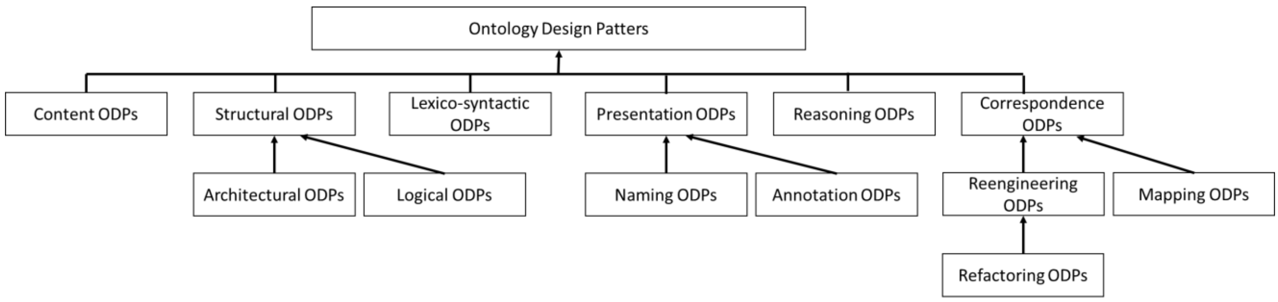

Fig. 2.3. Types of Ontology Design Patterns (adapted from [60]*).*

ODPs – and, in particular, Content ODPs – are documented with a set of features, among which it is important to underline:

- the Name, which specifies the name of a pattern;
- the intent (or Generic Use Case) addressed by the pattern;
- examples of the CQs the pattern helps answer;
- a UML representation of the pattern;
- the patterns to which a Content ODP is related (specialization, generalization, or composition of (an)other pattern(s)).

As of today, an online catalogue of ODPs – under the name of Ontology Design Pattern.org[1] – is maintained by the same research groups involved in the NeOn research project[2] (Sixth Framework Programme of the European Commission). The catalogue, structured as a Wiki, enlists known ODPs using the set of features mentioned above, while a forum allows users to submit new patterns and discuss controversial ones. The website shows an active community up to early 2022. Before 2009, the Manchester ODPs Public Catalog provided ODPs for ontologies in the biomedical domain, providing lists of patterns and their formalizations.

2.5.1 Using Content ODPs to engineer an ontology

As mentioned in the previous section, eXtreme Design is an authoring methodology that exploits ODPs to support ontologists in development issues. In particular, the family of ODPs exploited in eXtreme Design is the Content one, i.e., conceptual design patterns that are dependent on the domain in which they are exploited.

[1] http://ontologydesignpatterns.org/wiki/Main_Page.
[2] Project data available at: https://cordis.europa.eu/project/id/027595.

The dependence between a Content ODP (CODP) and the specific domain of knowledge represented by an ontology, in which the CODP is exploited, seems to contradict the possibility of reusing a COPD as a micro-ontology in several domains. However, conceptual modelling problems have two components: a domain and a use case (or task). In other words, a domain can have different use cases, and the same use case can be found in different domains. CODPs explicitly represent both a domain and a use case: considering that use cases can be various, a catalogue of COPDs can leverage on the feature of a Generic Use Case – i.e., a generalization of use cases that are provided as examples for modelling problems of a domain. In this way, a Generic Use Case represents a recurrent scenario in different domains. The general nature characterizing the Generic Use Case allows for tackling modelling problems by composing different Generic Use Cases.

Moreover, considering that ontologies are composed of domain tasks that can be captured by CQs, Generic Use Cases can be adopted (or composed) to solve the questions [52]. However, Generic Use Cases alone would merely have a "guideline" role – but the CODPs describing the Generic Use Case are actual modelling answers to the issues posed by CQs. From a practical perspective, the local problems an ontologist has to face can be represented as CQs – an activity that is common to most macro-level OEM in the Ontology management stage –, and the latter can be mapped with the CQs used to describe the Generic Use Cases. Since both target ontology's CQs and Generic Use Case's CQs are formulated in natural language, the ontology engineers can adopt different approaches to conduct the matching activity (e.g., a keyword-based approach or a manual approach in case the ontology engineers have a solid knowledge of COPDs).

2.5.2 Advantages and limitations of engineering with ODPs

Literature underlined how the reuse of ODPs could benefit OE. In particular, novice ontology engineers could reuse simple ODPs to avoid common pitfalls or leverage the ODPs catalogue to get support in making informed design choices while modelling [59, 62]. ODPs per se are "micro-ontology" that can be potentially reused and combined, but they can also be described using the implementation language directly (e.g., OWL). This makes the reuse of ODPs easier, not only as abstract conceptualization but as actual reusable components.

Blomqvist et al. [63] argue that ODPs may foster ontology reuse: an ODP seems to be a "middle ground" between reusing a complete ontology (which may be strongly committed on an ontological level, thus making it hard to be reused as a whole) and developing from scratch. ODPs, as "small components", are more likely to fit into a domain conceptualization than larger ontologies. Moreover, by reusing ODPs in an OE process, the target ontology can get a certain level of interoperability – i.e., the target ontology is interoperable with other models using the same ODPs.

At the same time, ODPs support heterogeneity, as they only ensure a minimum level of interoperability and, therefore, a minimum ontological commitment. In other words, it is still possible to freely represent any perspectives for any domain of knowledge while enhancing interoperability and fostering the practical reuse of small components. In some ways, ODPs promise to be a "common language" for ontologists, a language able to foster interoperability at the ontology level, which would also provide an inherent set of points for ontology alignment [63] – i.e., two ontologies adopting the same ODPs have some natural common constructs (the ODPs) that share the same ontological commitment and implementation structures.

Nonetheless, there are some limitations and barriers that hinder the widespread adoption of ODPs in OE practice. Among the criticisms raised against the use of ODPs, there is a lack of relevant patterns both at a generic and domain-specific level. In particular, identifying patterns that have been validated against specific requirements (or real use cases) is still a non-trivial task.

Another relevant limitation is related to hidden ODPs and legacy ontologies. In fact, many used and well-known ontologies were engineered before the introduction of patterns. These ontologies are upper-level models (e.g., SUMO) or widely-adopted ontologies in the field of bioengineering (e.g., BFO, SNOMED-CT). The tasks concerning the analysis of these ontologies to identify ODPs or re-engineer them are far from trivial. Also, these ontologies may rely on hidden ODPs that could potentially be extracted and reused to support OE practices (contributing to filling the gap between ODP-based OE and availability of domain-specific ODPs) – but those hidden ODPs remain out of reach. In particular, in the biomedical industry, several ontologies have been developed and are reused for different purposes: re-engineering those models would require a collective and impressive effort, followed by the (partial) re-engineering of the applications that rely on those ontologies. It has also been underlined how relevant biomedical ontologies reuse ODPs in a very limited way: Mortensen et al. [64] investigated a set of eight large biomedical ontologies to find out that only five of the documented patterns were retrieved in some of the ontologies. In detail, CODPs were very limited – which could be motivated by the domain-dependence of this type of ODPs – while few Structural ODPs were found.

Contrary to software engineering – where patterns are extracted from existing software – ODPs are identified using a top-down approach. Although this might be justified by the limited amount of ontologies available in the early 2000s, there is the need for bottom-up approaches to automatically extract the ODPs from a corpus of ontologies [63]. Ontologists also require access to a set of ODPs that is documented, maintained, and updated: although this is partially happening with the ODPs catalogue and the Manchester ODPs Public Catalog for bio-ontologies, the catalogues are not maintained anymore and present several domains that are not addressed.

2.6 Conclusions

The brief survey of works dedicated to the evaluation of OEMs shows that only macro-level methodologies are being evaluated. The papers presented differ widely in the definition of criteria for an evaluation and also in the identification of a solid quantitative method of evaluation. More importantly, very few works involved OEMs users, and therefore “human usability” and efficacy dimensions for each methodology remain uninvestigated – the only two cases of OEM evaluation that may refer to a methodology being used and evaluated by users [65, 66] adopt very different approaches and obtain incomparable results. The evaluation of OEMs is far from being a trivial task, as each OEM can be evaluated from several perspectives – process, stakeholders’ involvement, results, etc. – and therefore, more attention in identifying sharable criteria of evaluation should be dedicated.

Nonetheless, it emerges how macro-level methodologies can be a solid starting point for novice ontology developers who need to get a grasp on the steps that need to be undertaken to approach ontology development. However, as pointed out in some work interviewing expert ontologists [67], ontology engineers are looking for support in very specific aspects of the engineering process – aspects that are often neglected or only partially covered in current macro-level OEMs, such as support in the management of the development (in particular at the beginning of the OE activities), guidance in the retrieval and reuse of existing model, support for the definition-driven development of ontologies, support in error detection and debugging tasks. Micro-level OEMs can provide support only in very few of these issues (e.g., definition-driven development), as long as it is possible to generalize an authoring OEM’s instructions and to use them in domains different from the ones it was developed for.

The most recent type of OEM, agile, seems to be promising for companies facing the challenges related to digitalization: the agile paradigm seems to fit both novice developers and expert ontologists’ needs because of the flexibility in organizing the various tasks composing OE activities and the focus on cooperative development (including stakeholders and domain experts, who may not have any expertise in knowledge engineering). Agile OEMs may be perceived as the destination of almost two decades of OE, as they merge a flexible organization of steps together with a lifecycle approach in a collaborative environment. However, agile OEMs are still very limited, as there are very few methodologies described in literature, and each one stresses some specific aspects. Also, agile OEMs convey the same issues of macro-level methodologies – among which no agile OEM has been tested with novice and/or expert practitioners yet.

The need for closing the gap between OEMs and their evaluation has an impact on OE as a discipline: an evaluation process following throughout the engineering phases may help OE in stepping up from a “craft” to a structured methodology [8]. This would require OEMs encompassing management tools to support ontologists in the early phases, testing phases to be conducted both at the process level (i.e., during the “steps” composing the OEM) and at the output level (i.e., the prototypical ontology and its evolutions through the activities prescribed by the OEM). Finally, novice ontologists (but also expert ontology engineers), could also benefit from the reuse of ODPs to find support in modelling issues and to grant their ontologies more ways to be aligned with other existing models.

2.7 Appendix

OEM	Approach	Collaborative	Groups of activities			Specific perspective and main peculiarities
			Management	**Development & support**	**Use**	
Uschold & King [68]	Waterfall	no	Not proposed	Identify the purpose, select the language, coding, reusing existing ontologies, evaluation	Not proposed	A skeletal methodology starting with the identification of the purpose of the ontology and ending with documenting all the phases. It does not detail activities
TOVE [69]	Waterfall	no	Not proposed	Use of CQs and motivating scenarios to achieve logical formalization	Not proposed	Leverages motivating scenarios and CQs to identify terminology, then exploits formal CQs to formalize the model
KACTUS [70]	Waterfall	no	Not proposed	Specify application requirements, design using top-level ontological categories, refine and structure the final ontology	Not proposed	Within the context of application development, it suggests the reuse of existing ontologies (developed for specific applications)
Methontology [7]	Lifecycle	no	Planification phase (planning, project control, quality control)	Detailed steps range from specification to evaluation	Support and guidelines provided	A complete lifecycle methodology that guides the modeler in the development of an evolving prototype
DILIGENT [71]	Lifecycle	yes	Not proposed	Detailed steps starting from the identification of scenarios, specification and development	Guidelines in evolving and adapting the ontology prototype provided	A decentralized methodology that evolves a prototype and adapts it to specific uses. Particular emphasis on updating

OEM	Approach	Collaborative	Groups of activities			Specific perspective and main peculiarities
			Management	Development & support	Use	
On-To-Knowledge [72]	Lifecycle	no	Feasibility study and project management foreseen	Set of activities that can be conducted iteratively	Focused mostly on evaluation	Evolving prototype methodology leaning on three perspectives (software engineering, human issues, and knowledge meta-process)
HCOME [73, 74]	Lifecycle	yes	Not explicitly addressed	Detailed set of activities ranging from specification to development, including reuse	Details on exploitation (sharing, versioning)	Stress the human-centered approach in ontology development, underlines the cooperative approach throughout the three groups of activities
DOGMA [75]	Lifecycle / Waterfall	yes	Feasibility study and project management foreseen	Two main steps (preparatory and engineering) guided by a set of principles	Support in merging and solve ontological conflicts	Axioms and specifications of ontology's concepts are separated. Later evolved into DOGMA-MESS [76]
RapidOWL [77]	Agile	yes	Not proposed (intentionally)	Intentionally not detailed, relies on tools	Not proposed	Does not prescribe a set of modelling activities but focuses on a set of 8 principles for the engineering process and 10 practices to be reiterated in a rapidly evolving scenario in which prerequisites change
Melting Point [28]	Lifecycle	yes	Set of tasks for scheduling, control, interactions and quality	Provides principles to guide the iterative development process	Not proposed	Detailed on the management group of activities, this OEM does not detail a set of activities for development but suggests principles to drive the modelers in the development process according to their specific situations
NeOn [78]	Lifecycle / Waterfall	yes	Continuous management of project (not detailed)	Set of activities to be conducted within one of the nine scenarios	Support provided with NeOn toolkit	Focused on the reuse of both non-ontological and ontological resources, with details on different approaches

OEM	Approach	Collaborative	Groups of activities			Specific perspective and main peculiarities
			Management	**Development & support**	**Use**	
POEM [47]	Lifecycle	yes	Foresees a specific step for feasibility study, planning, and scheduling	Detailed set of activities ranging from specification to development, to be reiterated	Integration, training, merging, alignment, and documentation foreseen	Developed to divide ontology development activities from ontological-software products in a cooperative context
UponLite [79]	Agile	yes	Not proposed	Six steps ranging from lexicon identification to development. Ontology engineers are involved in the sixth	Not proposed	Stresses the involvement of domain experts and end-users and enables their roles in conceptualization through familiar tools. The aim is achieving rapid ontology engineering from a community perspective adopting an agile and cyclic approach of the six steps
SAMOD [80]	Agile	yes	Not proposed	Three main steps originating from knowledge elicitation and development of a first ontology, which is enriched with the emergence of new requisites through a set of test-cases	Not proposed	Small steps performed in an iterative process; requirements are elicited from a “bag” of test-cases at each iteration

Table 1. A summary of the main macro-level OEMs and their features, with a focus on the activities they describe (references in the table are temporally ordered).

Steps and evaluation criteria	Fernández-López et al. [40]	Corcho et al. [39]	Simperl et al. [23]	Garcia et al. [28]	Iqbal et al. [41]	Al-Baltah et al. [38]	Simperl et al. [43]	Sensuse et al. [44]	Kotis et al. [31]	Sattar et al. [42]
Ontology management										
Feasibility study	X							X		
OEM / ontology and development type				X	X				X	
Ontology development and support										
Domain experts / Knowledge workers / community involvement / roles / needs				X			X	X	X	X
Domain analysis			X					X	X	X
Knowledge acquisition	X	X	X			X	X	X		X
Design and specification (purpose, scope, users)	X	X				X		X		X
(Strategy for) level of application engagement / dependency				X	X		X			
Conceptualization (including arguments / discussions for identifying concepts)		X	X		X	X	X	X	X	X
Knowledge elicitation				X				X		
Knowledge formalization						X	X	X		
Ontology reuse, compare, merge, evolution					X		X	X	X	X
Implementation	X	X	X			X		X		X
Evaluation / Validation / Exploitation		X	X			X		X	X	X
Documentation / Detailed versioning		X				X			X	X
Ontology use										

Maintenance	X	X				X				X
Use / operation	X	X					X			
Support for interoperability					X					X
Others										
Inheritance from knowledge engineering / rooted in well-established methodologies				X			X			X
Level of detail of the methodology				X	X		X			X
Recommended tools, methods techniques support				X			X		X	
Collaborative engineering processing / construction	X				X		X		X	X
Recommended lifecycle	X			X	X		X			
Project management processes (IEEE standard)	X	X					X			X

Table 2. Criteria traced in reviews and surveys of OEMs, organized according to the stages of ontology engineering and other features considered (references in the table are temporally ordered.

Chapter 3 – Evaluation of agile Ontology Engineering Methodologies

The evaluation framework, the experiment, and its results are based on this candidate's work published in Spoladore, Daniele, and Pessot, Elena. "An evaluation of agile Ontology Engineering Methodologies for the digital transformation of companies." Computers in Industry 140 (2022): 103690 (DOI: 10.1016/j.compind.2022.103690).

The considerable amount of OEMs available, both macro-level and micro-level methodologies, may raise some questions regarding the possibility of evaluating OEMs' efficacy and efficiency in supporting ontology engineers along the OE process. Notwithstanding the interest in the OE discipline, only a few works address the issue of evaluating methodologies. Furthermore, each work selects different objects for the evaluation and adopts its own criteria and method to conduct the investigation. A survey of OEMs' evaluation works is provided in the following Section.

3.1 A survey of evaluations of OEMs

Researchers operating in knowledge engineering and Semantic Web could benefit from many different macro-level OEMs and fewer authoring methodologies in the early 2000s. The number of OEMs available kept growing throughout the last years of the 1990s and 2010 when the first agile OEMs started appearing, published in conferences and journal works. Figure 3.1 provides an overview of the main OEMs that appeared in the period between 1995 and 2022.

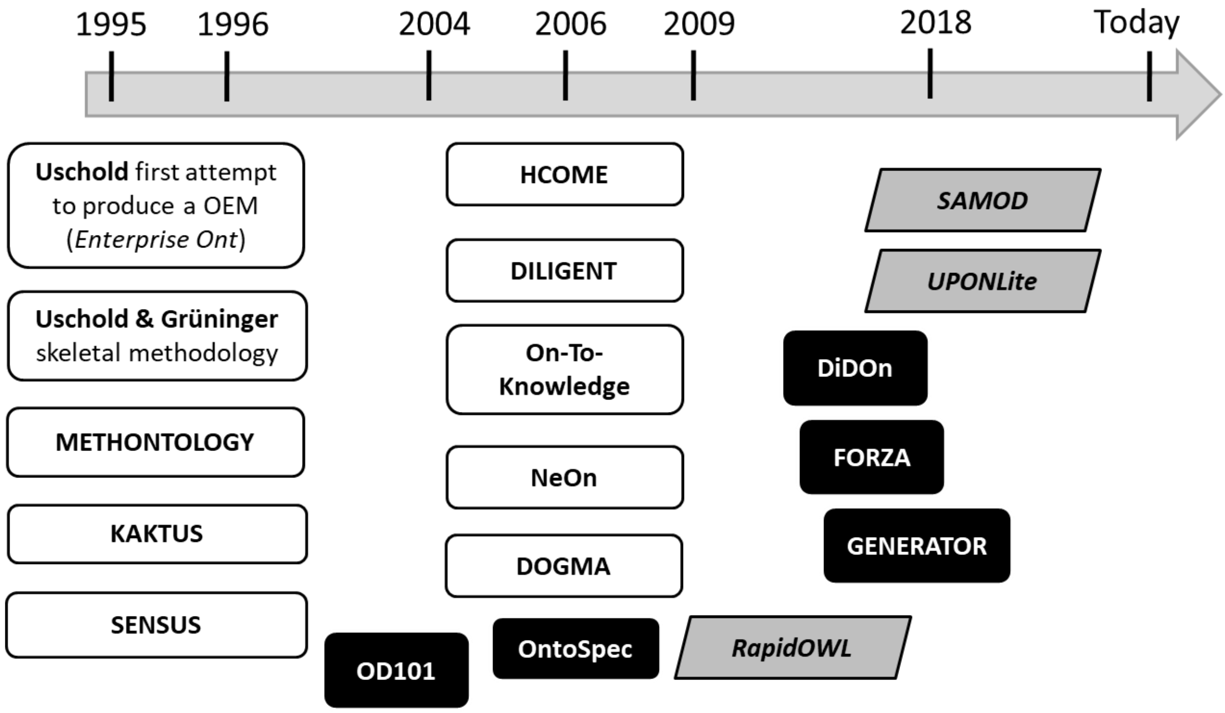

Fig. 3.1. A chart illustrating the temporal distribution of the main OEMs; white-background rectangles represent macro-level OEMs, black-background rectangles represent authoring methodologies, and gray-background parallelograms represent agile OEMs.

Researchers began to question whether having different OEMs could impact OE and if such a conspicuous amount of OEMs corresponded to the concrete community's necessities related to different ways of developing ontologies. However, only a very limited amount of works is dedicated to evaluating OEMs – most of them revolve around the evaluation of a methodology created by the authors of papers.

Tempich et al. [66] is an example of this category; the authors conducted an evaluation of their DILIGENT methodology, with particular attention to its engineering process. The methodology is put to the test with a group of legal experts, with the aim of evaluating the collaborative engineering of an ontology in the legal domain. Although the first attempt at evaluation did not produce results, a second attempt managed to produce a shared ontology even with legal experts geographically distributed, with a relevant speed of development and proper documentation of the results achieved by the team. The evaluation of the OEM was based mostly on qualitative considerations, such as the possibility to cooperate even in geographically distributed environments via a Wiki, participation in discussions, and testing the developed ontology against a set of CQs. Since the stakeholders involved in the OE took part as domain experts (a legal team participating in the domain analysis and conceptualization phases), this paper can draw conclusions on the quality of the collaborative process in some aspects of the Ontology development and support stage.

A proposal for a weighted evaluation of OEMs is described in [81], which allows quantifying seven aspects of a methodology – *Weltanschauung* (i.e., the philosophy of the OEM), coverage in process, coverage in product, reuse of product and process, participation of stakeholders, representation of product and process, and maturity. The seven aspects roughly correspond to the main activities provided by macro-level methodologies. The paper does not put the evaluation framework to the test; therefore, its contribution is somehow limited to a theory for OEM evaluation.

Similarly to [81], a set of qualitative criteria for the evaluation of an OEM are identified by Chimienti et al. [82]. Taking the IEEE Standard for Developing Software Life Cycle Processes (1074-1995) as a reference, the authors identify six criteria for benchmarking OEMs. The criteria encompass the definition of formal metrics for comparison purposes, an evaluation of the ease of use of the OEM, whether or not the methodology enables the modeler to take into account different perspectives, the flexibility of the OEM in a cooperative context, the focus on quality improvement and the necessity of empirically testing the methodology. It can be argued that rather than proposing an evaluation framework, this work sets the criteria for a qualitative evaluation of OEMs. However, whether criteria were drawn for macro or micro-level OEMs is unclear. Also, the criteria were not tested against any OEM.

The adoption and adaptation of a management tool, balanced scorecards, is at the center of the evaluation conducted in [65]. Balance scorecards – a tool from management – are adjusted to provide a quantitative evaluation for Unified Process for ONtology (UPON) development [83], a prototypical OEM for the agile UPONLite (described in Sect. 3.3.1). Four perspectives are evaluated: ontology engineers' satisfaction, the efficiency, and simplicity of the process proposed by the OEM being evaluated, the analysis of the competencies, and skills necessary for the people involved to perform the activities implied by the OEM, the evaluation of the ontology developed from a user's perspective. The methodology for acquiring data related to the perspectives relies on observations and qualitative interviews. Still, the paper does not provide any detail regarding the sample of participants and the process of combining the acquired data.

Three different works from the same author [84–86] are dedicated to evaluating agile methodologies. In [84], the focus is on assessing whether existing agile OEMs are appropriate for the modular development of ontologies. The agile OEMs selected are XP.K [87], Explode, and RapidOWL – however, the second methodology remains unknown since it is described within an M.Sc. thesis which is not accessible. The evaluation is performed by identifying requirements for which each OEM is qualitatively and empirically evaluated. The requirements are 1) the provision of tasks for dividing a domain into subdomains, 2) the provision of activities to model ontology modules, 3) support in the identification of knowledge to be reused, 4) the ability to independently model a single module, and 5) support in the evaluation of the developed module(s). The evaluation underlines that none of the methodologies considered provides specific activities for dividing a domain; only XP.K seems to address guidelines for modular development, while in the remaining requisites, the OEMs do not provide explicit support. In another work [86], the same three OEMs are evaluated with an adapted version of 4-DAT tool – originally designed to evaluate agile software engineering methods.

The aim is to evaluate the OEMs' adherence to the values of the Agile Manifesto, and the three methodologies (evaluated against a total of nine values) generally performed poorly (with XP.K presenting only three of the agile values investigated, while the other OEMs' scores attested on two). Nonetheless, the author underlines that the investigation presented is not complete and still ongoing. A similar approach is adopted in [85], in which XP.K, Explode and RapidOWL undergo a quantitative evaluation against some of the Agile Manifesto's values. Different from the previous work, the author adapts a quantitative framework for agile software engineering in this paper. The paper's conclusion underlines how RapidOWL is "the most agile" among the three analyzed OEMs, since it values cooperation and flexibility more than the others.

Kotis et al. [31] investigated whether the use of collaborative or non-collaborative OEMs impacted the liveness and evolution of selected ontologies. The research concludes that researchers prefer collaborative, tool-supported, decentralized methodologies to facilitate stakeholders' participation in the OE process. Custom and collaborative approaches to ontology development seem to impact the liveness of ontologies, thus fostering their evolution over time.

The survey highlights how OEM evaluations are inconsistent among different authors. However, even considering the diversity of the OEMs taken into account in the papers surveyed, there exist some dimensions that capture researchers' attention. The most represented is collaboration, which is investigated for DILIGENT and agile methodologies (Gobin's works): collaboration among stakeholders, domain experts, and ontology engineers (and, in general, decentralized cooperation) covers a pivotal role in OE. Therefore, evaluating OEMs to understand if they can support these features means investigating whether the actors involved could effectively contribute to the OE activities. It is also plausible to expect that good cooperative OEMs foster the development of good ontologies. However, no work links the evaluation of the OEMs' processes with some metrics of the developed ontologies. The focus on cooperative development is crucial in agile OEMs, where it is investigated with tools from the Agile Software Engineering paradigms: the works from Gobin undoubtedly stressed the importance of relying on quantitative and qualitative methods for OEMs evaluation, but they are minimal. In fact, the conclusions drawn from the application of different adapted evaluation frameworks are not supported by an investigation with OEMs' users, and the whole evaluation is performed by the author using the "instructions" or "guidelines" provided in papers. In other words, these works lack the ontology engineers' perspective and are confined to assessing the "degree of agility" of each methodology.

On the contrary, Chimienti et al. [65] adapted a management tool to explore user-related dimensions, e.g., satisfaction in using the OEM, the simplicity of the instructions provided, and the adequacy of involved actors' skills and expertise. While choosing balanced scorecards can be debatable, the paper does not offer any detailed information on the significance of gathered data (e.g., the number of participants in the experiment is missing, as well as details concerning their personal skills and previous experiences with OE or knowledge engineering processes, etc.). Moreover, it is unclear how qualitative data were translated into scores.

It is worth noticing that the reuse of existing ontologies or non-ontological models is not a feature taken into account in any of the surveyed works. Reuse can significantly simplify ontologists' work and is one of the cornerstones of the Semantic Web. Nonetheless, the papers do not investigate whether an OEM offered support or tools to identify existing ontologies to be reused.

Also, considering the purpose of OEMs – guiding ontologists in the development of ontologies through a set of steps – all the works surveyed lack identifying a way to evaluate the quality of the ontologies developed following OEMs' instructions or steps.

3.2 A novel framework for agile OEMs evaluation

The results from the survey discussed in the previous Section illustrate how previous efforts in evaluating OEMs could identify some dimensions of investigation while neglecting others. Besides the collaboration, another relevant dimension to be addressed is whether or not the instructions provided by the OEM are clear and simple or if they are specifically applicable only to some domains. Also, the evaluation cannot overlook the efficacy of the methodology, i.e., if the OEM analyzed can bring to the development of an ontology –

according to the purposes for which the ontology is developed. This is particularly important for agile OEMs, the latest "evolution" in terms of macro-level methodologies: while the agility of this methodology is expected to significantly help ontology engineers develop models, its efficacy in assisting ontologists in delivering good models has yet to be examined. Moreover, understanding whether an ontology is "good" or not against some constraints or requirements is a very debated topic – which goes under the discipline of Ontology Evaluation – that can count on different ontology evaluation frameworks [88] and tools [89].

Therefore, in order to effectively evaluate OEMs, two main aspects need to be taken into account: the *process* (i.e., the "journey" ontology engineers have to take to move from a set of informal requirements to a formal model, and ontologists' perspectives in following the methodology's instructions, using supportive tools, conducting activities as specified, etc.) and the *outcome* (i.e., whether the results of the process – the ontology and its characteristics – are in-line with the purposes declared for the ontology during the process, and whether the developed model is logically consistent) [90].

In this Section, a novel framework for the evaluation of agile OEMs is presented. The framework is dedicated explicitly to agile OEMs because they are defined to be implemented with less constricted but more creative efforts. The waterfall and lifecycle methodologies appear more challenging than "ready-to-use" agile approaches, which conversely should help ontologists rapidly produce an ontology prototype – in a collaborative and decentralized setting –including modifying domain knowledge's formalization according to stakeholders' needs. These methodologies stress the collaborative approach and the role of domain experts and stakeholders in defining the knowledge and the underlying conceptualization, thus reducing the activities of engineers in an efficient and scalable way. Agile OEMs are expected to help developers overcome many task-related barriers such as knowledge retrieving, creation of documentation, domain analysis and conceptualization, and reducing the learning period due to their structure (i.e., agile OEMs should be easy to learn and customizable). Also, agile OEMs do not require stakeholders and domain experts to know how to model an ontology. Instead, they suggest ontology engineers and participants use any supporting tool to identify and conceptualize knowledge (as described in Chapter 2). Considering that the agile paradigm itself does not foresee a sequence of steps or activities to be strictly followed – rather than the main areas of intervention in which an ontology engineer may want to intervene to get to a prototype – participants in the OE process can potentially participate in any moment of the process.

The proposed evaluation framework consists of two separate dimensions, outcome and process. The framework takes into account the contribution provided by three agile OEMs – namely UPONLite [79], SAMOD [80], and RapidOWL [77, 91] – which are among the most detailed agile methodologies in scientific publications, with explicit descriptions of steps and activities (when these are foreseen).

For the assessment of the *outcome* (i.e., the developed ontology), the basic ontology metrics from ontology evaluation (summarized in [88]) were selected. Other features included in this assessment concern whether formal or informal models were adopted during the OE process (reuse) [23, 92], the preparation of a set of documents to attest and describe the ontology (documentation delivery) [77, 79, 80], the number of times the OEM was browsed to achieve the delivery of the prototype (iterations) [77, 79, 80], the degree to which the ontology provides the information that is expected to be modelled (relevance of the model) [93, 94], the features in the selected ontology language that are used to engineer the ontology (structural measures) [88, 93], the consistency of the entities composing the model (logical consistency) [88, 93], and the amount of time spent to develop the ontology (time) [92, 95].

The assessment of the *process* leverages previous works investigating the features of OEMs and agile programming paradigms. The instructions provided by the evaluated OEM must be perceived as clear and simple (clarity and simplicity) [65, 96], but they must also be adaptable to different domains and flexible to changes characterizing the rapid evolution of the prototype (adaptability and flexibility) [85, 96, 97]. The role of documentation is essential in OE, since it can facilitate knowledge transfer: an OEM should foresee activities dedicated to the production of documentation (knowledge management support) [85, 98]. In agile methodologies, the focus on collaborative development is pronounced, and agile OEMs should therefore foster cooperation among stakeholders, domain experts, and ontologists (teamwork and cooperation) [65, 84, 86].

However, agility foresees the rapid prototyping of the ontology (developer perceived effort) [65] in a cooperative context where everyone's role is clear throughout the OE process (developer perceived role) [65, 85, 96]. Finally, differently from waterfall and lifecycle OEMs, agile methodologies are designed to nurture creativity and innovative approaches, also enabling the possibility to scrape part(s) of the developed ontology in favor of a new approach (innovation support) [85, 86, 96, 97]. Table 3 summarizes and details both the outcome and process features.

Features	Description	Reference
Outcome (ontology)		
Reused models	The ontology produced by participants reuses any existing model (ontological or not, in its entirety or parts of it), including reusing one or more fragments of a foundational ontology to start the development or any ODPs	[23, 92]
Documentation delivery	The ontology produced by participants comes with the documentation required by the OEM adopted. The documentation includes the List of Competency Questions, the Glossary or Lexicon, and the Conceptual map. There exist several types of documents that can attest the modelling choices, and the Ontology Requirements Specification Document can help in specifying several details regarding the developed ontology [99] – no agile OEM foresees this document	[77, 79, 80]
Iterations	The number of OEM full cycles required for participants to complete the ontology. Contrary to waterfall OEMs (and in part similarly to lifecycle OEMs), some agile OEMs foresee the possibility of reiterating part of their instructions. Each OEM specifies whether some activities are dependent on others	[77, 79, 80]
Relevance of the model	The degree to which the ontology provides the information that is expected to be modelled. These metrics assess whether the structural measures (classes, properties, individuals, etc.) composing the model contain the information identified in the requirements. It is evaluated in terms of the definition of Domain and Range of properties, Disjunctions, and Restrictions to model the requirements elicited, and in terms of presence or absence of Unsatisfiable concepts [100]	[93, 94]
Time	The number of hours the participants spent developing the ontology	[92, 95]
Structural measures	The number of features available in the ontology language that are used to model the ontologies. They include classes and subclasses, object properties, datatype properties, individuals, axioms, SWRL rules, and annotation properties conveniently adopted to represent the domain at hand	[88, 93]
Logical consistency	Checks with a reasoner if terms have a consistent meaning in the ontology	[88, 93]
Process (OEM)		
Clarity and simplicity	The possibility to rely on clear and simple instructions to help developers move from an informal representation of the domain to a formal one	[65, 96]

Adaptability and flexibility	One of the principles of agile methodologies, adaptability and flexibility of an OEM foresee the opportunity to accommodate changes at any level of the OE process	[85, 96, 97]
Knowledge management support	Documentation and reuse of existing ontologies or non-ontological resources cover a pivotal role in OE since it supports knowledge acquisition and transfer processes	[85, 98]
Teamwork and cooperation	Fundamental aspects of a collaborative OEM, teamwork and cooperation foresee developers and domain experts working together in different steps of the development process	[65, 85, 86]
Developer perceived effort	Perception of the effort spent by developers and the added value of iterations for the OE process. The developer may perceive an OEM as easy to follow or particularly strenuous in some aspects	[65]
Developer perceived role	The value perceived by individuals regarding their contributions and interactions during the whole collaborative OE process	[65, 85, 96]
Innovation support	The value perceived by the individuals on the possibility of testing novel solutions, adopting a *try-and-learn* approach in their implementation, fostering the creative efforts in the conceptual model and its refactoring	[85, 86, 96, 97]

Table 3. The list of outcome and process features of the evaluation framework for agile OEMs.

Outcome features are not exhaustive of the evaluation of a domain ontology. However, the selected features can provide a solid assessment of the quality of a domain ontology, in particular for small and specialized ontologies. The evaluation of the process of OE foreseen by an agile OEM builds on works and frameworks already introduced and provides adequate coverage of the main characteristics an agile OEM should provide.

3.3 Evaluating agile OEMs with the framework

Considering the outcome and process features described in the previous Section, this Chapter proceeds to the evaluation of three agile OEMs. The evaluation takes advantage of an experiment involving human participants, to which questionnaires were administered.

The remainder of this Section is devoted to introducing the three OEMs and their main characteristics; describing the experimental setting, the sample, and the methodologies adopted; presenting the results gathered; and finally, discussing the results and the limitations of the study. The Section closes with tables reporting the results of the experiment conducted.

3.3.1 The three agile OEMs: UPONLite, SAMOD, and RapidOWL

As mentioned in the previous Section, the three agile OEMs evaluated were selected mainly because they are provided with explicit descriptions of their processes and instructions. In fact, despite the growing interest in this type of methodology, very few papers are specifically dedicated to the explicit illustration of agile OEMs. Out of these works, three agile OEMs are thoroughly presented in papers that guide ontology engineers in the use of the respective methodology – namely UPONLite [79], SAMOD [80], and RapidOWL [77, 91]. Because of the availability of explicit instructions, these OEMs can represent a relevant resource to be employed as conductors or facilitators for developers aiming at successfully developing and applying ontologies in their knowledge management processes. In accordance with the agile OE principles, the OEMs address collaboration among domain experts, stakeholders, and ontologists. Most of them do not necessarily prescribe a specific set of steps to be followed – although there persist some dependencies in each OEM: for example, it is hardly conceivable to start an OE process without considering the requirements on which many other activities depend.

3.3.1.1 UPONLite (Unified Process for ONtology building Lightweight)

UPONLite is the only OEM that builds on a non-agile OEM, the Unified Process for Ontology building (UPON) [83], which relies on the Unified Process (UP) from software engineering. Designed to support ontologists in developing large-scale domain ontologies, UPON shares with its agile counterpart the iterative features and the aim of reducing the time and costs of the OE process. However, UPON commits to using UML as a "blueprint" for representing the domain knowledge and is structured as a lifecycle iterative and evolving-prototype OEM; also, although it underlines the necessity of relying on domain experts, UPON does not explicitly address the cooperative aspects of the OE process.

On the contrary, UPONLite stresses the possibility of reducing the need for ontology engineers by providing non-ontology specialists with a set of instructions that can support them in developing ontologies without relying on ontologists – until the very last step foreseen by the methodology. For this reason, the OEM does not refer to "ontologists" or "ontology engineers", but prefers the term "users" (of the methodology). Users cooperate with a socially-oriented approach (which may also use social media platforms) to run through the steps composing the OEM, with the goal of rapidly getting to a prototype.

1. UPONLite is organized as a sequence of six steps, where the outcome of each one is refined in the following steps. The first step consists in identifying the terminology of the domain of knowledge under investigation by producing a list (*Lexicon*). The outcome of this first step consists of a set of lexemes or in an information structure able to identify the "words" (nouns, verbs, adjectives) that are used in the domain; the criteria for including a lexeme in the list is the statistical evidence that a professional of the domain would recognize the term as relevant. This step can take advantage of existing lexicons and dictionaries, as well as the extraction of terminologies from other sources (handbooks, papers, etc.).
2. The lexicon is then enriched into a *Glossary*, which completes the lexemes with their textual meaning and, if needed, a set of synonyms. In this step, the users involved need to agree on the meaning of the terms identified in the previous step. As there may be terms with contradictory definitions, this OEM suggests publishing the Glossary with more definitions, leaving a "social-validation phase" to converge toward a unique term description.
3. In the third step, a *Taxonomy* is generated by organizing the terms composing the Glossary, using a generalization/specialization approach. This step requires users to be involved in a consistent knowledge-modelling effort using *is*-a relationships to specify the terms identified in the previous steps. The taxonomy may also require introducing more abstract terms that are generally not part of the domain, but they are useful in organizing domain knowledge.
4. Among glossary terms, the OEM suggests identifying those representing properties and connecting those terms to the entities they characterize (*Predication*). The fourth step is compared to the design phase of a database, as it is concentrated on explicit properties that characterize the entities that compose the domain. UPONLite suggests identifying complex atomic properties and complex properties (the first being similar to printable data fields – like a *unit price*, while the second can have a structure – like an *address*, composed of *street, postal code, city, state*), and properties that refer to other entities, called reference properties (corresponding to foreign keys in a database).
5. Also, complex entities should be connected to their components (or parts) (*Parthood*). The entities composing a more complex entity should be identified with *partOf* (and its inverse *hasPart*) to create a hierarchy of concepts.
6. Finally, the sixth step is the development of the ontology with a formal language on the basis of the conceptual knowledge gathered and organized in the previous five steps. The definition of cardinality constraints, domain and range are expected to come from the analysis of the output of step 4. This step also takes into account the possibility of evaluating the outcome ontology under syntactic and social criteria (which is addressed through the stepwise approach and social collaboration), semantic quality (which consists in checking the consistency of the ontology), and pragmatic quality (which is ensured by the involvement of users during the whole OE process).

UPONLite states the dependency holding between steps 1 and 2, and between 2 and 3, 4 and 5, but also stresses that there is no inherent dependency among *Taxonomy*, *Parthood,* and *Predication*. Since the OE process lacks a linear progression, each of this OEM's steps provides feedback to the previous ones (although this feedback process is not clearly stated in the OEM instruction). Nonetheless, users of UPONLite can skip one or more of the steps from 3 to 5, depending on their context and business purposes. Figure 3.2 summarizes the steps of UPONLite.

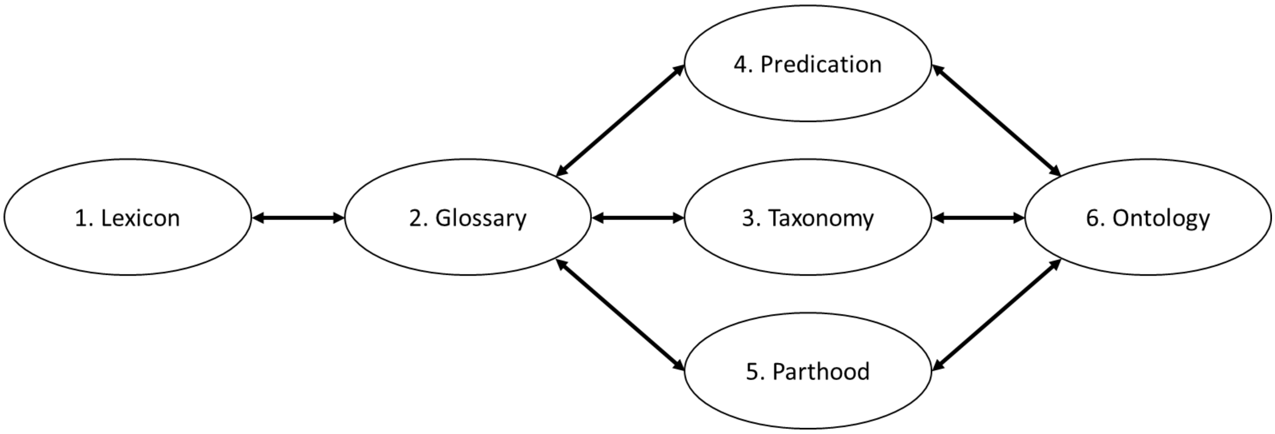

Fig. 3.2. The steps composing UPONLite and their dependencies (source [79]).

3.3.1.2 Challenges and strategies in UPONLite

UPONLite is motivated by the observation that the traditional idea of ontologists working with domain experts is time-consuming and often results in the necessity to modify the developed model with some domain-driven corrections; this also includes maintenance. This traditional approach, defined as "closed teams", fails to respond to the needs of the ontology to be developed. At the same time, the extensive involvement of end-users and stakeholders is considered the optimal solution.

This OEM also identifies some potential challenges that may emerge during the execution of its six steps and some countermeasures that can be adopted to mitigate the risks deriving from them. With regard to its steps, the creation of a lexicon and a glossary summarizes part of the traditional Domain analysis phase. UPONLite recognizes that deciding whether a term is relevant can be a challenge. Therefore, it suggests users evaluate if including terms from a "foreign domain" could be helpful or if referring to external and existing ontologies addressing the "foreign" terms is valuable. UPONLite suggests checking the inclusion or exclusion of terms and their meanings by means of social validation. This social validation should make use of social media platforms to debate and sanction lexemes and their meanings. The definition of a domain taxonomy is recognized as challenging, as it requires identifying general concepts and their specializations starting from the terms contained in the glossary. UPONLite suggests reusing existing structured lexical databases (such as WordNet) or thesauri to solve conflicts originating from diverging opinions, even though a social approach based on folksonomies is also advised. Similarly to the challenges indicated for the third step, the fourth step requires users to agree on a representation of properties, although typing of the atomic property can be delegated to ontologists in the final step. A challenge pertaining to the fifth step is understanding whether the relationship holding between two entities can be identified as an *is-a* relationship or *part of* relationship (a pretty common problem in knowledge engineering, in particular for novice ontologists [92]); social validation plays a central role also in solving this challenge. The sixth and final step is a prerogative of ontology engineers, who have to translate the outputs from the previous step into a coded ontology – therefore selecting a language expressive enough to represent the taxonomy, parthood, and predication.

UPONLite underlines cooperative and decentralized decision-making (by means of a social approach already experienced in OE [101]) and stresses the possibility for the users to adopt familiar tools (such as spreadsheets or textual documents) to finalize the outcomes of the first five steps. Nonetheless, this agile OEM is aware of the possibility of incurring modelling uncertainties, which – if not solved through the social approach – are devolved to ontologists. Also, users are not left alone in performing the steps from 1 to 5: the "ontology master" (an expert with the responsibility of monitoring and coordinating the advancement of ontology-engineering activities). The lack of dependency and the feedback regarding the outputs of the steps from 1 to 5 ensure the possibility of an iterative approach, which enables the possibility of modifying the already generated outputs.

3.3.1.3 Simplified Agile Methodology for Ontology Development (SAMOD)

SAMOD is an agile methodology inspired by test-driven development [102], designed to decrease the interactions between ontologists and domain experts to the necessary. The OEM is organized in small steps taking place in an iterative process and results in the development of a prototype that is incremented on each iteration.

The starting point for SAMOD consists of a set of activities that could be associated with the Domain analysis phase of non-agile OEMs. It requires identifying:

- a *motivating scenario*, identified by a name, a description, and one or more examples accorded to the description. It is significant and representative of the domain to be modelled
- a set of *competency questions* (CQs), represented in natural language and addressing the requirements within a particular domain. The CQs are organized hierarchically (higher-level CQs require answers to other lower-level questions) and are associated with examples of answers, expected outcomes, and identifiers. Also, each higher-level CQs lists the other lower-level CQs requiring an answer
- a *glossary of terms*, a list of term-definition pairs related to lexemes adopted to describe and discuss the domain at hand
- a *set of test cases*

The iterative steps composing SAMOD can be summarized as follows:

1. Ontologists, aided by domain experts, collect all the relevant information about the domain and build a *modelet*, a standalone model describing the portion of the domain encompassed in a test case (a modelet does not include entities from other models and is not included in other models). The modelet has the purpose of formalizing the test case, following certain ontology development principles, in a formal model (TBox) and its related dataset (ABox). The modelet is then tested against the CQs (including formal CQs expressed with SPARQL), and the model (TBox) and its related dataset (ABox) composing it are also checked to assess whether it describes completely all the examples accompanying the motivating scenario. If the tests are passed, a *milestone* is released (i.e., the outcome of the step, which in this step is the formalization of the modelet).
2. Ontology engineers identify a new motivating scenario (included in a test case) and produce the corresponding modelet (as described in the previous step). The modelet of the new test case is merged with the milestone (the modelet from the previous step). The merged modelet is then tested against formal requirements. A milestone (updated modelet) is released in case of a positive outcome.
3. Ontologists refactor the current model, with a particular focus on the last part (added as a consequence of the previous step); again, the model is tested, and – in case of a positive outcome – the milestone (consolidated modelet) is released. If there are more motivating scenarios, the OE process is concluded; otherwise, the process is iterated starting from the first step.

Figure 3.3 provides a graphical representation of the iterative steps composing SAMOD.

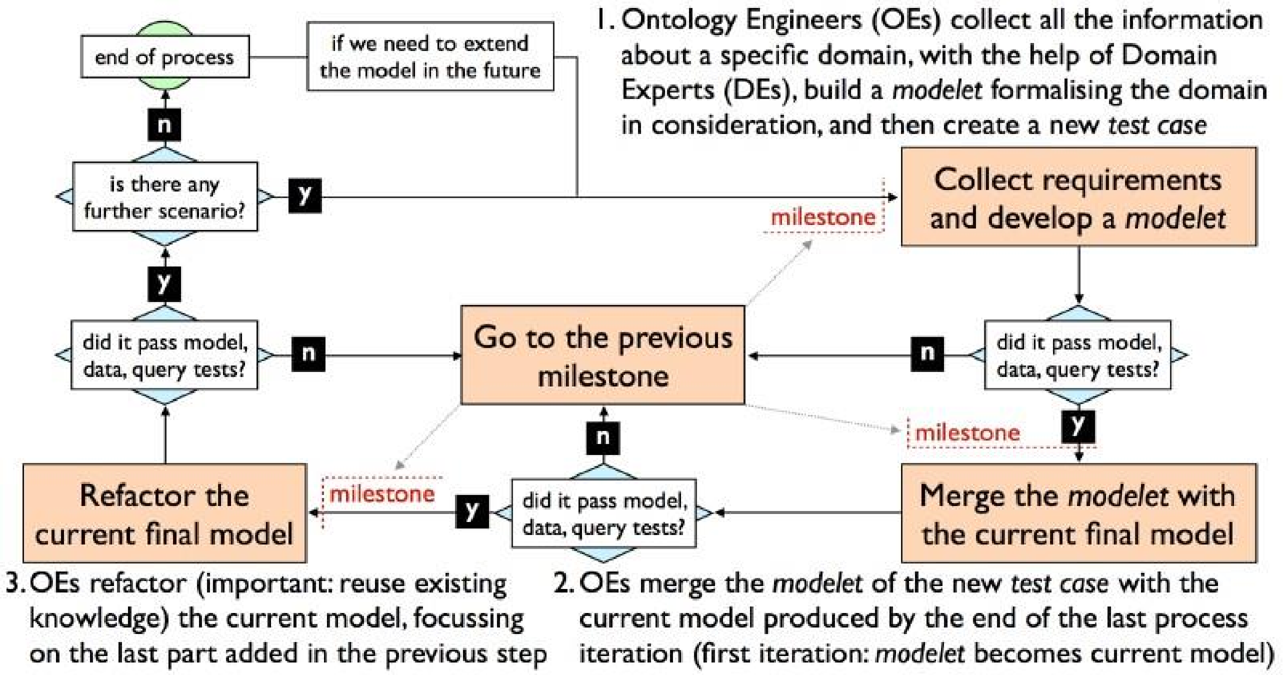

Fig. 3.3. The iterative steps composing SAMOD, starting from the collection of requirements and first modelet development to the refactoring of an outcome (source [80]).

Domain experts are involved exclusively in the definition of the motivating scenario, the list of informal CQs, and the glossary. The rest of the OE process is a prerogative of ontology engineers.

3.3.1.4 SAMOD's engineering principles and operative suggestions

As SAMOD's steps do not delve into modelling techniques, this OEM is detailed with a set of operative suggestions and principles that are supposed to guide the work of ontologists.

