

Methodological guidelines for reusing general ontologies

Mariano Fernández-López^{a,*}, Asunción Gómez-Pérez^{b,1}, Mari Carmen Suárez-Figueroa^{b,2}

^a Escuela Politécnica Superior, Universidad San Pablo CEU, Urbanización Montepríncipe, S/N, 28668 Boadilla del Monte, Madrid Spain

^b Departamento de Inteligencia Artificial, Facultad de Informática, Universidad Politécnica de Madrid, Campus de Montegancedo, 28660 Boadilla del Monte, Madrid Spain

A B S T R A C T

Currently, there is a great deal of well-founded explicit knowledge formalizing general notions, such as time concepts and the *part_of* relation. Yet, it is often the case that instead of reusing ontologies that implement such notions (the so-called general ontologies), engineers create procedural programs that implicitly implement this knowledge. They do not save time and code by reusing explicit knowledge, and devote effort to solve problems that other people have already adequately solved. Consequently, we have developed a methodology that helps engineers to: (a) identify the type of general ontology to be reused; (b) find out which axioms and definitions should be reused; (c) make a decision, using formal concept analysis, on what general ontology is going to be reused; and (d) adapt and integrate the selected general ontology in the domain ontology to be developed. To illustrate our approach we have employed use-cases. For each use case, we provide a set of heuristics with examples. Each of these heuristics has been tested in either OWL or Prolog. Our methodology has been applied to develop a pharmaceutical product ontology. Additionally, we have carried out a controlled experiment with graduated students doing a MCs in Artificial Intelligence. This experiment has yielded some interesting findings concerning what kind of features the future extensions of the methodology should have.

1. Introduction

General ontologies formalize notions that go beyond particular domains [1,2]. They are also known as foundational ontologies [3], and formal ontologies [4] in the sense that they are domain-neutral. For example, the *part_of* relation links objects in the mechanical domain (the spark plug is part of the motor) and in the domain of cultural activities (for example, the Radetzky March performance is part of the New Year's Concert). These general ontologies seem to have a great potential for reuse.

Nowadays, however, most ontology developers consider these types of ontologies hard to understand and, consequently, difficult to use. For this reason, they shy away from using the formal notions embedded in general ontologies. This point is confirmed by the results of a simple experiment. Using Watson on-line³ we searched for semantic documents that contain the term "part". Our search returned a great number of results (4031).⁴ Then, after examining a subset of ontologies (approximately 500), we realized that the relations searched, such as *has_part*, and *part_of*, were simply defined as object properties and that no background knowledge was provided with this kind of relation. In our analysis, we found only one ontology (*bio-zen.owl*⁵) that reuses the general ontology DOLCE-Lite.⁶ A search for ontologies containing the term "time" returned similar results. In this case,

* Corresponding author. Tel.: +34 913726439; fax: +34 91 372 40 49.

E-mail addresses: mfernandez.eps@ceu.es (M. Fernández-López), asun@fi.upm.es (A. Gómez-Pérez), mcsuarez@fi.upm.es (M.C. Suárez-Figueroa).

¹ Tel.: +34 913367417; fax: +34 913524819.

² Tel.: +34 913363672; fax: +34 913524819.

³ <http://watson.kmi.open.ac.uk/WatsonWUI/>.

⁴ Access carried out the 4th of November of 2011.

⁵ <http://neuroscientific.net/bio-zen.owl>.

⁶ <http://www.loa-cnr.it/ontologies/DOLCE-Lite.owl>.

we located 8800 semantic documents and found that only one of the ontologies analyzed (*bio-zen.owl*) reuses the notion