From an operative perspective, the number of ontology engineering involved in the OE process can vary from one to many. In the case of several ontologists, there is the possibility to split them into two different teams – one dedicated to the development of the modelets, the second responsible for testing them by creating data describing the test cases and providing feedback to the first team.

As for the modelling principles, they vary according to the different activity – whether a new test case and modelet are being developed, a new model is being merged with a modelet, or the merged model is being refactored.

For the first activity, SAMOD suggests keeping the number of developed ontology entities limited to the essential: in this way, it is always possible to assess whether a change implies some errors. Also, SAMOD – leveraging the findings on information theory by Miller [103] – explicitly states that ontology engineers responsible for the development of the modelet cannot hold in working memory more than a small number of objects (quantified in 7±2).

This principle is matched with an invitation to keep the modelets simple, thus avoiding modelling unnecessary information not pertaining to the motivating scenarios and the CQs. SAMOD advice is to focus on modelling the entities to provide a complete description of the motivating scenarios, avoiding concentrating on inference in this stage. This effort on simplicity should also ensure rapid development.

To help ontologists solve possible modelling issues and identify the best way to model a particular aspect of the domain, SAMOD encourages taking into account existing knowledge. This includes documented patterns (with particular reference to the Semantic Web Best Practices and Deployment Working Group ones and the Ontology Design Patterns portal) and existing vocabularies.

Modelled entities should be self-explanatory. It means that human users must be able to understand what an entity stands for simply by looking at its IRI. No labels and comments should be added while developing the modelets.

When merging a model with a modelet, the ontology engineers involved in the testing follow a three consecutive steps approach:

1. all the axioms from the current modelet and preexisting model must be merged, collapsing semantically-identical entities (based on their local IRI names);
2. update all the test cases, which includes updating all the TBox and ABox and the set of CQs to refer to the most recent version of the model;
3. test the merged model and, if the tests give positive outcomes, set the merged modelet as the new model – which may be merged again in successive iterations.

Finally, when the merging of all modelets is concluded and the final model is a representation of all test cases, ontologists need to refactor the TBox and all its Abox(es) and CQs. In doing this activity, they are suggested to rely on existing knowledge, reusing concepts and relationships in external entities, or align the model with other ontologies. In this activity, it is recommended to add labels and comments (e.g., `rdfs:isDefinedBy`) on ontological entities so that a natural language description of each is provided.

SAMOD clearly traces the boundaries between ontology engineers' work and domain experts' activity. The latter are involved exclusively in the activities involving the specification of the domain and its requirements. Through a lean and possibly iterative three-step methodology, this OEM aims to provide a fast-developed prototype of a part of the domain, which is increased at each iteration of the steps. General recommendations and operative suggestions are also provided to facilitate the ontologists' work – among which is the possibility of reusing some patterns (although the type of reuse is not specified).

3.3.1.5 RapidOWL

The oldest among the three agile OEMs investigated here, RapidOWL [77, 91] highlights the necessity of flexible and quick tools for the rapid prototyping of ontologies. This OEM is partially inspired by XP.K [87] and its Extreme Programming principles, with a focus on collaboration. RapidOWL recognizes that, in order to face rapidly changing requirements, a methodology must be flexible and adaptable to different needs.

RapidOWL focuses on small chunks of knowledge (e.g., RDF statements) and iteratively increments them through small changes leveraging on a cooperative approach until a stable state of the formalized knowledge is achieved. This agile OEM is grounded on the paradigms of a generic architecture of knowledge-based systems, and its process is characterized by values rather than instructions. On these values, principles are derived and used to guide the OE process. Principles are then mapped to practices. Figure 3.4 illustrates this OEM's structure.

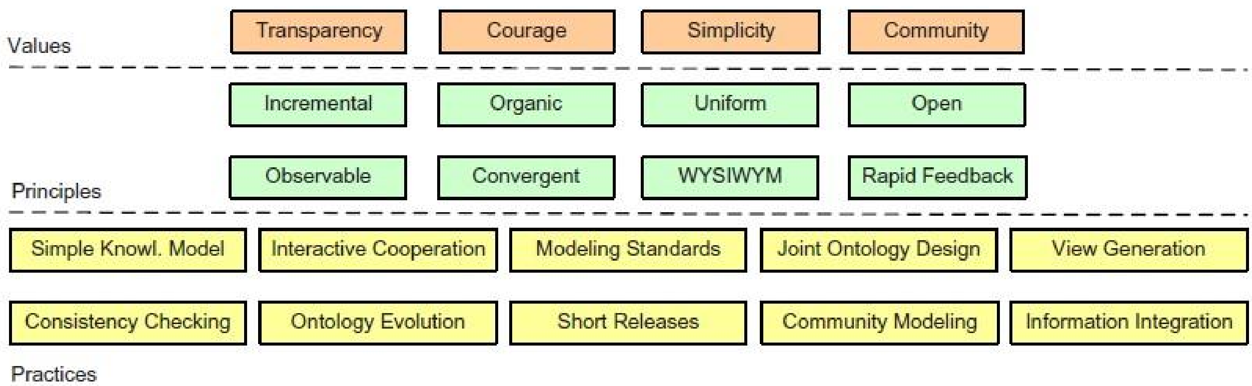

Fig. 3.4. The set of values, principles, and practices composing the backbone of RapidOWL (source [77]*).*

The values underlying RapidOWL are borrowed and adapted from Extreme Programming paradigms and dedicated to:

- enabling cooperative engineering of the ontology and its evolution through feedback (*Community*);
- increasing the ontology maintainability (*Simplicity*);
- promoting the early detection of modelling errors through the immediate publishing of the models and their modifications (*Transparency*);
- daring new modelling solutions (*Courage*).

These values represent the "long-term goals" of the OE process, while the development process is guided by principles – inspired by the design goals of the first Wiki system. The principles include:

- Open-world assumption (*Open*);
- the promotion of *incremental* changes, since concepts can refer to other concepts – including those that are not yet described: this principle is expected to produce changes by cooperating developers;
- the ontology must be open to change and evolution, and errors cannot disrupt what exists thanks to the adoption of roll-back mechanisms (*Organic evolution*).
- the adoption of a uniform authoring method for TBox development and data acquisition enables each ontologist to be both a modeler and an organizer at the same time (*Uniform*);
- ontology authoring is transparent and observable by all participants, which enables a social review mechanism and ensures the prompt publication of any change performed on the model (*Observable development*);
- *rapid feedback* is a consequence of the previous principles, as each contributor can provide his/her feedback;
- duplication of entities is discouraged, and duplicate entities are referred to related or existing concepts to keep the ontology simple and small (*Convergent*)
- relying on a visual representation of the model – "What You See Is What You Modeled" (*WYSIWYM*) – makes it easier to understand which information represents a statement, a class, or an instance.

The practices foreseen by RapidOWL are likewise inspired by those of Extreme Programming and by a limited number of those of XP.K. The practices have the purpose of enhancing the role of domain experts, enabling their active role in the OE process.

- *Joint ontology design* invites domain experts and ontology engineers to describe concepts using a simple "subject – predicate – object" paradigm; such statements can be then translated into RDF or OWL triples.
- When different information sources related to the same domain exist, it is difficult to maintain their interrelations – hence, the necessity of developing an ontology. The different information sources need to be integrated by importing them into the ontology or by linking them with the ontology (*Information integration*).
- *View generation* pertains to the possibility of generating different domain-specific views on an ontology to foster knowledge transfer between different stakeholders (human or software systems).
- As ontology is developed through a collaborative process, it may evolve over time (in the case of RapidOWL, in short amounts of time). Modifying and/or extending an ontology with RapidOWL can take advantage of an ad hoc approach for *ontology evolution* [104], which identifies a hierarchy of changes to facilitate the human reviewing process.
- Contrary to traditional software engineering adopting long-lasting develop-release cycles and leveraging on the Transparency value, RapidOWL fosters the immediate publication of any changes so that it is easier to discuss the modifications and collaboratively review them (*Short releases*).
- RapidOWL commits to a *simple knowledge model*, in which the fewer entities composing the ontology, the easier it is to ensure Transparency and collaborative revisions. The ontology's complexity is expected to increase over time (and following some iterations of the practices). At the beginning of the OE process, domain experts are required to make all the knowledge they believe to

be relevant explicit, but the ontology does not have to model that knowledge completely. Thus, the ontology does not need to anticipate all requirements a priori – relevant entities are added as needed.

- Taking into account the successive enrichment of the knowledge by domain experts, it may happen that some chunks are not DL-compliant. By performing *consistency checking* with DL reasoners on parts of the ontology, it is possible to identify how to modify the ontology to make it DL-compliant.
- *Community modelling* foresees domain experts exchanging opinions and voting on the statements identified in a collaborative way; to achieve this from a technical perspective, domain experts can adopt RDF reification to annotate RDF statements.
- The exchanges between domain experts and ontology engineers should happen in transparent *interactive cooperation* fostered by online tools. This should enhance collaboration in decentralized settings.
- *Modeling standards* indicates the adoption of agreed conventions (e.g., classes should be named with a capital letter; properties should start with a verb, etc.).

RapidOWL is focused on establishing shared conceptualizations to be modelled with an ontological language and does not foresee a sequence of activities: it is up to the ontology engineers, supported by the domain experts, to decide which practices to adopt and when, and none of the activities is explicitly dependent on the others (although, it is evident how *Consistency checking* requires an ontological model to check, which in RapidOWL implies that some *Community modelling* activity produced some *Simple knowledge model*). The complete lack of structure makes this agile OEM very flexible, to the point that domain experts and ontologists can personalize the methodology and promptly respond to changes in the requirements.

It is also worth noticing that RapidOWL stresses the quick prototyping and social validation of ontologies fostered by different cooperative dimensions – expressed values, principles, and practices.

3.3.2 General remarks on the three agile OEMs investigated

By reading the works in which UPONLite, SAMOD, and RapidOWL are described, some general considerations can be observed. First, two OEMs (UPONLite and RapidOWL) stress the cooperative aspects of OE: for UPONLite, cooperation should occur among domain experts to limit the activities of ontology engineers, while in RapidOWL, domain experts and ontologists work together. Contrary to the other two agile OEMs, SAMOD explicitly states that the role of domain experts should be reduced, thus diminishing the importance of a collaborative dimension among all the actors involved in the OE process.

Reuse is addressed in all OEMs, although with different perspectives: in UPONLite, it is suggested to reuse existing vocabularies, in particular when addressing possible conflicts regarding the structure of the model. SAMOD stresses the relevance of reuse at an authoring level, i.e., while refactoring the final version of the ontology, by adopting already well-known patterns. Finally, RapidOWL does not address reuse specifically, but this practice is somehow implied in *Information integration* and – if expert domain users or ontologists are involved in the OE process – in the cooperative development dimension (*Community modelling*).

Only UPONLite makes explicit references to a sequence of steps, underlining the dependencies among them. In SAMOD and RapidOWL, dependencies among OE activities are implied, although they are quite evident. In all the three OEMs, the possibility to repeat one or more steps and modifying an output is deemed to be relevant: this feature is explicitly addressed in RapidOWL by choice of not sequencing the practices composing the methodology, while in UPONLite, it is made explicit by stating it within the text. SAMOD, whose structure is reiterative, necessarily asks its users to perform its steps more than once. In all three OEMs, this feature is expected to increase modelling flexibility.

Concerning documenting the ontology, RapidOWL does not make any explicit references to this activity, while SAMOD considers "documenting" the ontology the third step of its reiterative sequence, and UPONLite does not explicitly mention it. However, both SAMOD and UPONLite start the OE process with some sort of domain analysis, which implies the writing of CQs and glossaries – a form of documentation that helps ontology users have a clear understanding of the entities composing the model.

A lack of examples illustrating one or more factual instances of OE with the methodologies is registered for all three OEMs. Illustrating examples are limited to explaining some steps of the OEMs.

In conclusion, it is necessary to point out that none of the agile OEMs considered foresees or suggests ways of dealing with the management of the OE process, thus making it harder to integrate it in companies or research activities that require a planning of processes.

3.3.3 Experiment setting and methodology

The three methodologies are tested against the framework described in Sect. 3.2 with an experimental case study since the case study methodology is particularly useful in investigating the "how" questions on the phenomenon of interest within authentic settings [105].

The experiment involved a group of company employees attending an *Advanced Professional Master course on Industry 4.0 enabling technologies* – which included disciplines like Semantic Web and OE, Artificial Intelligence, and Robotics. The sample of participants consisted in 21 members (20 males and 1 female; average age: 26.7 years). The whole sample was employed in Italian small and medium enterprises (SMEs) in middle-management roles. Before conducting the experiment, the participants were administered a 35-hour long course on Semantic Web, with elements of OE. Also, in various degrees, all participants had previous knowledge of Computer Science disciplines (SQL databases, basics of programming with Python, and elements of Logic and Computational thinking), as most of them attended technical and technological high schools.

Adopting a qualitative research approach, participants were actively involved in the development of ontologies with the three different agile OEMs described in the previous Section. The experiment was structured in four main phases (summarized in Figure 3.5):

I. in the first one, participants are divided into three groups (A, B, and C, each composed of seven members), and the experiment is described in detail. Participants were provided with information on materials (i.e., the papers with the OEMs' instructions), the ontology editor to use (Protégé 5.3, equipped with the DL reasoner Pellet), and the possibility of relying on domain experts.
II. In the second phase, each Group is administered with the three case-test domains (depicted in Table 4), to be modelled with the same agile OEM. Domains and case tests of the assignments focused on knowledge-based decision support systems.
III. The third phase is split into as many parts as the case tests: it consists of developing an ontology (one for each Group) using the OEM specified. Each Group of participants must develop one ontology for each test case. During the experimentation, the developers could rely on external experts that were accessible to all three groups: these included two trainers and one domain expert for each case-test domain (a product designer, an e-commerce platform manager, and a production planner, respectively). Cooperation was enabled by the co-presence of participants and domain experts for the whole duration of the development phases. At the end of each development task, participants are asked to answer a questionnaire (described in the following Section 3.3). Trainers should be able to collect a number of questionnaires equal to the number of participants times the number of case tests.
IV. Finally, once all the case tests are developed and both the questionnaires and the ontologies are collected, participants are administered a questionnaire for the overall evaluation of the OEMs they experienced.

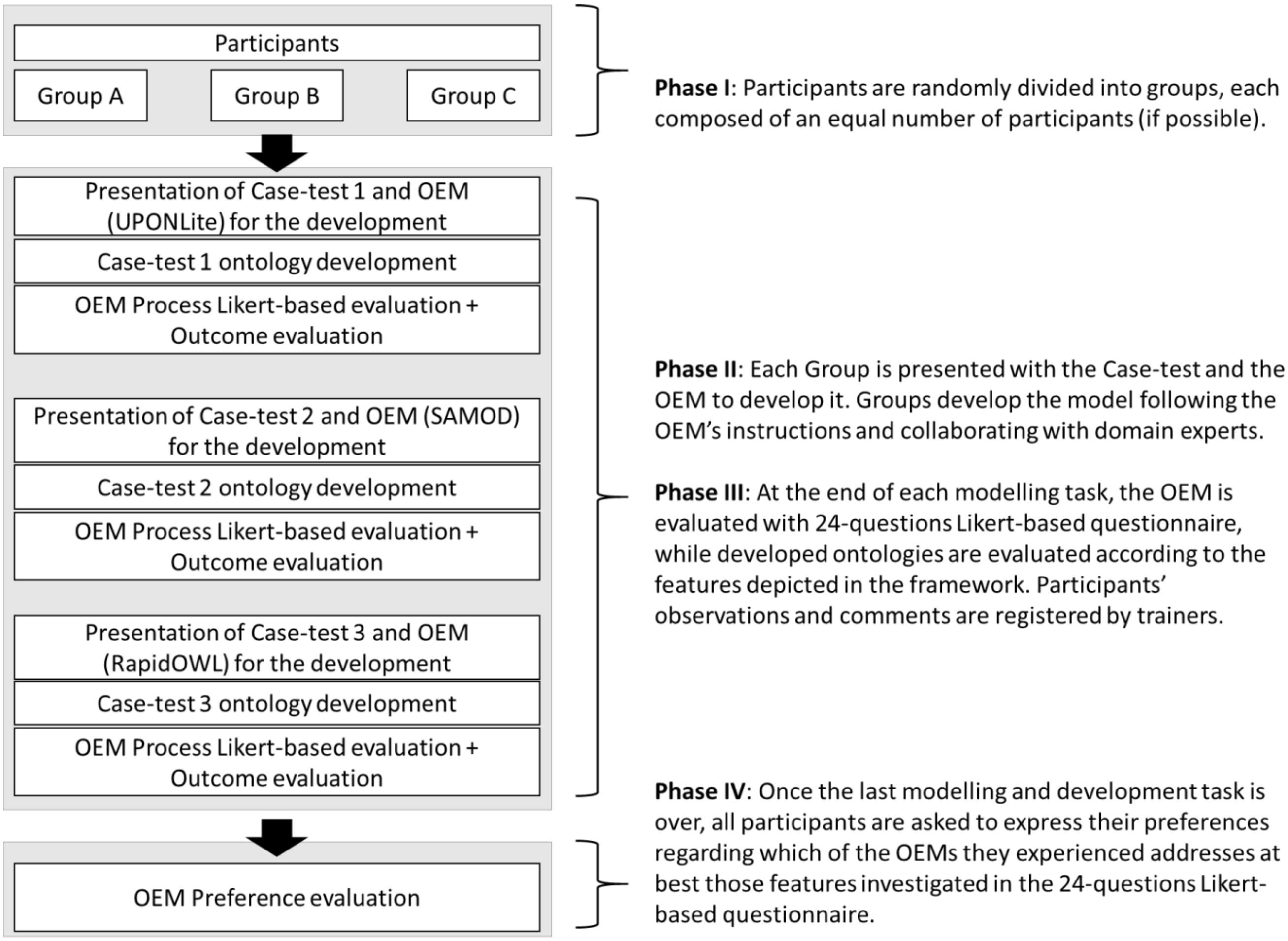

Fig. 3.5. A graphical representation of the phases composing the experiment.

The three case tests the participants were asked to model pertained domains that they partially knew, as discussed during the master course. Moreover, for each case test, a domain expert available in the presence and via online tools during the third phase was enrolled. The case tests were structured in accordance with domain experts so that they reflected a realistic scenario (Table 4). For each case test, participants were asked to model the domain ontology using a specific agile OEM (the same for all the groups)

Case-test	Domain	Case test	OEM
1	Product portfolio	An ontology for representing and managing a company's product portfolio producing custom packaging boxes. The model must be able to represent different types of products, including characteristics such as: • Raw materials; • Available size(s) for each product; • Available colours for each product; • Available shapes for each product; • Availability of finished products; • Availability of raw materials in the warehouse; • Processing time for each type product.	UPONLite

		The framework is designed to support B2B customers in understanding company's product portfolio and estimated time between an order and shipment (orders for available products, custom orders requiring production). **Domain expert**: orders department senior executive – local box factory, SME	
2	E-commerce platform	An online clothing store scenario. From a B2C perspective, the framework must be able to: • Represent different products and their information (type of product, colour, available sizes, price); • Represent customers; • Model previous orders performed by customers; • Suggest new products to customers according to their chromatic preferences, sizes of previously purchased products, products availability. From a backend perspective, the framework must be able to: • Model the cost for each product; • Calculate margin for each product; • Calculate overall margin on each order. The framework is designed to support e-commerce managers in automating recommended products marketing activities, taking into account bot customer preferences and margin-based benefits. **Domain expert**: e-commerce marketing & product manager – regional clothing franchise	SAMOD
3	Production planning	A framework to plan capacity utilization of shop floor resources able to: • Represent workers and their general information (name, surname, age, role in the shop floor); • Each worker's hourly cost, including additional costs for hours between 8:00 PM – 6:00 AM (night shift); • Represent shop floor's machinery, including information on their functioning (time to set up, amount of time for completing a task, required breaks); • Model working shifts (on three shifts: 6.00 AM - 2:00 PM; 2:00 PM - 10:00 PM; 10:00 PM - 6:00 AM); The framework is designed to provide information such as: • Identify machinery's periods of inactivity; • Calculate total personnel cost for each shift and for every working day; • Support organisation manager in planning working shifts (considering that a worker may work the same shift for an entire week). **Domain expert**: organizational manager – plastic molding SME	RapidOWL

Table 4 – Domains, case tests, and OEMs adopted in the study.

Each Group of participants was expected to develop three ontologies for each case test (for a total of nine ontologies for all case tests), to be completed within a prefixed amount of time (1 week for the whole OE process for each case test). During this time, the domain experts were available during working hours, both physically – in the classroom – and via online tools. Also, participants had access to the Web if they wanted to search for ontologies or other informal models they wanted to reuse.

3.3.4 Data collection

The process features of the framework were translated into a 24-item questionnaire (Table 5): each item of the questionnaire could be evaluated with a Likert scale ranging from a minimum of "1 – strongly disagree" to a maximum of "4 – strongly agree". Moreover, participants' comments, observations, and field notes raised during the development tasks were registered by two trainers (not belonging to the participants, therefore not directly involved in the sample taking part in the experiment). This secondary source data was then triangulated with features of the questionnaire (as shown in the following Section).

OEM process features	**Questionnaire item**	
Clarity and simplicity	1	The instructions provided by this methodology are clear
	2	This methodology is simple to learn and use
	3	This methodology identifies and details all the steps and activities needed to develop the ontology
	4	Every step of this methodology is clearly presented and detailed
Adaptability and flexibility	5	This methodology allows modifying the ontology at any moment of the development process
	6	This methodology can be applied to different domains of knowledge
	7	This methodology can be used to develop ontologies in domains characterized by different scales and complexity
	8	This methodology enables the personalization of steps/activities, including taking into account domain experts' feedback
Knowledge management support	9	This methodology provides support in identifying resources to be reused or re-engineered (e.g., other ontologies, taxonomies, conceptualizations)
	10	This methodology provides support in the creation of documentation
Teamwork and cooperation support	11	This methodology simplifies the cooperation between developers and domain experts
	12	This methodology requires the domain expert to have an active role in the development team
	13	This methodology eases the teamwork within the development team
Developer perceived effort	14	This methodology enables the development of an ontology in an adequate amount of time
	15	This methodology provides a substantial amount of steps to be followed

	16	The use of iterations foreseen by this methodology simplifies the development of the ontology
Developer perceived role	17	I felt engaged in the team and during the development process
	18	In my opinion, I was able to contribute to the development process using this methodology
	19	My role in the development team was always clear and explicit
Innovation support	20	Using this methodology, the development team is encouraged to adopt new ideas to achieve the scopes of the ontology.
	21	This methodology encourages developers to be creative.
	22	This methodology allows for making and fixing mistakes easily.
	23	Team members took the initiative to perform the tasks foreseen by this methodology.
	24	Using this methodology, the development team can freely make decisions at any moment of the modelling and development phases.

Table 5. The list of 24 items administered to the participants to evaluate the process features (on the left) composing the evaluation framework.

The questionnaire was administered at the end of each development phase (so each participant was supposed to answer the questionnaire three times throughout the experiment), while the same version of the questionnaire, asking for each item which agile OEM the participant preferred, was administered only after participants completed Phase III (therefore after they had experienced all the methodologies). A total of 63 questionnaires were expected at the end of Phase III, and 21 preference questionnaires were expected at the end of phase IV.

A total of nine ontologies (one for each Group and for each case test) were developed at the end of Phase III. The trainers (researchers in the field of Semantic Web and OE) evaluated each ontology by relying on the outcome features depicted in Table 3 (as further illustrated in Sect. 3.3.5).

3.3.5 Results

The participants produced nine ontologies developed for the case tests, which the two trainers analyzed following the features defined for the outcome. The results are reported in Table 5. By observing the model and the documentation provided by participants, it was possible to evaluate whether the ontologies *Reuse*d existing models or referred to non-ontological resources. *Documentation* was provided together with the ontologies in separate files, and participants stated on the questionnaire the number of *Iterations* for each OEM (participants were asked to keep track of the number of reiterations, according to the guidelines provided by the methodologies). Observing the ontologies also allows for assessing the *Relevance of the model* features and *Structural measures* while trainers keep track of the amount of *Time*. The *Logical consistency* of the model was tested by trainers using the Pellet reasoner.

It is possible to observe some interesting aspects. The reuse of resources (both existing ontologies and non-ontological resources) is very limited and concentrated in Case-test 1 (with UPONLite). Similarly, the most complete *Documentation delivery* is with UPONLite. At the same time, SAMOD and RapidOWL registered the production of fewer documents, i.e., the List of Competency Questions, the Glossary, and the Conceptual map. For example, at the beginning of Case-test 2 with SAMOD, participants struggled to identify the first scenario to develop the first modelet. A discrete amount of time was then dedicated to the Competency Questions to identify the scenarios correctly.

Concerning the metrics that determine *Relevance of the model*, all the models produced are generally very poor in capturing the complexity of the proposed domains and properly representing it in the ontologies. Not all the constructs (domain and range definitions, class restrictions, class disjunctions) were coherently adopted: the groups showed slightly more effort in clearly modelling the domain of the first case test. This result is reflected in *Structural measures*, with participants producing complete models for the first case test. For the other case tests, many incomplete models were reported – i.e., models lacking classes, properties, and individuals to represent what was required.

Only one case of *Unsatisfiable concept* [100] – i.e., concepts that cannot be true and are equivalent to empty sets, usually caused by modelling mistakes – was registered.

From the perspective of *Logical consistency*, the majority of models were consistent, while Case-test 3 reports one inconsistent ontology (GroupA, with inconsistencies originated by SWRL rules: the participants in this Group reported not being able to solve the inconsistency generated by the rules).

With regard to process evaluation, all 21 participants answered the questionnaire three times and entirely (for a total of 63 questionnaires). All participants also provided the preference questionnaires (21) foreseen in Phase IV. The results of the process feature evaluation and the number of preferences are summarized in Table 6. Considering that the study is focused on an experimental setting involving a small number of participants, descriptive statistics was adopted to analyze the data: mean values for each questionnaire item and standard deviation were calculated for data pertaining the Phase III, while a simple sum of preferences was adopted for Phase IV (bearing in mind that the minimum amount of preferences is 0 and the maximum is set to 21, the number of participants).

Also, the process features are integrated with a qualitative approach to provide more insights into the features differentiating the three OEMs and to gain a deeper understanding of the mean values and statistics of the three OEMs. This analysis was based on data from participants' spontaneous comments and observations on the OEM adopted (depicted in Table 7). Only those comments addressed directly from participants to trainers were registered and mapped to one process feature.

3.3.6 Discussion

The results reported in the previous subsection allow for drawing some conclusions. Although the construction of the ontologies has not been entirely satisfactory in the three test cases, the agile OEMs enabled appropriate learning mechanisms and continuous improvement, with participants properly taking advantage of the feedback generated during iterative cycles and the opportunity to broaden the semantic knowledge base during its operation [106]. Results of the study show that there is not a prevalent OEM that is representative in all features analyzed. Also, none of the OEMs proves to be notably effective for outcome features or process features. Considering the outcome, the ontologies developed were perfectible under both *Structural measures* and *Relevance of the model* feature, although some were more complete than others. Similarly, process features are not homogeneously appreciated by participants.

The development with UPONLite pointed out that –according to participants' scores for questionnaires in Phase III and preferences gathered during Phase IV – this OEM provides clear and structured steps. It is also appreciated to support participants in defining entities and clearly stating and sharing their meaning (see Table 7): this could be the reason why the *Documentation delivery* outcome for case test 1 was the most complete (all groups sketched a conceptual map, compiled *Glossary*, and two out of three groups provided a *List of CQs*). One of the three groups also adopted some concepts and relationships from an application ontology for furniture packaging [107] (in particular, the concepts of `Client`, `Box`, and `Material`, and some relationships such as `is_made_of` [range:`Material`], [domain:`Box`] `has_a` [range:`Stamp`]). Here the type of reuse adopted is "soft reuse", which consists in referring to the entities of the ontology that are being reused by referencing their URIs [108]. The other two groups relied on sources of information retrieved on the Web (websites selling cardboard boxes). Also, while following UPONLite, participants heavily relied on conceptual maps, paper notes, and spreadsheets, which were familiar tools adopted to ease the domain analysis and conceptualization (see also comments in Table 7). Participants' needs to find definitions pushed them into searching for existing models that could be adopted as a reference – a sort of impartial source for solving the issue of agreeing on the same meanings. As it emerges from Table 6, this OEM was particularly appreciated in those features that enable the support among teamwork members (including domain experts) and in the proper involvement and definition of each participant's role during the engineering process. These characteristics led to the overall quality of the ontologies produced with this OEM: in fact, from an outcome perspective, the ontologies developed with UPONLite present *Structural measures* adequate for the description of the concepts and relationships foreseen by the case test (R*elevance of the model*).

SAMOD was voted as the OEM that best enables modifications of the ontologies at any time due to its iterative characteristics (as pointed out in the results for features *Adaptability and* Flexibility of the OEM reported in Table 6 and in some comments in Table 7). The possibility to update the list of CQs (which was the only form of *Documentation delivery* that all three groups somehow provided) proved to be useful in modelling a domain, even complex ones – according to item 7. However, participants required more *Time* for the development of case test 2 ontologies when compared to the time invested in using UPONLite (see Table 6 – item 14, and Table 5): it can be argued that SAMOD was perceived as a more complicated OEM than UPONLite, which resulted in the methodology being perceived as composed of several steps (item 15). It is interesting to note that participants recognized that the iterative structure of SAMOD supports the development process (item 16), with a final effort dedicated to investigating whether there exist some knowledge sources that could be reused (item 9): two groups relied on different sources to model the domain at hand (while one group searched the Web for B2C e-commerce platforms, another group explicitly looked for ontologies describing the domain, ending up in being inspired by the work of [109] – especially in the organization of a few classes, namely Customer, Product, Category (of the products)). From an outcome perspective, ontologies developed with SAMOD resulted in being almost adequate in describing all of the concepts and relationships presented in case test 2 (one of the three ontologies did not model orders, nor the properties for costs and margins, thus not developing any SWRL rule to infer margins of products and orders). This may be due to the fact that participants struggled to identify the first modelet, investing a discrete amount of time in correctly identifying test cases and deriving CQs (two groups reported some difficulties in this task, as reported in Table 7). Rather than developing different modelets in a likewise number of small ontologies, the participants adopted the iterative structure to increment the existing modelet, thus reducing the time dedicated to merging two models; however, the resulting models were always tested for CQs identified with each test case.

The *Clarity and simplicity* process features are considerably penalized by Phase III and Phase IV questionnaires results when it comes to modelling with RapidOWL (Table 6). Participants faced difficulties in identifying "where to start" with this methodology (Table 7), whose extremely flexible structure seems to have hindered its comprehension by participants. Also, it is worth mentioning that RapidOWL was the only agile OEM that did not score one single preference for item 1, meaning that the complete lack of structure and explicit dependencies translated into uncertainty for its users. Participants were reported tackling the conceptualization of the domain foreseen in case test 3 directly using Protégé, in a collaborative development effort to capture the model using the ontology editor – thus neglecting the documentation preparation (which resulted in a poor *Documentation delivery*). This fact may be partially motivated by the *Community modelling* and *Joint ontology design* practices, which indicates making every ontology output immediately available (in this case, immediately developed with the ontology editor). Such conditions pushed participants into more creative efforts (items 20 and 21 of *Innovation support* process features), which, however, did not produce adequate ontologies under the *Relevance of the model* and *Structural measures* features: in fact, all the ontologies were lacking concepts and relationships to represent relevant characteristics of case test 3 (one model lacked the properties for describing workers, two models did not represent more than 1 type of machinery, all three ontologies were unable to provide a viable solution for modelling working shifts, and all models did not provide enough rules to calculate personnel costs and organizing working shift – although domain expert suggested a solution that could have been implemented). The difficulties registered at the beginning seem responsible for an increased amount of *Time* dedicated to the development, which was the highest for all groups. Interestingly, participants reported that one single iteration of RapidOWL's practices was sufficient to develop the model: it is plausible that without a clear starting point, it is difficult to understand how many times a set of practices was repeated.

It can be observed from the results gathered that the three OEMs are not interchangeable: UPONLite's structure supported ontologists in following a set of instructions to move from an informal set of information to a formal model. Vigo and colleagues already pointed out that expert ontology engineers do not obsequiously rely on OEMs' instructions but instead rely on their expertise [110]. However, participants in the study can be considered novice ontologists; therefore, they benefit from methodologies to guide them in the OE process. Participants' lack of expertise in OE may be the leading cause of RapidOWL's below-average scores in "Clarity and simplicity" and "Knowledge and management" features and for the poor quality of the outcomes, with ontologies developed by participants resulting incomplete. We can thus argue that RapidOWL can be a valuable asset for those organizations in which OE is somehow already in practice, with a systematic allocation of resources in terms of time and level of skills of personnel employed. From this perspective, RapidOWL's creative environment and innovation opportunities can achieve the goal of rapid and cooperative prototyping. RapidOWL is arguably the agile OEM that requires a "try-and-learn" approach to be properly mastered. It can be concluded that this OEM is dedicated to developers with some pre-existent experience in OE. Similarly, the results scored by SAMOD underline that this OEM can be useful for domains characterized by any degree of complexity. However, it should be adopted when appropriate time (and resources) for the OE process are dedicated (it is important to stress that participants recognized SAMOD as the most laborious OEM) or when access to domain experts is limited.

Creativity – a cornerstone of the agile paradigm – shows how SAMOD's reiterative structure encouraged participants to be creative, enabling a "try-and-learn" mechanism (allowing them to fix mistakes during the development) and involving them all. Similarly, RapidOWL – due to its lack of structure – forced participants to find creative solutions and to work together on a solution to model the domains. On the contrary, the more detailed structure of UPONLite penalized the methodology's score in the whole *Innovation support* features: it is easy to observe that (although steps 3, 4, and 5 are stated to be happening simultaneously) the dependencies among the various steps are clear in this OEM, even though the ontologists can always rely on the possibility to "come back" and reiterate one or more steps. However, participants did not take advantage of this possibility (as UPONLite scored only one reiteration of the whole methodology, thus using it almost as if it were a waterfall OEM).

The features of *Adaptability and flexibility* were appreciated in both UPONLite and SAMOD: although RapidOWL was recognized as suitable to modify the ontologies at any moment, it was not deemed to be adequate for modelling complex domains. Once again, it could probably be caused by the fact that participants had some difficulties understanding how to get started with RapidOWL, thus figuring that the same difficulties might persist with different domains. On the other hand, UPONLite was identified as the most rigid OEM (which does not allow to change the ontology easily at any time, as underlined in item 5 and item 22).

Teamwork and cooperation features investigate another cornerstone of agile OEMs. If UPONLite's constant involvement of the domain experts rewarded the scores in these features and preferences, SAMOD seems to suffer because of the limitations on the involvement of the domain experts – they can be involved only before starting the development case. However, participants were able to gather all the necessary information before commencing the development. Again, the difficulties experienced with RapidOWL caused the participants to identify this OEM as the least cooperative.

This experiment has theoretical implications at the intersection of the research streams on agile OEMs, OEMs evaluation, and in general knowledge management processes. Its results contribute to the literature on agile OEMs by illustrating that the paradigm of agility has shown valuable in fostering collaboration efforts between ontology developers and domain experts through iterative cycles. At the same time, it is more effective if the methodology depicts structured steps to guide developers. This is true, especially for ontology engineers adopting OEMs as a "guide" in their first development attempts: they mainly rely on OEMs' instructions, which are the usual context where OEMs are effectively used since they allow domain experts' contribution even in the absence of experienced ontology engineers (as highlighted by Vigo and colleagues [110]).

From an ontology engineer perspective, results illustrate that OEMs are not interchangeable, and specific features should be considered in the choice of the methodology. Developers should select an OEM according to their expertise and knowledge of OE, bearing in mind that some methodologies require more significant time and resources for development. Beyond dedicated effort and role, it is also important to state that OE is an activity usually included in a broader knowledge management framework. As a consequence, the choice of the OEM is also dictated by the activities in which the OE process is included (e.g., digitalization of companies, research projects, etc.).

The results also show that agility "comes with a cost": there is a trade-off between the level of agility (in terms of the possibility to be creative, to introduce innovation, to be adaptable and flexible along with the iterations), and the support provided by the OEM (with clear and simple steps to be followed, support to documentation and knowledge reuse, and a limited amount of time needed to deliver a prototype). The practice of reuse pays a higher price in those agile OEMs where the level of agility is higher (namely RapidOWL, followed by SAMOD). The agility is also maximized by neglecting the management activities (depicted in Chapter 2): agile OEMs analyzed seem to focus exclusively on Simperl's "ontology development" stage [23], thus avoiding taking into account "ontology management" and only partially considering "ontology use" – which is limited to RapidOWL, whose constant and collaborative OE approach with quickly published results may be regarded as a form of constant "ontology update" by all the stakeholders involved.

Finally, it is worth noticing that only SAMOD addressed the possibility of supporting ontologists in authoring their models. However, participants did not take this possibility into account while engineering their ontologies, preferring to rely on observations of other sources of knowledge to find modelling solutions. This way of reusing knowledge (being "inspired" by other sources rather than importing the entities of interest) is relevant, as it highlights how authoring is not properly integrated into any OEM investigated. On the contrary, participants made an effort while adopting UPONLite, by importing a (very limited) amount of entities.

3.3.7 Limitations of the study and its findings

The study presents a preliminary empirical study on the assessment of three agile OEMs. As such, it has some limitations that open up future research possibilities.

Firstly, the experiment was conducted in a learning setting with a limited number of participants. Due to this sample characteristic, mixing the three groups after each development phase was impossible – as this would have required a larger and more articulated set of case tests and tracking for the changes in each participant Group. However, groups worked in a controlled environment, which allowed participants to develop the ontologies in a more coherent and realistic context, like in a company team. Also, each case test was developed with only one OEM: future research could investigate whether different ontology engineers applying different OEMs to the same case test may get to considerably different ontologies.

Contrary to the existing literature on OEM evaluation, the participants' sample did not include experienced ontology developers. This could have biased the evaluation, considering that OEMs are mostly dedicated to guiding the overall OE process, and most expert ontology engineers do not rely obsequiously on OEMs' instructions. An interesting experimental setting could compare and contrast how agile OEM features assessment varies according to the years of experience in OE (and type of OEM, e.g., waterfall), but considering the same case tests.

Finally, the sample was limited to participants from Italy, all employed in SMEs: different settings (i.e., participants working in other contexts) might elicit different findings.

OEM outcome features		**Case-test 1 with UPONLite**			**Case-test 2 with SAMOD**			**Case-test 3 with RapidOWL**		
		Group A	***Group B***	***Group C***	***Group A***	***Group B***	***Group C***	***Group A***	***Group B***	***Group C***
Reused models	*Non-ontological*	no	Website	Website	no	no	Competitor website	no	no	no
	Ontologies	yes (ontology on furniture packaging)	no	no	no	yes (ontology for e-commerce search)	no	no	no	no
Documentation delivery	*List of Competency Questions*	yes	no	yes	yes	partial	yes	no	partial	no
	Glossary / Lexicon	yes	partial	yes	no	no	no	partial	no	no
	Conceptual map	yes	yes	yes	partial	no	yes	no	no	yes
Iterations		2	2	1	7	8	5	1	1	1
Relevance of the model	*Domain and range defined*	yes	yes	yes	yes	yes	partial	yes	yes	no
	Disjunctions defined	yes	no	no	no	no	no	yes	no	no
	Restrictions defined	no	yes	no	no	yes	no	no	no	no
	Unsatisfiable concepts	no	no	no	no	no	no	yes	no	no

Structural measures	34 classes, 11 object properties, 7 datatype properties, 67 individuals, 151 axioms, 11 SWRL rules	37 classes 8 object properties 4 datatype properties 30 individuals, 164 axioms 5 SWRL rules 4 annotation properties	31 classes 10 object properties 8 datatype properties 48 individuals, 7 SWRL rules	35 classes, 12 object properties, 9 datatype properties, 25 individuals, 136 axioms, 12 SWRL rules	12 classes, 5 object properties, 11 datatype properties, 12 individuals, 96 axioms, 6 SWRL rules	29 classes, 12 object properties, 14 datatype properties, 24 individuals, 146 axioms, 14 SWRL rules [incomplete model]	22 classes, 4 object properties, 19 datatype properties, 21 individuals, 147 axioms, 8 SWRL rules [incomplete model]	14 classes, 3 object properties, 13 datatype properties, 15 individuals, 134 axioms, 5 SWRL rules [incomplete model]	10 classes, 6 object properties, 10 datatype properties, 30 individuals, 125 axioms, 5 SWRL rules [incomplete model]
Time	12 hours	10 hours	11.5 hours	11 hours	12 hours	12 hours	14 hours	14 hours	14 hours
Logical consistency	yes	yes	yes	yes	yes	yes	no (2 SWRL rules generate logical inconsistencies)	yes	yes

Table 5. Results on OEMs outcome features.

OEM process features	Questionnaire item		Assessment of each OEM Mean (Standard Deviation)			Preference among OEMs		
			UPONLite	*SAMOD*	*RapidOWL*	*UPONLite*	*SAMOD*	*RapidOWL*
Clarity and simplicity	1	The instructions provided by this methodology are clear	**3.48** (0.60)	2.90 (0.62)	2.14 (0.48)	**17**	4	0
	2	This methodology is simple to learn and use	**3.29** (0.64)	2.71 (0.72)	2.71 (0.72)	**13**	6	2
	3	This methodology identifies and details all the steps and activities needed to develop the ontology	**3.33** (0.48)	3.19 (0.60)	1.62 (0.80)	9	**10**	2
	4	Every step of this methodology is clearly presented and detailed	**3.38** (0.74)	3.00 (0.63)	1.81 (0.68)	**10**	**10**	1
Adaptability and flexibility	5	This methodology allows modifying the ontology at any moment of the development process	2.76 (0.70)	**3.52** (0.81)	3.10 (0.62)	7	**10**	4
	6	This methodology can be applied to different domains of knowledge	**3.52** (0.51)	3.33 (0.48)	2.86 (0.57)	**11**	9	1
	7	This methodology can be used to develop ontologies in domains characterised by different scales and complexity	**3.19** (0.51)	**3.19** (0.51)	2.57 (0.75)	**10**	**10**	1
	8	This methodology enables the personalisation of steps/activities, including taking into account domain experts' feedback	3.00 (0.45)	**3.33** (0.48)	3.05 (0.74)	5	**11**	5

Knowledge management support	9	This methodology provides support in identifying resources to be reused or re-engineered (e.g., other ontologies, taxonomies, conceptualisations)	3.14 (0.79)	**3.43** (0.51)	2.33 (0.80)	**9**	**9**	3
	10	This methodology provides support in the creation of documentation	**3.05** (0.59)	2.95 (0.22)	1.95 (0.59)	**15**	4	2
Teamwork and cooperation support	11	This methodology simplifies the cooperation between developers and domain experts	**3.33** (0.58)	3.24 (0.44)	2.52 (0.60)	8	**10**	3
	12	This methodology requires the domain expert to have an active role in the development team	**3.33** (0.73)	2.76 (0.83)	1.95 (0.86)	**10**	7	4
	13	This methodology eases the teamwork within the development team	3.38 (0.50)	**3.43** (0.51)	3.14 (0.85)	**11**	8	2
Developer perceived effort	14	This methodology enables the development of an ontology in an adequate amount of time	**3.14** (0.48)	3.00 (0.55)	2.57 (0.75)	**14**	5	2
	15	This methodology provides a substantial amount of steps to be followed	2.62 (0.74)	**2.90** (0.77)	2.14 (0.65)	3	**15**	3
	16	The use of iterations foreseen by this methodology simplifies the development of the ontology	3.00 (0.45)	**3.14** (0.65)	1.86 (0.48)	7	**14**	0
Developer perceived role	17	I felt engaged in the team and during the development process	3.86 (0.36)	3.43 (0.51)	2.86 (0.65)	**11**	8	2
	18	In my opinion, I was able to contribute to the development process using this methodology	**3.29** (0.64)	3.10 (0.89)	2.90 (0.89)	**14**	5	2

	19	My role in the development team was always clear and explicit	3.24 (0.62)	**3.29** (0.72)	2.67 (0.80)	**10**	9	2
Innovation support	20	Using this methodology, the development team is encouraged to adopt new ideas to achieve the scopes of the ontology.	3.43 (0.51)	3.48 (0.51)	**3.52** (0.51)	3	8	**10**
	21	This methodology encourages developers to be creative.	3.19 (0.40)	**3.24** (0.44)	**3.24** (0.44)	4	5	**12**
	22	This methodology allows for making and fixing mistakes easily.	3.14 (0.48)	**3.57** (0.60)	3.24 (0.54)	8	**12**	1
	23	Team members took the initiative to perform the tasks foreseen by this methodology.	**3.14** (0.57)	**3.14** (0.57)	2.57 (0.68)	**8**	**8**	5
	24	Using this methodology, the development team can freely take decisions at any moment of the modelling and development phases.	2.86 (0.73)	**3.24** (0.70)	**3.24** (0.77)	5	**11**	5

Table 6. Table summarizing the results of the questionnaires administered to participants, expressed as the Mean Values and Standard Deviation in brackets. For each item, the higher score is highlighted in bold. The final three columns report participants' preferences for each of the 24 items (bold indicates the maximum value).

Group	Quotations	Case-test/OEM	Process feature
A	"We can use anything we want to sketch a conceptual map. It's a way to make everyone participate in the conceptualisation activity."	1/UPONLite	Teamwork and cooperation support
C	“We are doing this training on a software, but I have to keep in mind that if I have to do it in my company, some senior colleagues are still on the ‘paper and pencil’ approach”	1/UPONLite	Teamwork and cooperation support
A	“The glossary help all of us [*group participants*] in sharing the same meaning for each concept.”	1/UPONLite	Clarity and simplicity
B	“Documentation is not a problem with this OEM […] it’s basically the first thing you have to do and it is good for structuring the development.”	1/UPONLite	Knowledge management and support
B	“I know we should start from somewhere, but how do we know we identified an adequate scenario?”	2/SAMOD	Knowledge management and support
C	“Identifying the different scenarios and merging them to the previous modelets […] it takes a lot of time […] I had the impression that this may slow down the development process … but then we realised in this iteration we can fix a potential mistake”	2/SAMOD	Developer perceived effort
C	“This approach “from small to big” can work for any domain.”	2/SAMOD	Adaptability and flexibility
A	“If I had to perform the evaluation of the perceived effort in my company, I would take into account that there are project management practices in place, they would be meant to ensure the correct	2/SAMOD	Developer perceived role

	development of the ontology in terms of time and cost […] this would influence the perception.”		
C	“You can decide where to start […] you can also start directly with development and then modify the prototype to include more details.”	3/RapidOWL	Innovation support
A	“We were forced to think big with this methodology, then we realised we should have started with a conceptual map – we were even stretching too much our minds in extending the definition of the model.”	3/RapidOWL	Innovation support
B	“With this OEM it’s easier to fix a mistake or change anything […] because you don’t have to go through previous steps. ”	3/RapidOWL	Innovation support

Table 7. A table summarizing the comments provided by participants and their mapping to process features.

Chapter 4 – Proposal for a novel Agile, Simplified and Collaborative Ontology engineering methodology (AgiSCOnt)

The novel agile ontology engineering methodology (AgiSCOnt), the experiment, and its results are under revision for publication in the journal Computers in Industry *in a manuscript authored by* Spoladore, Daniele, Elena Pessot, and Alberto, Trombetta *titled* "A novel agile ontology engineering methodology for companies knowledge management".

The considerations drawn in the previous Chapter underline how the topmost agile OEMs present some limitations when investigated from users' perspectives. In particular, some of them may be unclear in their instructions or even strenuous in their use. Others adopt an intricate iterative process, while some try to avoid complexity by not specifying dependencies completely. All of them do not offer specific activities to support ontology engineers in modelling, thus leaving ontology reuse a marginal role. In this Chapter, a novel agile OEM is proposed with the aim of overcoming some of the limitations identified for existing agile methodologies while leveraging the cooperation and flexibility already assessed for them.

In particular, it is argued that an agile OEM needs to take into account not only those activities strictly devoted to the development but also those related to integrating the OE process in a larger framework – thus, the management of the OE process – and the activities that foresee an evolution of the developed ontology over the time, to address requirements modifications.

The proposed methodology, named Agile, Simplified and Collaborative Ontology engineering methodology (AgiSCOnt), originated from the considerations summarized in Chapter 2, from the remarks pointed out in Chapter 3, and from several OE activities conducted through the years 2015 and 2022 (e.g., those described in the fields of Ambient Assisted Living [111, 112] and health and wellbeing [113, 114]).

4.1 Introduction to AgiSCOnt – Agile, Simplified and Collaborative Ontology engineering methodology

AgiSCOnt divides the activities into three main steps, plus an optional step for defining a Management framework. The three steps are inspired by Simperl's general structure for macro-level OEMs [23], and in each step, a set of activities is identified and described. AgiSCOnt foresees a close collaboration (to be conducted both in the presence or in decentralized settings) among ontology engineers and domain experts: the latter are involved in all the steps of the OE process, with the sole exclusion of the *Development* activities in Step 2 – although, domain experts with experiences or knowledge of OE and its languages may take part to this activity, as well.

The role of domain experts is deemed pivotal in this OEM, as the assumption underlying AgiSCOnt's structure is that a solid definition of the domains involved and of the requirements is essential to the success of the whole OE output. Also, AgiSCOnt recognizes that knowledge elicitation and the consequent domain analysis activities are a bottleneck for the OE process, and for such reasons, it commits to techniques that have the twofold goal of 1) extracting knowledge from domain experts and their knowledge sources and 2) providing with the help of domain experts an informal conceptual map of the domain, to guide the following development activities.

With regard to the development activities foreseen in Step 2, AgiSCOnt concentrates on helping ontology engineers in moving from the informal conceptual map to a formal language. In this effort, the OEM underlines the role of ODPs as a form of reuse. The "target ontology" (i.e., the ontology being developed with this OEM) resulting is there tested against a set of use cases, leveraging once again on domain experts, which have the purpose of understanding whether the model contains all the relevant pieces of information or it requires further modifications.

A similar approach is at the foundation of Step 3 activities, which revolve around the use of ontology. Feedback from users and stakeholders is fundamental to identifying possible requirements modifications and updates to the target ontology.

The borders among the three steps are loose, as outputs from use case testings may highlight the necessity of adding or removing some concepts from the model, thus modifying the outputs of the previous Step 1. Similarly, users and stakeholders adopting the target ontology may provide technical feedback on the model, so asking to modify Step 2's outputs, or to update the knowledge underlying the target ontology, therefore intervening in Step 1's output.

The recursive characteristic of this agile OEM can thus enable both an evolving-prototype approach (similar to the one envisaged by SAMOD) or a model enrichment by reiteration paradigm (common to both UPONLite and RapidOWL). Also, similarly to what UPONLite suggests, AgiSCOnt suggests adopting a familiar tool to facilitate the domain analysis and conceptualization activities, referring to graphical conceptual maps to involve all domain experts and ontologists in these activities. In contrast with SAMOD – in which the test cases were adopted to identify modelets – test cases in AgiSCOnt serve as a means to validate the target ontology's outputs in a cooperative effort to underline whether the model could be enhanced.

The following Figure 4.1 sketches AgiSCOnt and its steps.

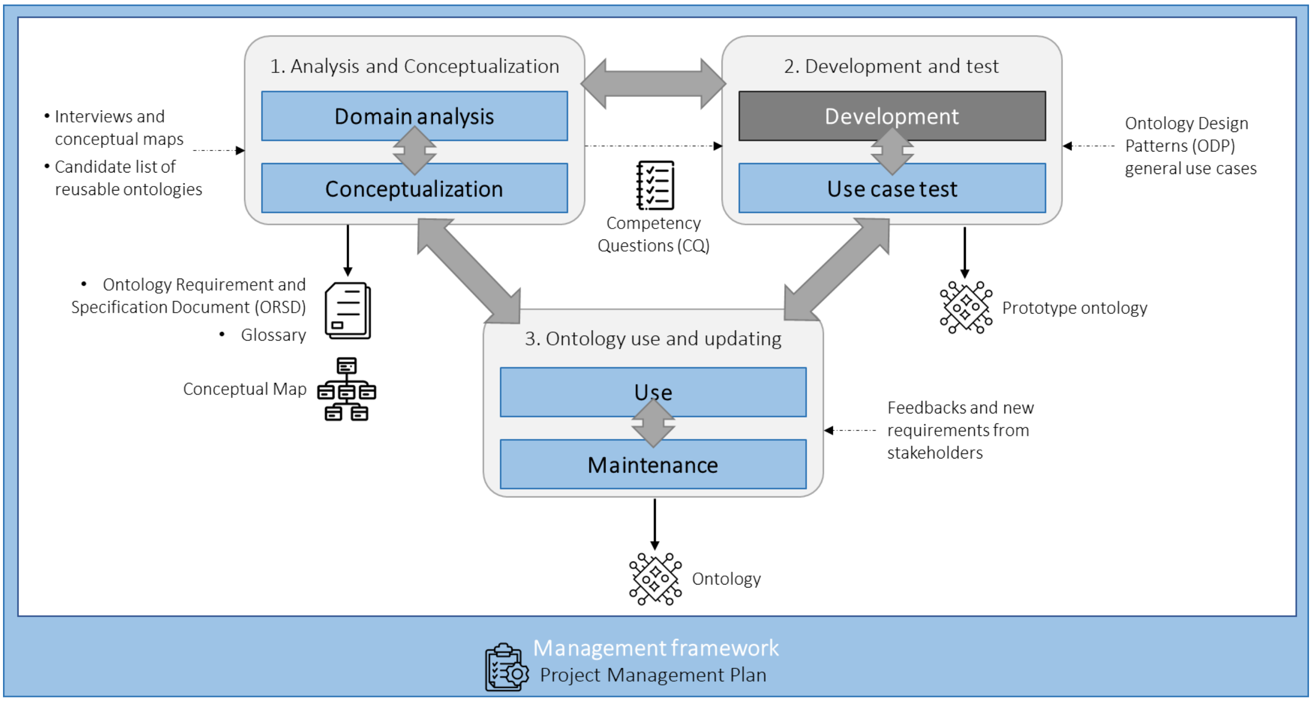

Fig. 4.1. The three steps composing the AgiSCOnt agile methodology, detailed with inputs (dashed black arrows) activities to be conducted (blue rectangles) and the outputs expected after each step (solid black arrows).

4.2 AgiSCOnt's steps and activities

In this Section, the Steps and the activities composing AgiSCOnt are described in detail, together with the motivations underlying specific choices.

4.2.1 Step 0: Management framework

Different from other agile OEMs, AgiSCOnt proposes Management activities to help ontologists in framing the OE process from a wider perspective. These activities are often referred to as "pre-development phase" [23] and serve the purpose of helping to identify the extent of the development effort to be undertaken, focusing on resources' availability, and establishing temporal constraints. Managerial activities can actively help ontology engineers in cases where the OE process needs to be carefully scheduled from an economic perspective and contextualized and documented in a larger framework (e.g., company digitalization activities, scientific research projects, etc.). Nonetheless, the activities do not necessarily have an impact on the OE process described by AgiSCOnt: on the contrary, the Management framework can help ontologists identify resources (not limited to economic resources but also including domain experts and other relevant stakeholders) and contextualize the OEM's steps in a project plan.

The management framework adopted by AgiSCOnt relies on a subset of International Standard ISO/IEC/IEEE 16236:2009(E) Systems and software engineering – Life cycle processes – Project management [115][3]. The ISO/IEC/IEEE 16236:2009 standard is thought with the purpose of supporting project managers in successfully achieving the results of projects related to software-intensive systems and products. To its full extent, the standard provides a Project Management Plan (PMP) composed of nine thematic elements (Project overview, References, Definitions, Project context, Project Planning, Project assessment and control, Product delivery, Supporting process plans, Additional plans), each subdivided into clauses (specifications of on element). The PMP aims to help project managers in specifying objectives, tools, and expected results of a project while scheduling working activities and planning budget allocations.

The management framework provided by AgiSCOnt reuses some of the ISO/IEC/IEEE 16236:2009 standard's elements and clauses that can support ontologists in managing the whole OE process with this methodology: in fact, AgiSCOnt and the ISO/IEC/IEEE 16236:2009 standard share some similarities with regards to fundamental aspects such as providing definitions, identifying the scope and objectives of the project, scheduling working activities, time-sequencing constraints for working activities, assessing and controlling of the outputs. The essential elements and clauses of the ISO/IEC/IEEE 16236:2009 standard that compose AgiSCOnt's management framework are summarized in Table 8.

Clause number	*Clause Name*	*Description*	*Commonalities with AgiSCOnt*
1	**Project Overview**		
1.1	Project summary		
1.1.1	Purpose, scope, and objectives	This clause enables OE process' participants to clearly state the ontology's purpose(s), scope, and objectives.	Step 1 (Ontology Ontology Requirement Specification Document)
1.1.2	Assumptions and constraints	Assumptions on which the ontology is developed should be stated; if some technological requirements or constraints emerge, they should also be detailed in this clause.	Step 1 (Ontology Ontology Requirement Specification Document)
1.1.3	Project deliverables	Work products (conceptual maps, Ontology Requirement Specification Document, Competency Questions, etc.) are listed in this clause, and provisional delivery dates are specified.	Step 1 (Ontology Ontology Requirement Specification Document and Documentation for the target ontology)

[3] The 2009 version of the International Standard ISO/IEC/IEEE 16236:2009(E) Systems and software engineering – Life cycle processes – Project management is supersed by the version released in 2019. However the clauses described in this Chapter remains unchanged in the 2019 version, as they are only reorganized in a different clause numbering.

1.1.4	Schedule and budget summary	The amount of budget foreseen for the OE process should be stated and scheduled throughout the whole duration of the process. See also Clause 5.	
1.2	Evolution of the plan	Considering the recursive nature of AgiSCOnt's activities, the management of the OE process should be flexible enough to foresee the possibility of updating the managerial plan.	
3	**Definitions**	This clause defines (and references documents, if necessary) all terms necessary for the understanding of the project.	This step is not the Glossary foreseen in AgiSCOnt's Step 1.
4	**Project context**		
4.1	Process model	The management framework should specifically refer to AgiSCOnt and its engineering methodology to identify major project work activities (Steps), their relationships, and the outputs foreseen.	
4.4	Methods, tools, and techniques	AgiSCOnt provides methods and techniques for all the Steps composing it. OE process' participants should state here whether or not they require different methods or tools to achieve their goals.	
4.6	Project Organization		
4.6.1	External interfaces	This clause identifies the boundaries between the project's external entities. Here, the identification of external participants as domain experts (and stakeholders such as observers) should be stated.	Essential clause to identify domain experts involved in Step 1.
4.6.2	Internal interfaces	This clause describes the internal resources that participate in the OE process with AgiSCOnt, including internal domain experts, stakeholders, and ontologists.	Essential clause to identify domain experts, stakeholders, and ontologists belonging to the organization and involved in the OE process.
5	**Project planning**		
5.1	Project initiation		
5.1.1	Estimation plan	In this clause, an estimation of the OE process cost, the amount of time required, and the resource requirements should be conducted.	
5.1.2	Staffing plan	The number of staff (and their skills) required for the OE process should be specified. This clause should also account for external interfaces (4.6.1) domain experts that are actively involved in the OE process.	
5.1.4	Project staff training plan	Internal and external staff may require time (and thus costs) to acquire domain-specific knowledge. The methods and costs of staff training are detailed in this clause.	
5.2	Project works plan		

5.2.1	Work activities	This clause specifies the work activities to be performed in the OE process. These should be extracted by the activities foreseen in each of AgiSCOnt's Steps.	AgiSCOnt's Steps provide each a set of activities that can be placed in this clause.
5.2.2	Schedule allocation	In this clause, scheduling relationships between the works activities can be specified to depict the time-sequencing constraints and illustrate the opportunities for concurrent activities.	AgiSCOnt's Steps are thought to be recursive; therefore project managers can a) provide a preliminary time schedule for each step or b) consider the whole activities from Step 1 to the delivery of the prototype (Step 3) as a one-time slot to be allocated according to OE process needs.
5.2.3	Resource allocation	The resources identified in clause 5.1.1 should be allocated to each of AgiSCOnt's work activities (identified in clause 5.2.1), including staff (clauses 4.6.1 and 4.6.2)	
5.2.4	Budget allocation	The budget for the OE process (clause 1.1.4) should be detailed for each of AgiSCOnt's work activities (identified in clause 5.2.1).	The budget allocation should be proportional to the efforts and resources identified in clause 5.2.3
6	**Project assessment and control**		
6.1	Requirements management plan	This clause details control mechanisms for measuring, reporting, and controlling changes to the OE process management plan.	This clause includes reporting the changes that affect the outputs of Step 1 and Step 2.
6.3	Schedule control plan	This clause specifies control mechanisms to measure the progress of work.	AgiSCOnt's expected outputs at the end of each Step can serve as milestones for this clause.
6.4	Budget control plan	This clause specifies control mechanisms to compare the cost of work completed with the costs foreseen in 5.1.1 and 5.2.3 and to identify possible corrective actions.	
7	**Product delivery**	This clause foresees plans to deliver the developed ontology.	This clause should take into account the requirements from AgiSCOnt's Step 1 and Step 3.
8	**Supporting process plan**		
8.2	Decision management	Considering the collaborative approach underlying AgiSCOnt, a decision mechanism should be determined by all OE process' participants.	
8.3	Risk management	This clause specifies the risk management plan for identifying, analyzing, and prioritizing project risk factors and describes the contingency plans.	

Table 8. The list of main elements and clauses from the ISO/IEC/IEEE 16236:2009 standard that compose AgiSCOnt's management framework.

As illustrated in Table 8, AgiSCOnt shares many commonalities with the ISO/IEC/IEEE 16236:2009. Clause number 1 (and its sub-clauses) requires the project manager to identify the project's goals, scope, and purposes, state constraints, assumptions, type, and schedule of the deliverables, together with a plan to evolve the PMP itself. Although beyond the Steps provided by AgiSCOnt, these sub-clauses can support some of the activities of Step 1 – specifically, the preparation of documentation for the target ontology. Moreover, the possibility to evolve the PMP in a flexible way commits to the recursive activities provided in AgiSCOnt's Steps.

Also, considering the pivotal role of domain experts and stakeholders in the methodology, sub-clauses 4.6.1 and 4.6.2 can actively support the project managers in identifying the (external and internal) expertise that can contribute to the Domain analysis and Conceptualization activities (Step 1 if AgiSCOnt), as well as taking part to the development activities and test phases (by proposing Competency Questions, test cases, and participating in the discussion of the results).

Clause 5 and its sub-clauses delve into project management, identifying a time schedule (5.2.2), budget allocation (5.2.4) and resources (human and technological) (5.2.2), and detailing the work activities (5.2.1): on this clause, AgiSCOnt's three-Steps structures (and its activities) can support the project manager in identifying and planning time and budget schedule. In particular, the schedule allocation sub-clause (5.2.2) could benefit from AgiSCOnt's recursive structure of activities: in this case, the project management can either provide a preliminary time schedule for each Step or consider the whole Steps as a single time slot to be allocated according to cogent needs emerging during the OE process (e.g., more time is needed for Domain analysis, less time is required for Conceptualization, an issue emerged during the Step 2 forces the activities to be allocated more time, etc.).

Assessment and control of the project are ensured by clause 6 and its sub-clauses. Also, in this case, AgiSCOnt's expected outputs at the end of each Step can serve as a milestone for assessing the achievements of the OE process (6.3) and to evaluate the costs of such activities (6.4). Control activities are also extended to changes on the PMP (6.1) that are caused by Step 1 and Step 2 recursive activities – e.g., if there emerges the necessity to allocate more budget, modify the PM by extending or reducing time schedules, etc.

Clause 7, dedicated to the delivery of the developed ontology, can support the activities of AgiSCOnt's Step 3 – as long as the flexibility of requirements foreseen for Step 1 is ensured in the previous clauses of the PM. Finally, clause 8 and its sub-clauses delve into supporting the whole OE process. In this case, relying on the collaborative nature of AgiSCOnt, decisions should be determined by the project manager together with involved stakeholders (8.2), while the identification of potential risks should be performed by the project manager and discussed in case these also involve domain experts or other stakeholders (8.3).

The adoption of the ISO/IEC/IEEE 16236:2009 standard answers to the necessity of providing clear instructions on the managerial side of the OE process. Although optional, the preparation of a PMP in AgiSCOnt can significantly contribute to some of the activities foreseen in the methodology, thus supporting the ontologists and developers in some tasks. Nonetheless, by adopting a reduced version of the whole standard – with clauses selected to provide a fast and flexible management framework while also serving and/or contributing to the work foreseen by AgiSCOnt – it is possible to describe the OE process of complex projects, in which the development of an ontology is the main – but not the only – output to be achieved.

In order to estimate costs related to the OE process, the clauses in the PMP can be filled with a consistent approach on which project management and project participants (including external OE process participants, if necessary) agree. As mentioned in Chapter 2, there are not many dedicated models or tools to estimate the costs related to ontology development activities: from the one hand, this may be due to the variety of activities to be conducted and some bottlenecks that are well-known in OE (see Step 1); on the other hand, the recursive structure of the activities foreseen by some OEMs makes it difficult to come to solid esteem of the costs. For example, considering that ONTOCOM [45] is one of the complete models to estimate the costs related to ontology development, adopting this model requires the ontologists to know the number of entities that the ontologists are going to model – which may not be so easy to have, especially if the Domain analysis activities have not been completed yet. Therefore, AgiSCOnt does not propose any particular cost estimation model. Nonetheless, it is worth noticing that the PMP provides the means to allocate budget to work activities, thus supporting the planning of the possible costs each Step may have.

4.2.2 Step 1: Analysis and Conceptualization

The first activity to be conducted is to agree on the purposes of the ontology and its requirements. This activity is the basis of **Domain analysis**, and it is necessary for **Conceptualization**. All domain experts and ontology engineers must agree on the purposes of the target ontology, what it is going to do, who is going to use it, and what is expected to be delivered by the ontology. Once these pieces of information are defined, another delicate and fundamental activity takes place: gathering the knowledge relevant to and related to the domain. This activity – indicated as “knowledge elicitation”, i.e., the adoption of a set of methods and techniques to extract knowledge from domain experts [116] – is represented in almost all of the macro-level OEMs since it constitutes the foundational basis for ontology development. As for AgiSCOnt, knowledge elicitation is an iterative activity that leverages specific knowledge engineering techniques and enables ontology engineers and domain experts to cooperatively provide a Domain analysis and a shared Conceptualization. These two phases happen simultaneously in AgiSCOnt, as the identification of ontology’s requirements (using CQs) is used to identify and define concepts in a continuous and collaborative discussion (which takes the form of an unstructured interview); concepts are then defined (in a Glossary) and linked together using Conceptual Maps. The two phases end when the domain experts involved believe in having reached a shared conceptualization of the domain (which serves the identified purposes of the ontology) and when all the relevant terms have been identified and defined. These two activities account for the highest impact on the total OE effort, as demonstrated by Simperl et al. [98]: it is, therefore, fundamental in AgiSCOnt to acquire shared knowledge in a collaborative way and to agree on a consequent conceptualization before starting the development step – although, the methodology is flexible enough to allow modifying the target ontology and its underlying conceptualization also in the development step.

Thus, the Conceptualization (through Conceptual maps) happens concurrently with the Domain analysis so that the Conceptual map, the Glossary, and the list of CQs evolve by means of the continuous debate among the stakeholders involved in the OE process. Also, these two phases are expected to deliver Documentation of the ontology, which consists of the latest version of the artifact mentioned above (and the addition of the Ontology Requirement Specification Document compiled by the ontologists).

4.2.2.1 Knowledge elicitation in OE: techniques and their use in OEMs

Knowledge elicitation (sometimes also referred to as “knowledge acquisition”) is very time-consuming and expensive, to the point where it has been defined as “knowledge acquisition bottleneck” since 1983 [117]: it takes longer to gather knowledge from experts and documentation than to write the software. Knowledge elicitation is both the core of OE and a very delicate activity, considering that the whole ontology depends on the quality of the acquired knowledge: in terms of actual knowledge elicitation, it is possible to acquire knowledge also from non-human resources, such as handbooks, technical reports, existing taxonomies, etc. However, for complex or multidisciplinary domains, human experts remain crucial stakeholders. Shadbolt et al. [116] argue that expert systems not informed by actual experts and their understandings are often poorer, referring to a slogan in the knowledge and cognitive engineering community – “*Gold is not in the documents*”. In large contexts (e.g., medium or large enterprises, organizations, multidisciplinary teams, etc.), the knowledge necessary to develop an ontology may be distributed among many experts, who may also have difficulties in making their knowledge explicit because it is routinized.

There exist several knowledge elicitation techniques involving domain experts and ontology engineers. The goal of techniques is to limit the knowledge acquisition bottleneck while minimizing the amount of time and effort necessary to elicit the knowledge and maximize the acquired knowledge (in terms of its quality and relevance for the OE process at hand). Knowledge elicitation techniques can be categorized into *natural* and *contrived* methods, where natural techniques are those that a domain expert can adopt while expressing or displaying his/her expertise [118], and contrived techniques require a domain expert to undertake a task that elicits knowledge in ways that are not usually familiar to the domain experts [119]. Examples of natural techniques are interviews (which can be unstructured, semi-structured, or structured) and Protocol Analysis, while examples of contrived techniques are Diagramming (with Conceptual Maps, Laddered Grids, and Process mapping), Sorting and rating methodologies, and Constrained processing methodologies.

Most OEMs do not rely on a specific knowledge elicitation technique nor suggest using specific tools. The *Domain analysis* phase was a primary concern in those methodologies that were developed between the end of the 1990s and early 2000s. However, the activity of knowledge elicitation is mostly taken for granted, i.e., it was supposed that ontologists and knowledge engineers were able to rely on consolidated elicitation techniques. Early macro-level OEMs (e.g., Uschold and Gruninger's skeletal methodology [120], METHONTOLOGY [7], and CommonKADS [121]) underlined the importance of acquiring knowledge for OE, although the techniques to elicit it are not specified: the goal of capturing knowledge is to build a common ground to get to a shared conceptualization. DILIGENT [71] tackles knowledge elicitation by means of discussion among domain experts (which could be traced back to the unstructured interview technique), in which experts decide when a topic is sufficiently discussed. A similar approach is also endorsed in DOGMA [75], where "classical" brainstorming techniques and scenario development are suggested. A slightly different approach is proposed in HCOME [122], where the role of knowledge workers is underlined: leveraging on the difference between personal knowledge and group knowledge, this OEM foster the development of individual-based conceptualizations of a (portion of a) domain, which are later put together and mapped in a synergic effort, in which discussions remain the main technique [73].

Together with the possibility of relying on experts' discussions, the role of Conceptual Maps in the knowledge elicitation activity is also relevant. Although not mentioned in early macro-level OEMs, conceptual modelling found a place in knowledge elicitation in the early 2000s. Conceptual maps are collections of propositions portrayed in a bi-dimensional graph, as they usually graphically represent concepts connected by lines and/or arrows – which can be labelled. This type of elicitation technique is widely adopted, and its efficacy was also assessed in different domains; moreover, it was highlighted how having experts cooperating on the same conceptual map leads to very efficient representations of a domain [123]. For these reasons, Conceptual maps have also been adopted in OE. In particular, Jarrar et al. [75] stated the benefits deriving from reusing conceptual modelling techniques in OE – i.e., maps can make ontologies more understandable, can facilitate their development phase, can foster their adoption, and ultimately conceptual maps can be mined or "ontologized". In particular, Conceptual maps seem to play a significant role in the early stages of the OE process, namely knowledge elicitation and capturing: Castro et al. [124] developed a macro-level OEM that heavily relies on Conceptual maps, which were used iteratively by domain experts to represent their knowledge in a graph form. This approach enabled experts to identify and represent "part-whole" and "is a" relationships, even though they initially represented specific test cases with instances rather than classes, using a bottom-up approach.

AgiSCOnt adopts unstructured group interviews and Conceptual maps for knowledge retrieval. Both techniques can be used in decentralized settings, thus allowing also remote and asynchronous cooperation. While unstructured interviews enable domain experts to discuss their expertise in a non-constrained way, Conceptual maps force them to agree on a shared conceptualization. AgiSCOnt does not prescribe any specific approach for discussing the domain at hand or developing the conceptual map: domain experts can tackle the problems from a bottom-up perspective (i.e., by delving into specific cases and then abstracting them into more general concepts and relationships), or top-down (i.e., by starting with the most general concepts and then detailing specific use cases). The use of these techniques is limited to the elicitation of those pieces of knowledge necessary to inform the ontology and its purposes.

4.2.2.2 Domain Analysis: identification of the CQs and ontology requirements

As mentioned above, the first step into Domain Analysis consists in gathering ontology requirements and purpose, using unstructured interviews involving the group of domain experts and ontology engineers as an elicitation technique. This discussion has the objectives of getting all participants at ease and starting to identify some preliminary questions that the ontology is supposed to answer. During the discussion, the general goal of the ontology (purpose) should be explicitly stated and annotated by the ontologists; the purpose must be agreed on by all participants.

The ontology engineers should also suggest identifying Competency Questions (CQs) to investigate the ontology's functional requirements. Ontologists may therefore propose a set of questions that the ontology should be able to answer. CQs (and their answers) are formulated in natural language and then used to identify the main concepts and relationships that inform the development of the domain's conceptual map.

In this phase, the ontologists can collect information to compile the Ontology Requirement and Specification Document (ORSD) [99, 125], a document that keeps track of the knowledge elicitation process, provides hints in finding candidate ontologies to be reused, and provides a means of verification of the ontology throughout its development. AgiSCOnt adopts a simplified version of the ORSD described by Suárez-Figueroa et al. [99], proposing a document that allows to state:

- Purpose of the ontology: the general goal of the ontology, upon which the maximum agreement should be reached by all domain experts and ontology engineers.
- Intended end-users: the persons and/or the applications expecting to be using the ontology.
- Intended use: what it is possible to do with the ontology, what it enables the users to do.
- Functional requirements: a list of content-specific ontology requirements expressed through CQs.

The ORSD is a document that must be updated as domain experts discuss, introduce new terms (and formulate new CQs), and sketch new concepts and relationships in the Conceptual map. Also, the ontology engineers should keep track of the most relevant terms that appear in the CQs and their answers, as well as of those terms that are indicated by domain experts: these terms constitute the Glossary, a document listing all the terms and their definitions – which are provided (and agreed on) by domain experts. The Glossary – the second document coming out of this phase – provides the definitions for concepts and relationships that are going to be modelled in the ontology and, therefore, must be kept updated.

4.2.2.3 Iterative Domain Analysis and Conceptualization

The discussion – which is conducted in the form of an unstructured interview – does not leave the ontology engineers as passive observers of the elicitation process. The ontologists have the responsibility to contribute to the outputs (i.e., the list of CQs and the ORSD, the Conceptual map, and the Glossary). In particular, by leveraging the combination of unstructured interview and Conceptual map, the ontologists can promptly observe if the domain experts are moving too far from topics relevant to the agreed purposes of the ontology or if there are concepts and relationships that are not clearly defined. In fact, while the interview is conducted in natural language, thus allowing for vagueness in the definition of entities, drafting the Conceptual map takes place in a graphical form: here, the ontology engineers play a pivotal role in explicitly asking whether a concept is self-defined or is related to other relevant concepts (which would be therefore added to the Glossary, if not already described), or whether an arrow, a line or a string have a particular meaning (i.e., if they stand for *part-whole* or *is-a* relationships).

The drafting of a Conceptual map does not rely on any particular modelling technique. Domain experts may use UML or "simpler" maps, such as those adopted in learning environments. As Conceptual maps are a common tool, it is here assumed that domain experts would be able to easily get a grasp on maps and their functioning. In particular, Novak and Gowin's [126, 127] Concept maps provide very intuitive schematic devices for the representation of concepts and the relationships holding among them: also, they can provide the means to sketch a preliminary hierarchy of concepts. However, AgiSCOnt stresses the importance of the granularity of the Conceptual map: the graphical representation of the domain should contain all the elements necessary to properly identify the knowledge and possibly the type of the entities being represented. In other words, the map should indicate what is *concept* – i.e., an abstraction, an idea, or a class –, what position a concept occupies with respect to other concepts (its "location" in the hierarchy of the concepts), which *relationships* a concept holds with the other concepts represent (how a concept is "linked" with the others and through what link). The map is completed with *examples*: domain experts, encouraged by ontology engineers, should provide factual examples of instances that are representative of a concept and of the relationships it holds with other concepts. The examples can be provided by use cases, which are "enacted" in the Conceptual map. Figure 4.2 provides an example of Conceptual map populated with some examples.

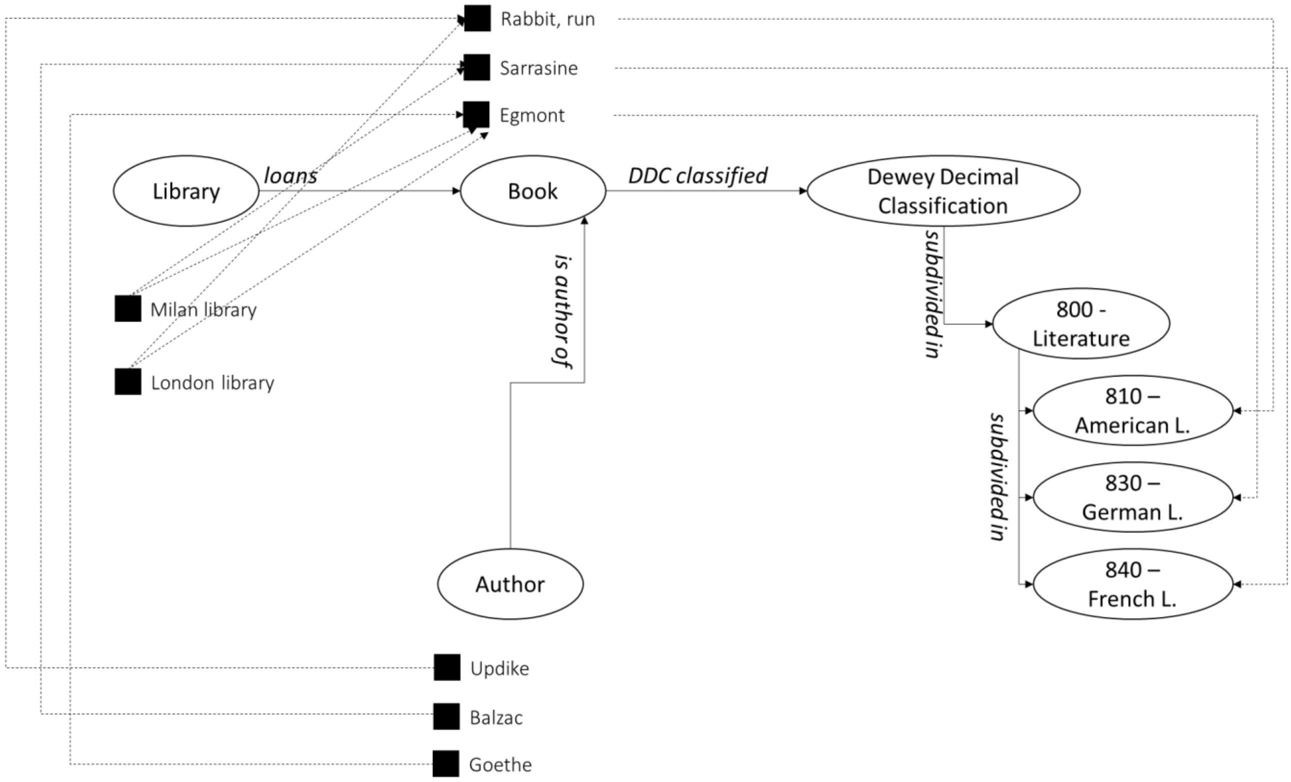

Fig. 4.2. A Conceptual map representing libraries, the books they loan to readers, their Dewey decimal classification, and their authors. The circles represent concepts, the labelled arrows represent the relationships holding among them, and the black squares represent the examples – for which the same relationships modelled for concepts hold.

While the Conceptual map populated with examples can help the domain experts and ontologists in the early identification of misrepresentation of concepts or relationships, it also provides some suggestions on *how to model* some facts in the subsequent step. In fact, referring to Figure 4.2, it seems that each *Author* is linked to a *Book* with a relationship (*is author of*) that could be modelled with an object property; the same thing happens for the *loans* relationship holding between a *Library* and a *Book*. However, the relationship between a Book and one of the sub-concepts under Dewey Decimal Classification is not clear – i.e., it could be modelled as an object property or each member of the concept Book could also be a member of one of the sub-concepts specifying the Dewey Decimal Classification, or else the whole classification could be modelled by recurring to datatype properties. Moreover, while the hierarchy of concepts underlying the Dewey Decimal Classification is pretty explicit, no such conclusion can be deduced for the concepts *Author*, *Book,* and *Library*, which are only related through different relationships.

4.2.2.4 Reuse of ontological and non-ontological resources

The activities of Domain analysis and Conceptualization could elicit information regarding knowledge, taxonomies, concepts, relationships, methods, questionnaires, etc., that are somehow involved in the domain at hand and used by experts regularly. It could also happen that some of this information is already modelled into ontological or non-ontological resources – i.e., part(s) of what the domain experts use or adopt during their professional life is already modelled in the form of an ontology or is modelled in a structured but non-ontological way (such as schemas, instructions, databases, documents, etc.). In this case, the ontologists should check whether this knowledge is already modelled into an ontology by searching in Ontology Libraries. If the searched knowledge does not appear in any ontological representation, the ontology engineers drive the Domain Analysis and Conceptualization activities to understand to what extent this knowledge should be included in the ontology and what features it has. Otherwise, ontologists should reuse the ontology containing the relevant pieces of information.

In the case of the reuse of existing ontologies, in this phase, the ontological models and/or their parts are just identified and further investigated. The selection of concepts and relationships to be reused is delegated to the Development phase, in which tested solutions for the reuse of ontologies (or their parts) are adopted to include those fragments that are believed to be relevant. Recalling the example from Figure 4.2, AgiSCOnt proposes to leverage the unstructured interview and the Conceptual map to identify those non-ontological and ontological resources that can be potentially reused in developing the ontology. In this case, domain experts and ontology engineers may realize that the Dewey Decimal Classification is an existing and widely-adopted standard for locating books according to their subject (a non-ontological resource that is also available in an ontological form [128]) and may also find in the Dublin Core vocabulary [129] some entities that could help in describing the relationship occurring among the ontology's concepts.

Ontology reuse is a key attribute of OE, and the majority of macro-level OEMs (including agile) mention it, but there are no established principles and practices so far. While ODPs emerge as a possibility for the reuse of bits of ontologies in ontology authoring, the selection of ontologies that can be (partially or completely) reused in a target ontology is far from being trivial task. As a result, many ontologies formalizing the same domain (or parts of it) are often not reused [108]. Although researchers have proposed different approaches to identify candidate ontologies to be reused [130–132], reuse remains very scarce and often limited to referencing some of the reused ontology elements via their URIs [133]. AgiSCOnt's approach is to identify key concepts emerging in this Step to search for reusable candidate ontologies in different Ontology Libraries and scientific literature. Candidate ontologies can then be discussed among the domain experts and ontologists to reach an agreement regarding the extent of the reuse.

4.2.2.5 Summary of outputs of Step 1

The activities of Domain analysis and Conceptualization, concurring simultaneously and composing the Analysis and Conceptualization step, generate the following outputs upon conclusion:

- The list of CQs
- The ORSD
- A Conceptual map of the domain to be modelled
- The preliminary indication of possible ontologies to be reused

These outputs inform the step of Implementation and Development.

4.2.3 Step 2: Development and Test

Once the outputs from the Step 1 Analysis and Conceptualization are consolidated, the Step 2 Development and Test can start. This step is the prerogative of ontology engineers, as they may be the only staff to have knowledge and expertise in ontological languages. Nonetheless, domain experts with experience or knowledge of OE and its languages may take part in this activity, as well. In this step, developers select the ontological languages for knowledge representation according to the domain complexity elicited in the previous step.

However, in AgiSCOnt, this Step is strictly connected to the previous one and cannot be conducted regardless of domain experts: in fact, whenever ontology engineers are dubious regarding how to model a particular entity (e.g., if it should be modelled as a class or a property), they should consult domain experts and presenting them with the various possibilities and their consequences. In this way, domain experts have the opportunity to further specify and clarify some details of the Conceptual map, while the ontologists can understand which modelling solutions best suit to solve the modelling problem at hand. In case minor modifications to the conceptualization are required, the Conceptual map (and all the outputs from Step 1 affected by the modification) should be updated with the new information.

The development can take advantage of the Conceptual map drafted in the previous step and leverages on ODPs to identify modelling solutions that can be reused. Considering that each ODP is described with a GUC (as mentioned in Chapter 2), ontology engineers can observe the target ontology Conceptual map and see whether parts of the map share structural or logical, or content similarities with one or more ODP. In this way, the ODP catalogs can provide ontologists with support in building the entities and relationships that populate the target ontology.

4.2.3.1 Reusing existing ontological resources

As mentioned above, ontology reuse is a cornerstone of the Semantic Web, although reuse is conducted in different ways. The majority of macro-level OEMs mention the "ontology reuse activity", but very few provide insight on how to perform it nor details regarding how to identify possible candidates to be reused. As underlined by different sources, there exist different types of reuse (usually identified as "ontology merging" and "ontology integration", with the first indicating the union of different existing ontologies representing the same domain into one single ontology, and the second consisting in assembling, adapting, expanding or specializing other existing ontologies into the target one) [26].

AgiSCOnt recognizes that reuse is a time-consuming activity and that ontologists may face one of the following scenarios when evaluating ontologies to be reused:

a) A complete domain ontology describes a portion of a conceptualization in a complete way; therefore it can be adopted completely (the reused ontology is integrated with the target ontology);
b) A domain ontology contains some concepts and relationships that, if reused in the target ontology, can satisfactorily describe a part of the conceptualization.

From a practical perspective, the first scenario consists in reusing a complete ontology: therefore, the target ontology imports the reused one. However, reusing an ontology in its entirety may be impracticable and/or very costly; therefore, ontology engineers may want to reuse a limited number of entities (second scenario): in this case, relying on the Minimum Information to Reference External Ontology Term (MIREOT) [134] techniques and its deriving tools – such as OntoFox [135] – can help ontologists in extracting a module from an ontology. The module contains the targeted entity, its unique identifier, the set of super-classes, and annotations. Another possibility for reusing parts of an ontology is to reference their reused entities URIs (soft reuse [108]): this way of reusing is effective in importing the entities that are relevant to the target ontology but requires ontologists to carefully take into account that constraints of the reused ontology may not be consistent anymore – and, as a consequence, they should be remodelled within the target ontology.

4.2.3.2 Reusing non-ontological resources

If, during Step 1, the unstructured interview and Conceptual map underline the possibility of reusing a classification, taxonomy, model, etc., that are not yet formalized in an ontology, the ontologists can include the parts of these non-ontological resources that are relevant to the conceptualization's documents.

In the given example, domain experts elicited that the Dewey Decimal Classification is relevant for the domain at hand. Therefore, the entities identified as necessary to provide a domain description must be modelled into the target ontology from scratch. If there exists any documentation of the non-ontological resources (for example, a Wikipedia page for each of the classes of the Dewey Decimal Classification), the ontology engineers can adopt RDFS annotation properties (e.g., `rdfs:seeAlso`, `rdfs:comment`) to link the documentation with the entities.

4.2.3.4 Reusing ODPs as a guide for authoring

AgiSCOnt recommends ontologists reuse and adopt ODPs – in particular, Content and Structural ODPs – to support the development phase at authoring level. For example, ontology engineers could take inspiration from the Content ODP "Collection", which (according to its Generic Use Case) can be used to state what entities are contained in a collection. The diagram for this ODP is reported in Figure 4.3 and shows two `owl:inverseOf` object properties – `:hasMember` and `:isMemberOf` – having the Collection class and the target Entity class as `rdfs:domain` and `rdfs:range`, respectively.

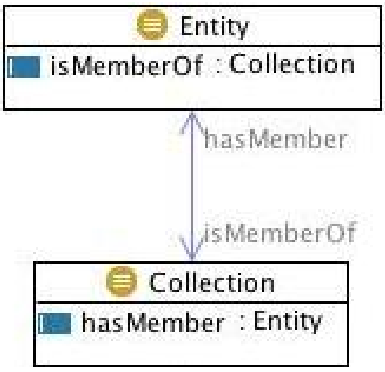

Fig. 4.3. The diagram provided for the Collection Content ODP [136].

This Content ODP can serve as a "guideline" to model the relationship between a Book and one of the subclasses of the Dewey Decimal Classification. Ontologists can here develop this portion of the domain in two ways:

a) The first consists of importing the whole Collection Content ODP, explicitly stating the equivalence between the property `ex:DDCClassified` and `:isMemberOf`, between the `ex:Dewey_Decimal_Classification` class and the `:Collection` class, and between the `ex:Book` and the `:Entity` classes.
b) The second consists of "mapping" the pattern emerging from the target ontology Conceptual map (which connects a *ex:Book* to its *ex:Dewey_Decimal_Classification* subclass) and re-building the Content ODP with the classes and relationships extracted from Step 1.

While importing the whole OPD would require more time – and, for larger ontologies, it may require extra work in finding the correct correspondences between the target ontology entities and the ODP's entities –, the second approach reuses ODP as a "guideline" to solve a domain-grounded modelling problem. As underlined in Chapter 2, the complete reuse of the ODP is a step towards (micro-) ontology reuse. Nonetheless, larger target ontologies may require a particular effort to adopt ODPs: considering that there is a lack of domain-specific ODPs and that Content ODPs may not have been validated against domain-specific requirements, ontology engineers should carefully select which ODPs can be effectively reused. A criterion for the selection of ODPs is that their inclusion in the target ontology does not require making major changes to the conceptualization deriving from Step 1 (this would result in a review of the outputs of the Analysis and Conceptualization Step).

The advantages of completely reusing one or more ODPs consist in developing a target ontology with a minimum level of interoperability (as described in Chapter 2), but ontologists should carefully assess whether costs overcome benefits. The second way provides a good trade-off between the necessity of relying on existing and solid authoring solutions and interoperability since the re-built ODPs can be later mapped to their original counterparts in Step 3.

4.2.3.5 Selecting the OWL profile

As mentioned in Chapter 1, the specifications for OWL 2 present three profiles (OWL 2 EL, OWL 2 QL, and OWL 2 RL). In this Step, ontologists may choose to select a specific OWL profile for the development of the ontology, bearing in mind the characteristics of each profile and considering the outputs of Step 1. In particular, the Conceptual map and the ORSD deriving from the previous activities should support ontologists in discerning whether the target ontology is very rich in classes (which are described through other classes, thus using class expressions) and presents a heavy use of classification (e.g., it contains a vast amount terms, which is something that can be observed by having a look at the glossary output from Step 1), or if the Conceptual map presents only those relationships that can be represented with an Entity-relationship model (e.g., the Conceptual map and the list of terms was gathered by relying on an existing SQL database) and the purpose of the ontology is to provide a semantic-enriched representation of a database schema. While in the first case, ontologists may be oriented toward the adoption of OWL 2 EL, in the second case, OWL 2 QL seems to be the best choice.

The selection of a profile should also take into account the reuse of other ontologies: if the target ontology integrates an existing ontology developed with OWL 2 Full, the ontologists should bear in mind that developing the ontology with a profile (i.e., a subset of OWL 2 Full) may bring some issues in reasoning – for example, an OWL 2 RL reasoner sacrifices some aspects of class expressiveness that an OWL 2 Full ontology may adopt to describe classes. Also, the selection of a specific profile depends on the availability of OWL tools (ontology editors and reasoners) able to properly adopt a profile's specifications.

However, considering that the three main profiles described in Chapter 1 share many similarities with OWL 2 DL and the fact that there exist several tools (both ontology editors and reasoners) able to manage the DL profile, ontology engineers can start the development activities with this profile.

4.2.3.6 Testing the prototype

The Development Step foresees the engineering of a prototype ontology. The prototype needs to be tested against a) the list of CQs gathered in Step 1 and b) a set of use cases that emerged from Step 1 and were provided by domain experts and target users of the ontology.

In order to verify that the developed ontology satisfies the requirements expressed with CQs (a), the prototype must be queried with SPARQL [137]. Converting natural-language CQs in SPARQL enables the retrieval of answers for the requirements underlying the CQs: AgiSCOnt leverages a well-established method to test the adherence of the model to its requirements [99]. This test ensures that the ontology is able to serve all the purposes it was developed for.

Considering that the Domain analysis and Conceptualization Step may have followed a bottom-up approach (i.e., domain experts started from real cases to elicit knowledge and conceptualize the domain) or it took advantage of examples to gradually verify the robustness of the conceptualization, the ontology should be tested to understand whether it is able to represent those examples and use cases satisfactorily (b). Therefore, ontologists need to work in close collaboration with domain experts to populate the target ontology with use cases and verify that the prototype provides all the necessary entities to model them. If the ontology does not provide all the necessary entities or domain experts recognize that some ODPs may be enhanced and/or modified to provide better expressiveness, ontology engineers can intervene with ad hoc modelling solutions. However, in case the testing with use cases underlines a lack of entities, the Conceptual map and CQs may be subjected to modifications. In this way, Step 2 is able to directly influence the outputs of Step 1 by re-opening the discussion among domain experts and ontologists to finalize the prototype (a discussion limited to the parts that require interventions and to those entities that are linked to those parts).

To ensure that use cases provided by domain experts are not tampered with the conceptualization developed in the prototype, AgiSCOnt suggests involving the domain experts in the use cases gathering process *before* showing them the finalized prototype. Also, if possible, use cases should be provided by (or gathered from) other stakeholders to which domain experts may have access. The use cases gathering process may lead to the elicitation of new requirements, but it may also happen that some stakeholders would provide some use cases that are not in the scope of the ontology. As a consequence, domain experts and ontology engineers must scrutinize use cases and discern those use cases (or parts of use cases) that can actually provide some relevant information to the target ontology.

In the given example, domain experts may ask other library experts (stakeholders) to provide some documentation about specific use cases they deem interesting. Some stakeholders might therefore underline that it could be interesting to divide the class of `ex:Library` into two subclasses `ex:Children_YoungAdult_Library` and `ex:Adult_Library`, so that it is possible to differentiate those libraries that are dedicated to only one type of readers (children and young adults and adults). Stakeholders could also present a use case in which, in addition to books lent to readers, the libraries organize different activities: in this case, domain experts and ontologists may decide that – considering the purpose of the target ontology – the suggestion deriving from the use cases and related to modelling two subclasses of `ex:Library` can deliver some useful insights and relevant concepts to the target ontology, while a representation of the activities does not fit the purposes of the ontology.

4.2.3.7 Summary of outputs of Step 2

Step 2 foresees the delivery of:

- A prototypical target ontology developed with ontological languages
- CQs test with SPARQL to assess the fulfillment of target ontology's requirements stated in the ORSD
- A set of use cases test modelled in the target ontology to verify the suitability of the model in properly representing the domain at hand and its complexity among domain experts and stakeholders

4.2.4 Step 3 Ontology use and updating

The main output of Step 2, the prototype ontology, is then adopted in the applications that foresee its use – and that are mentioned in the PMP (clause 7). According to the specific availability of the target ontology (i.e., if it is publicly accessible or not), the possibility of gathering feedback from external stakeholders can play a pivotal role in updating the ontology. In fact, a target ontology complete with documentation enables other ontology engineers and stakeholders to fully understand the modelling choices, the purpose, and the scope of the model. In this way, external stakeholders are encouraged to investigate the possibility of integrating new pieces of knowledge into the prototype so that it can match their objectives. In this case, AgiSCOnt refers the ontologists to its first Step so that the OE process can start with the aim of modifying the original ontology.

Similarly, if the ontologists and the domain experts who took part in the OE process of the original target ontology acknowledge the possibility of updating the ontology with new pieces of knowledge, AgiSCOnt's recursive structure enables the possibility of modifying the outputs of Step 1 (and, consequently, the outputs of Step 2).

In general, feedback generation for the target ontology is essential to update and refine it, in particular if it is expected to be reused by third parties (stakeholders that were not involved in the original OE process) or integrated into third-party applications. According to the PMP, once external feedback is gathered, the project manager(s) and/or the decision authority (identified in clause 8.2 of the PMP) choose the best course of action, taking into account also the risks (PMP clause 8.3) related to heavy re-engineering of the target ontology.

4.2.4.1 Fostering the dissemination and possible reuse of the target ontology

To foster the dissemination of the target ontology and its purposes, AgiSCOnt suggests that ontologists develop the target ontology including also annotations (e.g.: `rdfs:comment`). Annotations should indicate who developed the ontology, the `owl:versionInfo` of the target ontology, and other relevant information (e.g., the name and website of the research project for which the target ontology was developed, the name of the domain experts who took part to the engineering process, etc.). Similarly, the documentation should provide contact to key persons – who will be in charge of gathering feedback for updating activities and can be identified during the definition of the PMP.

According to the possibility of having the target ontology and its documentation public (which is something that should be identified at a PMP level, in clause 7), three different scenarios for the reuse of the model can be possible:

1. Both the ontology prototype and its documentation are publicly available (or available on request): in this scenario, it is possible that other ontologies may reuse the target ontology and/or map some of their entities to it. The availability of documentation makes it easy for external stakeholders to understand the purposes, scope, and modelling choices of the target ontology, thus fostering its reuse in other contexts.
2. Only the prototype ontology is available: in this scenario, if the target ontology is clear (i.e., if its concepts and relationships are described with class expressions, labels, comments, etc.), it is possible

that other ontologies may reuse or map to some of the target ontology's entities. Nonetheless, the absence of documentation might open the possibility of misinterpretation of parts of the target ontology.

3. Only the documentation is available: in this scenario, the reuse of parts of the target ontology (its ODPs or some excerpts that are replicated in other ontologies through soft reuse) is still possible, although limited to those parts represented in the documentation. Nonetheless, this is not an ideal case for fostering reuse, as it does not provide clear and precise indications of the target ontology.

4.2.4.2 Alignment with other ontologies

One of the outcomes of the interactions occurring among stakeholders, users, domain experts, and ontologists may be the possibility of aligning (or map) the developed target ontology with other existing models – or parts of them. The alignment activities have the purpose of generating a set of correspondences between entities of different ontologies [138].

This can foster the adoption of the developed target ontology within other contexts and highlights similarities between two models that describe (partially) the same domains. Ontology alignment exceeds the scope of OE and, in particular, the scope of AgiSCOnt. Nonetheless, it is worth noticing that adopting the best practices of documenting the ontology may support matching activities.

It may also happen that the discussion regarding the alignment among entities may underline the possibility of partially restructuring the target ontology: in this case, the recursive activities of AgiSCOnt can support ontology engineers (and domain experts) in such activities.

4.2.4.3 Summary of outputs of Step 3

At the end of Step 3, the following outputs are possible:

- A list of new requirements, originated by feedback collection, to be discussed by the ontologists and domain experts to update the target ontology

4.3 Evaluation of AgiSCOnt

AgiSCOnt needs to be evaluated according to the framework presented in Chapter 3, with the aim of assessing the quality of the process foreseen by its structure and the quality of the outcomes that are produced with this agile OEM. It is also interesting to understand whether AgiSCOnt performs better than other agile OEMs by supporting OE process participants in different activities and helping them deliver good ontologies.

For this purpose, this Section introduces a slightly modified version of the evaluation framework described in Chapter 3 and adopts it to test with participants AgiSCOnt and the three other agile OEMs already investigated (UPONLite, SAMOD, and RapidOWL). The results gathered from this experiment are then presented and discussed to underline the strong points and weaknesses of AgiSCOnt when compared to UPONLite, SAMOD, and RapidOWL.

4.3.1 Evaluation framework

The evaluation framework is very similar to the one presented in Chapter 3 (and in [90]), with the exception of an extra feature for the *process*: the *Operational support*, aimed at investigating whether an agile OEM provides concrete support in helping ontologists by performing some of its activities. The framework is therefore modified as represented in Table 9.

Features	Description
Outcome (ontology)	
Reused models	The ontology produced by participants reuses any existing model (ontological or not, in its entirety or parts of it), including reusing one or more fragments of a foundational ontology to start the development or any ODPs
Documentation delivery	The ontology produced by participants comes with the documentation required by the OEM adopted. The documentation includes the List of Competency Questions, the Glossary or Lexicon, and the Conceptual map. There exist several types of documents that can attest the modelling choices and the Ontology Requirements Specification Document can help in specifying several details regarding the developed ontology [99] – no agile OEM foresees this document
Iterations	The number of OEM full cycles required for participants to complete the ontology. Contrary to waterfall OEMs (and in part similarly to lifecycle OEMs), some agile OEMs foresee the possibility of reiterating part of their instructions. Each OEM specifies whether some activities are dependent on others
Relevance of the model	The degree to which the ontology provides the information that is expected to be modelled. These metrics assess whether the structural measures (classes, properties, individuals, etc.) composing the model contain the information identified in the requirements. It is evaluated in terms of the definition of Domain and Range of properties, Disjunctions, and Restrictions to model the requirements elicited, and in terms of presence or absence of Unsatisfiable concepts [100]
Time	The number of hours the participants spent to develop the ontology
Structural measures	The number of features available in the ontology language that are used to model the ontologies. They include classes and subclasses, object properties, datatype properties, individuals, axioms, SWRL rules, and annotation properties conveniently adopted to represent the domain at hand
Logical consistency	Checks with a reasoner if terms have a consistent meaning in the ontology
Process (OEM)	
Clarity and simplicity	The possibility to rely on clear and simple instructions to help developers move from an informal representation of the domain to a formal one
Adaptability and flexibility	One of the principles of agile methodologies, adaptability and flexibility of an OEM foresee the opportunity to accommodate changes at any level of the OE process
Knowledge management support	Documentation and reuse of existing ontologies or non-ontological resources cover a pivotal role in OE since it supports knowledge acquisition and transfer processes
Teamwork and cooperation	Fundamental aspects of a collaborative OEM, teamwork and cooperation foresee developers and domain experts working together in different steps of the development process
Developer perceived effort	Perception of the effort spent by developers and the added value of iterations for the OE process. The developer may perceive an OEM as easy to follow or particularly strenuous in some aspects

Developer perceived role	The value perceived by individuals regarding their contributions and interactions during the whole collaborative OE process
Innovation support	The value perceived by the individuals on the possibility of testing novel solutions, adopting a *try-and-learn* approach in their implementation, fostering the creative efforts in the conceptual model and its refactoring
Operational support	The value perceived by the individuals on the concrete possibility of being supported during the OE process in one or more tasks through the steps or activities foreseen by the methodology

Table 9. A table that summarizes and details both the outcome and process features, including the addition of operational support.

The *operational support* process feature attempts to answer a pivotal question: from a practical perspective, how does the agile OEM support (novel) ontologists in developing a model? Except for the clarity of instructions and the ways OE process participants can feel about using a methodology, it is important to research if an agile OEM is structured in such a way that it can actively help ontologists in performing one or more activities.

Contrary to the previous evaluation framework, the focus on reuse and authoring that is foreseen by AgiSCOnt can also be investigated in the other agile OEMs to assess whether they are perceived as supportive during the practical execution of the OEM's steps. As pointed out in Chapters 2 and 3, agile macro-level OEMs foresee tasks for the development of the ontologies throughout the whole OE process; however, very few of them dig into *how to* model something. They may lack an authoring perspective that can effectively guide ontology engineers in this regard.

4.3.2 Experiment setting and methodology

Similarly to Chapter 3, the four agile OEMs (UPONLite, SAMOD, RapidOWL, and AgiSCOnt) are tested against the framework described in Table 9, using the experimental use case study methodology already adopted in the previous Chapter.

Also in this experiment, a sample of participants was involved. The sample was composed of 16 company employees attending the *Advanced professional Master course on Sustainable Industry 4.0* – which included disciplines like Semantic Web and general notions on OE, Artificial Intelligence, Sustainability and energy efficiency, Smart buildings, and Building Information Modelling. The participants were all male (average age: 26.1 years) and all employed in Italian SMEs in middle-management roles (with the exception of one participant, employed in a Large Enterprise in the same role); they attended a 45-hours long course on Semantic Web, with an introduction on OE (including Ontology Design Patterns). Similarly to the sample recruited in the experiment of Chapter 3, all participants had previous knowledge of Computer Science disciplines (SQL databases, basics of programming with Python, and elements of Logic and Computational thinking), and all of them attended technical and technological institutes during their high school years; two participants had also enrolled in a bachelor degree program (architecture and computer science engineering) before withdrawing after a few months.

Once again, adopting the qualitative research approach described in Chapter 3, participants were divided into three groups (A, composed of 5 members, B, composed of 6 members, and C, composed of 5 members). They were asked to develop four ontologies adopting the four different OEMs. Figure 4.4 summarizes the main phases of the experiment. The development of the ontologies took advantage of four case tests (one more than the experiment described in the previous Chapter), one for each agile OEM that participants were asked to adopt.

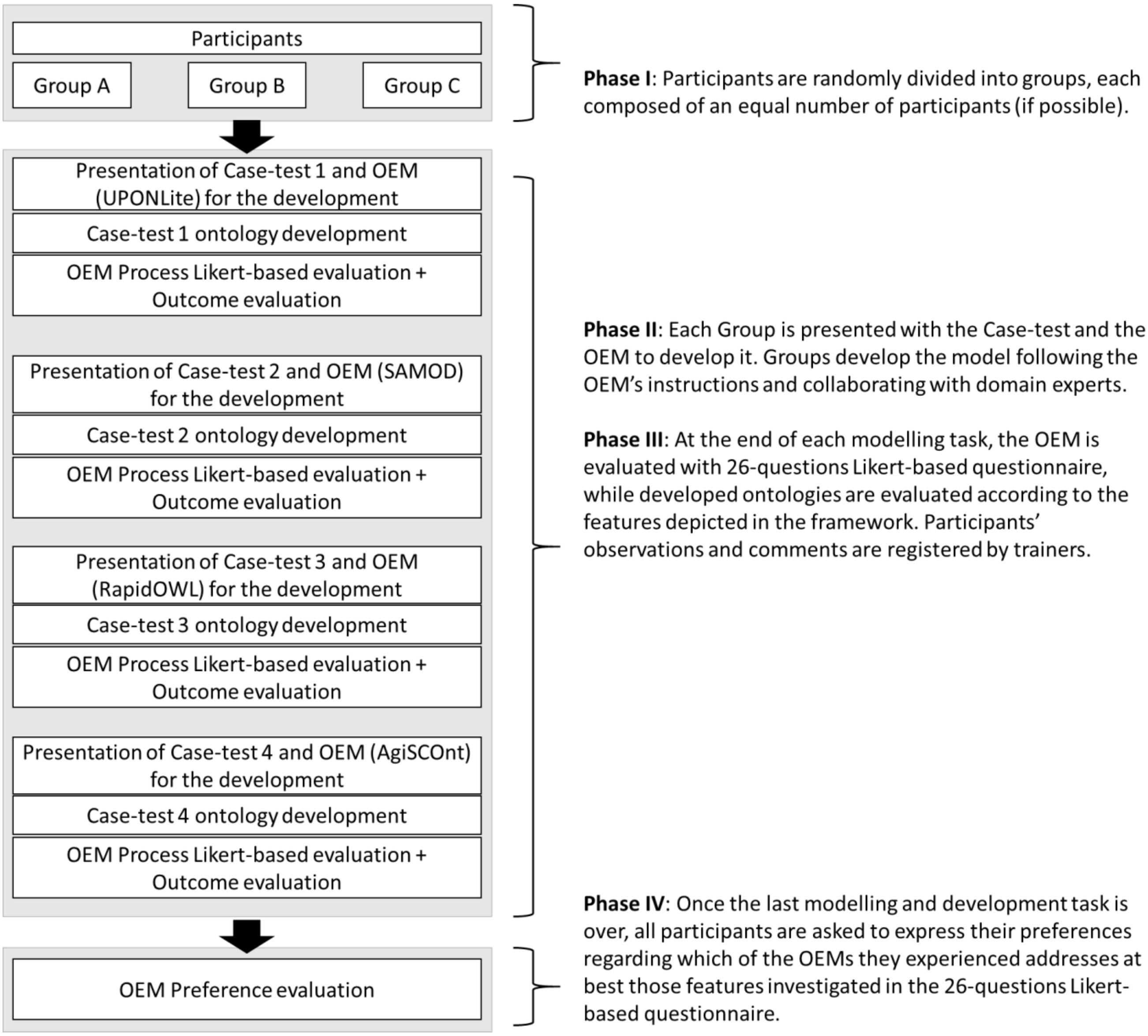

Fig. 4.4. A graphical representation of the phases composing the experiment.

Considering that in this study the number of agile OEMs to be evaluated and compared is increased by one, the case tests also are also increased. The disciplines and the topics addressed by participants during their course allowed them to have some knowledge about the domains of the test cases, which were structured with the support of domain experts in order for the case tests to reflect realistic scenarios. Three of the case tests are borrowed from the previous experiment, while the fourth is original. The case tests are summarized in Table 10.

Case-test	Domain	Case test	OEM
1	Product portfolio	An ontology for representing and managing a company's product portfolio producing custom packaging boxes. The model must be able to represent different types of products, including characteristics such as: • Raw materials; • Available size(s) for each product; • Available colors for each product; • Available shapes for each product;	UPONLite

<table>
<tr><td></td><td></td><td>Availability of finished products;
Availability of raw materials in the warehouse;
Processing time for each type of product.
The framework is designed to support B2B customers in understanding the company's product portfolio and estimated time between an order and shipment (orders for available products, custom orders requiring production).

Domain expert: orders department senior executive – local box factory, SME</td><td></td></tr>
<tr><td>2</td><td>E-commerce platform</td><td>An online clothing store scenario. From a B2C perspective, the framework must be able to:
Represent different products and their information (type of product, color, available sizes, price);
Represent customers;
Model previous orders performed by customers;
Suggest new products to customers according to their chromatic preferences, sizes of previously purchased products, and products availability.
From a backend perspective, the framework must be able to:
Model the cost for each product;
Calculate the margin for each product;
Calculate the overall margin on each order.
The framework is designed to support e-commerce managers in automating recommended products marketing activities, taking into account bot customer preferences and margin-based benefits.

Domain expert: e-commerce marketing & product manager – regional clothing franchise</td><td>SAMOD</td></tr>
<tr><td>3</td><td>Production planning</td><td>A framework to plan capacity utilization of shop floor resources able to:
Represent workers and their general information (name, surname, age, role on the shop floor);
Each worker’s hourly cost, including additional costs for hours between 8:00 PM – 6:00 AM (night shift);
Represent shop floor’s machinery, including information on their functioning (time to set up, amount of time for completing a task, required breaks);
Model working shifts (on three shifts: 6.00 AM - 2:00 PM; 2:00 PM - 10:00 PM; 10:00 PM - 6:00 AM);
The framework is designed to provide information such as:
Identify machinery’s periods of inactivity;</td><td>RapidOWL</td></tr>
</table>

		• Calculate total personnel cost for each shift and for every working day; • Support organization manager in planning working shifts (considering that a worker may work the same shift for an entire week). **Domain expert**: organizational manager – plastic molding SME	
4	Recipes recommender system	An ontology for recommending patients healthy recipes. The model must be able to: • Represent patients, their personal data, and their health conditions • Represent recipes and all the steps composing them, taking advantage of the "Recipe book" provided by the domain expert • Propose for each patient one or more recipes deemed suitable for his/her health condition The recommender system is designed with the aim of helping patients identify the most healthy recipes according to their health conditions **Domain expert**: biomedical engineer (involved in a nutrition research project); the domain expert provides a simplified version of a recipe book, containing for each recipe indications on the suitability of the dish for people characterized by specific health issues	AgiSCOnt

Table 10. The four domains, case tests, and OEMs adopted in the study.

Each group of participants was expected to develop four ontologies (for a total of twelve ontologies for all case tests), to be completed within 1 week for the whole OE process. During this time, participants had access to domain experts via online tools and in the classroom. Access to the Web was also granted to allow participants to search for existing ontologies, materials, and informal models that could be reused. For case test four, participants were provided with a simplified "recipe book" (consisting of an excerpt of five recipes in textual form, localized in the Italian language; an example of a recipe can be seen in Figure 4.5) listing recipes, their steps and the set of health issues for which each recipe could be suitable: the excerpt was prepared by the domain expert.

Fig. 4.5. An example of a recipe provided by the domain expert for case test 4, extracted from a recipe book. The upper-left box describes the steps for completing the recipe, while the bottom-left box identifies persons characterized by some health issues who could potentially benefit from the recipe.

For case test 4 with AgiSCOnt, participants were introduced during the course to OdPs and how to import ontologies using Protégé (the ontology editor adopted for this experiment); therefore, they were aware of the existence of ODPs portals describing them, as well as of the possibility to browse scientific literature to reproduce ODPs.

4.3.3 Data collection

The process features composing the evaluation framework were translated into a 26-item questionnaire (Table 11). Each item of the questionnaire could be evaluated with a Likert scale, ranging from a minimum of 1 (strongly disagree) to a maximum of 4 (strongly agree). Also, participants' comments, observations, and field notes raised during the development tasks were registered by two trainers (not belonging to the Groups of participants, therefore not directly involved in the sample taking part in the experiment). This secondary source data was then triangulated with features of the questionnaire (as shown in the following sub-section).

OEM process features	**Questionnaire item**	
Clarity and simplicity	1	The instructions provided by this methodology are clear
	2	This methodology is simple to learn and use
	3	This methodology identifies and details all the steps and activities needed to develop the ontology

	4	Every step of this methodology is clearly presented and detailed
Adaptability and flexibility	5	This methodology allows modifying the ontology at any moment of the development process
	6	This methodology can be applied to different domains of knowledge
	7	This methodology can be used to develop ontologies in domains characterized by different scales and complexity
	8	This methodology enables the personalization of steps/activities, including taking into account domain experts' feedback
Knowledge management support	9	This methodology provides support in identifying resources to be reused or re-engineered (e.g., other ontologies, taxonomies, conceptualizations)
	10	This methodology provides support in the creation of documentation
Teamwork and cooperation support	11	This methodology simplifies the cooperation between developers and domain experts
	12	This methodology requires the domain expert to have an active role in the development team
	13	This methodology eases the teamwork within the development team
Developer perceived effort	14	This methodology enables the development of an ontology in an adequate amount of time
	15	This methodology provides a substantial amount of steps to be followed
	16	The use of iterations foreseen by this methodology simplifies the development of the ontology
Developer perceived role	17	I felt engaged in the team and during the development process
	18	In my opinion, I was able to contribute to the development process using this methodology
	19	My role in the development team was always clear and explicit
Innovation support	20	Using this methodology, the development team is encouraged to adopt new ideas to achieve the scopes of the ontology.
	21	This methodology encourages developers to be creative.
	22	This methodology allows for making and fixing mistakes easily.
	23	Team members took the initiative to perform the tasks foreseen by this methodology.
	24	Using this methodology, the development team can freely make decisions at any moment of the modelling and development phases.
Operational support	25	The methodology provides me with operational support for the activities pertaining the analysis of the domain.
	26	The methodology provides me with operational support for the activities pertaining the development of the ontology.

Table 11: the list of 26 items administered to the participants to evaluate the process features (on the left) composing the evaluation framework.

The questionnaire was administered at the end of each development phase (so each participant was supposed to answer the questionnaire three times throughout the experiment), while the same version of the questionnaire, asking for each item which agile OEM the participant preferred, was administered only after participants completed Phase III (therefore after they had experienced all the methodologies). A total of 64 questionnaires were expected at the end of Phase III, and 16 preference questionnaires were expected at the end of phase IV.

With regard to the outcome evaluation, the same approach adopted for the experiment described in Chapter 3 was readopted here: trainers (researchers in the field of Semantic Web and OE) evaluated each ontology, relying on the outcome features composing the framework.

4.3.4 Results

The number of ontologies produced amounted to 12. The trainers analyzed the ontologies according to the features represented for the outcome: the results are summarized in Table 12. By the observation of the documentation and models provided by the three Groups, it is possible to observe whether participants complied with the *Reuse* practices and if they were able to produce enough *Documentation* – provided in separate files and physical materials, e.g., pieces of papers. By answering the process questionnaires, participants were also able to state the number of *Iterations* for each of the four agile OEMs (two Groups were unable to provide the number of iterations for RapidOWL).

Similarly, the observation of the ontologies allows for assessing the *Relevance of the model*s in capturing the domain at hand and the *Structural measures* they adopted. Trainers also kept track of the amount of *Time* that Groups devoted to the development of the ontologies, and – using the Pellet reasoner – they checked the *Logical consistency* of the models.

Results for the outcome clearly indicate that all the ontologies are very poor in terms of *Relevance of the model*: not all the available constructs were adopted to represent the complexity of the domains in a coherent way. As seen in the previous Chapter, this aspect also impacts the *Structural measures*, which suggest that half of the ontologies produced failed to represent all the characteristics of the domains underlying the four case tests.

No *Unsatisfiable concepts* were registered, as well as no *Logical inconsistent* ontologies were provided by participants. However, in one case (case test 1, Group A), the lack of quality in *Structural measures* led participants to develop SWRL rules, which in turn draw inferences that are not correct in the domain (e.g., by incorrectly modelling domains and range for some object properties led to the unwanted inference that some `:Raw_materials` are also `:Finished_products`, an issue participants could have realized by implementing disjunctions).

With regard to the *process* features, Table 13 summarized the answers provided by participants – registered as mean values with standard deviation, following the methodology already introduced in Chapter 3. Participants provided 64 Likert questionnaires to evaluate the agile OEMs foreseen for Phase III, plus 16 preference questionnaires for Phase IV.

Secondary data sources – i.e., the comments and observations provided by participants to trainers while they were adopting an OEM – are summarized in Table 14, where each comment is triangulated by trainers with process features to help understand the statistics of the three agile methodologies in this experiment.

4.3.5 Discussion

As the results illustrate, the developed ontologies were not entirely satisfying: half of the models provided by participants in the four case tests lacked entities to describe the complexity of the domains. In particular, this happened in Group A while using UPONLite (the ontology developed by participants did not provide any entity for developing the amount of time required for an order to be processed) and for both Groups A and B while using SAMOD (Group A did not provide any entity for calculating products' margins, while Group B provided only a few SWRL rules to suggest customers new products, limiting the recommendations to clothes' sizes and neglecting colors, previous orders, and products availability). All models provided for case test 3 were incomplete.

The development of case test 1 with UPONLite indicates that this methodology is perceived as clear: with the sole exception of one slightly incomplete model, all the ontologies provided describe in an acceptable way the domain at hand. Even though the *Structural measures* are indicative of the limited efforts conducted in developing the conceptualization (which resulted in poor *Relevance of the model* metrics, as well), there are no *Unsatisfiable concepts,* and all the ontologies are *Logical consistent*. The *Documentation* provided for these ontologies is adequate (the list of CQs and the Glossary were produced by all participants, although not always complete). In particular, it is worth noticing that the only incomplete model (Group A) is the one that has the least adequate documentation: this fact also impacted the *Relevance of the model* and *Structural measure* outcome features. Participants partially neglected the analysis of the domain, which had effects on the definition of object properties (which lacked domain and range definition); as a consequence, two SWRL rules ended up also producing undesired inferences (i.e., some individuals of the class `:Raw_materials` are also members of the class `:Finished_products`, a fact that could have also been observed by implementing disjunctions and running the DL reasoner). While developing with UPONLite, participants adopted spreadsheets to gather information from the domain expert, while some participants took notes on paper (which were later adopted to write the list of CQs). From a process perspective, this agile OEM satisfies participants for *Clarity and simplicity* and most of the features of *Adaptability and flexibility*, with scores that compete with those of SAMOD and AgiSCOnt.

Case test 2 with SAMOD resulted in two incomplete models. Participants complained about the fact that during the development phase, domain experts could not take a direct role (as prescribed by the OEM) and faced issues in identifying the test cases to develop the modelets (see Table 14). This could explain in part the paucity of Groups A and B's ontologies, which lack some of the *Structural measures* to properly represent some concepts (Group A did not model entities to calculate the overall margin on each order, while Group B missed some rules to enable customers' recommendations). The effort of defining domain, ranges, and class expressions was left to the last phase of the methodology but resulted in a very poor model from the perspective of the *Relevance of the model*. Nonetheless, all Groups investigated the existing e-commerce websites to gather some insights into the model structure (which can be included in *Reused* non-ontological *models*). From a *Documentation* perspective, all Groups provided partial deliveries, with some Groups trying to sketch the TBox of the ontology in a conceptual map; nonetheless, only one Group (C) updated the list of CQs each time a new test case was modelled, and merged into a modelet (Table 14). Each model was *Logically consistent* and lacked *Unsatisfiable concepts*. SAMOD was perceived as one of the most time-consuming agile OEMs (*Developer perceived effort*), although the amount of *Time* dedicated to the development is less than the hours used to develop with RapidOWL and AgiSCOn (Table 12). Finally, its flexibility was generally recognized by all participants (Tables 13 and 14).

The development with RapidOWL resulted in three incomplete ontologies, although all of them were *Logically consistent* and did not present *Unsatisfiable concepts*. Participants declared to have struggled in understanding "where to start" (Table 14) with the methodology, also because of the lack of *Clarity and simplicity* in using the OEM (Table 13). Two Groups were unable to identify when one iteration of the ontology started. Although the initial difficulties were noted, the participants still managed to adopt the methodology to develop the ontologies, which resulted incomplete from a *Relevance of the model* perspective – no Group described domain and range for properties, and class expressions were completely absent. The incompleteness of models did not produce any "mistake" – all SWRL rules were working correctly and were inferencing what participants expected. However, all Groups struggled with working shifts representation, while Group A and B did not model enough entities to provide a full representation of the workers (basic personal information, e.g., name, surname, age). Participants worked together with domain experts to gather knowledge, then focused on finding some ideas to properly and conveniently model the main characteristics of the domain (Table 13 *Innovation support* features, and Table 14): the development phase took the highest amount of *Time* (Table 12).

Finally, development with AgiSCOnt was able to provide three complete models. From a *Reuse* perspective, it was interesting to notice that all Groups asked the domain expert how to model health conditions, which resulted in them adopting the International Classification of Functioning, Disability and Health (ICF) [139] following the expert's suggestion. However, none of the Groups reused the existing ontology on ICF, but they decided to model the bits of the ICF they were interested in directly in the model (soft reuse). Participants were able to import and use some ODPs (namely: Group B and C imported the Sequence ODP [140], a submission to the Ontology Design Patterns Portal, to model the sequence of steps composing each recipe, and the ODP for health condition modelling with ICF, which was reused basing on an open access work that provides clear pictures of the ODP [112]. Group A only adopted the Sequence ODP, while relying on datatype properties to model different ICF codes). Participants engaged in a close discussion with the domain experts, discussing aspects of the Conceptual map and asking her opinions about the recipe book excerpt (Figure 4.5) she provided as a source of knowledge (Table 14). Although Groups reused patterns and existing sources to model their ontologies, it must be noticed that the *Relevance of the model* features domain and range were partially compiled by ODPs adopted – so the effort participants had to make was very limited. In general, the ontologies developed are still poor, even though *Structural measures* indicate a slight enhancement that allowed Groups to provide all the entities necessary to represent all the concepts and provide the recommendations. It is worth noticing that all Groups dedicated a considerable amount of *Time* for the development, the second highest if compared with other OEMs: this may, in part, be due to the fact that Groups had to browse the ODPs to understand the ones to reuse (Table 14). The ontologies are all *Logical consistent,* and no *Unsatisfiable concepts* are present. From a *Documentation* perspective, participants provided a full list of CQs, but only partial glossaries and a Conceptual map – which is "mandatory" when developing with AgiSCOnt: therefore, it was an expected output.

By taking into account each process feature, *Clarity and simplicity* shows that AgiSCOnt was particularly appreciated by participants, although UPONLite immediately follows: both OEMs provide a structured and limited set of steps with clear dependencies among them. This feature seems to make a difference in understanding the methodology (Table 13). Both UPONLite and AgiSCOnt also registered some comments connected to these process features. One Group struggled to understand how to divide the *Parthood* and *Predication* steps in UPONLite, declaring that they conducted steps from 3 to 5 all at the same time (Table 14): this fact also happened during the experiment depicted in Chapter 3. Dependencies holding among steps seem to cover a relevant role in the perception of the *Clarity* of an OEM: this might explain the very close scores gained by UPONLite and AgiSCOnt (Table 13). This might also provide an explanation for SAMOD and RapidOWL's mediocre score in these features (with participants stating to have faced issues in understanding how to start, as reported in Table 14). It is interesting to note that also in this experiment, participants underlined the same issues emerging with SAMOD and RapidOWL: the identification of the first modelet and the first practice to be tackled (among the set of RapidOWL's practices), respectively, are not clear to participants. This translates into the perception of a longer amount of *Time* (item 14) to be dedicated to understanding and using the OEMs.

Knowledge management and support, which includes the reuse of existing resources (both ontological and non-ontological) indicates AgiSCOnt as the most suitable OEM for providing support in identifying other resources (item 9) and generating documentation (item 10): AgiSCOnt was created with the specific aim of supporting the reuse of existing ontological resources, with a focus on ODPs (which were reused by all Groups), so the score gathered was not surprising for item 9; similarly, for item 10, AgiSCOnt foresees a Conceptual map as a means to have domain experts and ontology engineers to cooperate on building a shared conceptualization and eliciting knowledge. SAMOD and its iterative structure are recognized as useful for identifying other resources to be reused (item 9), while UPONLite is also appreciated for the generation of documentation (item 10). RapidOWL scored below average in this process feature – a fact that may be attributed to its practices and to the lack of dependency among them.

The results of the process feature *Adaptability and flexibility* illustrate a different situation. AgiSCOnt shares its dominant position in two features out of four. In particular, item 6 (the possibility to reuse an OEM for different domains) sees UPONLite and AgiSCOnt being almost equally appreciated by participants.

Conversely, item 5 sees SAMOD (and its iterative structure) being recognized as the OEM that better enables to modify the ontology at any moment. The personalization of activities in the OEMs (item 8) is a feature recognized in SAMOD, RapidOWL, and AgiSCOnt – with UPONLite outdistanced only for 0.06 points: these OEMs share an iterative structure, with SAMOD limiting the action of domain experts, while in UPONLite, RapidOWL and AgiSCOnt they are fully involved in the all activities. Therefore, it is not surprising that the participants recognized all OEMs with this feature of flexibility. The slight difference UPONLite holds might be motivated by the separation between its last step (6 - Development) and the other 5 steps, which require domain experts to be more involved. This fact is also underlined by the preferences participants attributed to each of these items, which – together with the very close scores registered in the whole *Adaptability and flexibility* features – indicate that all OEMs are very effective in being adaptable and flexible to developers' needs.

A similar scenario presents when analyzing the *Teamwork and cooperation* features. The less active role of domain experts penalized both SAMOD and RapidOWL (item 12), as pointed out by participants in some comments (Table 14): the two OEMs share the same scores, although not the same number of preferences. On the contrary, for the same item (12) UPONLite and AgiSCOnt registered the same score and the same amount of preferences as well. The cooperation between developers and domain experts was perceived as facilitated by AgiSCOnt, with UPONLite following very closely (item 11). According to participants, the teamwork in the development team is supported in the same way by UPONLite, RapidOWL, and AgiSCOnt, with RapidOWL being preferred. As noted during case test 3 in Phase III, RapidOWL is the agile OEM that "forces" participants to come together with solutions (Table 14).

In the feature *Developer perceived effort*, SAMOD is deemed the OEM that consists of more steps to be followed (item 15) but whose iterative structure can simplify the development of the ontology (item 16). UPONLite is the methodology that participants think can support the development of ontologies in an adequate amount of time (item 14). It is relevant to notice that AgiSCOnt did not score better than other OEMs in item 14, with participants noticing a deeper knowledge of ODPs (or, in general, of the knowledge about the existence of specific ODPs) could have made the development process faster.

Again, in *Developer perceived role*, AgiSCOnt and UPONLite present very similar scores. Although AgiSCOnt has slightly higher numbers, the participants' scores differ only for a few decimal points in all items. The amount of preferences is also very close for these two OEMs in item 17, while it is the same for item 19. It is worth noticing that item 18 indicates that participants do not have a clear preference among which of the agile OEMs fosters a better perception of personal contribution to the development activities.

With regard to the *Innovation support*, which portrays the creativity of the agile OEMs, RapidOWL surpasses the other methodologies: as also highlighted in the comments (Table 14), the absence of a clear dependency among this OEM's practices puts the creativity of participants to the test to achieve solutions (item 20), in a creative and collaborative effort to identify the best modelling solutions (item 21) without being constrained by the methodology's structure (item 24). This situation traces back to the results seen in the experiment illustrated in Chapter 3. The two methodologies that are recognized to support fixing mistakes in an easy way are SAMOD and AgiSCOnt (item 22) – a result that is similar to the experiment of the previous Chapter, with the exception of AgiSCOnt. The same situation presented in the previous experiment regarding participants taking the initiative is reproduced here (item 23): UPONLite and SAMOD gained the same score. It seems that AgiSCOnt does not emerge as a particularly creative OEM: this statement is also endorsed by some comments participants provided (Table 14), suggesting that relying on existing solutions (to be reused, like ODPs) reduces the creative efforts performed by the Groups of participants.

The *Operational support* feature – which was not included in the previous experiment – clearly indicates that AgiSCOnt is the most suitable OEM to provide practical support and suggestions to developers. Once again, this OEM is developed specifically with the aim of fostering ontology authoring, with suggestions to reuse ODPs and a Conceptual map that asks participants to sketch a TBox on paper. However, it is interesting to note that RapidOWL scored below average in this feature, in particular for the support to be provided during the development of the model (item 26), with participants commenting on the fact that for practical development, this methodology's practices are not particularly useful (Table 14). If SAMOD is as limited as RapidOWL in item 26, its structure and attention to case tests allowed it to gather a slightly higher score for item 25 – dedicated to the support in domain analysis. Similarly, UPONLite provides more practical support in domain analysis (item 25) rather than practical development support (item 26).

4.3.6 Considerations on AgiSCOnt

The results and their discussion allow the underlining of some characteristics of AgiSCOnt. Similar to UPONLite, this OEM is regarded as a simple and clear one. This fact – common to the two most "structured" methodologies – is due to the sequence of steps provided in the instructions. Nonetheless, the same findings that were retrieved in Chapter 3 experiment hold: while the users may perceive AgiSCOnt as supportive due to its sequence of steps, this fact may hinder the creative dimension – a cornerstone of agile engineering. This is also particularly relevant when considering the *Reuse* features of the outcome. Participants reused ODPs, but highlighted how the reuse of "ready-to-use" solutions might limit the efforts in creativity (Table 14). Therefore, the trade-off between OEMs' sequential structure and creativity presents itself again.

However, AgiSCOnt's aim to foster the reuse was somehow successful, as the participants adopted not only ontological solutions (the ODPs), but also non-ontological resources (ICF). It is important to underline that the participants' lack of knowledge (and expertise) with both ODPs and existent ontologies in the domain of health impacted their models: on the one hand, participants did not look for ICF ontological representations (although there exist a publicly available version of the ICF ontology on the BioPortal [141]); on the other hand, participants invested a relevant amount of time in finding out two ODPs, while others could have been potentially reused. This fact illustrates that with more expertise and know-how regarding ODPs, participants could have produced their ontologies in less time (potentially better ones). In other words, it is safe to assume that, although producing a relevant step towards reuse, AgiSCOnt becomes more and more effective with its users' growth in expertise. More acknowledged users in the field of ODPs and domain-related ontologies may translate into developing ontologies that better reuse existing knowledge and patterns.

The experiment also elicits that the Conceptual map covers a dual role. The creation of the map serves the knowledge elicitation activity with domain experts while also serving as a document – which is further used to model the TBox of the target ontology. This fact is relevant because AgiSCOnt was explicitly developed to limit knowledge elicitation efforts while eliciting any pertinent information. In fact, considering the requests included in case test 4, none of the produced ontology provided entities to model the nutrition facts of the recipes (which were provided by the domain expert in the bottom-right box of each recipe depicted in Figure 4.5). This fact indicates that participants discussed whether or not to include this information in the model and decided not to, as it was not stated in the case test 4 guidelines.

The collaborative dimension of AgiSCOnt is also well represented by the score the OEM gained in the process features *Teamwork and cooperation support*: this result is particularly important since AgiSCOnt was developed with the specific purpose of involving domain experts in every activity they could contribute.

Finally, compared to the other OEMs, AgiSCOnt is particularly appreciated for its *Operational support*. The domain analysis activities (in particular knowledge elicitation and conceptualization) are conducted in close cooperation with domain experts, who are exploited as direct sources of knowledge. Later on, the Conceptual map sketched (together with the suggestions to adopt ODPs and reuse existing models when possible) considerably supports ontologists in the practical development of the model. In fact, the Conceptual map in AgiSCOnt supports ontologists and domain experts in identifying "is-a" and "part-whole" relationships [124]. It is worth noting that this way of engineering ontologies is different from what UPONLite entails, as the latter asks domain experts to come to terms with activities (taxonomy, parthood, and predication) that they might be only partially familiar with. In AgiSCOnt, these aspects are faced through the Conceptual map and tackled by ontology engineers, who discuss with domain experts the implications of specific taxonomical, predicative and mereological settings.

4.3.7 Limitations of this study and its findings

This study presents a preliminary and empirical study to assess AgiSCOnt and other agile OEMs, and – as such – it has some limitations.

The limited number of participants taking part in the experiment and the learning environment did not enable the possibility of mixing groups after each development phase (Phase III). This would have required a longer amount of time and a significantly higher number of case tests, as well as keeping track of the changes in each participants' Group. Nonetheless, the learning environment enabled the possibility to acquire comments and notes from participants while granting them the opportunity to work in a coherent and realistic controlled environment (simulating a team in a company).

Similar to the limitations of the experiment illustrated in the experiment of Chapter 3, each case test was developed relying on a single OEM: a future research line could investigate whether the application of different OEMs may lead to different ontologies for the same case test. As one of AgiSCOnt's limitations for its use "at its best" consists in the knowledge of ODPs and domain-related model to be possibly reused, it could be investigated whether more expert ontology engineers – with a solid knowledge of patterns and where to find them – would evaluate the methodology differently.

Finally, the ontologies developed with AgiSCOnt (as well as developed with any other OEM) should be accepted by the community or researchers and stakeholders. This means that the developed ontologies should undergo a process of "peer reviewing" – partially foreseen by Step 3 of AgiSCOnt – and should be evaluated in their quality by other (expert) ontologists. This type of evaluation takes time and is out of the scope of this dissertation.

To conclude, the sample of participants – similar to the one enrolled for the experiment reported in Chapter 3 – is limited to Italian employees. It could be investigated whether different working settings might elicit different findings.

OEM outcome features		Case-test 1 with UPONLite			Case-test 2 with SAMOD			Case-test 3 with RapidOWL			Case-test 4 with AgiSCOnt		
		Group A	***Group B***	***Group C***	***Group A***	***Group B***	***Group C***	***Group A***	***Group B***	***Group C***	***Group A***	***Group B***	***Group C***
Reused models	*Non-ontological*	Website	Website	Website	Website	Website	Website	no	no	no	ICF	ICF	ICF
	Ontologies	no	no	no	no	no	no	no	no	no	ODPs (1 imported, 1 soft reuse)	ODPs (2 soft reuse)	ODPs (2 imported)
Documentation delivery	*List of Competency Questions*	partial	yes	yes	no	partial	yes	no	partial	partial	yes	yes	yes
	Glossary / Lexicon	partial	yes	yes	yes	partial	no	no	no	no	partial	partial	partial
	Conceptual map	no	no	no	no	no	partial	no	yes	no	yes	yes	yes
Iterations		2	1	3	10	7	6	2	n.d.	n.d.	3	4	3
Relevance of the model	*Domain and range defined*	partial	no	partial	partial	no	partial	no	no	no	partial	yes	partial
	Disjunctions defined	no	no	no	no	no	no	no	no	no	no	yes	yes
	Restrictions defined	no	no	yes	no	no	no	no	no	no	no	no	yes
	Unsatisfiable concepts	no	no	no	no	no	no	no	no	no	no	no	no

Structural measures	17 classes, 5 object properties, 1 datatype properties, 21 individuals, 79 axioms, 2 SWRL rules [incomplete model]	14 classes 9 object properties 4 datatype properties 33 individuals, 143 axioms 1 SWRL rules	26 classes 3 object properties 5 datatype properties 28 individuals, 147 axioms, 6 SWRL rules, 3 annotation properties	25 classes, 12 object properties, 11 datatype properties, 29 individuals, 166 axioms, 6 SWRL rules [incomplete model]	12 classes, 5 object properties, 11 datatype properties, 12 individuals, 96 axioms, 4 SWRL rules [incomplete model]	21 classes, 8 object properties, 12 datatype properties, 41 individuals, 256 axioms, 7 SWRL rules	12 classes, 3 object properties, 15 datatype properties, 29 individuals, 149 axioms, 9 SWRL rules [incomplete model]	17 classes, 5 object properties, 9 datatype properties, 19 individuals, 121 axioms, 5 SWRL rules [incomplete model]	19 classes, 7 object properties, 6 datatype properties, 33 individuals, 153 axioms, 6 SWRL rules [incomplete model]	12 classes, 8 object properties, 5 datatype properties, 42 individuals, 165 axioms, 10 SWRL rules	14 classes, 8 object properties, 4 datatype properties, 43 individuals, 153 axioms, 8 SWRL rules	5 classes, 9 object properties, 4 datatype properties, 44 individuals, 149 axioms, 8 SWRL rules
Time	9.5 hours	9.5 hours	10 hours	12.5 hours	12.5 hours	11.5 hours	14 hours	14 hours	14 hours	13 hours	13.5 hours	13.5 hours
Logical consistency	yes (several incorrect inferences)	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes

Table 12. Results on OEMs outcome features, including development with AgiSCOnt.

OEM process features	Questionnaire item		Assessment of each OEM Mean (Standard Deviation)				Preference among OEMs			
			UPONLite	*SAMOD*	*RapidOWL*	*AgiSCOnt*	*UPONLite*	*SAMOD*	*RapidOWL*	*AgiSCOnt*
Clarity and simplicity	1	The instructions provided by this methodology are clear	3.13 (0,72)	2.63 (0.50)	2.13 (0.62)	**3.38** (0.62)	**7**	0	0	**9**
	2	This methodology is simple to learn and use	3.19 (0.66)	2.44 (0.73)	2.44 (0.81)	**3.25** (0.68)	5	1	1	**9**
	3	This methodology identifies and details all the steps and activities needed to develop the ontology	3.06 (0.57)	2.88 (0.72)	1.63 (0.50)	**3.25** (0.45)	6	1	1	**8**
	4	Every step of this methodology is clearly presented and detailed	3.13 (0.50)	2.81 (0.66)	1.81 (0.66)	**3.44** (0.63)	**7**	1	0	**8**
Adaptability and flexibility	5	This methodology allows modifying the ontology at any moment of the development process	2.63 (0.50)	**3.25** (0.68)	3.19 (0.75)	3.06 (0.68)	1	**5**	**5**	**5**
	6	This methodology can be applied to different domains of knowledge	**3.31** (0.48)	3.25 (0.45)	2.81 (0.54)	**3.31** (0.60)	**6**	4	0	**6**
	7	This methodology can be used to develop ontologies in domains characterised by different scales and complexity	3.19 (0.54)	3.13 (0.50)	2.56 (0.51)	**3.38** (0.50)	5	5	0	**6**
	8	This methodology enables the personalisation of steps/activities, including taking into account domain experts' feedback	3.00 (0.37)	**3.06** (0.44)	**3.06** (0.57)	**3.06** (0.77)	3	3	**5**	**5**
Knowledge management support	9	This methodology provides support in identifying resources to be reused or re-engineered (e.g., other ontologies, taxonomies, conceptualisations)	2.94 (0.57)	3.00 (0.52)	1.81 (0.75)	**3.44** (0.51)	4	4	0	**8**
	10	This methodology provides support in the creation of documentation	3.06 (0.57)	2.75 (0.45)	1.63 (0.62)	**3.56** (0.51)	**7**	1	1	**7**
Teamwork and cooperation support	11	This methodology simplifies the cooperation between developers and domain experts	3.19 (0.40)	3.06 (0.44)	2.56 (0.63)	**3.50** (0.52)	7	0	1	**8**
	12	This methodology requires the domain expert to have an active role in the development team	**3.13** (0.72)	2.81 (0.66)	2.81 (0.75)	**3.13** (0.62)	**6**	1	3	**6**

	13	This methodology eases the teamwork within the development team	3.31 (0.48)	3.06 (0.57)	**3.38** (0.50)	3.31 (0.60)	4	3	**5**	4
Developer perceived effort	14	This methodology enables the development of an ontology in an adequate amount of time	**3.06** (0.44)	2.19 (0.54)	2.13 (0.81)	2.56 (0.51)	**7**	1	3	5
	15	This methodology provides a substantial amount of steps to be followed	2.75 (0.58)	**3.00** (0.63)	2.13 (0.81)	2.06 (0.85)	2	**8**	3	3
	16	The use of iterations foreseen by this methodology simplifies the development of the ontology	2.94 (0.57)	**3.13** (0.62)	2.38 (0.72)	2.81 (0.66)	2	**10**	2	2
Developer perceived role	17	I felt engaged in the team and during the development process	3.25 (0.45)	3.06 (0.57)	2.69 (0.60)	**3.56** (0.51)	6	2	1	**7**
	18	In my opinion, I was able to contribute to the development process using this methodology	3.31 (0.48)	3.25 (0.58)	3.00 (0.63)	**3.44** (0.51)	**4**	**4**	**4**	**4**
	19	My role in the development team was always clear and explicit	3.31 (0.48)	3.25 (0.58)	2.69 (0.48)	**3.38** (0.62)	**6**	2	2	**6**
Innovation support	20	Using this methodology, the development team is encouraged to adopt new ideas to achieve the scopes of the ontology.	3.13 (0.62)	3.25 (0.58)	**3.31** (0.48)	2.94 (0.68)	3	3	**7**	3
	21	This methodology encourages developers to be creative.	3.06 (0.44)	3.19 (0.40)	**3.25** (0.58)	3.00 (0.37)	2	6	**5**	3
	22	This methodology allows for making and fixing mistakes easily.	3.13 (0.50)	**3.25** (0.45)	3.19 (0.66)	**3.25** (0.59)	2	**6**	4	4
	23	Team members took the initiative to perform the tasks foreseen by this methodology.	**3.19** (0.40)	**3.19** (0.54)	3.06 (0.57)	3.00 (0.52)	**6**	5	2	3
	24	Using this methodology, the development team can freely take decisions at any moment of the modelling and development phases.	2.94 (0.57)	3.13 (0.62)	**3.19** (0.54)	3.13 (0.81)	3	4	**6**	3
Operational support	25	The methodology provides me with operational support for the activities pertaining the analysis of the domain.	2.69 (0.60)	2.31 (0.48)	1.75 (0.45)	**3.31** (0.48)	4	3	0	**9**
	26	The methodology provides me with operational support for the activities	2.00 (0.52)	1.94 (0.68)	1.56 (0.63)	**3.50** (0.52)	4	**2**	0	**10**

		pertaining to the development of the ontology.								

Table 13. A table summarizing the results of questionnaires administered to participants expressed as the Mean Value and Standard Deviation in brackets. For each item, the higher score is highlighted in bold. The final four columns report participants' preferences for each of the 26 items (bold indicates the maximum value).

Group	Quotations	Case-test/OEM	Process feature
A	"It is interesting that we can use any tool we want to make a conceptualization […] This can make some of the work easier and faster."	1/UPONLite	Teamwork and cooperation support
C	"It is not clear when *Parthood* and *Predication* start. We ended up doing them at the same time."	1/UPONLite	Clarity and simplicity
C	"Most of [*UPONLite*] steps require everyone to participate and share their opinion, otherwise the ontology may not be shared by everyone"	1/UPONLite	Teamwork and cooperation support
B	"This methodology makes you prepare more than half of the documentation you need."	1/UPONLite	Knowledge management and support
A	"Most of the time effort is dedicated to steps from 1 to 5."	1/UPONLite	Developer perceived effort
A	"In practice there are no suggestions on how to identify test cases and model them"	2/SAMOD	Operational support
B	"There are not enough instructions in identifying the first modelet"	2/SAMOD	Clarity and simplicity
C	"This methodology requires a lot of time […] test case identification for the modelets is not easy"	2/SAMOD	Developer perceived effort
C	"Each time you merge [*a modelet*] to the model you have to modify the whole documentation."	2/SAMOD	Knowledge management and support
B	"We would have preferred the domain expert could take part to the development activities […] he could have checked what we were doing while we were doing it"	2/SAMOD	Teamwork and cooperation support
A	"I am not sure [*SAMOD*] could be adopted in my company. It takes too much time."	2/SAMOD	Developer perceived effort
C	"It is not clear where we should start to model this case test."	3/RapidOWL	Clarity and simplicity

A	“[*RapidOWL*] does not limit you, the group must find solutions because the methodology does not give you any hint on how to find them.”	3/RapidOWL	Innovation support
B	“There is no structure, so we can do everything at any time.”	3/RapidOWL	Innovation support
C	“Without guidance you cannot understand when an iteration is over […] it is too fluid”	3/RapidOWL	Developer perceived effort
C	“When you have to develop with Protégé […] there are not guidelines at all.”	3/RapidOWL	Operational support
B	“It is not hard to follow [*AgiSCOnt*]’s steps because they are only three.”	4/AgiSCOnt	Clarity and simplicity
C	“If we had a deeper knowledge of ontology design patterns we could be faster in developing the ontology”	4/AgiSCOnt	Developer perceived effort
B	“[*AgiSCOnt*] tells you to reuse patterns […] in this way there is less room for creativity, because we are adopting an existing solution”.	4/AgiSCOnt	Innovation support
A	“When you sketch the conceptual map with the domain experts you are basically developing the ontology […] they are the key to everything with this methodology.”	4/AgiSCOnt	Teamwork and cooperation support
C	“Using ontology design patterns helps in detailing the ontology […] they [*ODPs*] are “ready for use” bits of our ontology.”	4/AgiSCOnt	Operational support
B	“With the patterns you always reuse something.”	4/AgiSCOnt	Knowledge management and support
A	“Documentation is done by asking questions to domain experts.”	4/AgiSCOnt	Knowledge management and support

Table 14. Table summarizing the comments provided by participants and their mapping to process features.

Chapter 5 – Developing ontologies with AgiSCOnt: two examples from the health industry

This Chapter introduces two novel ontology-based Decision Support Systems (DSSs) in the health field, with a particular focus on nutrition. The development of the ontologies underlying the DSSs is performed using AgiSCOnt, and the ontology engineering processes are described. The DSSs are one of the outputs of a scientific research project and are developed in collaboration with clinical personnel. Each system aims to tackle a different chronic health problem through specific clinical and nutritional recommendations, to alleviate some of the issues' symptoms deriving from exacerbation or chronic conditions. Each ontology is described separately, with particular attention dedicated to the outcomes deriving from three steps composing the agile OEM. The *Appendix* Section presents full result tables from relevant queries of the two ontologies.

5.1 Ontologies and decision support systems

Decision Support and the development of Decision Support Systems (DSSs) can be seen as an application area for Semantic Web technologies. The two areas share many similarities: they apply technologies originally developed in the context of Artificial Intelligence, they are both focused on models, and they touch upon Information Retrieval. In the first decade of the 2000s, the development of semantic-based DSSs was particularly investigated [142], with ontology-based DSSs covering the majority of this share. Also, the amount of ontology-based DSSs applied to the healthcare domain was significant. Ontologies and rules adopted in these works, and the deriving applications, can be seen as a continuation of the expert systems tradition that adopts Semantic Web standards (in particular, ontological languages) for knowledge representation, thus replacing older representation conventions or special-purpose languages.

The type of DSSs leveraging on ontologies and rules can be classified as *Knowledge-driven DSS*, developed to recommend or suggest actions to end-users [142, 143]. These systems try to perform (a part of) the actual decision-making for the end-user.

In Sections 3 and 4, two examples of ontology-based DSS (knowledge-driven DSS) are presented: their ontologies are developed with AgiSCOnt.

5.2 The HUB sPATIALS[3] research project

The DSSs described in the following Sections are one of the outcomes of the research project HUB sPATIALS[3] [144]. The project starts with recognizing that nutrition plays a pivotal role in human well-being, especially in people characterized by specific health conditions that could be exacerbated by malnutrition. Therefore, the agri-food market needs to be able to meet the demands and expectations of consumers in terms of food quality, safety, and sustainability, as well as foods capable of generating healthy effects on the body.

To achieve these goals, the HUB sPATIALS[3] project leverages research institutes, companies, and hospitals in the Lombardy Region (Italy) to develop novel nutraceutical foods, investigate their quality, and exploit digital technologies to support both patients and clinical personnel in fighting some chronic health conditions with nutrition. The DSSs presented in Sections 3 and 4 fall under the use of digital solutions in nutrition. In fact, they were developed to support patients (and their caregivers) and clinicians in understanding how nutrition can affect two chronic diseases' exacerbation and providing their target users with tailored suggestions.

5.3 Recipes for patients affected by Dysphagia

The first DSS described tackles the problem of dysphagia, a condition that causes the individual to face difficulties in swallowing foods and liquids. This condition affects more than 10% of the Italian population (more than 6 million individuals), and in particular, it can lead to further health issues related to malnutrition.

5.3.1 Background information on dysphagia

Dysphagia is defined as an objective impairment in swallowing, resulting in a delay in the transit of a liquid or solid bolus. This delay may be caused by an extension of the oropharyngeal or esophageal swallowing phases [145]. This health issue is very common in the elderly, particularly in patients older than 50 years old and those who have faced surgical interventions to the mouth or to the neck. It may also be a symptom of neurological diseases. The dysphagia can be persistent or intermittent and is the primary cause of complications such as food aspiration (and aspiration-induced pneumonia), malnutrition, increased morbidity, and mortality. Considering that dysphagia often characterizes populations that cannot completely live autonomously or are affected by neuromuscular diseases, it may result in a relevant problem, both from a health and social perspective.

From a clinical perspective, dysphagia has to be evaluated by clinical personnel (ENT doctors, otolaryngologists) using standardized methods. The evaluation may also require diagnostic imaging. The evaluation aims to identify food consistencies that are not tolerated by the patient and suggest modifications to his/her diet. It is also relevant to assess whether dysphagia occurs as a consequence of neuromuscular diseases, although the treatment and dietary modifications may not change.

In particular, patients affected by dysphagia have difficulty swallowing one or more food consistencies or liquids. This condition can cause them to have *penetration* (the passage of material into the larynx that does not pass below the vocal folds) or *aspiration* (the passage of material below the level of the vocal folds) [146]. In both cases, the patient may intervene by expelling (through cough) the materials penetrated or aspirated. However, dysphagia may also cause them not to perceive the intrusion of materials in the airways, resulting in pneumonia or asphyxiation. In patients aware of their dysphagia, the disease can cause a significant loss of weight, chest pain, regurgitation, and cardiovascular problems [147]. Another relevant consequence is malnutrition, which is a state of nutrition in which a deficiency of energy, protein, and other nutrients causes measurable adverse effects on the composition of tissues and organs. It may also derive from the partial or total absence of one or more essential nutrients [148]. Dysphagia can thus impact significantly frail people's quality of life, and malnutrition may be related to dietary changes patients (or their caregivers) produce on their own: in fact, patients may deliberately decide to reduce or avoid specific food consistencies (because they are not perceived as safe), thus limiting their diet and inadvertently contributing to weight loss and malnutrition – both major risk factors for frail patients [149]. Such dietary changes often result in unbalanced nutrients intake. Still, they may be motivated by a lack of knowledge in preparing and processing food in an appropriate way. There exist different non-invasive solutions that can be actively support patients and their caregivers in delivering a nutritional-balanced diet to people affected by dysphagia.

The DSS presented in the following subsection describes an example of a digital application dedicated to providing patients with dysphagia and their caregivers with an everyday tool to ensure a nutrition-balanced and satisfying diet.

5.3.2 Domain experts and team

For the development of this ontology, the following team members were involved:

- 1 ontologist with experience in modelling with agile OEMs
- 1 biomedical engineer with previous experience in OE
- 1 neurologist with expertise in treating dysphagia, head of research of the NEMO Lab, a clinical center in Niguarda General Hospital specialized in the design and development of technological solutions for healthcare of neuromuscular diseases
- 1 ENT doctor with specific expertise in treating dysphagia
- 1 ENT senior doctor with specific expertise in diagnosis and treating of dysphagia in elderly patients characterized by chronic and neuromuscular conditions

The team was composed of clinical personnel (with specific expertise in the domain), one member dedicated explicitly to OE and one biomedical engineer who had previous experiences with ontology-based applications and the OE process in general. The clinical personnel was able to provide anonymized test cases.

The team mainly interacted online (due to the pandemic situation characterizing Italy between 2021 and 2022) using MS PowerPoint to produce and sketch the Conceptual map, while MS Teams was adopted to conduct the unstructured interviews. Physical meetings at the NEMO Lab were also scheduled.

5.3.3 OE process with AgiSCOnt for the Dysphagia ontology

This Section focuses on the OE process with AgiSCOnt, underlining the most relevant outcomes for each of the agile OEM's steps.

From a management perspective, the activities pertaining to the development of this DSS were included in the management framework provided by the HUB sPATIALS[3] research project, which allowed four months for domain analysis, two months for development and testing, and two months for the DSS application development.

5.3.3.1 Domain analysis

This step took advantage of frequent and numerous team interactions (18 online and 3 physical meetings). In the beginning, an appropriate introduction to the problem of dysphagia and how it is clinically treated in patients was necessary to help all team members have a solid and shared vision of the domain at hand. Then the discussions focused on what the ontology should have been able to represent and deliver. The following problems emerged at the end of the debate:

- Patients characterized by dysphagia need to face consistency modifications in their diets; in elderly patients, this fact often results in malnutrition – i.e., modifications of the consistency result in patients limiting their diet because of consistencies they cannot eat.
- Patients' caregivers are often unaware of inadvertently causing weight loss or malnutrition because of limitations they introduce in the diet. This issue is particularly relevant for patients characterized by comorbidities, e.g., frailty.
- Meals and preparations may be nutritionally unbalanced, causing patients to develop diseases in the medium and long term.
- From a clinical perspective, dysphagia can be assessed and measured using a variety of clinical scales: there is no unified standard for the evaluation of dysphagia. Different clinical scales can assess different aspects of the disease. In Italy, the treatment of dysphagia is managed according to regional guidelines.
- Clinicians and multidisciplinary laboratories make significant efforts to develop guidelines and instructions for balanced and safe nutrition of patients affected by dysphagia. These efforts may result in tailored recipe books. One of these books was written by NEMO Lab clinical personnel (experts in nutrition and dysphagia) in collaboration with two chefs [150].

According to the considerations and problems identified above, the team decided that the ontology should have focused on representing an assessment of the dysphagia and suggesting, for each patient, a set of nutritionally-balanced options for their diet. The suggestions should be tailored according to each patient's specific dysphagia, i.e., according to the assessment clinical personnel conducted on the patient. The team, encouraged by the necessities highlighted by the clinical personnel, opted for an ontology-based DSS application, which patients should use autonomously during their meals. The discussion then dealt with how to assess and represent the dysphagia and what type of outputs (nutritionally-balanced options) to suggest.

Clinicians pointed out that there exist several scales adopted in clinical practice to assess dysphagia (e.g., Dysphagia Outcome and Severity Scale – DOSS [151], Penetration-Aspiration Scale – PAS [146], Functional Oral Intake Scale – FOIS [152], Test of Masticating and Swallowing Solids – TOMASS [153]). All the clinicians involved in the team adopt two scales, the PAS – for identifying food consistencies a patient can or cannot swallow safely – and DOSS – which measures the severity of the dysphagia. Leveraging on physicians' experience and expertise, a combination of the two scales can provide a solid assessment of the disease while also providing indications of those consistencies that should be avoided. DOSS investigates three factors (level of independence, level of nutrition, and level of diet modification) ranging from 1 to 7 (according to Table. 15). This scale's output consists of one integer number ranging from 1 to 7, which is comprehensive of the three aspects investigated by DOSS.

Factor	**Range**	
Level of independence	7	Normal
	6	Modified independence
	5	Distant supervision
	4	Intermittent supervision
	3	Total supervision
	2	Maximum assistance
	1	Non-per-oral nutrition
Factor	**Range**	
Level of nutrition	7	Full oral nutrition
	6	
	5	
	4	
	3	
	2	Non-oral nutrition
	1	
Factor	**Range**	
Diet modification	7	Normal consistency
	6	
	5	1 diet consistency restriction
	4	1< diet consistency restrictions ≤ 2
	3	>2 diet consistency restrictions
	2	Artificial nutrition
	1	

Table 15. A table summarizing the factor investigated by DOS clinical scale and the values indicating, for each factor, the patient's status (source [151]).

In order to identify the food consistencies that may cause issues for patients, the PAS scale proposes a model that, for each food consistency, asks the clinicians to evaluate whether or not there is penetration or aspiration of materials, whether these phenomena occur below, to, or above folds. The food consistencies being analyzed are Liquids (Li), Semi-solid (Se), Semi-liquid (SeLi), and Solid (So). They are recognized in Italy as a clinical standard in the "Terminology for foods and liquids consistencies". For each consistency, an integer score ranging from 1 (aspiration below folds with an absence of a reflexive or conscious attempt to expel bolus, also known as "Silent Aspiration") to 8 is given (no penetration, no aspiration). The output of this scale is a score for each of the four food consistencies. Figure 5.1 illustrates the PAS scale structure.

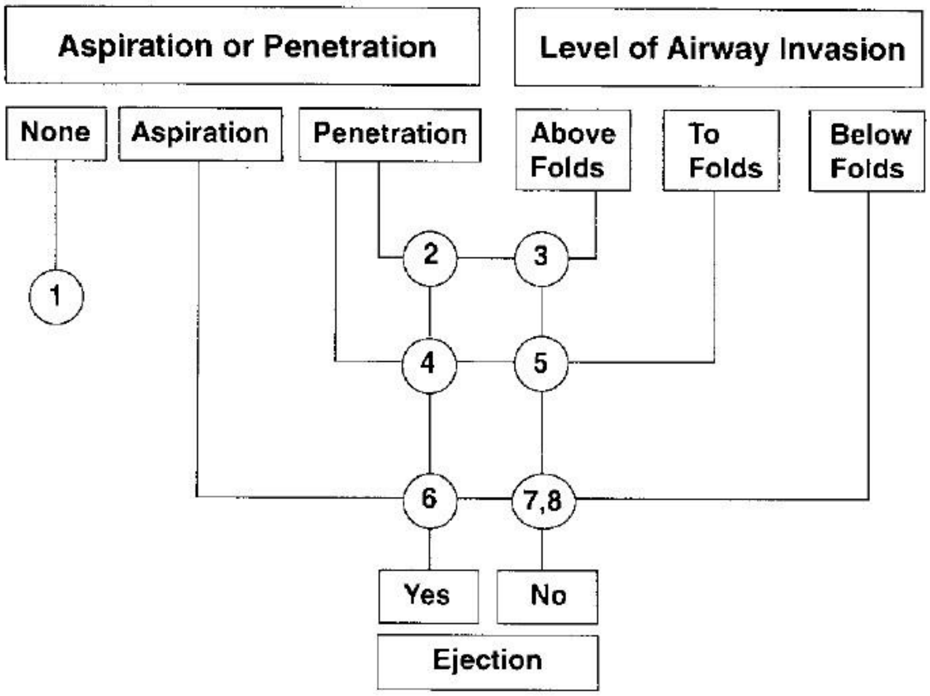

Fig. 5.1. A schematic representation of PAS scale score system (source: [146]*).*

With regard to the output to be provided, the team agreed that recipes would have served as an output for patients who can live independently (or to patients that are not completely independent but live with caregivers who are in charge of meal preparation) and have to face dysphagia and its consequences. Severe cases of dysphagia (which often results in artificial and non-oral nutrition) are related to neuromuscular diseases that cannot be managed autonomously. On the contrary, patients not characterized by severe dysphagia may benefit from support in coping with their impairment while keeping balanced and healthy nutrition. However, recipes designed for patients with dysphagia require food processing aimed at modifying their consistency by adding one or more ingredients and crushing or diluting them. Therefore, patients (and their caregivers) need to learn new food preparation techniques to prepare safe dishes. For this purpose, recipe books dedicated to patients affected by dysphagia can provide support. The clinical personnel suggested adopting a book (to which they cooperated) in which recipes are thoroughly described with nutritionally-balanced compositions [150]. Each recipe contains nutrition facts, a step-by-step textual guide, and pictures to help patients and caregivers prepare dishes.

The discussion also underlined that each recipe must be associated with one (and only one) food consistency: this is pivotal, as dishes characterized by several consistencies may cause patients problems swallowing. The team agreed it is essential to ascribe a specific consistency to each recipe. However, this may not be enough to guarantee patients' safety; therefore, clinical personnel reviewed each recipe and its consistency to ensure they respected the safety criteria (in terms of DOSS and PAS scores). They took note when exceptions were found (i.e., a recipe that has a specific consistency but needs to be restrained for some patients having certain DOSS and PAS values). This process was conducted on a spreadsheet and resulted in an annotated list of recipes, each indicating the minimum requirements (in terms of DOSS and PAS scores) for a patient to safely consume the recipe (Figure 5.2).

ID	NOME RICETTA	CONSISTENZA	DOSS	PAS
r01	BIANCOMANGIARE AI LAMPONI	SEMISOLIDA C	≥ 3	Se So ≤2
r02	CANNELLONI DI CRESPELLA CON PATATE,PORRO E SALSA POMODORO	SOLIDA D	≥6	So ≤ 2
r03	CARNE DI VITELLA IN PANZANELLA	SEMISOLIDA C	≥ 3	Se So ≤2
r04	COCKTAIL DI GAMBERI	SEMISOLIDA C	≥ 3	Se So ≤2
r05	CONIGLIO ALLA GRIGLIA CON SALSA BBQ	SEMISOLIDA C	≥ 3	Se So ≤2
r06	CONIGLIO IN PORCHETTA	SEMISOLIDA C	≥ 3	Se So ≤2
r07	CREM CARAMEL	SEMISOLIDA C	≥ 4	Se So ≤2
r08	CREMA DI CAROTE E ZUCCHINE	SEMILIQUIDA B	≥ 3	Se Li ≤ 2
r09	CREMA DI LEGUMI	SEMILIQUIDA B	≥ 3	Se Li ≤ 2
r10	DOLCE CREMOSO ALLO YOGURT	SEMILIQUIDA B	≥ 3	Se Li ≤ 2
r11	GELATINA DI MOSCATO CON FRAGOLINE DI BOSCO	SOLIDA D	≥3	So ≤ 2
r12	GIRELLO DI VITELLO CON SALSA TONNATA	SEMISOLIDA C	≥ 3	Se So ≤2

Fig. 5.2. An excerpt of the recipes attributed their consistencies and DOSS and PAS values (from right to left: recipe ID, name of the recipe, consistency, DOSS values, PAS values).

The clinicians warned that a recipe could be deemed safe for a patient if suitable under a severity perspective (DOSS scale) and, simultaneously, a consistency perspective (PAS scale). However, clinical personnel might grant individual patients some exceptions if they assess a patient may tolerate other recipes.

Concerning food consistencies, as mentioned above, clinical personnel were compact in adopting the *Terminology for foods and liquids consistencies*, which the Italian Study Group on Dysphagia developed (part of the European Study Group for Dysphagia and Globus, later renamed as European Society for Swallowing Disorders) [154]. The classification foresees six different consistencies (liquid, semi-liquid A and B, semi-solid, solid D, and E), with differences in consistencies determined by food density and processing. Since it has been used in clinical practice for many years, it was preferred to other classifications.

The outputs of the domain and analysis step with AgiSCOnt were a Conceptual map of the ontology (Figure 5.3), the list of Competency questions (CQs) (Table 16), and the Ontology Requirement Specification Document (ORSD).

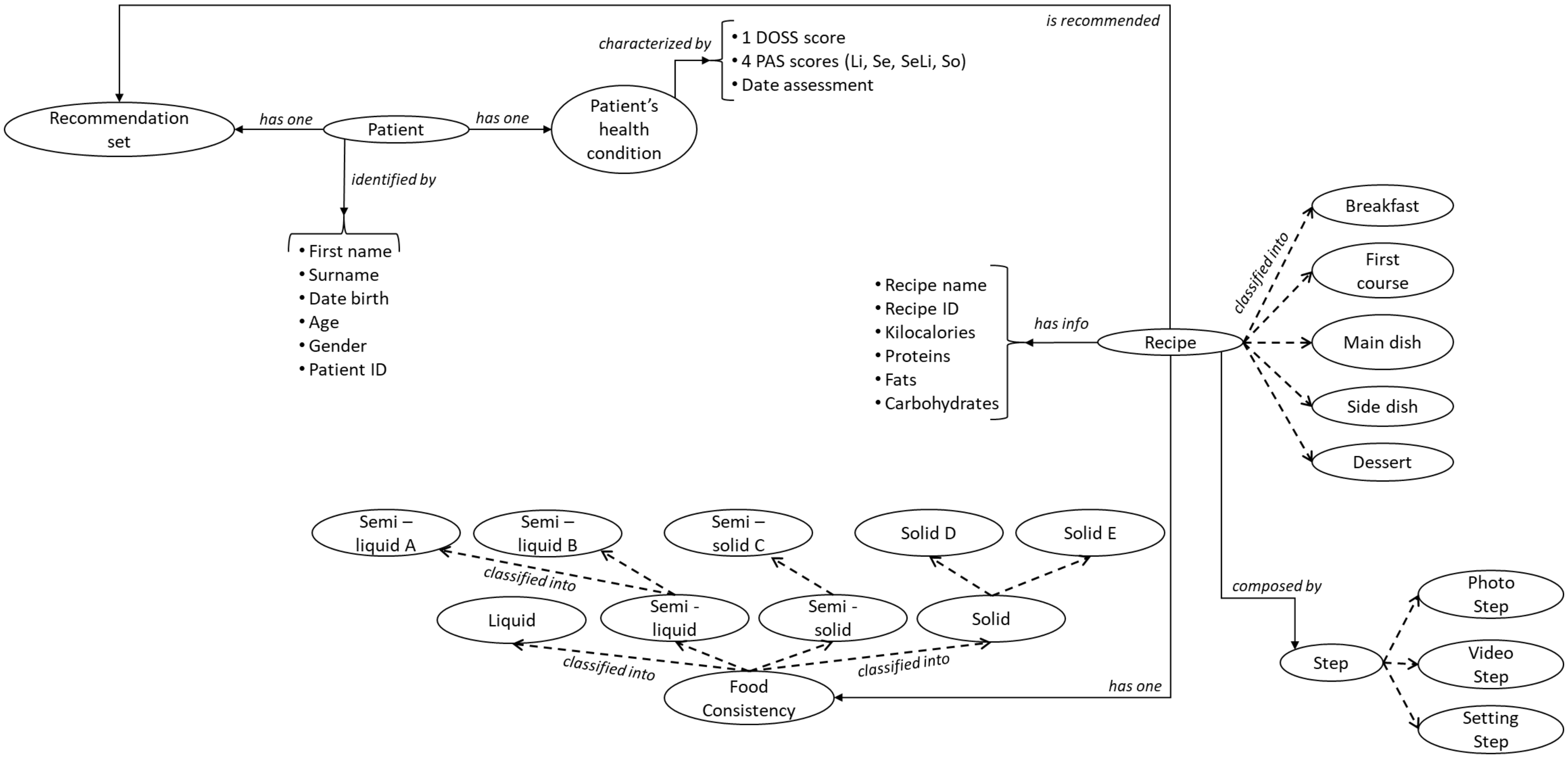

Fig. 5.3. The Conceptual map developed by the team involved in the dysphagia DSS ontology engineering process.

Competency Questions
1. What are the information characterizing the patient?
The information characterizing the patient are: the first name, the surname, the date of birth, the age, the gender (to be selected between “male” and “female”), an ID provided by clinical personnel, and a health condition.
2. How is a patient’s health condition assessed?
Patient’s health condition is assessed in a specific day and time by clinical personnel and, for the purpose of this DSS, it reports the DOSS scale score and the PAS scores (one for each food consistency, if available) evaluated by clinical personnel during visits
3. What is the recommendation provided by this ontology? *3a. How are recipes found to be recommandable by the ontology?*
The recommendation provided by this ontology foresees, for each patient, a list of recipes that are inferred to be suitable according to his/her health condition. The inference takes advantage of DOSS and PAS scores characterizing patients' health conditions; these values are compared with requirements represented for each recipe (requirements concerning the severity of the dysphagia – assessed with DOSS, and the issues with specific food consistencies – assessed with PAS). If a patient’s health condition meets all the requirements represented for a recipe, then he/she is suggested that recipe.
4. How are recipes characterized? *4a. How many consistencies does a recipe have?*
Recipes are characterized by a set of information consisting of the name of the recipe (in Italian), the recipe unique ID, the amount of kilocalories it provides per portion, the amount of proteins, carbohydrates and fats it provides per portion; each recipe is also associated to one and only one food consistency and is categorized as Breakfast recipe, First-course recipe, Main dish recipe, Side dish recipe, or Dessert recipe. Each recipe is described through a set of steps which needs to be followed in a precise order to get the recipe.
5. What are the food consistencies that recipes can have?
According to the *Terminology for foods and liquids consistencies* a food can have one among these five consistencies: Semi-liquid A, Semi-liquid B, Semi-solid C, Solid D and Solid E.
6. What are the steps composing a recipe characterized? *6a. Which are the steps composing a recipe?*
The steps composing a recipe are all provided with a textual description, which consists in the instruction a patient must follow; moreover, the Steps can be classified as: Setting Steps (which provides the list and amount of ingredients with a representative picture of all of them); Photo Steps (which provides a picture illustrating the content of the step); Video Steps (which provides a short video illustrating the content of the step).

Table 16. The list of CQs and answers for the dysphagia DSS ontology engineering process.

5.3.3.2 Development

The development phase took advantage of the outputs of the previous step. In particular, considering the need to develop rules to indicate the severity (DOSS) and consistency (PAS) scores that prevent patients from consuming specific recipes, the OWL profile selected was OWL 2 DL. The ontology was developed using the Protégé ontology editor, which also allows for illustrating most of the results in a graphical form; in this way, it was possible to involve domain experts in validating different modelling choices.

The TBox of the Dysphagia ontology (prefixed as `dis:`) is structured in 24 classes, which reproduce the concepts illustrated in the Conceptual map (Figure 5.3) – with the addition of the meta-class `owl:Thing` class. Classes representing fundamental concepts in the ontology are restricted: for example, a `dis:Patient` is any object that holds exactly 1 `dis:isInHealthCondition` relationship with an object that *is-a* `dis:Health_Condition` (reusing and ODP introduced in Chapter 4).

```
###  http://www.stiima.cnr.it/sPATIALS3-Disfagia#User
dis:User rdf:type owl:Class ;
         owl:equivalentClass [ rdf:type owl:Restriction ;
                               owl:onProperty dis:isInHealthCondition ;
                               owl:qualifiedCardinality
"1"^^xsd:nonNegativeInteger ;
                               owl:onClass dis:Health_Condition
                             ] ;
         rdfs:label "User"@en ,
                    "Utente"@it .
```

Similarly, a `dis:Recipe` is an object that `dis:isCompsoedOf` of some `dis:Steps`, and that has exactly 1 `dsi:FoodConsistency` and exactly 1 `dis:recipedID` integer value.

```
###  http://www.stiima.cnr.it/sPATIALS3-Disfagia#Recipe
dis:Recipe rdf:type owl:Class ;
           owl:equivalentClass [ rdf:type owl:Restriction ;
                                 owl:onProperty dis:isComposedOf ;
                                 owl:someValuesFrom dis:Step
                               ] ,
                               [ rdf:type owl:Restriction ;
                                 owl:onProperty dis:hasConsistency ;
                                 owl:qualifiedCardinality
"1"^^xsd:nonNegativeInteger ;
                                 owl:onClass dis:Food_Consistency
                               ] ,
                               [ rdf:type owl:Restriction ;
                                 owl:onProperty dis:recipeID ;
                                 owl:qualifiedCardinality
"1"^^xsd:nonNegativeInteger ;
                                 owl:onDataRange xsd:int
                               ] ;
           owl:disjointWith dis:User ;
           rdfs:label "Recipe"@en ,
                      "Ricetta"@it .
```

Each class is labelled with `rdfs:label` annotation to provide both English and Italian description of the concepts' names; furthermore, classes that describe consistencies adopt the annotation property `dis:description` to provide a string describing the consistency, according to the *Terminology for foods and liquids consistencies*, and some `dis:example` providing a string with examples of foods that fall into that consistency.

```
###  http://www.stiima.cnr.it/sPATIALS3-Disfagia#Semi-solid_C
dis:Semi-solid_C rdf:type owl:Class ;
```

```
                 rdfs:subClassOf dis:Semi-solid_Consistency ;
                 dis:description "Risultano di consistenza omogenea, densa,
talvolta sono derivati da alimenti frullati e setacciati, possono essere mangiati
con la forchetta, mantengono la forma nel piatto, non richiedono masticazione."@it
;
                 dis:examples "Budini, omogeneizzati di carne e derivati,
omogeneizzati di pesce e di formaggi, formaggi freschi tipo ricotta, robiola,
mousses salate e dolci, polenta, flan, semolino compatto, panna cotta, carni crude
frullate insieme a gelatine, gelatine salate e dolci, uova alla coque, acqua
gelificata"@it ;
                 rdfs:label "Semi-solid C"@en ,
                            "Semisolida C"@it .
```

The ontology adopts 13 object properties and 20 datatype properties to describe individuals. The `owl:FunctionalProperty` predicate was adopted to ensure relevant facts that the domain experts pointed out during the Domain analysis Step, for example, the fact that one recipe can have one and only one consistency:

```
###  http://www.stiima.cnr.it/sPATIALS3-Disfagia#hasConsistency
dis:hasConsistency rdf:type owl:ObjectProperty ,
                            owl:FunctionalProperty ;
                   rdfs:domain dis:Recipe ;
                   rdfs:range dis:Food_Consistency ;
                   rdfs:comment "Links a recipe to its food consistency." ;
                   rdfs:label "ha consistenza"@it .
```

Also, for object and datatype properties, annotation properties are adopted to allow human users to quickly comprehend the relationships that populate the ontology.

With regard to the ABox, it consists of 124 individuals, which are used to represent 12 patients (called `dis:User` to avoid stigmatization and reported in Appendix 1) and an equal number of `dis:Health_Condition` and `dis:Recipe_Recommendation`, 60 recipes, and some of their steps – so far, not all of the recipes have been completed represented with `dis:Step`, as it is a process that requires the preparation of videos and pictures (see further Sect. 3.3.3).

To conveniently represent the logical dependencies among the `dis:Step` composing a `dis:Recipe`, the ontology reused the Content ODP *Sequence* [140]. This ODP is fully documented and available as a "reusable building block"; it allows to represent the notion of transitive and intransitive precedence and their inverses, and it is used to represent processes. The reuse of this ODP enables the possibility of linking the `dis:Step` that composes the sequence of operations that need to be enforced for a patient to prepare a `dis:Recipe` (Figure 5.4 provides an example of the use of the ODP Sequence for a `dis:Recipe` and its `dis:Step`).

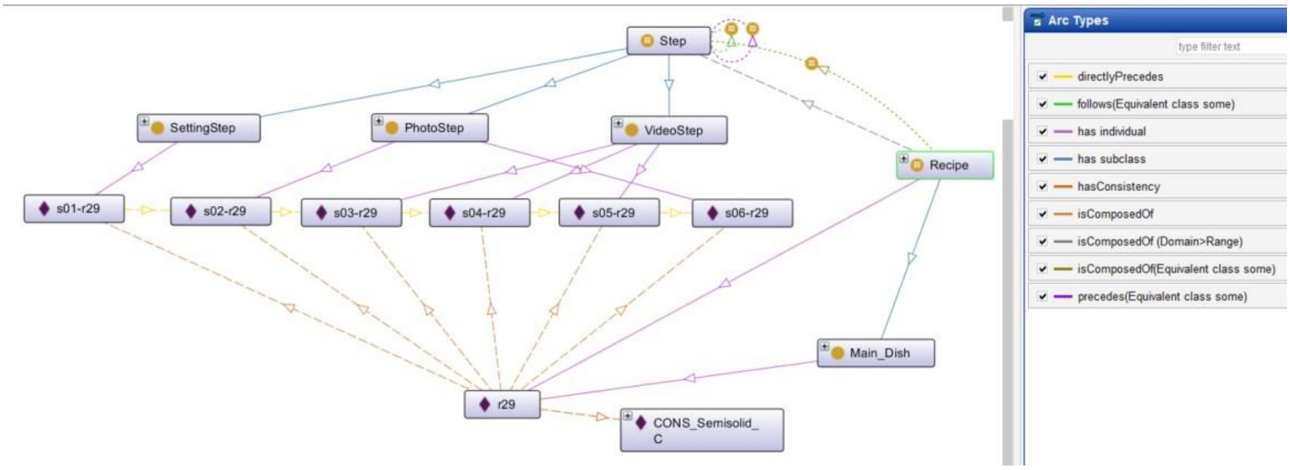

Fig. 5.4: An excerpt of the Dysphagia ontology (obtained using OntoGraf plugin for Protégé [155]*) illustrating a recipe (the individual `dis:r29`) that `dis:isComposedOf` its six dis:Step, each of which `:directlyPrecedes` its successor.*

Each `dis:Recipe` is also classified according to the course type it is supposed to belong (`dis:Breakfast`, `dis:First_Course`, `dis:Main_Dish`, `dis:Side_Dish` and `dis:Dessert`). This can help patients identify the type of dish for their daily needs, supporting them in composing their meals.

As recipes' instructions may contain key activities – especially those related to grinding solid foods to get a semi-solid or liquid consistency – that are pivotal for altering the texture of the dish, the team deemed it essential to provide short videos illustrating these delicate activities. As a consequence, the TBox of the ontology reflected this need by classifying `dis:Step` into `dis:SettingStep` (which presents the ingredients and their quantities, the preparation and cooking time), `dis:PhotoSteps` (which provides a representative picture of the step), and `dis:VideoStep` (which provides a video illustrating the step); each step is completed with a `dis:stepDescription`, providing a textual description of the instructions (in the Italian language).

The ontology needs to represent the conditions under which a patient may or may not consume a specific recipe: these conditions were stated by the clinical personnel involved in the OE process (as depicted in Figure 5.2). At the beginning of the definition of such conditions, the clinicians studied the recipe book (the source of knowledge for recipes) and investigated whether it was possible to state general rules preventing patients from consuming a recipe according to the recipe's specific consistency; however, food consistencies are *indicative*, since a dish can have a consistency because it originally had that consistency, but its consistency can be modified through food processing (for example, water has a `dis:Liquid_A` consistency and can be transformed into a `dis:Solid_E` through a freezing process or into a `dis:Semi-solid_C`, as in the case of aquagel). Therefore, for each recipe, it was necessary to state the minimum levels for a patient to be safely allowed to eat a dish. Consequently, each recipe can be safe under a severity perspective (evaluated with DOSS score) and a consistency perspective (evaluated with PAS scores). If a recipe is safe under both perspectives, it is completely safe for a patient.

These considerations are translated into the ontology using SWRL rules. Recipes are evaluated for a `dis:severitySafeRecipe` perspective:

```
User(?u), isInHealthCondition(?u, ?hc), DOSS(?hc, ?x), greaterThanOrEqual(?x, 3),
Recipe(?r), hasConsistency(?r, ?c), Semi-liquid_B(?c), isRecommended(?u, ?rec) -
> severitySafeRecipe(?rec, ?r)
```

And for a `dis:consistencySafeRecipe` perspective:

```
User(?u), isInHealthCondition(?u, ?hc), PAS_SeLi(?hc, ?x), lessThanOrEqual(?x,
2), isRecommended(?u, ?rec), Recipe(?r), hasConsistency(?r, ?c), Semi-
liquid_B(?c) -> consistencySafeRecipe(?rec, ?r)
```

As the rules illustrate, the comparison is performed by comparing the DOSS and PAS scores describing a patient's health condition. If the antecedent holds true, then also the consequence is true. These consequences are constituted of a triple in the form *`dis:user's_recommendation dis:consistencySafeRecipe dis:rXX`* or *`dis:user's_recommendation dis:severitySafeRecipe dis:rXX`*. Then, another SWRL rule allows concluding that:

```
User(?u), isRecommended(?u, ?rec), severitySafeRecipe(?rec, ?r1),
consistencySafeRecipe(?rec, ?r2), recipeID(?r1, ?id1), recipeID(?r2, ?id2),
equal(?id1, ?id2) -> safeRecipe(?rec, ?r1)
```

In other words, a recipe that is `dis:consistencySafeRecipe` and `dis:severitySafeRecipe` for a patient is a completely `dis:safeRecipe`. This ensures that the only recipes being inferred as safe by the ontology have been evaluated under both severity and consistencies perspectives through the respective scales.

SWRL rules also allow to represent some exceptions – e.g., the case of `dis:r07`, which, contrary to other recipes that have a `dis:Semi-solid_C` consistency, is indicated for patients with a DOSS score higher or equal to 4.

```
User(?u), isInHealthCondition(?u, ?hc), DOSS(?hc, ?x), greaterThanOrEqual(?x, 3), isRecommended(?u, ?rec), Recipe_Recommendation(?rec), recipeID(?r, ?id), notEqual(?id, 7), Recipe(?r), hasConsistency(?r, ?c), Semi-solid_C(?c) -> severitySafeRecipe(?rec, ?r)

User(?u), isInHealthCondition(?u, ?hc), DOSS(?hc, ?x), greaterThanOrEqual(?x, 4), isRecommended(?u, ?rec), Recipe_Recommendation(?rec), recipeID(?r, ?id), equal(?id, 7), Recipe(?r), hasConsistency(?r, ?c), Semi-solid_C(?c) -> severitySafeRecipe(?rec, ?r)
```

In this case, a patient characterized by a DOSS severity equal or greater than 3 can eat any semi-solid recipe, with the sole exception of the recipe with id equal to 7, because that recipe requires a DOSS score greater or equal than 4 – i.e., recipe 7 is recommended for patients with a slightly better dysphagia severity.

The ontology makes use of 14 SWRL rules, which can account for all the characteristics of the recipes that the clinical personnel identified during the previous Step with AgiSCOnt.

The recommendation is represented as individuals, instances of the class `dis:Recipe_Recommendation`. After reasoning with a DL reasoner, the inferences materialize as triples in the form *`dis:recommendation dis:safeRecipe dis:recipe`*; however, also `dis:consistencySafeRecipe` and `dis:severitySafeRecipe` are reported. This is important to illustrate to clinical personnel which recipes have been selected as completely safe, and which are safe only under one perspective (severity or consistency). In this way, it is possible to query the ontology to find out only the `dis:safeRecipe`, and also `dis:severitySafeRecipe` and `dis:consistencySafeRecipe` (see Sect. 3.3.3). The following Table 17 provides an excerpt of a SPARQL query to identify for each user the list of `dis:safeRecipe` inferred.

```
SELECT ?user ?id ?recipe ?recipeName WHERE {
      ?user a dis:User ;
               dis:patientID ?id ;
               dis:isRecommended ?rec .
                  ?rec dis:safeRecipe ?recipe .
                  ?recipe rdfs:label ?recipeName .
             }
```

?user	**?id**	**?recipe**	**?recipeName**
…	…	…	…
dis:Us_CDC	02	dis:r46	CREMA DI PATATE@it
dis:Us_CDC	02	dis:r02	CANNELLONI DI CRESPELLA CON PATATE,PORRO E SALSA POMODORO@it
dis:Us_CDC	02	dis:r03	CARNE DI VITELLA IN PANZANELLA@it
dis:Us_CDC	02	dis:r47	PASSATO DI CECI E FAGIOLI@it
dis:Us_CDC	02	dis:r44	PASSATO DI VERDURA SENZA POMODORO@it
dis:Us_CDC	02	dis:r45	PASSATO DI VERDURA@it
dis:Us_CDC	02	dis:r01	BIANCOMANGIARE AI LAMPONI@it
dis:Us_CDC	02	dis:r42	FRULLATO DI BANANA E BISCOTTI@it

dis:Us_CDC	02	dis:r43	SEMOLINO AL LIMONE@it
dis:Us_CDC	02	dis:r08	CREMA DI CAROTE E ZUCCHINE@it
dis:Us_CDC	02	dis:r09	CREMA DI LEGUMI@it
dis:Us_CDC	02	dis:r06	CONIGLIO IN PORCHETTA@it
dis:Us_CDC	02	dis:r07	CREM CARAMEL@it
dis:Us_CDC	02	dis:r51	ZUPPA DI CAROTE E CECI@it
dis:Us_CDC	02	dis:r52	MOUSSE DI PROSCIUTTO@it
dis:Us_CDC	02	dis:r50	PAPPA CON POMODORO@it
dis:Us_FM	04	dis:r15	GNOCCHI DI SEMOLINO VERDE, ASPARAGO CON BURRO E SALVIA@it
dis:Us_FM	04	dis:r59	CUPOLA DI YOGURT ALLA PESCA@it
dis:Us_FM	04	dis:r16	LASAGNE ALLE VERDURE@it
dis:Us_FM	04	dis:r57	CREMA PASTICCERA@it
dis:Us_FM	04	dis:r13	GNOCCHI DI PATATE,VERZA E PECORINO@it
dis:Us_FM	04	dis:r58	CREMA PASTICCERA VELOCE@it
dis:Us_FM	04	dis:r14	GNOCCHI DI RICOTTA "AIO OIO E BOTTARGA"@it
dis:Us_FM	04	dis:r55	MOUSSE DI ASPARAGI E FORMAGGIO@it
dis:Us_FM	04	dis:r11	GELATINA DI MOSCATO CON FRAGOLINE DI BOSCO@it
dis:Us_FM	04	dis:r56	PURÈ DI FAVE CON SCAROLA E PECORINO@it
…	…	…	…

Table 17. A fragment of the results provided by a SPARQL query aimed at retrieving all patients and the list of recipes inferred to be completely safe (both from consistency and severity perspective).

As the adoption of use cases and tests is pivotal in AgiSCOnt to verify and further modify the target ontology, the developed model was tested with the 12 patients (and their health conditions) provided by the clinical personnel (Appendix 1). To ensure patient anonymity, the datatype properties developed to identify patients (i.e., `dis:FirstName`, `dis:Surname`, `dis:Age`, `dis:dateOfBirth`) were not compiled. Users were identified exclusively by the `dis:patientID`, while the name attributed to `owl:Individuals` representing them was provided by clinical personnel. Not all health conditions were complete (some of them lacked one or more PAS scores), while three patients were out of the scope of the ontology – as a high severity characterizes their health condition, therefore they rely to non-oral nutrition: for these patients, the ontology was expected to retrieve no inferred data. The ontology was then queried to identify the `dis:safeRecipe` with SPARQL:

```
SELECT ?user ?id ?recipe ?recipeName ?cons WHERE {
      ?user a dis:User ;
               dis:patientID ?id ;
               dis:isRecommended ?rec .
                  ?rec dis:safeRecipe ?recipe .
                  ?recipe rdfs:label ?recipeName ;
                            dis:hasConsistency ?cons .
              }
```

and the results were analyzed by clinical personnel to asses whether all results were in line with the expectations and the clinical recommendations. As mentioned, two patients (ID 05 and ID 12) were linked to a recommendation individual that did not provide any inference. Similarly, a third patient (ID 3) is also characterized by a severe health condition, which also lacks data regarding three consistencies (PAS scores); thus, the few inferences characterizing his/her diet were only related to `dis:consistencySafeRecipe` – but were disregarded as the patient is fed via non-oral nutrition, as pointed out also by the lack of `dis:safeRecipe` inferences. Clinical personnel confirmed that these three patients are characterized by a level of severity that requires non-oral nutrition. Appendix 1 illustrates the full results of the query. The ontology was tested using the snapSPARQL plugin for Protégé [156] and the Stardog Enterprise RDF triple-store (with the SL reasoning type, as it supports both OWL 2 DL and SWRL) [157].

5.3.3.3 Use and maintenance

After the first round of tests with the 12 patients, clinical personnel asked for a slight modification to the ontology: the possibility to manually add one or more recipes to the list of recipes inferred. This would reflect the possibility for the clinicians to intervene in patients' diet, especially in those cases in which a recipe is not deemed as `dis:safeRecipe` because of a single score (as in the case of recipe `dis:07`, which differs from other semi-solid dished because it is recommended for patients with a DOS score greater than or equal to 4). As a consequence, the `dis:prescribedRecipe` object property was added to enable clinical personnel to indicate which recipes should be added in addition to those that are inferred.

As mentioned in previous Sections, the ontology serves as a base for an application that is expected to be used by patients daily to help them cope with the issues caused by dysphagia while maintaining healthy and satisfying nutrition. The application, connected to the Stardog triple store, will provide users only those recipes that are inferred to be `dis:safeRecipe` and dis:prescribedRecipe. However, before deploying and testing the application with patients, a preliminary test in the NEMO Lab is scheduled to assess the system's usability and willingness to adopt the technology on a daily basis.

Moreover, it is also necessary to increase the number of patients currently represented in the knowledge base and to extend the validation to even more experts in the dysphagia field: this would enable the possibility to receive feedback from other domain experts to further detail and extend the representation of the issues related to dysphagia and, eventually, increase the number of recipes to be provided to patients.

While the application development is an ongoing activity (it requires pictures and videos for recipes to be recorded), the revised prototype of the ontology is solid.

5.3.4 Discussion

The ontology presented in the previous Section, developed with AgiSCOnt, managed to reach the purpose of representing patients characterized by dysphagia – according to the clinical standards adopted in Italy – and to propose to these patients a set of recipe options, inferred to be adequate for their particular health condition.

The test conducted with the 12 patients and their health conditions provided by clinical personnel underlined that the ontology was able to suggest recipes to patients correctly. In some cases, it was impossible to infer recipes that were found to be safe under both a consistency and a severity perspective because patients were not eligible for the system (their severity was such that no recipe could be adequate for them). This fact is relevant to mention as it indicates that the knowledge modelled in the ontology does not provide undesired or unforeseen inferences, which may result in incorrect (or harmful) patient suggestions. The full list of inferred recipes for each eligible patient is presented in Appendix 2.

It is also worth mentioning that with a SPARQL query, it is possible to access the complete list of `dis:severitySafeRecipe`, `dis:consistencySafeRecipe`, and `dis:safeRecipe` that were inferred. This is important for clinical personnel that might want to reuse the ontology (and its deriving application) since it enables them to trace back the reasons why some recipes are inferred as safe, and some are not. In fact, by analyzing the results provided by the following query:

```
SELECT distinct ?user ?id ?prop ?recipe ?label WHERE {
      ?user rdf:type dis:User ;
            dis:patientID ?id ;
            dis:isRecommended ?recom .
      ?recom rdf:type dis:Recipe_Recommendation .
      ?recipe a dis:Recipe ;
            rdfs:label ?label .
      ?prop a owl:ObjectProperty .
```

```
    ?recom ?prop ?recipe .

}
ORDER BY ?id ?prop
```

it is possible to observe the full list of inferred recipes. This also includes the possibility to observe those inferences that did not originate any `dis:safeRecipe` inferences – such as the case of the patient with ID equal to 3, whose health condition and data prevent him/her from having any safe recipe at all, but only a few `dis:severitySafeRecipes`. Table 18 presents an excerpt of the results of the query described above.

?user	?id	?prop	?recipe	?label
…	…	…	…	…
dis:Us_CDB	1	dis:consistencySafeRecipe	dis:r52	MOUSSE DI PROSCIUTTO@it
dis:Us_CDB	1	dis:consistencySafeRecipe	dis:r50	PAPPA CON POMODORO@it
…	…	…	…	…
dis:Us_CDB	1	dis:safeRecipe	dis:r15	GNOCCHI DI SEMOLINO VERDE, ASPARAGO CON BURRO E SALVIA@it
dis:Us_CDB	1	dis:safeRecipe	dis:r59	CUPOLA DI YOGURT ALLA PESCA@it
dis:Us_CDB	1	dis:safeRecipe	dis:r16	LASAGNE ALLE VERDURE@it
dis:Us_CDB	1	dis:safeRecipe	dis:r57	CREMA PASTICCERA@it
dis:Us_CDB	1	dis:safeRecipe	dis:r13	GNOCCHI DI PATATE,VERZA E PECORINO@it
…	…	…	…	…
dis:Us_CO-LO	3	dis:consistencySafeRecipe	dis:r55	MOUSSE DI ASPARAGI E FORMAGGIO@it
dis:Us_CO-LO	3	dis:consistencySafeRecipe	dis:r12	GIRELLO DI VITELLO CON SALSA TONNATA@it
dis:Us_CO-LO	3	dis:consistencySafeRecipe	dis:r53	MOUSSE DI TONNO E MELANZANE@it
dis:Us_CO-LO	3	dis:consistencySafeRecipe	dis:r54	MOUSSE DI GORGONZOLA@it
dis:Us_CO-LO	3	dis:consistencySafeRecipe	dis:r24	PANCOTTO "DEL PRETE"@it
dis:Us_CO-LO	3	dis:consistencySafeRecipe	dis:r25	PANNA COTTA AL CARAMELLO@it
dis:Us_CO-LO	3	dis:consistencySafeRecipe	dis:r21	MERENDA IN CAMPAGNA CON FAVA E PECORINO@it
…	…	…	…	…
dis:Us_CO-LO	3	dis:consistencySafeRecipe	dis:r52	MOUSSE DI PROSCIUTTO@it
dis:Us_CO-LO	3	dis:consistencySafeRecipe	dis:r50	PAPPA CON POMODORO@it
dis:Us_FM	4	dis:consistencySafeRecipe	dis:r15	GNOCCHI DI SEMOLINO VERDE, ASPARAGO CON BURRO E SALVIA@it
dis:Us_FM	4	dis:consistencySafeRecipe	dis:r59	CUPOLA DI YOGURT ALLA PESCA@it
dis:Us_FM	4	dis:consistencySafeRecipe	dis:r16	LASAGNE ALLE VERDURE@it
…	…	…	…	…

Table 18. An excerpt of the results provided by the query that retrieves any inferred recipe. This query enables the possibility to observe patients (e.g., `dis:US_CO-LO`) who only have `dis:severitySafeRecipe` or `dis:consistencySafeRecipe` inferred, but no `dis:safeRecipes`.

With the possibility of intervening on inferred recipes (through the `dis:prescribedRecipe` relationship), clinical personnel retains their central role in defining a patient's diet. Such possibility brought the clinical personnel involved in the project to evaluate the possibility of having a "clinician application", i.e., a version of the ontology-based application that ENT doctors and clinicians treating dysphagia may adopt to suggest patients tailored recipes, helping them to cope with the issues caused by their condition.

The development with AgiSCOnt through the shared Conceptual map brought the team to focus on those aspects that were strictly related to the clinical assessment of the dysphagia and the outcomes they were expecting from the ontology. The cooperation was pivotal for getting to a shared conceptualization of the domain, following the general principle of focusing only on the relevant information that would have been used from a clinical perspective and from an application perspective. The development resulted in a simple and efficient model enriched with annotation properties to foster human readability of the ontology. With 12 patients and 12 health conditions (and 10 recipes modelled with their `dis:Steps`), the reasoning process conducted with the Pellet reasoner [158] was completed correctly within 6 seconds (5976 ms). Further tests with a larger ABox will be conducted with the Stardog triple-store to assess the scalability of the ontology. In particular, considering the limited amount of `owl:Individuals` represented in this ontology, all the instances representing patients are developed in the same ABox, while (for safety and privacy reasons, as well as for reasoning speed) it might be necessary to restructure the ontology into one TBox and as much Aboxes as the patients to be represented are.

5.4 Clinical nutritional recommendations for COPD patients

The second ontology presented tackles Chronic obstructive pulmonary disease (COPD), a chronic condition that affects the respiratory system. It is reported that in Italy, COPD affects 3.5 million people (5.6% of the adult population, i.e., 40 years old or older), and it is responsible for 55% of deaths from respiratory diseases.

5.4.1 Background information on COPD

COPD is a chronic disease characterized by the inflammation of the airways and lungs. The disease results in making it harder to breathe and is characterized by one or more of chronic emphysema (which affects the lungs' alveoli by damaging their walls, reducing the amount of oxygen that a person can absorb), bronchitis (which affects cilia, causing mucus creation and respiratory difficulties), and refractory asthma (which may also be nonreversible). The causes and risk factors for COPD consist of long-term exposure to substances that can cause lung irritation (among which: tobacco smoke, occupational dust, vapors and fumes, and indoor and outdoor air pollutants) [159]. However, other risk factors – e.g., ageing and pulmonary infections – may contribute to the insurgence of COPD.

COPD manifests with cough – chronic and persistent wheezing while breathing, shortness of breath (exacerbated during physical activity), frequent flu or colds, weight loss (often involuntary), and a sense of energy loss. In the USA, COPD is the third cause of death (3.3 million deaths, affecting patients with an average age of 42.5 years), while in European countries, up to 10% of adults aged 40 or more years old are affected by COPD (with a prevalence in men, rather than in women) [160].

Pneumonologists are the doctors that treat COPD in its various stages, with medications (bronchodilators, inhalers, steroids) and different therapies (oxygen therapy and pulmonary rehabilitation programs). The purpose of COPD treatment is to relieve the symptoms and slow the progression of the disease, preventing complications. When symptoms become stronger – even while the patient is undergoing treatment – and are sustained, the COPD undergoes a phase of exacerbation. These phases may require modifying the quantity and type of medications up to having the patient hospitalized because of rapidly deteriorating conditions [161].

COPD also affects nutrition: appetite and eating can be heavily impacted in COPD patients, who can undergo weight loss, pain, and digestive problems [162]. Nutritional depletion can contribute significantly to negative impacts on the lungs and their functions, as well as aggravating physical condition in elderlies; also, nutrition and an unhealthy diet can accelerate the decline of COPD patients [163] – as a consequence, diet is a modifiable risk factor for COPD and has a role in its prevention and treatment [164]. However, nutrition and dietary patterns play a central role in treating COPD. For example, increasing the intake of fresh fruits and vegetables (for prolonged periods of time) may reflect positively on systemic inflammation and physical functions [165].

In patients characterized by COPD's most severe consequences, such as sarcopenia – a gradual loss of strength and functions due to muscle loss – and cachexia – a syndrome preventing muscles, fat, and liver from being insulin-resistant, with the consequent inability to absorb glucose –, nutritional supplementation of essential amino acids and BCAAs (branched-chain amino acids), and of vitamins and minerals may reduce the risk of COPD exacerbations [166]. In general, a diet balanced for re-gaining weight or losing excessive weight and characterized by less simple carbohydrates and sugars and a specific amount of proteins (which can vary according to the severity of COPD) is advised to help patients in reducing the risk of exacerbation and in maintaining (or acquiring) adequate nutrition.

However, nutrition is not always taken properly into account in clinical practice, as pneumologists are not required to be trained in clinical nutrition. Also, patients may not be aware of the role played by nutrition in their condition. Nonetheless, COPD should be considered a systemic disease with extra-pulmonary manifestations, and a tailored diet can significantly contribute to increasing a patient's quality of life and limit the severe symptoms related to exacerbations.

Therefore, the DSS presented in the following subsection tackles the issue of identifying the correct daily amount of nutrients for COPD patients, by taking into account their health conditions and the specific needs it entails. Leveraging on the experience of domain experts, the ontology is primarily dedicated to pneumonologists, to help them in offering patients a tailored diet to avoid exacerbations.

5.4.2 Domain experts and team

The development of this ontology takes advantage of the following team:

- 1 ontologist with experience in modelling with agile OEMs
- 1 biomedical engineer with previous experience in OE and knowledge of COPD
- 2 senior Dieticians with clinical experience in COPD patients
- 1 pneumologist

The team was composed of clinical personnel from Universities (dieticians) and a research and cure center (IRCCS) with specialists in COPD (pneumologist) and a long experience in treating such patients.

The team interacted mostly online (due to the pandemic situation) using MS Teams for meetings, while MS PowerPoint was adopted to support the collaborative process of drafting the Conceptual Map. Physical meetings were not possible throughout 2021 and the first half of 2022, as the clinical personnel involved in this OE process was also involved in other pandemic-related activities.

5.4.3 OE process with AgiSCOnt for COPD & Nutrition ontology

This Section focuses on the OE process with AgiSCOnt for the COPD & Nutrition ontology (prefixed as `copd`), underlining the most relevant outcomes for each of the agile OEM's steps.

From a management perspective, the activities pertaining to the development of this DSS were included in the management framework provided by the HUB sPATIALS[3] research project, which allowed six months for domain analysis, four months for development and testing, and two months for the DSS application development.

5.4.3.1 Domain analysis

This step of AgiSCOnt relied on several team interactions (25 online meetings). The clinical experts and the biomedical engineer provided a comprehensive introduction to COPD, how the disease reflects on patients' quality of life and how it is treated. They also delved into the role nutrition and diet play in tackling this disease, with examples from literature and clinical trials in which the clinical personnel was involved. The discussion was then oriented to identify some issues that the ontology was expected to answer. Therefore, the following problems emerged at the end of the debate:

- Patients characterized by COPD can benefit from tailored nutrition to avoid exacerbation; however, in clinical practice, nutritional and dietetic aspects are often disregarded – mainly because pneumologists (the first line against COPD) do not have complete knowledge of the effects of nutrition in COPD patients.
- COPD, especially in elderlies, is often characterized by a situation of sarcopenia and cachexia. Both situations need to be faced with specific nutritional advice and supplements.
- COPD, in particular for severe cases, often results in weight loss – which is a situation that may lead to malnutrition, sarcopenia, and cachexia and therefore needs to be tackled promptly.
- There are several guidelines used in clinical practice to tackle nutritional problems: these guidelines can be adopted to provide an answer for patients affected by COPD. It is, therefore, relevant to assess the nutrition risk of patients to provide better customizations.
- COPD is characterized by four stages: mild, moderate, severe, and very severe. The results from the spirometry test allow to classify a patient's health condition into one of these stages [167].
- There exist several indicators of COPD, as well as indications of the nutritional status of the patient: this information significantly contributes to diet definition and customization.

According to the considerations and problems identified above, the team decided that the ontology should have focused on representing the patients' health condition and the stage of their COPD. The purpose of the ontology is to illustrate, for each patient, a tailored percentage of macro-nutrients they are advised to assume on a daily basis to avoid exacerbation. Considering the necessity of supporting pneumologists in suggesting their patients an adequate and tailored diet, the team opted for an ontology-based DSS to support clinicians in their activity: the output of the system should consist in tailored nutritional guidelines for each patient, which clinicians can use to explain to patients what to eat and why.

The representation of COPD is based on GOLD classification [167], which leverages on the amount of Forced Expiratory Volume in the 1st second (FEV1): this indicator is the volume of air (expressed in liters) exhaled in the first second during forced exhalation after maximal inspiration. CODP stages are defined as illustrated in Table 19.

COPD Stage	FEV1 (%)
Mild	≥ 80
Moderate	50 ≤ FEV1 < 80
Severe	30 ≤ FEV1 < 50
Very severe	< 30

Table 19. A table illustrating the cut-offs identifying the COPD stages based on the FEV1 values.

Also, the evaluation of the risk related to nutrition that a patient faces should be assessed using a standard. Clinical personnel suggested the Nutritional Risk Index Profile (NRI) [168], which is used in clinical practice to assess the level of malnutrition – in particular for protein energy malnutrition. The assessment of this index is based on an equation:

$$NRI = (1.519 \times sereum\ albumin) + 41.7 \times \left(\frac{current\ weight}{usual\ weight}\right)$$

This equation implies that the levels of serum albumin (expressed in g/L), the weight of the patients at the moment of the assessment (current weight, expressed in Kg), and the weight the patient had six months before the assessment (or the baseline weight, expressed in Kg). These clinical values can easily be obtained during the clinical evaluation of the patient and blood tests. The result of the equation is then compared to the cut-offs reported in Table 20, thus identifying the nutritional risk profile of the patient.

NRI	Index
Absence of risk	> 100
Mild risk	97.5 ≤ NRI ≤ 100
Moderate risk	83.5 ≤ NRI < 97.5
Severe risk	< 83.5

Table 20. A table illustrating the cut-offs identifying the four levels of nutritional risk based on the NRI.

Clinical personnel underlined that for COPD patients, according to both literature and clinical practice, it is essential to be able to provide an adequate daily caloric intake (in particular for sarcopenic and cachectic patients). It is also essential to understand whether a patient is a state of underweight or overweight, which heavily influences the amount of calories. Therefore, they proposed to rely on an anthropometric phenotype classification based on Body Mass Index (BMI) values – reported in Table 21.

Nutritional status	**BMI**
Underweight	< 18.5
Normal weight	18.5 ≤ BMI ≤ 24.9
Overweight (pre-obesity)	25.0 ≤ BMI < 29.9
Obesity degree I	30.0 ≤ BMI < 34.9
Obesity degree II	35.0 ≤ BMI < 39.9
Obesity degree III	≥ 40

Table 21: World Health Organization (WHO) cut-offs for nutritional status categories based on BMI values [169].

In a similar way, the diagnosis of sarcopenia is based on the analysis of specific patient's values. The first one is the Appendicular Skeletal Muscular Mass (ASMM), which is calculated according to:

$$ASMM = -3.964 + \left(0.227 \times \left(\frac{height^2}{Resistance}\right)\right) + (0.095 \times current\ weight) + (1.384 \times gender) + (0.064 \times Reactance)$$

Together with other indicators that contribute to defining the sarcopenic condition of a patient (the hand grip and gait speed), it is possible to classify a patient according to Table 22 [170].

Criteria	**Cut-off for male**	**Cut-off for female**
a) Low muscular strenght – Hand grip	< 27 Kg	< 16 Kg
b) Low muscular quantity	ASMM < 20 Kg ASMM / height^2 < 7 Kg/m^2	ASMM < 15 Kg ASMM / height^2 < 5.5 Kg/m^2
c) Poor physical performance	≤ 0.8 m/s	

Table 22. Criteria for the classification of sarcopenia in patients.

If criterion a) holds, then the patient is *Probable sarcopenic*; if criteria a) and b) hold, then the patient is *Diagnosed sarcopenic*; if the three criteria hold, then the patient is *Severe sarcopenic*.

A similar approach was adopted to identify whether a patient is *Cachectic*: if a patient's polymerase chain reaction (PCR) is > 10, the level of Iron transport is < 150, the level of Albuminemia is < 3.5, and the patient is *Sarcopenic*, then he/she is also *Cachectic* [171]. As such, it is safe to infer that cachexia is a particular case of Sarcopenia.

Considering the parameters and indicators involved in the correct identification of a patient's condition, the discussion among clinical personnel underlined the necessity of acquiring specific patient data to ensure the possibility of identifying correct amounts of macro-nutrients that should characterize the diet of a COPD patient. These data are usually acquired during pneumologic visits and tests (such as spirometry and blood tests):

- General patient data, such as age, gender, height (in meters), current weight and usual weight
- FEV1 and Partial pressure of Carbon dioxide (PaCO2) – the latter being an indicator useful to determine nutrients' amount in a patient's diet
- Resistance and Reactance
- Iron transport, albuminemia, polymerase chain reaction (PCR)
- Hang grip and gait speed

The dieticians then worked on a table that, taking into account the anthropometric phenotype of a patient, his/her sarcopenic and cachectic conditions, the gender, and the stage of the COPS he/she is affected by, provides amount of macro-nutrients to support pneumologists in providing patients with a balanced and tailored diet (Table 23). The table was redacted taking into account clinical practice and literature findings, and identifies the minimum amounts (in percentage) of fats and carbohydrates for each phenotype; it also provides instructions regarding how to calculate the amount of proteins (in grams, later converted in a percentage). The basal metabolic rate is calculated according to Mifflin or Benedict-Harris equations (different equations for males and females) [172], then it is corrected according to a correction factor that is given to each phenotype and the presence of sarcopenia and cachexia. For each phenotype, disregarding the presence of sarcopenia and cachexia, the indication of dividing the corrected caloric intake into 5 or 6 meals (breakfast, lunch, dinner, and two snacks) is given.

Therefore, the team decided that the output the ontology should give consists of a set of recommendations that illustrate, for each patient and on a daily basis:

- The basal metabolic rate and the corrected caloric intake amount
- The minimum and maximum shares of carbohydrates, fats, fibers
- The maximum percentage of saturated fats and sugar allowed in the patient's diet
- The share of proteins recommended
- The amount of cholesterol and sodium (salt) allowed
- Whether the patient is in need of BCAA supplementation or other energy-protein supplements

AgiSCOnt's outputs for the Domain analysis phase consisted of the Conceptual map reported in Figure 5.5 and a list of CQs (Table 24).

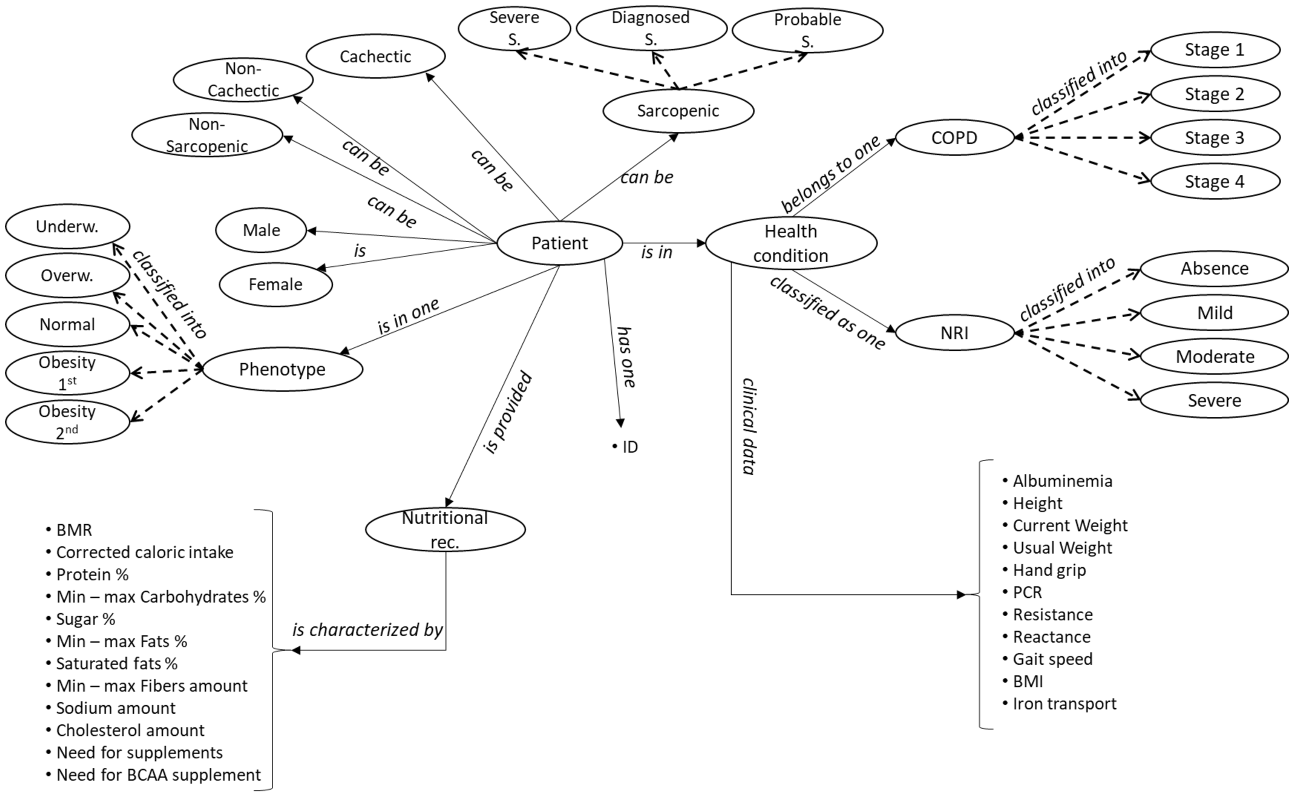

Fig. 5.5. The Conceptual map developed by the team involved in the ontology engineering process for the clinical COPD & Nutrition ontology.

Phenotyoe	Basal metabolic rate (BMR)	Correction	Energy	Protein	BCAA*	Energy-protein supplement	Carbos (min – max)	Sugar	Fiber (min – max)	Fats (min – max)	Saturated fats	Cholesterol	Sodium	Calcium
Normal w. (no sarcopenic. no cachectic)	Harris-Benedict	1.5 (COPD 1-2)	MBR x 1.5	1.2 g/Kg/die	No	No	45-50%	<15%	12.6g/1000Kcal max 25g/die	30-35%	<10%	<300 mg/die	<2.4 g/die (Salt <6 g/die)	M: 1200 mg/die
		1.8 (COPD 3-4)	MBR X 1.8	1.5 g/Kg/die	Yes	No				if PaCO2 >50 35-50%				F: 1500 mg/die
Normal w.. sarcopenic	Harris-Benedict	1.5 (COPD 1-2)	MBR x 1.5	1.5 g/Kg/die	Yes	Yes	45-50%	<15%	12.6g/1000Kcal max 25g/die	30-35%		<300 mg/die		M: 1200 mg/die
		1.8 (COPD 3-4)	MBR X 1.8							if PaCO2 >50 35-50%				F: 1500 mg/die
Normal w.. cachectic	Harris-Benedict	1.8	MBR x 1.8	25%	Yes	Yes	45-50%	<15%	12.6g/1000Kcal max 25g/die	30-35%		<300 mg/die		M: 1200 mg/die
										if PaCO2 >50 35-50%				F: 1500 mg/die
Overw. (no sarcopenic. no cachectic)	Harris-Benedict	1.5 (COPD 1-2)	MBR x 1.5	1.2 g/Kg/die	No	No	45-50%	<15%	12.6g/1000Kcal max 25g/die	30-35%		<300 mg/die		M: 1200 mg/die
		1.8 (COPD 3-4)	MBR X 1.8	1.5 g/Kg/die	Yes	No				if PaCO2 >50 35-50%				F: 1500 mg/die
Overw. sarcopenic	Mifflin	1.5 (COPD 1-2)	MBR x 1.5	1.5/Kg/die	Yes	No*	45-50%	<15%	12.6g/1000Kcal max 25g/die	30-35%		<300 mg/die		M: 1200 mg/die
		1.8 (COPD 3-4)	MBR X 1.8							if PaCO2 >50 35-50%				F: 1500 mg/die
Overw.t cachectic	Mifflin	1.8	MBR x 1.8	25%	Yes	Yes	45-50%	<15%	12.6g/1000Kcal max 25g/die	30-35%		<300 mg/die		M: 1200 mg/die
										if PaCO2 >50 35-50%				F: 1500 mg/die

*Table 23. An excerpt (illustrating only normal weight and overweight) of the table provided by dieticians to calculate the amount of macronutrients for each anthropometric phenotype, according to COPD stage and other conditions. The symbol * represents the indication: "if caloric intake is not reached with meals".*

Competency Questions
1. What information identify a patient? *1a. What basic information are used to identify the patient?* *1b. What clinical information are used to identify the patient?*
A patient is identified by an ID and the gender. Each patient is associated to one health condition and to one anthropometric phenotype (defined on BMI cut-offs). Each patient can be classified as a sarcopenic or cachectic patient, or as a non-cachectic or non-sarcopenic patient.
2. How is COPD evaluated?
COPD is evaluated according to the criteria defined in GOLD standard: it can be Mild, Moderate, Severe and Very severe. The criterion to be analysed is the FEV1.
3. How is Sarcopenia evaluated?
The status of sarcopenia is evaluated according to clinical standards (operational definition of sarcopenia): the first criterion consists of low muscle strength; the second criterion consists of a low muscle quantity or quality; the third criterion consists of low physical performance. The presence of the first criterion alone indicates probable sarcopenia; the presence of both the first and second criteria indicates diagnosed sarcopenia; the presence of all three criteria indicates severe sarcopenia.
4. How is cachexia evaluated?
Cachexia is evaluated by means of biochemical indicators, according to [171]. Albuminemia, iron transport, polymerase chain reaction criteria must be co-present to indicate a cachexia diagnosis.
5. Which data characterize the patient's health condition? *5a How is Nutritional risk index assessed?*
Patients' health conditions must indicate the stage of COPD and the Nutritional Risk Index profile characterizing the patient. Each health condition must illustrate anthropometric measures (current weight, usual weight, height in meters, BMI), physical performance indicators (hand grip and gait speed), and biochemical indicators (albuminemia, PCR, Resistance, Reactance, Iron transport). The Nutritional risk index assessment is performed following clinical standards.
6. What recommendations are given to clinical personnel?
The recommendations provided to clinical personnel indicate (for each patient) the basal metabolic rate and the corrected caloric intake, the daily macro-nutrients shares (protein, minimum and maximum share of carbohydrates, minimum and maximum share of fats, minimum and maximum shares of fibers, maximum share of sugar, maximum share of saturated fats), the amount of cholesterol and sodium, the indication of whether the patient should increase his/her caloric intake by means of BCAA or enegy-protein supplementations.
7. How are recommendation values calculated?
The indications provided by the patient's recommendation are calculated according to clinical standards and differentiated according to: patient's gender; stage of COPD; anthropometric phenotype.

Table 24: The list of CQs and answers for the COPD & Nutrition ontology engineering process.

5.4.3.2 Development

The development step adopted the Conceptual map and CQs produced in the previous step to guide the whole development process and the discussion on whether to model some concepts as pertaining the patients or their health conditions. Clinicians explicitly asked to be illustrated any significant advancement in the development of TBox and Abox (e.g., patient modelling, health condition characterization, recommendation modelling, etc.) to ensure that undisered entailments were modelled in the ontology. From a reuse perspective, the only ODP reused in this ontology is the one introduced in Chapter 4 that relates a `copd:Patient` to his/her `copd:Health_Condition` via the `copd:isInHealthCondition` object property.

The development started with the identification of concepts that could be translated into `owl:Classes`. The concept of `copd:Patient` is pivotal in this ontology. Each patient is defined by exactly one `copd:patientID`, is recommended at least one `copd:Nutritional_Recommendation`, and is in a `copd:Health_Condition`.

```
copd:Patient rdf:type owl:Class ;
             owl:equivalentClass [ rdf:type owl:Restriction ;
                                   owl:onProperty copd:hasRecommendation ;
                                   owl:someValuesFrom
copd:Nutritional_Recommendation
                                 ] ,
                                 [ rdf:type owl:Restriction ;
                                   owl:onProperty copd:isInHealthCondition ;
                                   owl:someValuesFrom copd:Health_Condition
                                 ] ,
                                 [ rdf:type owl:Restriction ;
                                   owl:onProperty copd:patientID ;
                                   owl:qualifiedCardinality
"1"^^xsd:nonNegativeInteger ;
                                   owl:onDataRange xsd:string
                                 ] .
```

Each patient needs to be classified as `copd:Female` or `copd:Male` – which are disjoint classes – and as `copd:Cachectic` (or its complement `copd:non-Cachectic`) or `copd:Sarcopenic` (or its complement `copd:non-Sarcopenic`). Sarcopenia and Cachexia are modelled as attributes of the patient – and not of his/her health condition: the clinical personnel deemed essential to state that these two conditions have the role of systemic status, therefore, they characterize the individual as a whole. The class `copd:Sarcopenic` is further detailed into the subclassed `copd:Probable_Sarcopenic`, `copd:Diagnosed_Sarcopenic`, and `copd:Severe_Sarcopenic` to reflect the operational definition standard provided by clinicians.

In the same way, the `copd:Anthropometric_Phenotype` are characteristics of the `copd:Patient`, and this class lists five subclasses for the representation of the WHO phenotypes.

Similarly to `copd:Patient`, the development of the TBox pertaining to the patient's health condition was discussed among the team members: each health condition is characterized by an NRI profile, but in general, not necessarily a `copd:Health_Condition` is characterized by COPD. The terms adopted in the Conceptual map to sketch the relationships holding between a `copd:Health_Condition`, `copd:Nutritional_Risk_Index_Profile` and `copd:COPD_HC` were found indicative of the clinicians' perspective: both NRI and COPD are considered particular attributes of a health condition – i.e., there could be health conditions characterized only by a NRI profile but lacking COPD. Therefore, the classes `copd:Nutritional_Risk_Index_Profile` and `copd:COPD_HC` were modelled as `rdfs:subclassOf` `copd:Health_Condition`. This decision was also encouraged by the fact that the datatype properties `copd:FEV1` and `copd:nutritionalRiskIndex` have `copd:Health_Condition` as domain.

The subclasses of copd:Nutritional_Risk_Index_Profile and copd:COPD_HC are characterized by restrictions that allow classifying individual health conditions whose copd:nutrtionalRiskIndex and copd:FEV1 object values fall under specific restrictions. For example, the copd:Stage1 – a rdfs:subclassOf copd:COPD_HC – is defined as:

```
copd:Stage1 rdf:type owl:Class ;
            owl:equivalentClass [ rdf:type owl:Restriction ;
                                  owl:onProperty copd:FEV1 ;
                                  owl:someValuesFrom [ rdf:type rdfs:Datatype ;
                                                       owl:onDatatype xsd:int ;
                                                       owl:withRestrictions  (  [
xsd:maxInclusive "30"^^xsd:int
                                                                                  ]
                                                                                )
                                                     ]
                                ] ;
            rdfs:subClassOf copd:COPD_HC .
```

Each copd:Health_Condition is described by a set of datatype properties, which represent the clinical data that need to be acquired for each patient through blood test and spirometry to enable his/her classification and recommendations. Each owl:Individual belonging to this class also materialized inferred triples related to the copd:AppendicularSkeletalMuscleMass, the copd:ResistiveIndex, and the copd:nutritionRiskIndex. While the copd:nutritionalRiskIndex is calculated using SWRL rules and used to classify each copd:Health_Condition into one of NRI's subclasses, the copd:ResistiveIndex (RI) is a piece of information necessary to calculate the copd:AppendicularSkeletalMuscleMass (both are inferred as the result of two different SWRL rules). Figure 5.6 illustrates an example of copd:Health_Condition completed with all its datatype properties (both asserted and inferred).

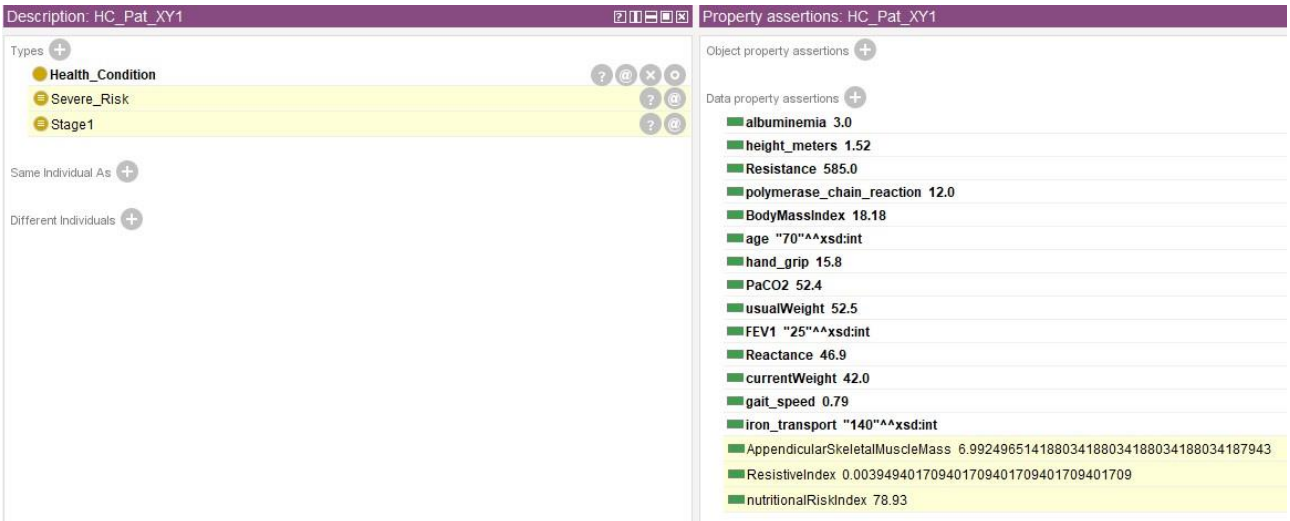

Fig. 5.6. The complete datatype property set for a copd:Health_Condition. The datatype properties with yellow background represent inferred values.

The ontology makes use of 39 datatype properties and 2 object properties (copd:isInHealthCondition and copd:hasRecommendation): almost all datatype properties are used to provide values for the patient's health condition and nutritional recommendation.

The COPD & Nutrition ontology contains 79 SWRL rules, which are largely used to represent the tuples of Table 23 – depicting the conditions that determine the shares and amounts of nutrients characterizing a patient's diet. The equations adopted to calculate the BMR and the corrected caloric intake were adapted with

SWRL using mathematical built-ins. Taking as example a copd:Overweight, copd:non-Sarcopenic, and copd:non-Cachectic male patient characterized by copd:Stage2 disease, the BMR is inferred by the following rule:

```
Male(?p), Overweight(?p), (not (Cachectic))(?p), (not (Sarcopenic))(?p), hasRecommendation(?p, ?rec), isInHealthCondition(?p, ?hc), age(?hc, ?age), currentWeight(?hc, ?kg), height_meters(?hc, ?m), multiply(?a, ?kg, 13.75), multiply(?b, ?m, 5, ?100), multiply(?c, 6.78, ?age), add(?d, 66.5?a, ?b), subtract (?e, ?d, ?c), roud(?f, ?e) -> regularRecommendedCaloricIntake(?rec, ?f)
```

Then the correction is applied:

```
(Normal_Weight or Obesity_1st_Degree or Overweight or Underweight)(?p), hasRecommendation(?p, ?rec), isInHealthCondition(?p, ?hc), (Stage1 or Stage2)(?hc), (not (Cachectic))(?p), (not (Sarcopenic))(?p), regularRecommendedCaloricIntake(?rec, ?reg), multiply(?corin, ?reg, 1.5), round(?f, ?corin) -> correctedRecommendedCaloricIntake(?rec, ?f)
```

The definition of the share of proteins that a COPD patient needs is calculated by identifying the amount (in grams) of proteins. With the sole exception of cachectic patients – which are given 25% protein share for clinical reasons –, for each patient, the daily quantity of proteins is calculated according to their weight:

```
(Normal_Weight or Overweight)(?p), (not (Cachectic))(?p), (not (Sarcopenic))(?p), hasRecommendation(?p, ?rec), isInHealthCondition (?p, ?hc), currentWeight (?hc, ?w), multiply (?pgra, ?w, 1.2) -> proteinsGrams (?rec, ?pgra)
```

This amount is then converted in calories, bearing in mind that 1 protein is equal to 4 Kcal:

```
Patient(?p), hasRecommendation(?p, ?rec), proteinGrams(?rec, ?pg), multiply(?pcal, 4.0, ?pg) -> proteinCalories(?rec, ?pcal)
```

And the daily protein share is then calculated by the following SWRL rule:

```
(Normal_Weight or Overweight)(?p), (not (Cachectic))(?p), (not (Sarcopenic))(?p), hasRecommendation(?p, ?rec), correctedRecommendedCaloricIntake(?rec, ?in), proteinCalories(?rec, ?prcal), multiply(?a, prcal, 100.0), divide(?prshare, ?a, ?in) -> proteinShare(?rec, ?prshare)
```

This approach also enables the possibility to correct the amount of proteins for particular classes of patients: for example, dieticians indicated that copd:non-Sarcopenic, copd:non-Cachectic, and copd:Underweight patients should have their protein share calculated taking into account a different BMI (which is set to a higher value, in order to fight their underweight condition, and is established at 22.5). SWRL rules can enable this inference:

```
Underweight(?p), (not (Cachectic))(?p), (not (Sarcopenic))(?p), hasRecommendation(?p, ?rec), isInHealthCondition(?, ?hc), height_meters(?hc, ?m), multiply(?ifweight, 22.5, ?a), multiply (?a, ?m, ?m), multiply(?prot, ?ifweight, 1.2) -> proteinGrams(?rec, ?prot)
```

SWRL rules are also adopted to represent the criteria for the definition of Sarcopenia. While to be classified as a copd:Probable_Sarcopenic patient, an individual's hand grip is compared to a parameter, the definition of a copd:Diagnosed_Sarcopenic requires the ASMM equation to be modelled (with different equations for males and females – in the example)

```
Female(?p), HealthCondition(?hc), isInHealthCondition(?p, ?hc), Reactance(?hc, ?xc), ResistiveIndex(?hc, ?ri), currentWeight(?hc, ?w), multiply(?a, ?ri, 0.227), multiply(?b, ?w, 0.095), multiply (?c, ?xc, 0.064), add (?asmm, ?a, ?b, ?c) → AppendicularSkeletalMuscleMass(?hc, ?asmm)
```

and then to be compared to cut-offs:

```
Female(?p), Probable_Sarcopenic(?p), isInHealthCondition(?p, ?hc), AppendicularSkeletalMuscleMass(?hc, ?asmm), lessThan(?asmm, 15.0) -> Diagnosed_Sarcopenic(?p)
```

The case of `copd:Severe_Sarcopenic` patient compares a `copd:Diagnosed_Sarcopenic` patient's `copd:gait_speed` to the cut-off to infer whether or not the patient is characterized by severe sarcopenia.

Similarly, the inference about the cachectic status of a patient is performed with a single SWRL rule checking that all three conditions characterizing this condition are present:

```
Patient(?p), Sarcopenic(?p), isInHealthCondition(?p, ?hc), HealthCondition(?hc), albuminemia(?hc, ?al), lessThan(?al, 3.5), iron_transport(?hc, ?iron), lessThan(?iron, 150.0), polymerase_chain_reaction(?hc, ?pcr), greaterThan(?pcr, 10.0) -> Cachectic(?p)
```

BCAA supplementations are indicated as suitable for specific categories of patients:

```
(Cachectic or Sarcopenic)(?p), (Normal_Weight or Obesity_1st_Degree or Obesity_2nd_Degree or Overweight or Underweight)(?p), hasRecommendation(?p, ?rec) -> BCAAsupplement(?rec, "yes "^^xsd:string)
```

With regard to `copd:fats`, the amount of this nutrient can vary according to the level of `copd:PaCO2` registered during spirometry and reported in the patient's health condition. Two rules determine the minimum and maximum amount of fats, taking into account the parameter:

```
Patient(?p), hasRecommendation(?p, ?r), isInHealthCondition(?p, ?hc), PaCO2(?hc, ?x), lessThanOrEqual(?x, 50.0) -> fatsMAXshare(?r, 35), fatsMINshare(?r, 30)
```

```
Patient(?p), hasRecommendation(?p, ?r), isInHealthCondition(?p, ?hc), PaCO2(?hc, ?y), greaterThan(?y, 50.0) -> fatsMAXshare(?r, 50), fatsMINshare(?r, 35)
```

Fiber plays a central role in the diet of COPD patients, and clinical personnel indicated a fixed maximum amount of 25 grams per day, while the minimum amount depends on corrected caloric intake. However, it is important that the amount of fiber does not exceed 25 grams; therefore two SWRL rules were modelled to indicate this condition:

```
Patient(?p), isInHealthCondition(?p, ?hc), hasRecommendation(?p, ?rec), correctedRecommendedCaloricIntake(?rec, ?cal), divide(?x, ?cal, 1000), multiply(?b, 12.6, ?x), lessThan(?b, 25.0) -> fiberMINamount(?rec, ?b)
```

```
Patient(?p), isInHealthCondition(?p, ?hc), hasRecommendation(?p, ?rec), correctedRecommendedCaloricIntake(?rec, ?cal), divide(?x, ?cal, 1000), multiply(?b, 12.6, ?x), greaterThanOrEqual (?b, 25.0) -> fiberMINamount(?rec, 25.0)
```

As mentioned above, the COPD & Nutrition ontology provides enough SWRL rules to model all the information identified by the domain expert and elicited in Table 23. As foreseen by the development step in AgiSCOnt, the ontology underwent a test phase with 16 patients – provided by clinicians, who also provided anonymized data regarding the patient's health conditions. The test was divided into two phases: the first was dedicated to assessing whether the ontology provided a correct classification of the patients (i.e., if it identifies `copd:Sarcopenic` and `copd:Cachectic` status for each patient, and if the stage of COPD and the `copd:Nutritional_Risk_Index_Profile` were correctly inferred). The following SPARQL query investigates each patient's status and his/her health condition:

```
SELECT distinct ?id  ?status ?antrPhen ?copdstage ?nri  WHERE {
      ?x a owl:NamedIndividual;
          a copd:Patient ;
```

```
            copd:patientID ?id ;
            a ?antrPhen ;
            a ?status .
        ?status rdfs:subClassOf copd:Patient .
                     FILTER (?status != copd:Patient).

        ?antrPhen rdfs:subClassOf copd:Anthropometric_Phenotypes .
                    FILTER (?antrPhen != copd:Anthropometric_Phenotypes).

         ?x copd:isInHealthCondition ?hc .
         ?hc a copd:COPD_HC ;
               a copd:Nutritional_Risk_Index_Profile ;
               a ?copdstage ;
               a ?nri .

         ?copdstage rdfs:subClassOf copd:COPD_HC.
         ?nri rdfs:subClassOf copd:Nutritional_Risk_Index_Profile .

                           FILTER (?copdstage != copd:Health_Condition)
                           FILTER (?copdstage != copd:COPD_HC)
                           FILTER (?nri != copd:Nutritional_Risk_Index_Profile)

} ORDER BY ?id ?status
```

An excerpt of the results of the query is reported in Table 25 (the full list of patients and health conditions is reported in Appendix 3).

?id	**?status**	**?antrPhen**	**?copdstage**	**?nri**
001	copd:Cachectic	copd:Underweight	copd:Stage1	copd:Severe_Risk
001	copd:Diagnosed_Sarcopenic	copd:Underweight	copd:Stage1	copd:Severe_Risk
001	copd:Female	copd:Underweight	copd:Stage1	copd:Severe_Risk
001	copd:Probable_Sarcopenic	copd:Underweight	copd:Stage1	copd:Severe_Risk
001	copd:Sarcopenic	copd:Underweight	copd:Stage1	copd:Severe_Risk
001	copd:Severe_Sarcopenic	copd:Underweight	copd:Stage1	copd:Severe_Risk
…	…	…	…	…
BB	copd:Female	copd:Normal_Weight	copd:Stage4	copd:Absence_of_Risk
BB	copd:non-Cachetic	copd:Normal_Weight	copd:Stage4	copd:Absence_of_Risk
BB	copd:non-Sarcopenic	copd:Normal_Weight	copd:Stage4	copd:Absence_of_Risk
CV	copd:Male	copd:Obesity_1st_Degree	copd:Stage3	copd:Mild_Risk
CV	copd:non-Cachetic	copd:Obesity_1st_Degree	copd:Stage3	copd:Mild_Risk
CV	copd:non-Sarcopenic	copd:Obesity_1st_Degree	copd:Stage3	copd:Mild_Risk
…	…	…	…	…

Table 25. An excerpt of the results retrieved for the query investigating, for each patient, their ID, their status (whether or not they are affected by sarcopenia or cachexia), the stage of COPD, their NRI profile. For patients characterized by sarcopenia, all the subclasses of `copd:Sarcopenic` *are illustrated, so that clinical personnel can easily see the importance of this condition.*

The pneumologist and dieticians verified the correctness of the classification for each patient. All 16 individuals representing patients were found to be correctly classified. The second phase deals with the retrieval of nutritional suggestions and their evaluation by the clinical personnel, with the aim of assessing the validity of the SWRL rules modelled in the COPD & Nutrition ontology. For this purpose, the following SPARQL query retrieves all the nutrient minimum and maximum shares, as well as quantities, deemed important for COPD patients:

The query retrieved the results reported in Appendix 4. Each `copd:Nutritional_Recommendation` and its inferred nutrient's values were evaluated by clinical personnel and were found corrected – although, for some values such as the `copd:proteinShare` and `copd:fiberMINamount` a rounding of the decimal was suggested by dieticians.

5.4.3.3 Maintenance and use

The COPD & Nutrition ontology described in the previous subsection is a tested prototype, able to classify patients properly and suggest clinicians with nutrition-related recommendations. Before the presented round of tests with 16 patients, the ontology was tested with 10 different patients – whose health conditions were lacking significant data, such as `copd:FEV1` and `copd:PaCO2`, as well as the `copd:usualWeight` information. Following the partial results the ontology was able to retrieve, domain experts underlined the necessity of having full health conditions and patient data to be able to draw significant inferences from the ontology. Although the knowledge underlying the ontology is based on clinical practice and literature, it was found essential to invite other pneumologists and nutrition-domain experts to validate the results provided by the system. To this aim, another sample of patients will be recruited and screened (within the HUB sPATIALS[3] project) to provide a more solid test-case base. Once this evaluation – and the integration of possible feedback – is completed, the clinician application will be developed and tested with a sample of pneumologists and dieticians.

Clinical personnel also noted that while the inferences produced by the ontology are relevant for pneumologists, the DSS may also give some relevant indications to patients. Therefore, the ontology will serve as a base for an application able to support clinicians and for an application dedicated to patients. The patients' application is expected to indicate on a daily basis the amount of calories the subject should intake (the `copd:correctedRecommendedCaloriIntake`) and the shares of main nutrients: considering that COPD can manifest in adults older than 40, such a patient application might prove useful to support them in managing COPD exacerbation through healthy and tailored nutrition.

Finally, during the Domain analysis step, it emerged that many COPD patients might be characterized by other chronic co-morbidities that affect nutrition and nutrients' absorption (e.g., diabetes mellitus, dyslipidemia, etc.). It was then suggested that a second – and wider – version of the prototype ontology should take into account also the main co-morbidities characterizing COPD patients. This fact would open the possibility of partially re-engineering the model for the patient's health condition to include the possibility of representing health conditions characterized by more than one disease.

5.4.4 Discussion

The COPD & Nutrition ontology has the purpose of classifying COPD patients to assess their health status and to illustrate a tailored diet, composed of shares of macro-nutrients and specific recommended amounts for some nutrients. The test with 16 patients confirmed the possibility for the ontology to correctly infer the information set as a goal, and the clinical personnel involved in the OE process validated the correctness of the inferred data.

The domain analysis step covered a pivotal importance in the engineering of this ontology: two different types of clinicians, with disciplines that rarely intersect, were able to collaborate and acquire knowledge through the unstructured interviews and to "put it together" in the Conceptual map. The development step was conducted by presenting clinical personnel with each of the relevant advancements in the modelling of TBox and ABox, as the domain experts were very careful in monitoring that entailed knowledge was not providing misrepresentations of patients or their health conditions. As foreseen in AgiSCOnt, this led to the (partial) modification of the Conceptual map, with more details added to the answers to the CQs list. Domain experts also underlined that the first aim of the OE process was to get to a working prototype of the ontology, asking to postpone any mapping and limiting the reuse of existing knowledge sources to those deemed relevant.

In particular, a preliminary version of the COPD & Nutrition ontology foresaw the possibility to re-use the International Classification of Diseases (ICD) – which also exists in ontological form – to correctly identify

COPD. However, the cost of reusing that ontology, compared to the benefit that might have come from its implementation, was found significant. Nonetheless, a version of the COPD & Nutrition ontology taking into account other co-morbidities (such as diabetes mellitus, dyslipidemia, heart, and circulation-related diseases, and metabolic syndromes) may not avoid relying on a solid and established knowledge source to properly identify diseases.

It is worth noticing that the COPD & Nutrition ontology can infer some nutrition-related information also in case of incomplete health conditions (i.e., those `copd:Health_Condition` lacking some data): however, considering the primary aim of the DSS, it was decided to test the ontology only using fully-described health conditions.

Finally, reasoning process (conducted with Pellet) with the COPD & Nutrition ontology (and 16 patients) was completed within 18 seconds). Further tests with a larger ABox representing more patients and their health conditions will be conducted with the Stardog triple-store to assess the scalability of the ontology. Similarly to the ontology described in the previous Section, a larger number of `copd:Patients` and their health conditions may require restructuring the ontology into one general TBox and as many Aboxes as the patients to be represented are.

5.5 Appendix

User	Patient ID	DOSS score	PAS scores			
			PAS Liquid	*PAS Semi-liquid*	*PAS Semi-solid*	*PA Solid*
dis:Us_CDB	1	5	5	1	1	1
dis:Us_CDC	2	5	1	1	1	1
dis:Us_CO-LO	3	1	-	-	1	-
dis:Us_FM	4	6	1	1	1	1
dis:Us_FO-MA	5	1	7	7	3	7
dis:Us_GIO-PA	6	5	3	1	1	1
dis:Us_MF	7	6	1	1	1	1
dis:Us_PA-LE	8	7	1	1	1	1
dis:Us_ROPRO	9	5	3	1	1	1
dis:Us_RU-MA	10	4	1	-	1	1
dis:Us_SOCE	11	4	3	1	3	3
dis:Us_VB	12	2	6	1	6	6

Appendix 1. A table representing the patients and their health conditions used to validate the inferences performed by the ontology. Blank values correspond to a lack of data.

?user	?id	?recipe	?recipeName	?cons
dis:Us_CDC	2	dis:r15	GNOCCHI DI SEMOLINO VERDE, ASPARAGO CON BURRO E SALVIA@it	dis:CONS_Solid_D
dis:Us_CDC	2	dis:r59	CUPOLA DI YOGURT ALLA PESCA@it	dis:CONS_Semisolid_C
dis:Us_CDC	2	dis:r16	LASAGNE ALLE VERDURE@it	dis:CONS_Solid_D
dis:Us_CDC	2	dis:r57	CREMA PASTICCERA@it	dis:CONS_Semiliquid_B
dis:Us_CDC	2	dis:r13	GNOCCHI DI PATATE,VERZA E PECORINO@it	dis:CONS_Solid_D
dis:Us_CDC	2	dis:r58	CREMA PASTICCERA VELOCE@it	dis:CONS_Semiliquid_B
dis:Us_CDC	2	dis:r14	GNOCCHI DI RICOTTA "AIO OIO E BOTTARGA"@it	dis:CONS_Solid_D
dis:Us_CDC	2	dis:r55	MOUSSE DI ASPARAGI E FORMAGGIO@it	dis:CONS_Semisolid_C
dis:Us_CDC	2	dis:r11	GELATINA DI MOSCATO CON FRAGOLINE DI BOSCO@it	dis:CONS_Solid_D
dis:Us_CDC	2	dis:r56	PURÈ DI FAVE CON SCAROLA E PECORINO@it	dis:CONS_Semiliquid_B
dis:Us_CDC	2	dis:r12	GIRELLO DI VITELLO CON SALSA TONNATA@it	dis:CONS_Semisolid_C
dis:Us_CDC	2	dis:r53	MOUSSE DI TONNO E MELANZANE@it	dis:CONS_Semisolid_C
dis:Us_CDC	2	dis:r10	DOLCE CREMOSO ALLO YOGURT@it	dis:CONS_Semiliquid_B
dis:Us_CDC	2	dis:r54	MOUSSE DI GORGONZOLA@it	dis:CONS_Semisolid_C
dis:Us_CDC	2	dis:r17	LASAGNE BIANCHE DI MARE CON PESCE BIANCO, SEPPIA, GAMBERO E PESTO DI BASILICO@it	dis:CONS_Solid_D
dis:Us_CDC	2	dis:r18	LASAGNE DI CARNE ALLA MARCHIGIANA@it	dis:CONS_Solid_D
dis:Us_CDC	2	dis:r60	SEMIFREDDO ALLA PESCA@it	dis:CONS_Semisolid_C
dis:Us_CDC	2	dis:r26	PASSATA DI TOPINANBUR@it	dis:CONS_Semiliquid_B
dis:Us_CDC	2	dis:r27	PASSATELLI CON PUREA DI SEDANO RAPA,STRACCHINO E SALSA NOCI@it	dis:CONS_Solid_D
dis:Us_CDC	2	dis:r24	PANCOTTO "DEL PRETE"@it	dis:CONS_Semisolid_C
dis:Us_CDC	2	dis:r25	PANNA COTTA AL CARAMELLO@it	dis:CONS_Semisolid_C
dis:Us_CDC	2	dis:r22	MILKSHAKE BANANA E PISTACCHIO@it	dis:CONS_Semiliquid_A
dis:Us_CDC	2	dis:r23	MI-TIRI-SU!@it	dis:CONS_Solid_D
dis:Us_CDC	2	dis:r20	MELONE INVERNALE E ARANCIA@it	dis:CONS_Semiliquid_A
dis:Us_CDC	2	dis:r21	MERENDA IN CAMPAGNA CON FAVA E PECORINO@it	dis:CONS_Semisolid_C
dis:Us_CDC	2	dis:r28	PETTO DI TACCHINO BOLLITO CON SALSA VERDE@it	dis:CONS_Semisolid_C
dis:Us_CDC	2	dis:r29	POLLO ALLA GRIGLIA CON SALSA LIMONE WORCESTER E PEPERONE GRIGLIATO@it	dis:CONS_Semisolid_C
dis:Us_CDC	2	dis:r30	POLENTA DI MAIS CON BACCALÀ, OLIVE E CAPPERI E POMODORI DATTERINO@it	dis:CONS_Solid_D
dis:Us_CDC	2	dis:r37	VITELLONE ALLA CALABRESE@it	dis:CONS_Semisolid_C
dis:Us_CDC	2	dis:r38	ZUPPA INGLESE@it	dis:CONS_Solid_D
dis:Us_CDC	2	dis:r35	SPIGOLA CON SALSA DI POMODORI ARROSTO@it	dis:CONS_Semisolid_C
dis:Us_CDC	2	dis:r36	VERDURA AGRODOLCE@it	dis:CONS_Solid_D

dis:Us_CDC	2	dis:r33	QUENELLE DI FEGATO@it	dis:CONS_Semisolid_C
dis:Us_CDC	2	dis:r34	SPIGOLA AL VAPORE CON BIETOLA AL LIMONE@it	dis:CONS_Semisolid_C
dis:Us_CDC	2	dis:r31	POLPETTE AL SUGO DI POMODORO FRESCO@it	dis:CONS_Solid_D
dis:Us_CDC	2	dis:r32	POLPETTE DI CARNE@it	dis:CONS_Solid_D
dis:Us_CDC	2	dis:r39	LATTE E BISCOTTI PRIMA INFANZIA@it	dis:CONS_Semiliquid_B
dis:Us_CDC	2	dis:r40	THE E BISCOTTI PRIMA INFANZIA@it	dis:CONS_Semiliquid_B
dis:Us_CDC	2	dis:r41	SPREMUTA D'ARANCIA E BISCOTTI PRIMA INFANZIA@it	dis:CONS_Semiliquid_B
dis:Us_CDC	2	dis:r48	VELLUTATA DI ZUCCHINE E QUENELLE DI FORMAGGIO@it	dis:CONS_Semiliquid_B
dis:Us_CDC	2	dis:r04	COCKTAIL DI GAMBERI@it	dis:CONS_Semisolid_C
dis:Us_CDC	2	dis:r49	ZUPPA DI PORRI E PATATE@it	dis:CONS_Semiliquid_A
dis:Us_CDC	2	dis:r05	CONIGLIO ALLA GRIGLIA CON SALSA BBQ@it	dis:CONS_Semisolid_C
dis:Us_CDC	2	dis:r46	CREMA DI PATATE@it	dis:CONS_Semiliquid_B
dis:Us_CDC	2	dis:r02	CANNELLONI DI CRESPELLA CON PATATE,PORRO E SALSA POMODORO@it	dis:CONS_Solid_D
dis:Us_CDC	2	dis:r03	CARNE DI VITELLA IN PANZANELLA@it	dis:CONS_Semisolid_C
dis:Us_CDC	2	dis:r47	PASSATO DI CECI E FAGIOLI@it	dis:CONS_Semiliquid_B
dis:Us_CDC	2	dis:r44	PASSATO DI VERDURA SENZA POMODORO@it	dis:CONS_Semiliquid_A
dis:Us_CDC	2	dis:r45	PASSATO DI VERDURA@it	dis:CONS_Semiliquid_A
dis:Us_CDC	2	dis:r01	BIANCOMANGIARE AI LAMPONI@it	dis:CONS_Semisolid_C
dis:Us_CDC	2	dis:r42	FRULLATO DI BANANA E BISCOTTI@it	dis:CONS_Semiliquid_B
dis:Us_CDC	2	dis:r43	SEMOLINO AL LIMONE@it	dis:CONS_Semiliquid_B
dis:Us_CDC	2	dis:r08	CREMA DI CAROTE E ZUCCHINE@it	dis:CONS_Semiliquid_B
dis:Us_CDC	2	dis:r09	CREMA DI LEGUMI@it	dis:CONS_Semiliquid_B
dis:Us_CDC	2	dis:r06	CONIGLIO IN PORCHETTA@it	dis:CONS_Semisolid_C
dis:Us_CDC	2	dis:r07	CREM CARAMEL@it	dis:CONS_Semisolid_C
dis:Us_CDC	2	dis:r51	ZUPPA DI CAROTE E CECI@it	dis:CONS_Semiliquid_B
dis:Us_CDC	2	dis:r52	MOUSSE DI PROSCIUTTO@it	dis:CONS_Semisolid_C
dis:Us_CDC	2	dis:r50	PAPPA CON POMODORO@it	dis:CONS_Semisolid_C
dis:Us_FM	4	dis:r15	GNOCCHI DI SEMOLINO VERDE, ASPARAGO CON BURRO E SALVIA@it	dis:CONS_Solid_D
dis:Us_FM	4	dis:r59	CUPOLA DI YOGURT ALLA PESCA@it	dis:CONS_Semisolid_C
dis:Us_FM	4	dis:r16	LASAGNE ALLE VERDURE@it	dis:CONS_Solid_D
dis:Us_FM	4	dis:r57	CREMA PASTICCERA@it	dis:CONS_Semiliquid_B
dis:Us_FM	4	dis:r13	GNOCCHI DI PATATE,VERZA E PECORINO@it	dis:CONS_Solid_D
dis:Us_FM	4	dis:r58	CREMA PASTICCERA VELOCE@it	dis:CONS_Semiliquid_B

dis:Us_FM	4	dis:r14	GNOCCHI DI RICOTTA "AIO OIO E BOTTARGA"@it	dis:CONS_Solid_D
dis:Us_FM	4	dis:r55	MOUSSE DI ASPARAGI E FORMAGGIO@it	dis:CONS_Semisolid_C
dis:Us_FM	4	dis:r11	GELATINA DI MOSCATO CON FRAGOLINE DI BOSCO@it	dis:CONS_Solid_D
dis:Us_FM	4	dis:r56	PURÈ DI FAVE CON SCAROLA E PECORINO@it	dis:CONS_Semiliquid_B
dis:Us_FM	4	dis:r12	GIRELLO DI VITELLO CON SALSA TONNATA@it	dis:CONS_Semisolid_C
dis:Us_FM	4	dis:r53	MOUSSE DI TONNO E MELANZANE@it	dis:CONS_Semisolid_C
dis:Us_FM	4	dis:r10	DOLCE CREMOSO ALLO YOGURT@it	dis:CONS_Semiliquid_B
dis:Us_FM	4	dis:r54	MOUSSE DI GORGONZOLA@it	dis:CONS_Semisolid_C
dis:Us_FM	4	dis:r17	LASAGNE BIANCHE DI MARE CON PESCE BIANCO, SEPPIA, GAMBERO E PESTO DI BASILICO@it	dis:CONS_Solid_D
dis:Us_FM	4	dis:r18	LASAGNE DI CARNE ALLA MARCHIGIANA@it	dis:CONS_Solid_D
dis:Us_FM	4	dis:r60	SEMIFREDDO ALLA PESCA@it	dis:CONS_Semisolid_C
dis:Us_FM	4	dis:r26	PASSATA DI TOPINANBUR@it	dis:CONS_Semiliquid_B
dis:Us_FM	4	dis:r27	PASSATELLI CON PUREA DI SEDANO RAPA,STRACCHINO E SALSA NOCI@it	dis:CONS_Solid_D
dis:Us_FM	4	dis:r24	PANCOTTO "DEL PRETE"@it	dis:CONS_Semisolid_C
dis:Us_FM	4	dis:r25	PANNA COTTA AL CARAMELLO@it	dis:CONS_Semisolid_C
dis:Us_FM	4	dis:r22	MILKSHAKE BANANA E PISTACCHIO@it	dis:CONS_Semiliquid_A
dis:Us_FM	4	dis:r23	MI-TIRI-SU!@it	dis:CONS_Solid_D
dis:Us_FM	4	dis:r20	MELONE INVERNALE E ARANCIA@it	dis:CONS_Semiliquid_A
dis:Us_FM	4	dis:r21	MERENDA IN CAMPAGNA CON FAVA E PECORINO@it	dis:CONS_Semisolid_C
dis:Us_FM	4	dis:r28	PETTO DI TACCHINO BOLLITO CON SALSA VERDE@it	dis:CONS_Semisolid_C
dis:Us_FM	4	dis:r29	POLLO ALLA GRIGLIA CON SALSA LIMONE WORCESTER E PEPERONE GRIGLIATO@it	dis:CONS_Semisolid_C
dis:Us_FM	4	dis:r30	POLENTA DI MAIS CON BACCALÀ, OLIVE E CAPPERI E POMODORI DATTERINO@it	dis:CONS_Solid_D
dis:Us_FM	4	dis:r37	VITELLONE ALLA CALABRESE@it	dis:CONS_Semisolid_C
dis:Us_FM	4	dis:r38	ZUPPA INGLESE@it	dis:CONS_Solid_D
dis:Us_FM	4	dis:r35	SPIGOLA CON SALSA DI POMODORI ARROSTO@it	dis:CONS_Semisolid_C
dis:Us_FM	4	dis:r36	VERDURA AGRODOLCE@it	dis:CONS_Solid_D
dis:Us_FM	4	dis:r33	QUENELLE DI FEGATO@it	dis:CONS_Semisolid_C
dis:Us_FM	4	dis:r34	SPIGOLA AL VAPORE CON BIETOLA AL LIMONE@it	dis:CONS_Semisolid_C
dis:Us_FM	4	dis:r31	POLPETTE AL SUGO DI POMODORO FRESCO@it	dis:CONS_Solid_D
dis:Us_FM	4	dis:r32	POLPETTE DI CARNE@it	dis:CONS_Solid_D
dis:Us_FM	4	dis:r39	LATTE E BISCOTTI PRIMA INFANZIA@it	dis:CONS_Semiliquid_B
dis:Us_FM	4	dis:r40	THE E BISCOTTI PRIMA INFANZIA@it	dis:CONS_Semiliquid_B
dis:Us_FM	4	dis:r41	SPREMUTA D'ARANCIA E BISCOTTI PRIMA INFANZIA@it	dis:CONS_Semiliquid_B

dis:Us_FM	4	dis:r48	VELLUTATA DI ZUCCHINE E QUENELLE DI FORMAGGIO@it	dis:CONS_Semiliquid_B
dis:Us_FM	4	dis:r04	COCKTAIL DI GAMBERI@it	dis:CONS_Semisolid_C
dis:Us_FM	4	dis:r49	ZUPPA DI PORRI E PATATE@it	dis:CONS_Semiliquid_A
dis:Us_FM	4	dis:r05	CONIGLIO ALLA GRIGLIA CON SALSA BBQ@it	dis:CONS_Semisolid_C
dis:Us_FM	4	dis:r46	CREMA DI PATATE@it	dis:CONS_Semiliquid_B
dis:Us_FM	4	dis:r02	CANNELLONI DI CRESPELLA CON PATATE,PORRO E SALSA POMODORO@it	dis:CONS_Solid_D
dis:Us_FM	4	dis:r03	CARNE DI VITELLA IN PANZANELLA@it	dis:CONS_Semisolid_C
dis:Us_FM	4	dis:r47	PASSATO DI CECI E FAGIOLI@it	dis:CONS_Semiliquid_B
dis:Us_FM	4	dis:r44	PASSATO DI VERDURA SENZA POMODORO@it	dis:CONS_Semiliquid_A
dis:Us_FM	4	dis:r45	PASSATO DI VERDURA@it	dis:CONS_Semiliquid_A
dis:Us_FM	4	dis:r01	BIANCOMANGIARE AI LAMPONI@it	dis:CONS_Semisolid_C
dis:Us_FM	4	dis:r42	FRULLATO DI BANANA E BISCOTTI@it	dis:CONS_Semiliquid_B
dis:Us_FM	4	dis:r43	SEMOLINO AL LIMONE@it	dis:CONS_Semiliquid_B
dis:Us_FM	4	dis:r08	CREMA DI CAROTE E ZUCCHINE@it	dis:CONS_Semiliquid_B
dis:Us_FM	4	dis:r09	CREMA DI LEGUMI@it	dis:CONS_Semiliquid_B
dis:Us_FM	4	dis:r06	CONIGLIO IN PORCHETTA@it	dis:CONS_Semisolid_C
dis:Us_FM	4	dis:r07	CREM CARAMEL@it	dis:CONS_Semisolid_C
dis:Us_FM	4	dis:r51	ZUPPA DI CAROTE E CECI@it	dis:CONS_Semiliquid_B
dis:Us_FM	4	dis:r52	MOUSSE DI PROSCIUTTO@it	dis:CONS_Semisolid_C
dis:Us_FM	4	dis:r50	PAPPA CON POMODORO@it	dis:CONS_Semisolid_C
dis:Us_RU-MA	10	dis:r37	VITELLONE ALLA CALABRESE@it	dis:CONS_Semisolid_C
dis:Us_RU-MA	10	dis:r15	GNOCCHI DI SEMOLINO VERDE, ASPARAGO CON BURRO E SALVIA@it	dis:CONS_Solid_D
dis:Us_RU-MA	10	dis:r38	ZUPPA INGLESE@it	dis:CONS_Solid_D
dis:Us_RU-MA	10	dis:r16	LASAGNE ALLE VERDURE@it	dis:CONS_Solid_D
dis:Us_RU-MA	10	dis:r13	GNOCCHI DI PATATE,VERZA E PECORINO@it	dis:CONS_Solid_D
dis:Us_RU-MA	10	dis:r35	SPIGOLA CON SALSA DI POMODORI ARROSTO@it	dis:CONS_Semisolid_C
dis:Us_RU-MA	10	dis:r36	VERDURA AGRODOLCE@it	dis:CONS_Solid_D
dis:Us_RU-MA	10	dis:r14	GNOCCHI DI RICOTTA "AIO OIO E BOTTARGA"@it	dis:CONS_Solid_D
dis:Us_RU-MA	10	dis:r55	MOUSSE DI ASPARAGI E FORMAGGIO@it	dis:CONS_Semisolid_C
dis:Us_RU-MA	10	dis:r33	QUENELLE DI FEGATO@it	dis:CONS_Semisolid_C
dis:Us_RU-MA	10	dis:r11	GELATINA DI MOSCATO CON FRAGOLINE DI BOSCO@it	dis:CONS_Solid_D
dis:Us_RU-MA	10	dis:r34	SPIGOLA AL VAPORE CON BIETOLA AL LIMONE@it	dis:CONS_Semisolid_C
dis:Us_RU-MA	10	dis:r12	GIRELLO DI VITELLO CON SALSA TONNATA@it	dis:CONS_Semisolid_C

dis:Us_RU-MA	10	dis:r31	POLPETTE AL SUGO DI POMODORO FRESCO@it	dis:CONS_Solid_D
dis:Us_RU-MA	10	dis:r53	MOUSSE DI TONNO E MELANZANE@it	dis:CONS_Semisolid_C
dis:Us_RU-MA	10	dis:r54	MOUSSE DI GORGONZOLA@it	dis:CONS_Semisolid_C
dis:Us_RU-MA	10	dis:r32	POLPETTE DI CARNE@it	dis:CONS_Solid_D
dis:Us_RU-MA	10	dis:r17	LASAGNE BIANCHE DI MARE CON PESCE BIANCO, SEPPIA, GAMBERO E PESTO DI BASILICO@it	dis:CONS_Solid_D
dis:Us_RU-MA	10	dis:r18	LASAGNE DI CARNE ALLA MARCHIGIANA@it	dis:CONS_Solid_D
dis:Us_RU-MA	10	dis:r04	COCKTAIL DI GAMBERI@it	dis:CONS_Semisolid_C
dis:Us_RU-MA	10	dis:r27	PASSATELLI CON PUREA DI SEDANO RAPA,STRACCHINO E SALSA NOCI@it	dis:CONS_Solid_D
dis:Us_RU-MA	10	dis:r05	CONIGLIO ALLA GRIGLIA CON SALSA BBQ@it	dis:CONS_Semisolid_C
dis:Us_RU-MA	10	dis:r24	PANCOTTO "DEL PRETE"@it	dis:CONS_Semisolid_C
dis:Us_RU-MA	10	dis:r02	CANNELLONI DI CRESPELLA CON PATATE,PORRO E SALSA POMODORO@it	dis:CONS_Solid_D
dis:Us_RU-MA	10	dis:r25	PANNA COTTA AL CARAMELLO@it	dis:CONS_Semisolid_C
dis:Us_RU-MA	10	dis:r03	CARNE DI VITELLA IN PANZANELLA@it	dis:CONS_Semisolid_C
dis:Us_RU-MA	10	dis:r23	MI-TIRI-SU!@it	dis:CONS_Solid_D
dis:Us_RU-MA	10	dis:r01	BIANCOMANGIARE AI LAMPONI@it	dis:CONS_Semisolid_C
dis:Us_RU-MA	10	dis:r21	MERENDA IN CAMPAGNA CON FAVA E PECORINO@it	dis:CONS_Semisolid_C
dis:Us_RU-MA	10	dis:r06	CONIGLIO IN PORCHETTA@it	dis:CONS_Semisolid_C
dis:Us_RU-MA	10	dis:r28	PETTO DI TACCHINO BOLLITO CON SALSA VERDE@it	dis:CONS_Semisolid_C
dis:Us_RU-MA	10	dis:r29	POLLO ALLA GRIGLIA CON SALSA LIMONE WORCESTER E PEPERONE GRIGLIATO@it	dis:CONS_Semisolid_C
dis:Us_RU-MA	10	dis:r07	CREM CARAMEL@it	dis:CONS_Semisolid_C
dis:Us_RU-MA	10	dis:r52	MOUSSE DI PROSCIUTTO@it	dis:CONS_Semisolid_C
dis:Us_RU-MA	10	dis:r30	POLENTA DI MAIS CON BACCALÀ, OLIVE E CAPPERI E POMODORI DATTERINO@it	dis:CONS_Solid_D
dis:Us_RU-MA	10	dis:r50	PAPPA CON POMODORO@it	dis:CONS_Semisolid_C
dis:Us_CDB	1	dis:r15	GNOCCHI DI SEMOLINO VERDE, ASPARAGO CON BURRO E SALVIA@it	dis:CONS_Solid_D
dis:Us_CDB	1	dis:r59	CUPOLA DI YOGURT ALLA PESCA@it	dis:CONS_Semisolid_C
dis:Us_CDB	1	dis:r16	LASAGNE ALLE VERDURE@it	dis:CONS_Solid_D
dis:Us_CDB	1	dis:r57	CREMA PASTICCERA@it	dis:CONS_Semiliquid_B
dis:Us_CDB	1	dis:r13	GNOCCHI DI PATATE,VERZA E PECORINO@it	dis:CONS_Solid_D
dis:Us_CDB	1	dis:r58	CREMA PASTICCERA VELOCE@it	dis:CONS_Semiliquid_B
dis:Us_CDB	1	dis:r14	GNOCCHI DI RICOTTA "AIO OIO E BOTTARGA"@it	dis:CONS_Solid_D
dis:Us_CDB	1	dis:r55	MOUSSE DI ASPARAGI E FORMAGGIO@it	dis:CONS_Semisolid_C
dis:Us_CDB	1	dis:r11	GELATINA DI MOSCATO CON FRAGOLINE DI BOSCO@it	dis:CONS_Solid_D
dis:Us_CDB	1	dis:r56	PURÈ DI FAVE CON SCAROLA E PECORINO@it	dis:CONS_Semiliquid_B

dis:Us_CDB	1	dis:r12	GIRELLO DI VITELLO CON SALSA TONNATA@it	dis:CONS_Semisolid_C
dis:Us_CDB	1	dis:r53	MOUSSE DI TONNO E MELANZANE@it	dis:CONS_Semisolid_C
dis:Us_CDB	1	dis:r10	DOLCE CREMOSO ALLO YOGURT@it	dis:CONS_Semiliquid_B
dis:Us_CDB	1	dis:r54	MOUSSE DI GORGONZOLA@it	dis:CONS_Semisolid_C
dis:Us_CDB	1	dis:r17	LASAGNE BIANCHE DI MARE CON PESCE BIANCO, SEPPIA, GAMBERO E PESTO DI BASILICO@it	dis:CONS_Solid_D
dis:Us_CDB	1	dis:r18	LASAGNE DI CARNE ALLA MARCHIGIANA@it	dis:CONS_Solid_D
dis:Us_CDB	1	dis:r60	SEMIFREDDO ALLA PESCA@it	dis:CONS_Semisolid_C
dis:Us_CDB	1	dis:r26	PASSATA DI TOPINANBUR@it	dis:CONS_Semiliquid_B
dis:Us_CDB	1	dis:r27	PASSATELLI CON PUREA DI SEDANO RAPA,STRACCHINO E SALSA NOCI@it	dis:CONS_Solid_D
dis:Us_CDB	1	dis:r24	PANCOTTO "DEL PRETE"@it	dis:CONS_Semisolid_C
dis:Us_CDB	1	dis:r25	PANNA COTTA AL CARAMELLO@it	dis:CONS_Semisolid_C
dis:Us_CDB	1	dis:r22	MILKSHAKE BANANA E PISTACCHIO@it	dis:CONS_Semiliquid_A
dis:Us_CDB	1	dis:r23	MI-TIRI-SU!@it	dis:CONS_Solid_D
dis:Us_CDB	1	dis:r20	MELONE INVERNALE E ARANCIA@it	dis:CONS_Semiliquid_A
dis:Us_CDB	1	dis:r21	MERENDA IN CAMPAGNA CON FAVA E PECORINO@it	dis:CONS_Semisolid_C
dis:Us_CDB	1	dis:r28	PETTO DI TACCHINO BOLLITO CON SALSA VERDE@it	dis:CONS_Semisolid_C
dis:Us_CDB	1	dis:r29	POLLO ALLA GRIGLIA CON SALSA LIMONE WORCESTER E PEPERONE GRIGLIATO@it	dis:CONS_Semisolid_C
dis:Us_CDB	1	dis:r30	POLENTA DI MAIS CON BACCALÀ, OLIVE E CAPPERI E POMODORI DATTERINO@it	dis:CONS_Solid_D
dis:Us_CDB	1	dis:r37	VITELLONE ALLA CALABRESE@it	dis:CONS_Semisolid_C
dis:Us_CDB	1	dis:r38	ZUPPA INGLESE@it	dis:CONS_Solid_D
dis:Us_CDB	1	dis:r35	SPIGOLA CON SALSA DI POMODORI ARROSTO@it	dis:CONS_Semisolid_C
dis:Us_CDB	1	dis:r36	VERDURA AGRODOLCE@it	dis:CONS_Solid_D
dis:Us_CDB	1	dis:r33	QUENELLE DI FEGATO@it	dis:CONS_Semisolid_C
dis:Us_CDB	1	dis:r34	SPIGOLA AL VAPORE CON BIETOLA AL LIMONE@it	dis:CONS_Semisolid_C
dis:Us_CDB	1	dis:r31	POLPETTE AL SUGO DI POMODORO FRESCO@it	dis:CONS_Solid_D
dis:Us_CDB	1	dis:r32	POLPETTE DI CARNE@it	dis:CONS_Solid_D
dis:Us_CDB	1	dis:r39	LATTE E BISCOTTI PRIMA INFANZIA@it	dis:CONS_Semiliquid_B
dis:Us_CDB	1	dis:r40	THE E BISCOTTI PRIMA INFANZIA@it	dis:CONS_Semiliquid_B
dis:Us_CDB	1	dis:r41	SPREMUTA D'ARANCIA E BISCOTTI PRIMA INFANZIA@it	dis:CONS_Semiliquid_B
dis:Us_CDB	1	dis:r48	VELLUTATA DI ZUCCHINE E QUENELLE DI FORMAGGIO@it	dis:CONS_Semiliquid_B
dis:Us_CDB	1	dis:r04	COCKTAIL DI GAMBERI@it	dis:CONS_Semisolid_C
dis:Us_CDB	1	dis:r49	ZUPPA DI PORRI E PATATE@it	dis:CONS_Semiliquid_A
dis:Us_CDB	1	dis:r05	CONIGLIO ALLA GRIGLIA CON SALSA BBQ@it	dis:CONS_Semisolid_C

dis:Us_CDB	1	dis:r46	CREMA DI PATATE@it	dis:CONS_Semiliquid_B
dis:Us_CDB	1	dis:r02	CANNELLONI DI CRESPELLA CON PATATE,PORRO E SALSA POMODORO@it	dis:CONS_Solid_D
dis:Us_CDB	1	dis:r03	CARNE DI VITELLA IN PANZANELLA@it	dis:CONS_Semisolid_C
dis:Us_CDB	1	dis:r47	PASSATO DI CECI E FAGIOLI@it	dis:CONS_Semiliquid_B
dis:Us_CDB	1	dis:r44	PASSATO DI VERDURA SENZA POMODORO@it	dis:CONS_Semiliquid_A
dis:Us_CDB	1	dis:r45	PASSATO DI VERDURA@it	dis:CONS_Semiliquid_A
dis:Us_CDB	1	dis:r01	BIANCOMANGIARE AI LAMPONI@it	dis:CONS_Semisolid_C
dis:Us_CDB	1	dis:r42	FRULLATO DI BANANA E BISCOTTI@it	dis:CONS_Semiliquid_B
dis:Us_CDB	1	dis:r43	SEMOLINO AL LIMONE@it	dis:CONS_Semiliquid_B
dis:Us_CDB	1	dis:r08	CREMA DI CAROTE E ZUCCHINE@it	dis:CONS_Semiliquid_B
dis:Us_CDB	1	dis:r09	CREMA DI LEGUMI@it	dis:CONS_Semiliquid_B
dis:Us_CDB	1	dis:r06	CONIGLIO IN PORCHETTA@it	dis:CONS_Semisolid_C
dis:Us_CDB	1	dis:r07	CREM CARAMEL@it	dis:CONS_Semisolid_C
dis:Us_CDB	1	dis:r51	ZUPPA DI CAROTE E CECI@it	dis:CONS_Semiliquid_B
dis:Us_CDB	1	dis:r52	MOUSSE DI PROSCIUTTO@it	dis:CONS_Semisolid_C
dis:Us_CDB	1	dis:r50	PAPPA CON POMODORO@it	dis:CONS_Semisolid_C
dis:Us_MF	7	dis:r15	GNOCCHI DI SEMOLINO VERDE, ASPARAGO CON BURRO E SALVIA@it	dis:CONS_Solid_D
dis:Us_MF	7	dis:r59	CUPOLA DI YOGURT ALLA PESCA@it	dis:CONS_Semisolid_C
dis:Us_MF	7	dis:r16	LASAGNE ALLE VERDURE@it	dis:CONS_Solid_D
dis:Us_MF	7	dis:r57	CREMA PASTICCERA@it	dis:CONS_Semiliquid_B
dis:Us_MF	7	dis:r13	GNOCCHI DI PATATE,VERZA E PECORINO@it	dis:CONS_Solid_D
dis:Us_MF	7	dis:r58	CREMA PASTICCERA VELOCE@it	dis:CONS_Semiliquid_B
dis:Us_MF	7	dis:r14	GNOCCHI DI RICOTTA "AIO OIO E BOTTARGA"@it	dis:CONS_Solid_D
dis:Us_MF	7	dis:r55	MOUSSE DI ASPARAGI E FORMAGGIO@it	dis:CONS_Semisolid_C
dis:Us_MF	7	dis:r11	GELATINA DI MOSCATO CON FRAGOLINE DI BOSCO@it	dis:CONS_Solid_D
dis:Us_MF	7	dis:r56	PURÈ DI FAVE CON SCAROLA E PECORINO@it	dis:CONS_Semiliquid_B
dis:Us_MF	7	dis:r12	GIRELLO DI VITELLO CON SALSA TONNATA@it	dis:CONS_Semisolid_C
dis:Us_MF	7	dis:r53	MOUSSE DI TONNO E MELANZANE@it	dis:CONS_Semisolid_C
dis:Us_MF	7	dis:r10	DOLCE CREMOSO ALLO YOGURT@it	dis:CONS_Semiliquid_B
dis:Us_MF	7	dis:r54	MOUSSE DI GORGONZOLA@it	dis:CONS_Semisolid_C
dis:Us_MF	7	dis:r17	LASAGNE BIANCHE DI MARE CON PESCE BIANCO, SEPPIA, GAMBERO E PESTO DI BASILICO@it	dis:CONS_Solid_D
dis:Us_MF	7	dis:r18	LASAGNE DI CARNE ALLA MARCHIGIANA@it	dis:CONS_Solid_D
dis:Us_MF	7	dis:r60	SEMIFREDDO ALLA PESCA@it	dis:CONS_Semisolid_C

dis:Us_MF	7	dis:r26	PASSATA DI TOPINANBUR@it	dis:CONS_Semiliquid_B
dis:Us_MF	7	dis:r27	PASSATELLI CON PUREA DI SEDANO RAPA,STRACCHINO E SALSA NOCI@it	dis:CONS_Solid_D
dis:Us_MF	7	dis:r24	PANCOTTO "DEL PRETE"@it	dis:CONS_Semisolid_C
dis:Us_MF	7	dis:r25	PANNA COTTA AL CARAMELLO@it	dis:CONS_Semisolid_C
dis:Us_MF	7	dis:r22	MILKSHAKE BANANA E PISTACCHIO@it	dis:CONS_Semiliquid_A
dis:Us_MF	7	dis:r23	MI-TIRI-SU!@it	dis:CONS_Solid_D
dis:Us_MF	7	dis:r20	MELONE INVERNALE E ARANCIA@it	dis:CONS_Semiliquid_A
dis:Us_MF	7	dis:r21	MERENDA IN CAMPAGNA CON FAVA E PECORINO@it	dis:CONS_Semisolid_C
dis:Us_MF	7	dis:r28	PETTO DI TACCHINO BOLLITO CON SALSA VERDE@it	dis:CONS_Semisolid_C
dis:Us_MF	7	dis:r29	POLLO ALLA GRIGLIA CON SALSA LIMONE WORCESTER E PEPERONE GRIGLIATO@it	dis:CONS_Semisolid_C
dis:Us_MF	7	dis:r30	POLENTA DI MAIS CON BACCALÀ, OLIVE E CAPPERI E POMODORI DATTERINO@it	dis:CONS_Solid_D
dis:Us_MF	7	dis:r37	VITELLONE ALLA CALABRESE@it	dis:CONS_Semisolid_C
dis:Us_MF	7	dis:r38	ZUPPA INGLESE@it	dis:CONS_Solid_D
dis:Us_MF	7	dis:r35	SPIGOLA CON SALSA DI POMODORI ARROSTO@it	dis:CONS_Semisolid_C
dis:Us_MF	7	dis:r36	VERDURA AGRODOLCE@it	dis:CONS_Solid_D
dis:Us_MF	7	dis:r33	QUENELLE DI FEGATO@it	dis:CONS_Semisolid_C
dis:Us_MF	7	dis:r34	SPIGOLA AL VAPORE CON BIETOLA AL LIMONE@it	dis:CONS_Semisolid_C
dis:Us_MF	7	dis:r31	POLPETTE AL SUGO DI POMODORO FRESCO@it	dis:CONS_Solid_D
dis:Us_MF	7	dis:r32	POLPETTE DI CARNE@it	dis:CONS_Solid_D
dis:Us_MF	7	dis:r39	LATTE E BISCOTTI PRIMA INFANZIA@it	dis:CONS_Semiliquid_B
dis:Us_MF	7	dis:r40	THE E BISCOTTI PRIMA INFANZIA@it	dis:CONS_Semiliquid_B
dis:Us_MF	7	dis:r41	SPREMUTA D'ARANCIA E BISCOTTI PRIMA INFANZIA@it	dis:CONS_Semiliquid_B
dis:Us_MF	7	dis:r48	VELLUTATA DI ZUCCHINE E QUENELLE DI FORMAGGIO@it	dis:CONS_Semiliquid_B
dis:Us_MF	7	dis:r04	COCKTAIL DI GAMBERI@it	dis:CONS_Semisolid_C
dis:Us_MF	7	dis:r49	ZUPPA DI PORRI E PATATE@it	dis:CONS_Semiliquid_A
dis:Us_MF	7	dis:r05	CONIGLIO ALLA GRIGLIA CON SALSA BBQ@it	dis:CONS_Semisolid_C
dis:Us_MF	7	dis:r46	CREMA DI PATATE@it	dis:CONS_Semiliquid_B
dis:Us_MF	7	dis:r02	CANNELLONI DI CRESPELLA CON PATATE,PORRO E SALSA POMODORO@it	dis:CONS_Solid_D
dis:Us_MF	7	dis:r03	CARNE DI VITELLA IN PANZANELLA@it	dis:CONS_Semisolid_C
dis:Us_MF	7	dis:r47	PASSATO DI CECI E FAGIOLI@it	dis:CONS_Semiliquid_B
dis:Us_MF	7	dis:r44	PASSATO DI VERDURA SENZA POMODORO@it	dis:CONS_Semiliquid_A
dis:Us_MF	7	dis:r45	PASSATO DI VERDURA@it	dis:CONS_Semiliquid_A
dis:Us_MF	7	dis:r01	BIANCOMANGIARE AI LAMPONI@it	dis:CONS_Semisolid_C

dis:Us_MF	7	dis:r42	FRULLATO DI BANANA E BISCOTTI@it	dis:CONS_Semiliquid_B
dis:Us_MF	7	dis:r43	SEMOLINO AL LIMONE@it	dis:CONS_Semiliquid_B
dis:Us_MF	7	dis:r08	CREMA DI CAROTE E ZUCCHINE@it	dis:CONS_Semiliquid_B
dis:Us_MF	7	dis:r09	CREMA DI LEGUMI@it	dis:CONS_Semiliquid_B
dis:Us_MF	7	dis:r06	CONIGLIO IN PORCHETTA@it	dis:CONS_Semisolid_C
dis:Us_MF	7	dis:r07	CREM CARAMEL@it	dis:CONS_Semisolid_C
dis:Us_MF	7	dis:r51	ZUPPA DI CAROTE E CECI@it	dis:CONS_Semiliquid_B
dis:Us_MF	7	dis:r52	MOUSSE DI PROSCIUTTO@it	dis:CONS_Semisolid_C
dis:Us_MF	7	dis:r50	PAPPA CON POMODORO@it	dis:CONS_Semisolid_C
dis:Us_PA-LE	8	dis:r15	GNOCCHI DI SEMOLINO VERDE, ASPARAGO CON BURRO E SALVIA@it	dis:CONS_Solid_D
dis:Us_PA-LE	8	dis:r59	CUPOLA DI YOGURT ALLA PESCA@it	dis:CONS_Semisolid_C
dis:Us_PA-LE	8	dis:r16	LASAGNE ALLE VERDURE@it	dis:CONS_Solid_D
dis:Us_PA-LE	8	dis:r57	CREMA PASTICCERA@it	dis:CONS_Semiliquid_B
dis:Us_PA-LE	8	dis:r13	GNOCCHI DI PATATE,VERZA E PECORINO@it	dis:CONS_Solid_D
dis:Us_PA-LE	8	dis:r58	CREMA PASTICCERA VELOCE@it	dis:CONS_Semiliquid_B
dis:Us_PA-LE	8	dis:r14	GNOCCHI DI RICOTTA "AIO OIO E BOTTARGA"@it	dis:CONS_Solid_D
dis:Us_PA-LE	8	dis:r55	MOUSSE DI ASPARAGI E FORMAGGIO@it	dis:CONS_Semisolid_C
dis:Us_PA-LE	8	dis:r11	GELATINA DI MOSCATO CON FRAGOLINE DI BOSCO@it	dis:CONS_Solid_D
dis:Us_PA-LE	8	dis:r56	PURÈ DI FAVE CON SCAROLA E PECORINO@it	dis:CONS_Semiliquid_B
dis:Us_PA-LE	8	dis:r12	GIRELLO DI VITELLO CON SALSA TONNATA@it	dis:CONS_Semisolid_C
dis:Us_PA-LE	8	dis:r53	MOUSSE DI TONNO E MELANZANE@it	dis:CONS_Semisolid_C
dis:Us_PA-LE	8	dis:r10	DOLCE CREMOSO ALLO YOGURT@it	dis:CONS_Semiliquid_B
dis:Us_PA-LE	8	dis:r54	MOUSSE DI GORGONZOLA@it	dis:CONS_Semisolid_C
dis:Us_PA-LE	8	dis:r17	LASAGNE BIANCHE DI MARE CON PESCE BIANCO, SEPPIA, GAMBERO E PESTO DI BASILICO@it	dis:CONS_Solid_D
dis:Us_PA-LE	8	dis:r18	LASAGNE DI CARNE ALLA MARCHIGIANA@it	dis:CONS_Solid_D
dis:Us_PA-LE	8	dis:r60	SEMIFREDDO ALLA PESCA@it	dis:CONS_Semisolid_C
dis:Us_PA-LE	8	dis:r26	PASSATA DI TOPINANBUR@it	dis:CONS_Semiliquid_B
dis:Us_PA-LE	8	dis:r27	PASSATELLI CON PUREA DI SEDANO RAPA,STRACCHINO E SALSA NOCI@it	dis:CONS_Solid_D
dis:Us_PA-LE	8	dis:r24	PANCOTTO "DEL PRETE"@it	dis:CONS_Semisolid_C
dis:Us_PA-LE	8	dis:r25	PANNA COTTA AL CARAMELLO@it	dis:CONS_Semisolid_C
dis:Us_PA-LE	8	dis:r22	MILKSHAKE BANANA E PISTACCHIO@it	dis:CONS_Semiliquid_A
dis:Us_PA-LE	8	dis:r23	MI-TIRI-SU!@it	dis:CONS_Solid_D
dis:Us_PA-LE	8	dis:r20	MELONE INVERNALE E ARANCIA@it	dis:CONS_Semiliquid_A

dis:Us_PA-LE	8	dis:r21	MERENDA IN CAMPAGNA CON FAVA E PECORINO@it	dis:CONS_Semisolid_C
dis:Us_PA-LE	8	dis:r28	PETTO DI TACCHINO BOLLITO CON SALSA VERDE@it	dis:CONS_Semisolid_C
dis:Us_PA-LE	8	dis:r29	POLLO ALLA GRIGLIA CON SALSA LIMONE WORCESTER E PEPERONE GRIGLIATO@it	dis:CONS_Semisolid_C
dis:Us_PA-LE	8	dis:r30	POLENTA DI MAIS CON BACCALÀ, OLIVE E CAPPERI E POMODORI DATTERINO@it	dis:CONS_Solid_D
dis:Us_PA-LE	8	dis:r37	VITELLONE ALLA CALABRESE@it	dis:CONS_Semisolid_C
dis:Us_PA-LE	8	dis:r38	ZUPPA INGLESE@it	dis:CONS_Solid_D
dis:Us_PA-LE	8	dis:r35	SPIGOLA CON SALSA DI POMODORI ARROSTO@it	dis:CONS_Semisolid_C
dis:Us_PA-LE	8	dis:r36	VERDURA AGRODOLCE@it	dis:CONS_Solid_D
dis:Us_PA-LE	8	dis:r33	QUENELLE DI FEGATO@it	dis:CONS_Semisolid_C
dis:Us_PA-LE	8	dis:r34	SPIGOLA AL VAPORE CON BIETOLA AL LIMONE@it	dis:CONS_Semisolid_C
dis:Us_PA-LE	8	dis:r31	POLPETTE AL SUGO DI POMODORO FRESCO@it	dis:CONS_Solid_D
dis:Us_PA-LE	8	dis:r32	POLPETTE DI CARNE@it	dis:CONS_Solid_D
dis:Us_PA-LE	8	dis:r39	LATTE E BISCOTTI PRIMA INFANZIA@it	dis:CONS_Semiliquid_B
dis:Us_PA-LE	8	dis:r40	THE E BISCOTTI PRIMA INFANZIA@it	dis:CONS_Semiliquid_B
dis:Us_PA-LE	8	dis:r41	SPREMUTA D'ARANCIA E BISCOTTI PRIMA INFANZIA@it	dis:CONS_Semiliquid_B
dis:Us_PA-LE	8	dis:r48	VELLUTATA DI ZUCCHINE E QUENELLE DI FORMAGGIO@it	dis:CONS_Semiliquid_B
dis:Us_PA-LE	8	dis:r04	COCKTAIL DI GAMBERI@it	dis:CONS_Semisolid_C
dis:Us_PA-LE	8	dis:r49	ZUPPA DI PORRI E PATATE@it	dis:CONS_Semiliquid_A
dis:Us_PA-LE	8	dis:r05	CONIGLIO ALLA GRIGLIA CON SALSA BBQ@it	dis:CONS_Semisolid_C
dis:Us_PA-LE	8	dis:r46	CREMA DI PATATE@it	dis:CONS_Semiliquid_B
dis:Us_PA-LE	8	dis:r02	CANNELLONI DI CRESPELLA CON PATATE,PORRO E SALSA POMODORO@it	dis:CONS_Solid_D
dis:Us_PA-LE	8	dis:r03	CARNE DI VITELLA IN PANZANELLA@it	dis:CONS_Semisolid_C
dis:Us_PA-LE	8	dis:r47	PASSATO DI CECI E FAGIOLI@it	dis:CONS_Semiliquid_B
dis:Us_PA-LE	8	dis:r44	PASSATO DI VERDURA SENZA POMODORO@it	dis:CONS_Semiliquid_A
dis:Us_PA-LE	8	dis:r45	PASSATO DI VERDURA@it	dis:CONS_Semiliquid_A
dis:Us_PA-LE	8	dis:r01	BIANCOMANGIARE AI LAMPONI@it	dis:CONS_Semisolid_C
dis:Us_PA-LE	8	dis:r42	FRULLATO DI BANANA E BISCOTTI@it	dis:CONS_Semiliquid_B
dis:Us_PA-LE	8	dis:r43	SEMOLINO AL LIMONE@it	dis:CONS_Semiliquid_B
dis:Us_PA-LE	8	dis:r08	CREMA DI CAROTE E ZUCCHINE@it	dis:CONS_Semiliquid_B
dis:Us_PA-LE	8	dis:r09	CREMA DI LEGUMI@it	dis:CONS_Semiliquid_B
dis:Us_PA-LE	8	dis:r06	CONIGLIO IN PORCHETTA@it	dis:CONS_Semisolid_C
dis:Us_PA-LE	8	dis:r07	CREM CARAMEL@it	dis:CONS_Semisolid_C
dis:Us_PA-LE	8	dis:r51	ZUPPA DI CAROTE E CECI@it	dis:CONS_Semiliquid_B

dis:Us_PA-LE	8	dis:r52	MOUSSE DI PROSCIUTTO@it	dis:CONS_Semisolid_C
dis:Us_PA-LE	8	dis:r50	PAPPA CON POMODORO@it	dis:CONS_Semisolid_C
dis:Us_ROPRO	9	dis:r15	GNOCCHI DI SEMOLINO VERDE, ASPARAGO CON BURRO E SALVIA@it	dis:CONS_Solid_D
dis:Us_ROPRO	9	dis:r59	CUPOLA DI YOGURT ALLA PESCA@it	dis:CONS_Semisolid_C
dis:Us_ROPRO	9	dis:r16	LASAGNE ALLE VERDURE@it	dis:CONS_Solid_D
dis:Us_ROPRO	9	dis:r57	CREMA PASTICCERA@it	dis:CONS_Semiliquid_B
dis:Us_ROPRO	9	dis:r13	GNOCCHI DI PATATE,VERZA E PECORINO@it	dis:CONS_Solid_D
dis:Us_ROPRO	9	dis:r58	CREMA PASTICCERA VELOCE@it	dis:CONS_Semiliquid_B
dis:Us_ROPRO	9	dis:r14	GNOCCHI DI RICOTTA "AIO OIO E BOTTARGA"@it	dis:CONS_Solid_D
dis:Us_ROPRO	9	dis:r55	MOUSSE DI ASPARAGI E FORMAGGIO@it	dis:CONS_Semisolid_C
dis:Us_ROPRO	9	dis:r11	GELATINA DI MOSCATO CON FRAGOLINE DI BOSCO@it	dis:CONS_Solid_D
dis:Us_ROPRO	9	dis:r56	PURÈ DI FAVE CON SCAROLA E PECORINO@it	dis:CONS_Semiliquid_B
dis:Us_ROPRO	9	dis:r12	GIRELLO DI VITELLO CON SALSA TONNATA@it	dis:CONS_Semisolid_C
dis:Us_ROPRO	9	dis:r53	MOUSSE DI TONNO E MELANZANE@it	dis:CONS_Semisolid_C
dis:Us_ROPRO	9	dis:r10	DOLCE CREMOSO ALLO YOGURT@it	dis:CONS_Semiliquid_B
dis:Us_ROPRO	9	dis:r54	MOUSSE DI GORGONZOLA@it	dis:CONS_Semisolid_C
dis:Us_ROPRO	9	dis:r17	LASAGNE BIANCHE DI MARE CON PESCE BIANCO, SEPPIA, GAMBERO E PESTO DI BASILICO@it	dis:CONS_Solid_D
dis:Us_ROPRO	9	dis:r18	LASAGNE DI CARNE ALLA MARCHIGIANA@it	dis:CONS_Solid_D
dis:Us_ROPRO	9	dis:r60	SEMIFREDDO ALLA PESCA@it	dis:CONS_Semisolid_C
dis:Us_ROPRO	9	dis:r26	PASSATA DI TOPINANBUR@it	dis:CONS_Semiliquid_B
dis:Us_ROPRO	9	dis:r27	PASSATELLI CON PUREA DI SEDANO RAPA,STRACCHINO E SALSA NOCI@it	dis:CONS_Solid_D
dis:Us_ROPRO	9	dis:r24	PANCOTTO "DEL PRETE"@it	dis:CONS_Semisolid_C
dis:Us_ROPRO	9	dis:r25	PANNA COTTA AL CARAMELLO@it	dis:CONS_Semisolid_C
dis:Us_ROPRO	9	dis:r22	MILKSHAKE BANANA E PISTACCHIO@it	dis:CONS_Semiliquid_A
dis:Us_ROPRO	9	dis:r23	MI-TIRI-SU!@it	dis:CONS_Solid_D
dis:Us_ROPRO	9	dis:r20	MELONE INVERNALE E ARANCIA@it	dis:CONS_Semiliquid_A
dis:Us_ROPRO	9	dis:r21	MERENDA IN CAMPAGNA CON FAVA E PECORINO@it	dis:CONS_Semisolid_C
dis:Us_ROPRO	9	dis:r28	PETTO DI TACCHINO BOLLITO CON SALSA VERDE@it	dis:CONS_Semisolid_C
dis:Us_ROPRO	9	dis:r29	POLLO ALLA GRIGLIA CON SALSA LIMONE WORCESTER E PEPERONE GRIGLIATO@it	dis:CONS_Semisolid_C
dis:Us_ROPRO	9	dis:r30	POLENTA DI MAIS CON BACCALÀ, OLIVE E CAPPERI E POMODORI DATTERINO@it	dis:CONS_Solid_D
dis:Us_ROPRO	9	dis:r37	VITELLONE ALLA CALABRESE@it	dis:CONS_Semisolid_C
dis:Us_ROPRO	9	dis:r38	ZUPPA INGLESE@it	dis:CONS_Solid_D
dis:Us_ROPRO	9	dis:r35	SPIGOLA CON SALSA DI POMODORI ARROSTO@it	dis:CONS_Semisolid_C

dis:Us_ROPRO	9	dis:r36	VERDURA AGRODOLCE@it	dis:CONS_Solid_D
dis:Us_ROPRO	9	dis:r33	QUENELLE DI FEGATO@it	dis:CONS_Semisolid_C
dis:Us_ROPRO	9	dis:r34	SPIGOLA AL VAPORE CON BIETOLA AL LIMONE@it	dis:CONS_Semisolid_C
dis:Us_ROPRO	9	dis:r31	POLPETTE AL SUGO DI POMODORO FRESCO@it	dis:CONS_Solid_D
dis:Us_ROPRO	9	dis:r32	POLPETTE DI CARNE@it	dis:CONS_Solid_D
dis:Us_ROPRO	9	dis:r39	LATTE E BISCOTTI PRIMA INFANZIA@it	dis:CONS_Semiliquid_B
dis:Us_ROPRO	9	dis:r40	THE E BISCOTTI PRIMA INFANZIA@it	dis:CONS_Semiliquid_B
dis:Us_ROPRO	9	dis:r41	SPREMUTA D'ARANCIA E BISCOTTI PRIMA INFANZIA@it	dis:CONS_Semiliquid_B
dis:Us_ROPRO	9	dis:r48	VELLUTATA DI ZUCCHINE E QUENELLE DI FORMAGGIO@it	dis:CONS_Semiliquid_B
dis:Us_ROPRO	9	dis:r04	COCKTAIL DI GAMBERI@it	dis:CONS_Semisolid_C
dis:Us_ROPRO	9	dis:r49	ZUPPA DI PORRI E PATATE@it	dis:CONS_Semiliquid_A
dis:Us_ROPRO	9	dis:r05	CONIGLIO ALLA GRIGLIA CON SALSA BBQ@it	dis:CONS_Semisolid_C
dis:Us_ROPRO	9	dis:r46	CREMA DI PATATE@it	dis:CONS_Semiliquid_B
dis:Us_ROPRO	9	dis:r02	CANNELLONI DI CRESPELLA CON PATATE,PORRO E SALSA POMODORO@it	dis:CONS_Solid_D
dis:Us_ROPRO	9	dis:r03	CARNE DI VITELLA IN PANZANELLA@it	dis:CONS_Semisolid_C
dis:Us_ROPRO	9	dis:r47	PASSATO DI CECI E FAGIOLI@it	dis:CONS_Semiliquid_B
dis:Us_ROPRO	9	dis:r44	PASSATO DI VERDURA SENZA POMODORO@it	dis:CONS_Semiliquid_A
dis:Us_ROPRO	9	dis:r45	PASSATO DI VERDURA@it	dis:CONS_Semiliquid_A
dis:Us_ROPRO	9	dis:r01	BIANCOMANGIARE AI LAMPONI@it	dis:CONS_Semisolid_C
dis:Us_ROPRO	9	dis:r42	FRULLATO DI BANANA E BISCOTTI@it	dis:CONS_Semiliquid_B
dis:Us_ROPRO	9	dis:r43	SEMOLINO AL LIMONE@it	dis:CONS_Semiliquid_B
dis:Us_ROPRO	9	dis:r08	CREMA DI CAROTE E ZUCCHINE@it	dis:CONS_Semiliquid_B
dis:Us_ROPRO	9	dis:r09	CREMA DI LEGUMI@it	dis:CONS_Semiliquid_B
dis:Us_ROPRO	9	dis:r06	CONIGLIO IN PORCHETTA@it	dis:CONS_Semisolid_C
dis:Us_ROPRO	9	dis:r07	CREM CARAMEL@it	dis:CONS_Semisolid_C
dis:Us_ROPRO	9	dis:r51	ZUPPA DI CAROTE E CECI@it	dis:CONS_Semiliquid_B
dis:Us_ROPRO	9	dis:r52	MOUSSE DI PROSCIUTTO@it	dis:CONS_Semisolid_C
dis:Us_ROPRO	9	dis:r50	PAPPA CON POMODORO@it	dis:CONS_Semisolid_C
dis:Us_GIO-PA	6	dis:r15	GNOCCHI DI SEMOLINO VERDE, ASPARAGO CON BURRO E SALVIA@it	dis:CONS_Solid_D
dis:Us_GIO-PA	6	dis:r59	CUPOLA DI YOGURT ALLA PESCA@it	dis:CONS_Semisolid_C
dis:Us_GIO-PA	6	dis:r16	LASAGNE ALLE VERDURE@it	dis:CONS_Solid_D
dis:Us_GIO-PA	6	dis:r57	CREMA PASTICCERA@it	dis:CONS_Semiliquid_B
dis:Us_GIO-PA	6	dis:r13	GNOCCHI DI PATATE,VERZA E PECORINO@it	dis:CONS_Solid_D

dis:Us_GIO-PA	6	dis:r58	CREMA PASTICCERA VELOCE@it	dis:CONS_Semiliquid_B
dis:Us_GIO-PA	6	dis:r14	GNOCCHI DI RICOTTA "AIO OIO E BOTTARGA"@it	dis:CONS_Solid_D
dis:Us_GIO-PA	6	dis:r55	MOUSSE DI ASPARAGI E FORMAGGIO@it	dis:CONS_Semisolid_C
dis:Us_GIO-PA	6	dis:r11	GELATINA DI MOSCATO CON FRAGOLINE DI BOSCO@it	dis:CONS_Solid_D
dis:Us_GIO-PA	6	dis:r56	PURÈ DI FAVE CON SCAROLA E PECORINO@it	dis:CONS_Semiliquid_B
dis:Us_GIO-PA	6	dis:r12	GIRELLO DI VITELLO CON SALSA TONNATA@it	dis:CONS_Semisolid_C
dis:Us_GIO-PA	6	dis:r53	MOUSSE DI TONNO E MELANZANE@it	dis:CONS_Semisolid_C
dis:Us_GIO-PA	6	dis:r10	DOLCE CREMOSO ALLO YOGURT@it	dis:CONS_Semiliquid_B
dis:Us_GIO-PA	6	dis:r54	MOUSSE DI GORGONZOLA@it	dis:CONS_Semisolid_C
dis:Us_GIO-PA	6	dis:r17	LASAGNE BIANCHE DI MARE CON PESCE BIANCO, SEPPIA, GAMBERO E PESTO DI BASILICO@it	dis:CONS_Solid_D
dis:Us_GIO-PA	6	dis:r18	LASAGNE DI CARNE ALLA MARCHIGIANA@it	dis:CONS_Solid_D
dis:Us_GIO-PA	6	dis:r60	SEMIFREDDO ALLA PESCA@it	dis:CONS_Semisolid_C
dis:Us_GIO-PA	6	dis:r26	PASSATA DI TOPINANBUR@it	dis:CONS_Semiliquid_B
dis:Us_GIO-PA	6	dis:r27	PASSATELLI CON PUREA DI SEDANO RAPA,STRACCHINO E SALSA NOCI@it	dis:CONS_Solid_D
dis:Us_GIO-PA	6	dis:r24	PANCOTTO "DEL PRETE"@it	dis:CONS_Semisolid_C
dis:Us_GIO-PA	6	dis:r25	PANNA COTTA AL CARAMELLO@it	dis:CONS_Semisolid_C
dis:Us_GIO-PA	6	dis:r22	MILKSHAKE BANANA E PISTACCHIO@it	dis:CONS_Semiliquid_A
dis:Us_GIO-PA	6	dis:r23	MI-TIRI-SU!@it	dis:CONS_Solid_D
dis:Us_GIO-PA	6	dis:r20	MELONE INVERNALE E ARANCIA@it	dis:CONS_Semiliquid_A
dis:Us_GIO-PA	6	dis:r21	MERENDA IN CAMPAGNA CON FAVA E PECORINO@it	dis:CONS_Semisolid_C
dis:Us_GIO-PA	6	dis:r28	PETTO DI TACCHINO BOLLITO CON SALSA VERDE@it	dis:CONS_Semisolid_C
dis:Us_GIO-PA	6	dis:r29	POLLO ALLA GRIGLIA CON SALSA LIMONE WORCESTER E PEPERONE GRIGLIATO@it	dis:CONS_Semisolid_C
dis:Us_GIO-PA	6	dis:r30	POLENTA DI MAIS CON BACCALÀ, OLIVE E CAPPERI E POMODORI DATTERINO@it	dis:CONS_Solid_D
dis:Us_GIO-PA	6	dis:r37	VITELLONE ALLA CALABRESE@it	dis:CONS_Semisolid_C
dis:Us_GIO-PA	6	dis:r38	ZUPPA INGLESE@it	dis:CONS_Solid_D
dis:Us_GIO-PA	6	dis:r35	SPIGOLA CON SALSA DI POMODORI ARROSTO@it	dis:CONS_Semisolid_C
dis:Us_GIO-PA	6	dis:r36	VERDURA AGRODOLCE@it	dis:CONS_Solid_D
dis:Us_GIO-PA	6	dis:r33	QUENELLE DI FEGATO@it	dis:CONS_Semisolid_C
dis:Us_GIO-PA	6	dis:r34	SPIGOLA AL VAPORE CON BIETOLA AL LIMONE@it	dis:CONS_Semisolid_C
dis:Us_GIO-PA	6	dis:r31	POLPETTE AL SUGO DI POMODORO FRESCO@it	dis:CONS_Solid_D
dis:Us_GIO-PA	6	dis:r32	POLPETTE DI CARNE@it	dis:CONS_Solid_D
dis:Us_GIO-PA	6	dis:r39	LATTE E BISCOTTI PRIMA INFANZIA@it	dis:CONS_Semiliquid_B
dis:Us_GIO-PA	6	dis:r40	THE E BISCOTTI PRIMA INFANZIA@it	dis:CONS_Semiliquid_B

dis:Us_GIO-PA	6	dis:r41	SPREMUTA D'ARANCIA E BISCOTTI PRIMA INFANZIA@it	dis:CONS_Semiliquid_B
dis:Us_GIO-PA	6	dis:r48	VELLUTATA DI ZUCCHINE E QUENELLE DI FORMAGGIO@it	dis:CONS_Semiliquid_B
dis:Us_GIO-PA	6	dis:r04	COCKTAIL DI GAMBERI@it	dis:CONS_Semisolid_C
dis:Us_GIO-PA	6	dis:r49	ZUPPA DI PORRI E PATATE@it	dis:CONS_Semiliquid_A
dis:Us_GIO-PA	6	dis:r05	CONIGLIO ALLA GRIGLIA CON SALSA BBQ@it	dis:CONS_Semisolid_C
dis:Us_GIO-PA	6	dis:r46	CREMA DI PATATE@it	dis:CONS_Semiliquid_B
dis:Us_GIO-PA	6	dis:r02	CANNELLONI DI CRESPELLA CON PATATE,PORRO E SALSA POMODORO@it	dis:CONS_Solid_D
dis:Us_GIO-PA	6	dis:r03	CARNE DI VITELLA IN PANZANELLA@it	dis:CONS_Semisolid_C
dis:Us_GIO-PA	6	dis:r47	PASSATO DI CECI E FAGIOLI@it	dis:CONS_Semiliquid_B
dis:Us_GIO-PA	6	dis:r44	PASSATO DI VERDURA SENZA POMODORO@it	dis:CONS_Semiliquid_A
dis:Us_GIO-PA	6	dis:r45	PASSATO DI VERDURA@it	dis:CONS_Semiliquid_A
dis:Us_GIO-PA	6	dis:r01	BIANCOMANGIARE AI LAMPONI@it	dis:CONS_Semisolid_C
dis:Us_GIO-PA	6	dis:r42	FRULLATO DI BANANA E BISCOTTI@it	dis:CONS_Semiliquid_B
dis:Us_GIO-PA	6	dis:r43	SEMOLINO AL LIMONE@it	dis:CONS_Semiliquid_B
dis:Us_GIO-PA	6	dis:r08	CREMA DI CAROTE E ZUCCHINE@it	dis:CONS_Semiliquid_B
dis:Us_GIO-PA	6	dis:r09	CREMA DI LEGUMI@it	dis:CONS_Semiliquid_B
dis:Us_GIO-PA	6	dis:r06	CONIGLIO IN PORCHETTA@it	dis:CONS_Semisolid_C
dis:Us_GIO-PA	6	dis:r07	CREM CARAMEL@it	dis:CONS_Semisolid_C
dis:Us_GIO-PA	6	dis:r51	ZUPPA DI CAROTE E CECI@it	dis:CONS_Semiliquid_B
dis:Us_GIO-PA	6	dis:r52	MOUSSE DI PROSCIUTTO@it	dis:CONS_Semisolid_C
dis:Us_GIO-PA	6	dis:r50	PAPPA CON POMODORO@it	dis:CONS_Semisolid_C
dis:Us_SOCE	11	dis:r26	PASSATA DI TOPINANBUR@it	dis:CONS_Semiliquid_B
dis:Us_SOCE	11	dis:r48	VELLUTATA DI ZUCCHINE E QUENELLE DI FORMAGGIO@it	dis:CONS_Semiliquid_B
dis:Us_SOCE	11	dis:r49	ZUPPA DI PORRI E PATATE@it	dis:CONS_Semiliquid_A
dis:Us_SOCE	11	dis:r57	CREMA PASTICCERA@it	dis:CONS_Semiliquid_B
dis:Us_SOCE	11	dis:r46	CREMA DI PATATE@it	dis:CONS_Semiliquid_B
dis:Us_SOCE	11	dis:r58	CREMA PASTICCERA VELOCE@it	dis:CONS_Semiliquid_B
dis:Us_SOCE	11	dis:r47	PASSATO DI CECI E FAGIOLI@it	dis:CONS_Semiliquid_B
dis:Us_SOCE	11	dis:r44	PASSATO DI VERDURA SENZA POMODORO@it	dis:CONS_Semiliquid_A
dis:Us_SOCE	11	dis:r22	MILKSHAKE BANANA E PISTACCHIO@it	dis:CONS_Semiliquid_A
dis:Us_SOCE	11	dis:r56	PURÈ DI FAVE CON SCAROLA E PECORINO@it	dis:CONS_Semiliquid_B
dis:Us_SOCE	11	dis:r45	PASSATO DI VERDURA@it	dis:CONS_Semiliquid_A
dis:Us_SOCE	11	dis:r42	FRULLATO DI BANANA E BISCOTTI@it	dis:CONS_Semiliquid_B

dis:Us_SOCE	11	dis:r20	MELONE INVERNALE E ARANCIA@it	dis:CONS_Semiliquid_A
dis:Us_SOCE	11	dis:r10	DOLCE CREMOSO ALLO YOGURT@it	dis:CONS_Semiliquid_B
dis:Us_SOCE	11	dis:r43	SEMOLINO AL LIMONE@it	dis:CONS_Semiliquid_B
dis:Us_SOCE	11	dis:r08	CREMA DI CAROTE E ZUCCHINE@it	dis:CONS_Semiliquid_B
dis:Us_SOCE	11	dis:r09	CREMA DI LEGUMI@it	dis:CONS_Semiliquid_B
dis:Us_SOCE	11	dis:r39	LATTE E BISCOTTI PRIMA INFANZIA@it	dis:CONS_Semiliquid_B
dis:Us_SOCE	11	dis:r51	ZUPPA DI CAROTE E CECI@it	dis:CONS_Semiliquid_B
dis:Us_SOCE	11	dis:r40	THE E BISCOTTI PRIMA INFANZIA@it	dis:CONS_Semiliquid_B
dis:Us_SOCE	11	dis:r41	SPREMUTA D'ARANCIA E BISCOTTI PRIMA INFANZIA@it	dis:CONS_Semiliquid_B

Appendix 2. The results of a SPARQL query indicating for each (eligible) patient the list of recipes that were inferred as safe (both under consistency and severity perspective), with the indication of the recipe name and consistency. Clinical personnel, knowing the health condition of the patients, evaluated each tuple for suitability. Clinical personnel concluded that the recipes suggested for each patient are suitable according to their health condition.

?id	?status	?antrPhen	?copdstage	?nri
001	copd:Cachectic	copd:Underweight	copd:Stage1	copd:Severe_Risk
001	copd:Diagnosed_Sarcopenic	copd:Underweight	copd:Stage1	copd:Severe_Risk
001	copd:Female	copd:Underweight	copd:Stage1	copd:Severe_Risk
001	copd:Probable_Sarcopenic	copd:Underweight	copd:Stage1	copd:Severe_Risk
001	copd:Sarcopenic	copd:Underweight	copd:Stage1	copd:Severe_Risk
001	copd:Severe_Sarcopenic	copd:Underweight	copd:Stage1	copd:Severe_Risk
002	copd:Cachectic	copd:Underweight	copd:Stage2	copd:Severe_Risk
002	copd:Diagnosed_Sarcopenic	copd:Underweight	copd:Stage2	copd:Severe_Risk
002	copd:Female	copd:Underweight	copd:Stage2	copd:Severe_Risk
002	copd:Probable_Sarcopenic	copd:Underweight	copd:Stage2	copd:Severe_Risk
002	copd:Sarcopenic	copd:Underweight	copd:Stage2	copd:Severe_Risk
002	copd:Severe_Sarcopenic	copd:Underweight	copd:Stage2	copd:Severe_Risk
003	copd:Cachectic	copd:Underweight	copd:Stage2	copd:Severe_Risk
003	copd:Diagnosed_Sarcopenic	copd:Underweight	copd:Stage2	copd:Severe_Risk
003	copd:Female	copd:Underweight	copd:Stage2	copd:Severe_Risk
003	copd:Probable_Sarcopenic	copd:Underweight	copd:Stage2	copd:Severe_Risk
003	copd:Sarcopenic	copd:Underweight	copd:Stage2	copd:Severe_Risk
003	copd:Severe_Sarcopenic	copd:Underweight	copd:Stage2	copd:Severe_Risk
BB	copd:Female	copd:Normal_Weight	copd:Stage4	copd:Absence_of_Risk
BB	copd:non-Cachetic	copd:Normal_Weight	copd:Stage4	copd:Absence_of_Risk
BB	copd:non-Sarcopenic	copd:Normal_Weight	copd:Stage4	copd:Absence_of_Risk
CV	copd:Male	copd:Obesity_1st_Degree	copd:Stage3	copd:Mild_Risk
CV	copd:non-Cachetic	copd:Obesity_1st_Degree	copd:Stage3	copd:Mild_Risk
CV	copd:non-Sarcopenic	copd:Obesity_1st_Degree	copd:Stage3	copd:Mild_Risk
DMP	copd:Female	copd:Obesity_2nd_Degree	copd:Stage3	copd:Moderate_Risk
DMP	copd:non-Cachetic	copd:Obesity_2nd_Degree	copd:Stage3	copd:Moderate_Risk
DMP	copd:non-Sarcopenic	copd:Obesity_2nd_Degree	copd:Stage3	copd:Moderate_Risk
FA	copd:Female	copd:Underweight	copd:Stage2	copd:Absence_of_Risk
FA	copd:non-Cachetic	copd:Underweight	copd:Stage2	copd:Absence_of_Risk
FA	copd:non-Sarcopenic	copd:Underweight	copd:Stage2	copd:Absence_of_Risk
FG	copd:Male	copd:Obesity_2nd_Degree	copd:Stage2	copd:Moderate_Risk
FG	copd:non-Cachetic	copd:Obesity_2nd_Degree	copd:Stage2	copd:Moderate_Risk
FG	copd:non-Sarcopenic	copd:Obesity_2nd_Degree	copd:Stage2	copd:Moderate_Risk
HB	copd:Female	copd:Normal_Weight	copd:Stage2	copd:Absence_of_Risk
HB	copd:non-Cachetic	copd:Normal_Weight	copd:Stage2	copd:Absence_of_Risk
HB	copd:non-Sarcopenic	copd:Normal_Weight	copd:Stage2	copd:Absence_of_Risk
LA	copd:Diagnosed_Sarcopenic	copd:Underweight	copd:Stage3	copd:Moderate_Risk
LA	copd:Female	copd:Underweight	copd:Stage3	copd:Moderate_Risk
LA	copd:Probable_Sarcopenic	copd:Underweight	copd:Stage3	copd:Moderate_Risk
LA	copd:Sarcopenic	copd:Underweight	copd:Stage3	copd:Moderate_Risk
LA	copd:non-Cachetic	copd:Underweight	copd:Stage3	copd:Moderate_Risk
MM	copd:Male	copd:Overweight	copd:Stage2	copd:Absence_of_Risk
MM	copd:non-Cachetic	copd:Overweight	copd:Stage2	copd:Absence_of_Risk
MM	copd:non-Sarcopenic	copd:Overweight	copd:Stage2	copd:Absence_of_Risk
SN	copd:Female	copd:Obesity_1st_Degree	copd:Stage3	copd:Absence_of_Risk
SN	copd:non-Cachetic	copd:Obesity_1st_Degree	copd:Stage3	copd:Absence_of_Risk
SN	copd:non-Sarcopenic	copd:Obesity_1st_Degree	copd:Stage3	copd:Absence_of_Risk

SP	copd:Female	copd:Obesity_1st_Degree	copd:Stage3	copd:Mild_Risk
SP	copd:non-Cachetic	copd:Obesity_1st_Degree	copd:Stage3	copd:Mild_Risk
SP	copd:non-Sarcopenic	copd:Obesity_1st_Degree	copd:Stage3	copd:Mild_Risk
TC	copd:Female	copd:Overweight	copd:Stage3	copd:Absence_of_Risk
TC	copd:non-Cachetic	copd:Overweight	copd:Stage3	copd:Absence_of_Risk
TC	copd:non-Sarcopenic	copd:Overweight	copd:Stage3	copd:Absence_of_Risk
TM	copd:Male	copd:Overweight	copd:Stage1	copd:Absence_of_Risk
TM	copd:non-Cachetic	copd:Overweight	copd:Stage1	copd:Absence_of_Risk
TM	copd:non-Sarcopenic	copd:Overweight	copd:Stage1	copd:Absence_of_Risk
VM	copd:Female	copd:Normal_Weight	copd:Stage4	copd:Absence_of_Risk
VM	copd:non-Cachetic	copd:Normal_Weight	copd:Stage4	copd:Absence_of_Risk
VM	copd:non-Sarcopenic	copd:Normal_Weight	copd:Stage4	copd:Absence_of_Risk

Appendix 3. A table reporting the full list of patients used for the testing of the COPD & Nutrition ontology ontology, with the indication of their status (whether or not they are sarcopenic – and in which severity – or cachectic) and their health condition, including the Nutritional Risk Index profile inferred.

?id	?rec	?bmr	?kcalInt	?prot	?minCar	?maxCar	?sugar	?minFat	?maxFat	?satFat	?col	?minFib	?maxFib	?bcaa
001	copd:Rec_Pat_XY1	1010	1818	25.0	45.0	50.0	14.9	35	50	9.9	300	22.90	25.0	yes
002	copd:Rec_Pat_XY2	1030	1854	25.0	45.0	50.0	14.9	30	35	9.9	300	23.36	25.0	yes
003	copd:Rec_Pat_XY3	1202	2164	25.0	45.0	50.0	14.9	30	35	9.9	300	25.0	25.0	yes
BB	copd:Rec_Pat_BB	946	1703	12.11	45.0	50.0	14.9	30	35	9.9	300	21.45	25.0	no
CV	copd:Rec_Pat_CV	1590	2862	14.79	45.0	50.0	14.9	35	50	9.9	300	25.0	25.0	only if does not reach protein intake with meals
DMP	copd:Rec_Pat_DMP	1498	2696	14.78	45.0	50.0	14.9	30	35	9.9	300	25.0	25.0	only if does not reach protein intake with meals
FA	copd:Rec_Pat_FA	947	1421	19.45	45.0	50.0	14.9	30	35	9.9	300	17.90	25.0	no
FG	copd:Rec_Pat_FG	2081	3122	15.05	45.0	50.0	14.9	35	50	9.9	300	25.0	25.0	only if does not reach protein intake with meals
HB	copd:Rec_Pat_HB	1140	1710	15.85	45.0	50.0	14.9	30	35	9.9	300	21.54	25.0	no
LA	copd:Rec_Pat_LA	941	1694	17.69	45.0	50.0	14.9	30	35	9.9	300	21.34	25.0	yes
MM	copd:Rec_Pat_MM	1506	2259	16.89	45.0	50.0	14.9	30	35	9.9	300	25.0	25.0	no
SN	copd:Rec_Pat_SN	1354	2437	14.78	45.0	50.0	14.9	30	35	9.9	300	25.0	25.0	only if does not reach protein intake with meals
SP	copd:Rec_Pat_SP	1519	2734	15.3	45.0	50.0	14.9	30	35	9.9	300	25.0	25.0	only if does not reach protein intake with meals
TC	copd:Rec_Pat_TC	1215	2187	14.7	45.0	50.0	14.9	30	35	9.9	300	25.0	25.0	no
TM	copd:Rec_Pat_TM	1463	2195	16.92	45.0	50.0	14.9	35	50	9.9	300	25.0	25.0	no
VM	copd:Rec_Pat_VM	1115	2007	12.53	45.0	50.0	14.9	30	35	9.9	300	25.0	25.0	no

Appendix 4. A table reporting the nutritional recommendations for each patient modelled in the COPD & Nutrition ontology.

Conclusions and future work

This Thesis investigated the status of Ontology Engineering, underlining the main research questions that remain unsolved. A literature review analyzed the evolution of the discipline and its methodologies, illustrating some of the key issues and how they were tackled by other researchers over the last two decades.

With the emergence of agile paradigm-based methodologies, which seem to represent a promising solution to some issues related to Ontology Engineering, a new (and, at the same time, old) research question arose: are the latest agile methodologies efficient in supporting ontologists in developing their models? In which ways do these methodologies actively support their end user, and what are the results of adopting them?

These questions were the driver for an investigation of agile methodologies, which eventually led to the development of a novel Agile, Simplified and Collaborative Ontology engineering methodology (AgiSCOnt) – and its evaluation. AgiSCOnt was then adopted for the development of two new ontologies in the field of health dedicated to nutrition for patients characterized by chronic conditions.

The investigation of existing agile methodologies underlined that the level of agility of a methodology is correlated to the clarity and simplicity of its instructions, as well as to the quality of the resulting ontology: the higher the level of agility, the less clear is the application of the methodology's instructions for novice ontology engineer – which reflects negatively on the quality of the ontologies. Moreover, existing agile approaches do not take into account the possibility of providing instructions at an authoring level, and often they neglect to provide comprehensive guidance in reusing existing sources of knowledge.

Leveraging on these findings, AgiSCOnt provided a different agile approach. Starting with identifying a possible solution to tackle the problem of "knowledge acquisition bottleneck," this new methodology relies on well-known and familiar tools to guide the domain analysis step and facilitate the successive development test. Devoted to collaborative engineering – which foresees an active role for domain experts in domain analysis and, possibly, in the development step – AgiSCOnt proved itself efficient in supporting novice ontologists and in helping them adopt some Ontology Design Patterns.

When used in a research project context, AgiSCOnt fostered cooperation between the ontologist and domain experts, reaching the goal of delivering two ontologies able to answer the requirements identified.

As discussed in Chapters 3 and 4, although promising for a small sample of novice ontologists, AgiSCOnt's evaluation could benefit from a broader evaluation involving a larger and more heterogeneous sample. The novel methodology could also benefit from feedback from the scientific community, in particular expert ontologists. In this regard, future research directions will be dedicated to testing (and refining) AgiSCOnt with a different sample and acquiring essential feedback to enable the enhancement of the methodology. Also, a comparison and evaluation between AgiSCOnt and non-agile methodologies could help elicit some aspects that were not considered in this work.

Bibliography

1. Doan, A., Halevy, A., Ives, Z., Doan, A., Halevy, A., Ives, Z.: Schema matching and mapping. Principles of Data Integration. 121–160 (2012).
2. Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge acquisition. 5, 199–220 (1993).
3. Borst, P., Akkermans, H., Top, J.: Engineering ontologies. International journal of human-computer studies. 46, 365–406 (1997).
4. Studer, R., Benjamins, V.R., Fensel, D.: Knowledge engineering: principles and methods. Data & knowledge engineering. 25, 161–197 (1998).
5. Guarino, N., Oberle, D., Staab, S.: What is an ontology? In: Handbook on ontologies. pp. 1–17. Springer (2009).
6. Pereira, C.S.: Collaborative ontology specification. In: Doctoral Symposium on Informatics Engineering (2008).
7. Fernández-López, M., Gómez-Pérez, A., Juristo, N.: Methontology: from ontological art towards ontological engineering. (1997).
8. Soldatova, L., Panov, P., Džeroski, S.: Ontology engineering: from an art to a craft. In: International Experiences and Directions Workshop on OWL. pp. 174–181. Springer (2015).
9. Pulido, J., Ruiz, M., Herrera, R., Cabello, E., Legrand, S., Elliman, D.: Ontology languages for the semantic web: A never completely updated review. Knowledge-Based Systems. 19, 489–497 (2006).
10. Su, X., Ilebrekke, L.: A comparative study of ontology languages and tools. In: International Conference on Advanced Information Systems Engineering. pp. 761–765. Springer (2002).
11. Hayes, P., Menzel, C.: A semantics for the knowledge interchange format. In: IJCAI 2001 Workshop on the IEEE Standard Upper Ontology. p. 145 (2001).
12. Kifer, M., Lausen, G.: F-logic: a higher-order language for reasoning about objects, inheritance, and scheme. In: Proceedings of the 1989 ACM SIGMOD international conference on Management of data. pp. 134–146 (1989).
13. Foxvog, D.: Cyc. In: Theory and applications of ontology: Computer applications. pp. 259–278. Springer (2010).
14. Pan, J.Z.: Resource description framework. In: Handbook on ontologies. pp. 71–90. Springer (2009).
15. McBride, B.: The resource description framework (RDF) and its vocabulary description language RDFS. In: Handbook on ontologies. pp. 51–65. Springer (2004).
16. Horrocks, I., Fensel, D., Broekstra, J., Decker, S., Erdmann, M., Goble, C., Harmelen, F. van, Klein, M., Staab, S., Studer, R., others: The ontology inference layer OIL, (2000).
17. Horrocks, I., others: DAML+OIL: A Description Logic for the Semantic Web. IEEE Data Eng. Bull. 25, 4–9 (2002).
18. Antoniou, G., Harmelen, F. van: Web ontology language: Owl. In: Handbook on ontologies. pp. 67–92. Springer (2004).
19. W3C: OWL 2 Profiles - available online: https://www.w3.org/2007/OWL/wiki/Profiles.
20. W3C: OWL 2 Web Ontology Language - Structural Specification and Functional-Style Syntax (Second Edition) - available online: https://www.w3.org/TR/owl2-syntax/.
21. Alamri, A., Bertok, P.: Distributed store for ontology data management. In: Computer and Information Science 2012. pp. 15–35. Springer (2012).
22. Mizoguchi, R.: Tutorial on ontological engineering Part 2: Ontology development, tools and languages. New Generation Computing. 22, 61–96 (2004).
23. Simperl, E.P.B., Tempich, C.: Ontology engineering: A reality check. In: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems. pp. 836–854. Springer (2006).
24. Grüninger, M., Fox, M.S.: The role of competency questions in enterprise engineering. In: Benchmarking—Theory and practice. pp. 22–31. Springer (1995).
25. Boicu, M., Tecuci, G., Stanescu, B., Balan, C., Popovici, E.: Ontologies and the knowledge acquisition bottleneck. In: OIS@ IJCAI (2001).
26. Pinto, H.S., Martins, J.: Reusing ontologies. In: AAAI 2000 Spring Symposium on Bringing Knowledge to Business Processes. p. 7. Karlsruhe, Germany: AAAI (2000).
27. Keet, M.: An introduction to ontology engineering. Maria Keet Cape Town (2018).
28. Garcia, A., O'Neill, K., Garcia, L.J., Lord, P., Stevens, R., Corcho, O., Gibson, F.: Developing ontologies within decentralised settings. In: Semantic e-Science. pp. 99–139. Springer (2010).
29. Neuhaus, F., Neuhaus, F., Ray, S., Sriram, R.D.: Toward ontology evaluation across the life cycle. US Department of Commerce, National Institute of Standards and Technology (2014).
30. Spoladore, D., Pessot, E.: Collaborative Ontology Engineering Methodologies for the Development of Decision Support Systems: Case Studies in the Healthcare Domain. Electronics. 10, 1060 (2021).
31. Kotis, K.I., Vouros, G.A., Spiliotopoulos, D.: Ontology engineering methodologies for the evolution of living and reused ontologies: status, trends, findings and recommendations. The Knowledge Engineering Review. 35, (2020).

32. Rebele, T., Suchanek, F., Hoffart, J., Biega, J., Kuzey, E., Weikum, G.: YAGO: A multilingual knowledge base from wikipedia, wordnet, and geonames. In: International semantic web conference. pp. 177–185. Springer (2016).
33. Calbimonte, J.-P., Dubosson, F., Hilfiker, R., Cotting, A., Schumacher, M.: The MedRed ontology for representing clinical data acquisition metadata. In: International Semantic Web Conference. pp. 38–47. Springer (2017).
34. Narula, G.S., Yadav, U., Duhan, N., Jain, V.: Evolution of FOAF and SIOC in Semantic Web: A Survey. In: Big Data Analytics. pp. 253–263. Springer (2018).
35. Salatino, A.A., Thanapalasingam, T., Mannocci, A., Osborne, F., Motta, E.: The computer science ontology: a large-scale taxonomy of research areas. In: International Semantic Web Conference. pp. 187–205. Springer (2018).
36. Tommasini, R., Sedira, Y.A., Dell'Aglio, D., Balduini, M., Ali, M.I., Le Phuoc, D., Della Valle, E., Calbimonte, J.-P.: VoCaLS: vocabulary and catalog of linked streams. In: International Semantic Web Conference. pp. 256–272. Springer (2018).
37. Arndt, N., Naumann, P., Radtke, N., Martin, M., Marx, E.: Decentralized collaborative knowledge management using git. Journal of Web Semantics. 54, 29–47 (2019).
38. Al-Baltah, I.A., Ghani, A.A.A., Rahman, W.N.W.A., Atan, R.: A comparative study on ontology development methodologies towards building semantic conflicts detection ontology for heterogeneous web services. Research Journal of Applied Sciences, Engineering and Technology. 7, 2674–2679 (2014).
39. Corcho, O., Fernández-López, M., Gómez-Pérez, A.: Methodologies, tools and languages for building ontologies. Where is their meeting point? Data & knowledge engineering. 46, 41–64 (2003).
40. Fernández-López, M., Gómez-Pérez, A.: Overview and analysis of methodologies for building ontologies. The knowledge engineering review. 17, 129–156 (2002).
41. Iqbal, R., Murad, M.A.A., Mustapha, A., Sharef, N.M., others: An analysis of ontology engineering methodologies: A literature review. Research journal of applied sciences, engineering and technology. 6, 2993–3000 (2013).
42. Sattar, A., Surin, E.S.M., Ahmad, M.N., Ahmad, M., Mahmood, A.K.: Comparative analysis of methodologies for domain ontology development: A systematic review. Int. J. Adv. Comput. Sci. Appl. 11, 99–108 (2020).
43. Simperl, E., Luczak-Rösch, M.: Collaborative ontology engineering: a survey. The Knowledge Engineering Review. 29, 101–131 (2014).
44. Sensuse, D.I., Sucahyo, Y.G., Silalahi, M., Wulandari, I.A., Akmaliah, I.F., Noprisson, H.: Toward to operationalization of socio-technical ontology engineering methodology. In: 2017 5th International Conference on Cyber and IT Service Management (CITSM). pp. 1–7. IEEE (2017).
45. Paslaru Bontas Simperl, E., Tempich, C., Sure, Y.: Ontocom: A cost estimation model for ontology engineering. In: International Semantic Web Conference. pp. 625–639. Springer (2006).
46. Staab, S., Studer, R., Schnurr, H.-P., Sure, Y.: Knowledge processes and ontologies. IEEE Intelligent systems. 16, 26–34 (2001).
47. Ali, S., Khusro, S.: POEM: Practical ontology engineering model for semantic web ontologies. Cogent Engineering. 3, 1193959 (2016).
48. Keet, C.M.: Transforming semi-structured life science diagrams into meaningful domain ontologies with DiDOn. Journal of biomedical informatics. 45, 482–494 (2012).
49. Noy, N., McGuinness, D.L., others: Ontology development 101. Knowledge Systems Laboratory, Stanford University. 2001, (2001).
50. Kassel, G.: Integration of the DOLCE top-level ontology into the OntoSpec methodology. arXiv preprint cs/0510050. (2005).
51. Guarino, N., Welty, C.A.: An overview of OntoClean. Handbook on ontologies. 151–171 (2004).
52. Presutti, V., Daga, E., Gangemi, A., Blomqvist, E.: eXtreme design with content ontology design patterns. In: Proc. Workshop on Ontology Patterns. pp. 83–97 (2009).
53. Keet, C.M., Khan, M.T., Ghidini, C.: Guided ENtity reuse and class Expression geneRATOR. In: Proceedings of the seventh international conference on Knowledge capture. pp. 143–144 (2013).
54. Keet, C.M., Khan, M.T., Ghidini, C.: Ontology authoring with FORZA. In: Proceedings of the 22nd ACM international conference on Information & Knowledge Management. pp. 569–578 (2013).
55. Davis, B., Iqbal, A.A., Funk, A., Tablan, V., Bontcheva, K., Cunningham, H., Handschuh, S.: Roundtrip ontology authoring. In: International Semantic Web Conference. pp. 50–65. Springer (2008).
56. Denaux, R., Holt, I., Corda, I., Dimitrova, V., Dolbear, C., Cohn, A.G.: ROO: A Tool to Assist Domain Experts with Ontology Construction. In: ESWC (Posters) (2008).
57. Dimitrova, V., Denaux, R., Hart, G., Dolbear, C., Holt, I., Cohn, A.G.: Involving domain experts in authoring OWL ontologies. In: International Semantic Web Conference. pp. 1–16. Springer (2008).
58. Keet, C.M., Lawrynowicz, A.: Test-driven development of ontologies. In: European Semantic Web Conference. pp. 642–657. Springer (2016).

59. Blomqvist, E., Gangemi, A., Presutti, V.: Experiments on pattern-based ontology design. In: Proceedings of the fifth international conference on Knowledge capture. pp. 41–48 (2009).
60. Gangemi, A., Presutti, V.: Ontology design patterns. In: Handbook on ontologies. pp. 221–243. Springer (2009).
61. Alexander, C., others: The timeless way of building. New york: Oxford university press (1979).
62. Blomqvist, E., Presutti, V., Daga, E., Gangemi, A.: Experimenting with eXtreme design. In: International Conference on Knowledge Engineering and Knowledge Management. pp. 120–134. Springer (2010).
63. Blomqvist, E., Hitzler, P., Janowicz, K., Krisnadhi, A., Narock, T., Solanki, M.: Considerations regarding Ontology Design Patterns. Semantic Web. 7, 1–7 (2016).
64. Mortensen, J.M., Horridge, M., Musen, M.A., Noy, N.F.: Applications of ontology design patterns in biomedical ontologies. In: AMIA Annual Symposium Proceedings. p. 643. American Medical Informatics Association (2012).
65. Chimienti, M., Dassisti, M., De Nicola, A., Missikoff, M.: Evaluation of Ontology Building Methodologies-A Method based on Balanced Scorecards. In: International Conference on Knowledge Engineering and Ontology Development. pp. 141–146. SCITEPRESS (2009).
66. Tempich, C., Pinto, H.S., Sure, Y., Vrandecic, D., Casellas, N., Casanovas, P.: Evaluating DILIGENT ontology engineering in a legal case study. In: IVR 22nd World Congress–Law and Justice in a Global Society. International Association for Philosophy of Law and Social Philosophy (2005).
67. Vigo, M., Jay, C., Stevens, R.: Design insights for the next wave ontology authoring tools. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 1555–1558 (2014).
68. Uschold, M., King, M.: Towards a methodology for building ontologies. Citeseer (1995).
69. Grüninger, M., Fox, M.S.: Methodology for the design and evaluation of ontologies. (1995).
70. Multiple Use (FP3-ESPRIT 3), M.K.A.C.T.S. for: The KACTUS Booklet version 1.0. Esprit Project 8145 KACTUS,. (1996).
71. Pinto, H.S., Staab, S., Tempich, C.: DILIGENT: Towards a fine-grained methodology for DIstributed, Loosely-controlled and evolvInG Engineering of oNTologies. In: ECAI. p. 393. Citeseer (2004).
72. Sure, Y., Staab, S., Studer, R.: On-to-knowledge methodology (OTKM). In: Handbook on ontologies. pp. 117–132. Springer (2004).
73. Kotis, K., Vouros, G.A., Alonso, J.P.: HCOME: A tool-supported methodology for engineering living ontologies. In: International Workshop on Semantic Web and Databases. pp. 155–166. Springer (2004).
74. Zachila, K., Kotis, K., Paparidis, E., Ladikou, S., Spiliotopoulos, D.: Facilitating Semantic Interoperability of Trustworthy IoT Entities in Cultural Spaces: The Smart Museum Ontology. IoT. 2, 741–760 (2021).
75. Jarrar, M., Meersman, R.: Ontology engineering–the DOGMA approach. In: Advances in Web Semantics I. pp. 7–34. Springer (2008).
76. De Moor, A., De Leenheer, P., Meersman, R.: DOGMA-MESS: A meaning evolution support system for interorganizational ontology engineering. In: International Conference on Conceptual Structures. pp. 189–202. Springer (2006).
77. Auer, S., Herre, H.: RapidOWL—An agile knowledge engineering methodology. In: International Andrei Ershov Memorial Conference on Perspectives of System Informatics. pp. 424–430. Springer (2006).
78. Suárez-Figueroa, M.C., Gómez-Pérez, A., Fernandez-Lopez, M.: The NeOn Methodology framework: A scenario-based methodology for ontology development. Applied ontology. 10, 107–145 (2015).
79. De Nicola, A., Missikoff, M.: A lightweight methodology for rapid ontology engineering. Communications of the ACM. 59, 79–86 (2016).
80. Peroni, S.: A simplified agile methodology for ontology development. In: OWL: Experiences and Directions–Reasoner Evaluation. pp. 55–69. Springer (2016).
81. Hakkarainen, S., Strasunskas, D., Hella, L., Tuxen, S.: Weighted Evaluation of Ontology Building Methods. In: CAiSE Short Paper Proceedings (2005).
82. Chimienti, M., Dassisti, M., De Nicola, A., Missikoff, M.: Benchmarking Criteria to Evaluate Ontology Building Methodologies. In: EMOI-INTEROP (2006).
83. De Nicola, A., Missikoff, M., Navigli, R.: A software engineering approach to ontology building. Information systems. 34, 258–275 (2009).
84. Gobin, B.: Assessing the suitability of existing Agile Ontology Engineering Methodologies for ontology module development.
85. Gobin, B.: A Quantitative Framework for assessing Agile Ontology Engineering Methodologies. In: Proc. International Conference on Web and Information Systems. Citeseer (2014).
86. Gobin, B.A.: Using the 4-DAT Tool to evaluate Agile Ontology Engineering Methodologies.
87. Knublauch, H.: An agile development methodology for knowledge-based systems including a Java framework for knowledge modeling and appropriate tool support, (2002).
88. Lourdusamy, R., John, A.: A review on metrics for ontology evaluation. In: 2018 2nd International Conference on Inventive Systems and Control (ICISC). pp. 1415–1421. IEEE (2018).
89. Hlomani, H., Stacey, D.: Approaches, methods, metrics, measures, and subjectivity in ontology evaluation: A survey. Semantic Web Journal. 1, 1–11 (2014).

90. Spoladore, D., Pessot, E.: An evaluation of agile Ontology Engineering Methodologies for the digital transformation of companies. Computers in Industry. 140, 103690 (2022).
91. Auer, S.: The RapidOWL Methodology–Towards Agile Knowledge Engineering. In: 15th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'06). pp. 352–357. IEEE (2006).
92. Keet, C.M.: The use of foundational ontologies in ontology development: an empirical assessment. In: Extended Semantic Web Conference. pp. 321–335. Springer (2011).
93. Burton-Jones, A., Storey, V.C., Sugumaran, V., Ahluwalia, P.: A semiotic metrics suite for assessing the quality of ontologies. Data & Knowledge Engineering. 55, 84–102 (2005).
94. Gangemi, A., Catenacci, C., Ciaramita, M., Lehmann, J.: A theoretical framework for ontology evaluation and validation. In: SWAP. p. 16 (2005).
95. Paslaru Bontas Simperl, E., Mochol, M.: Cost estimation for ontology development. In: Business Information Systems–9th International Conference on Business Information Systems (BIS 2006). Gesellschaft für Informatik eV (2006).
96. Soundararajan, S., Balci, O., Arthur, J.D.: Assessing an organization's capability to effectively implement its selected agile method (s): An objectives, principles, strategies approach. In: 2013 Agile Conference. pp. 22–31. IEEE (2013).
97. Soundararajan, S., Arthur, J.D., Balci, O.: A methodology for assessing agile software development methods. In: 2012 Agile Conference. pp. 51–54. IEEE (2012).
98. Simperl, E., Mochol, M., Bürger, T.: Achieving maturity: the state of practice in ontology engineering in 2009. International Journal of Computer Science and Applications. 7, 45–65 (2010).
99. Suárez-Figueroa, M.C., Gómez-Pérez, A., Villazón-Terrazas, B.: How to write and use the ontology requirements specification document. In: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems. pp. 966–982. Springer (2009).
100. Kalyanpur, A., Parsia, B., Sirin, E., Cuenca-Grau, B.: Repairing unsatisfiable concepts in OWL ontologies. In: European Semantic Web Conference. pp. 170–184. Springer (2006).
101. Barbagallo, A., De Nicola, A., Missikoff, M.: eGovernment ontologies: Social participation in building and evolution. In: 2010 43rd Hawaii International Conference on System Sciences. pp. 1–10. IEEE (2010).
102. Beck, K.: Test-driven development: by example. Addison-Wesley Professional (2003).
103. Miller, G.A.: The magical number seven, plus or minus two: Some limits on our capacity for processing information. Psychological review. 63, 81 (1956).
104. Auer, S., Herre, H.: A versioning and evolution framework for RDF knowledge bases. In: International Andrei Ershov Memorial Conference on Perspectives of System Informatics. pp. 55–69. Springer (2006).
105. Yin, R.K.: Case study research: Design and methods. sage (2009).
106. Colomo-Palacios, R., Garcıa-Crespo, Á., Soto-Acosta, P., Ruano-Mayoral, M., Jiménez-López, D.: A case analysis of semantic technologies for R&D intermediation information management. International Journal of Information Management. 30, 465–469 (2010).
107. Tacla, C.A., Caliari, F.M., Gimenez-Lugo, G.A.: An application ontology for furniture packaging design. In: 2007 11th International Conference on Computer Supported Cooperative Work in Design. pp. 728–733. IEEE (2007).
108. Fernández-López, M., Poveda-Villalón, M., Suárez-Figueroa, M.C., Gómez-Pérez, A.: Why are ontologies not reused across the same domain? Journal of Web Semantics. 57, 100492 (2019).
109. Alaa, R., Gawish, M., Fernández-Veiga, M.: Improving recommendations for online retail markets based on ontology evolution. Electronics. 10, 1650 (2021).
110. Vigo, M., Bail, S., Jay, C., Stevens, R.: Overcoming the pitfalls of ontology authoring: Strategies and implications for tool design. International Journal of Human-Computer Studies. 72, 835–845 (2014).
111. Spoladore, D., Mahroo, A., Trombetta, A., Sacco, M.: DOMUS: A domestic ontology managed ubiquitous system. Journal of Ambient Intelligence and Humanized Computing. 13, 3037–3052 (2022).
112. Spoladore, D., Sacco, M.: Semantic and dweller-based decision support system for the reconfiguration of domestic environments: RecAAL. Electronics. 7, 179 (2018).
113. Arlati, S., Spoladore, D., Mottura, S., Zangiacomi, A., Ferrigno, G., Sacchetti, R., Sacco, M.: Analysis for the design of a novel integrated framework for the return to work of wheelchair users. Work. 61, 603–625 (2018).
114. Spoladore, D., Cilsal, T., Mahroo, A., Trombetta, A., Sacco, M.: Towards an Ontology-Based Decision Support System to Support Car-Reconfiguration for Novice Wheelchair Users. In: International Conference on Computers Helping People with Special Needs. pp. 445–452. Springer (2022).
115. ISO/IEC/IEEE 16326:2009. Systems and software engineering — Life cycle processes — Project management, (2009).
116. Shadbolt, N.R., Smart, P.R., Wilson, J., Sharples, S.: Knowledge elicitation. Evaluation of human work. 163–200 (2015).
117. Hayes-Roth, F., Waterman, D.A., Lenat, D.B.: Building expert systems. Addison-Wesley Longman Publishing Co., Inc. (1983).

118. Cooke, N.J.: Varieties of knowledge elicitation techniques. International journal of human-computer studies. 41, 801–849 (1994).
119. Shadbolt, N., Burton, M.: Knowledge elicitation: a systematic approach. Evaluation of human work: a practical ergonomics methodology. 406–440 (1995).
120. Uschold, M., Gruninger, M.: Ontologies: Principles, methods and applications. The knowledge engineering review. 11, 93–136 (1996).
121. Schreiber, A.T., Schreiber, G., Akkermans, H., Anjewierden, A., Shadbolt, N., Hoog, R. de, Velde, W. Van de, Wielinga, B., Nigel, R., others: Knowledge engineering and management: the CommonKADS methodology. MIT press (2000).
122. Kotis, K., Vouros, G.A.: Human-centered ontology engineering: The HCOME methodology. Knowledge and Information Systems. 10, 109–131 (2006).
123. Hoffman, R.R., Coffey, J.W., Ford, K.M., Carnot, M.J.: Storm-lk: A human-centered knowledge model for weather forecasting. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting. pp. 752–752 (2001).
124. Castro, A.G., Rocca-Serra, P., Stevens, R., Taylor, C., Nashar, K., Ragan, M.A., Sansone, S.-A.: The use of concept maps during knowledge elicitation in ontology development processes–the nutrigenomics use case. BMC bioinformatics. 7, 1–14 (2006).
125. Suárez-Figueroa, M.C., Gómez-Pérez, A.: Ontology requirements specification. In: Ontology Engineering in a Networked World. pp. 93–106. Springer (2012).
126. Novak, J.D., Cañas, A.J.: Concept mapping. International Guide to Student Achievement. 362–365 (2013).
127. Novak, J.D., Gowin, D.B., Bob, G.D.: Learning how to learn. cambridge University press (1984).
128. Dar, A.R., Razzaq, S.: Building Ontology For Library Management System Using Dewey Decimal Classification Scheme. Presented at the (2018).
129. Weibel, S.L., Koch, T.: The Dublin core metadata initiative. D-lib magazine. 6, 1082–9873 (2000).
130. Alani, H., Brewster, C.: Ontology ranking based on the analysis of concept structures. In: Proceedings of the 3rd international conference on Knowledge capture. pp. 51–58 (2005).
131. Buitelaar, P., Eigner, T., Declerck, T.: OntoSelect: A dynamic ontology library with support for ontology selection. In: In Proceedings of the Demo Session at the International Semantic Web Conference. Citeseer (2004).
132. Park, J., Oh, S., Ahn, J.: Ontology selection ranking model for knowledge reuse. Expert Systems with Applications. 38, 5133–5144 (2011).
133. Poveda Villalón, M., Suárez-Figueroa, M.C., Gómez-Pérez, A.: The landscape of ontology reuse in linked data. (2012).
134. Courtot, M., Gibson, F., Lister, A.L., Malone, J., Schober, D., Brinkman, R.R., Ruttenberg, A.: MIREOT: The minimum information to reference an external ontology term. Applied Ontology. 6, 23–33 (2011).
135. Xiang, Z., Courtot, M., Brinkman, R.R., Ruttenberg, A., He, Y.: OntoFox: web-based support for ontology reuse. BMC research notes. 3, 1–12 (2010).
136. Gangemi, A.: Content Ontology Design Pattern: Collection - available online: http://ontologydesignpatterns.org/wiki/Submissions:Collection.
137. Consortium, W.W.W., others: SPARQL 1.1 overview. (2013).
138. Euzenat, J., Shvaiko, P., others: Ontology matching. Springer (2007).
139. Organization, W.H., others: IFC: International Classification of Functioning, Disability and Health. (2001).
140. Gangemi, A.: Content Ontology Design Patterns: Sequence - available online: http://ontologydesignpatterns.org/wiki/Submissions:Sequence.
141. International Classification of Functioning, Disability and Health (ICF) Ontology - available onlines: https://bioportal.bioontology.org/ontologies/ICF, (2012).
142. Blomqvist, E.: The use of Semantic Web technologies for decision support–a survey. Semantic Web. 5, 177–201 (2014).
143. Antunes, F., Freire, M., Costa, J.P.: Semantic web and decision support systems. Journal of Decision Systems. 25, 79–93 (2016).
144. sPATIALS3 An integrated HUB of RESEARCH, DEVELOPMENT and INNOVATION in the context of NUTRITION at the service of Companies and Citizens - Research project website: https://hubspatials3.it/en/, (2021).
145. Spieker, M.R.: Evaluating dysphagia. American Family Physician. 61, 3639–3648 (2000).
146. Rosenbek, J.C., Robbins, J.A., Roecker, E.B., Coyle, J.L., Wood, J.L.: A penetration-aspiration scale. Dysphagia. 11, 93–98 (1996).
147. Tibbling, L., Gustafsson, B.: Dysphagia and its consequences in the elderly. Dysphagia. 6, 200–202 (1991).
148. Gallegos, C., Brito-De La Fuente, E., Clavé, P., Costa, A., Assegehegn, G.: Nutritional aspects of dysphagia management. Advances in food and nutrition research. 81, 271–318 (2017).
149. Cichero, J.A.: Age-related changes to eating and swallowing impact frailty: Aspiration, choking risk, modified food texture and autonomy of choice. Geriatrics. 3, 69 (2018).

150. Ugliassi, P. Mario andPiaggiesi: Nutrirsi con gusto. Centro Clinico NEMO (2013).
151. O'Neil, K.H., Purdy, M., Falk, J., Gallo, L.: The dysphagia outcome and severity scale. Dysphagia. 14, 139–145 (1999).
152. Battel, I., Calvo, I., Walshe, M.: Functional Oral Intake Scale–Italian Version. Folia Phoniatrica et Logopaedica: International Journal of Phoniatrics, Speech Therapy and Communication Pathology.
153. Huckabee, M.-L., McIntosh, T., Fuller, L., Curry, M., Thomas, P., Walshe, M., McCague, E., Battel, I., Nogueira, D., Frank, U., others: The Test of Masticating and Swallowing Solids (TOMASS): reliability, validity and international normative data. International Journal of Language & Communication Disorders. 53, 144–156 (2018).
154. Bernardes, R.A., Cruz, A., Neves, H., Parola, V., Catela, N.: Screening Tools Designed to Assess and Evaluate Oropharyngeal Dysphagia in Adult Patients: A Scoping Review. Nursing Reports. 12, 245–258 (2022).
155. Falconer, S.: OntoGraf plugin for Protégé - available online: https://protegewiki.stanford.edu/wiki/OntoGraf, (2010).
156. Horridge, M., Musen, M.: Snap-SPARQL: a java framework for working with SPARQL and OWL. In: International Experiences and Directions Workshop on OWL. pp. 154–165. Springer (2015).
157. Enterprise, S.: Troubleshooting Reasoning - available online: https://docs.stardog.com/inference-engine/troubleshooting, (2022).
158. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A practical owl-dl reasoner. Journal of Web Semantics. 5, 51–53 (2007).
159. Mannino, D.M., Buist, A.S.: Global burden of COPD: risk factors, prevalence, and future trends. The Lancet. 370, 765–773 (2007).
160. Mannino, D.M., Kiri, V.A.: Changing the burden of COPD mortality. International journal of chronic obstructive pulmonary disease. 1, 219 (2006).
161. Rodriguez-Roisin, R.: Toward a consensus definition for COPD exacerbations. Chest. 117, 398S–401S (2000).
162. Nordén, J., Grönberg, A., Bosaeus, I., Forslund, H.B., Hulthén, L., Rothenberg, E., Karlsson, J., Wallengren, O., Slinde, F.: Nutrition impact symptoms and body composition in patients with COPD. European journal of clinical nutrition. 69, 256–261 (2015).
163. Iersel, L.E. van, Beijers, R.J., Gosker, H.R., Schols, A.M.: Nutrition as a modifiable factor in the onset and progression of pulmonary function impairment in COPD: A systematic review. Nutrition reviews. 80, 1434–1444 (2022).
164. Scoditti, E., Massaro, M., Garbarino, S., Toraldo, D.M.: Role of diet in chronic obstructive pulmonary disease prevention and treatment. Nutrients. 11, 1357 (2019).
165. Furulund, E., Bemanian, M., Berggren, N., Madebo, T., Rivedal, S.H., Lid, T.G., Fadnes, L.T.: Effects of nutritional interventions in individuals with chronic obstructive lung disease: a systematic review of randomized controlled trials. International Journal of Chronic Obstructive Pulmonary Disease. 16, 3145 (2021).
166. Hsieh, M.-J., Yang, T.-M., Tsai, Y.-H.: Nutritional supplementation in patients with chronic obstructive pulmonary disease. Journal of the Formosan Medical Association. 115, 595–601 (2016).
167. Calverley, P.M.: The GOLD classification has advanced understanding of COPD. American journal of respiratory and critical care medicine. 170, 211–212 (2004).
168. Prasad, N., Sinha, A., Gupta, A., Bhadauria, D., Manjunath, R., Kaul, A., Sharma, R.: Validity of nutrition risk index as a malnutrition screening tool compared with subjective global assessment in end-stage renal disease patients on peritoneal dialysis. Indian journal of nephrology. 26, 27 (2016).
169. Aronne, L.J.: Classification of obesity and assessment of obesity-related health risks. Obesity research. 10, 105S–115S (2002).
170. Cruz-Jentoft, A.J., Bahat, G., Bauer, J., Boirie, Y., Bruyère, O., Cederholm, T., Cooper, C., Landi, F., Rolland, Y., Sayer, A.A., others: Sarcopenia: revised European consensus on definition and diagnosis. Age and ageing. 48, 16–31 (2019).
171. Evans, W.J., Morley, J.E., Argilés, J., Bales, C., Baracos, V., Guttridge, D., Jatoi, A., Kalantar-Zadeh, K., Lochs, H., Mantovani, G., others: Cachexia: a new definition. Clinical nutrition. 27, 793–799 (2008).
172. Amirkalali, B., Hosseini, S., Heshmat, R., Larijani, B.: Comparison of Harris Benedict and Mifflin-ST Jeor equations with indirect calorimetry in evaluating resting energy expenditure. Indian journal of medical sciences. 62, 283–290 (2008).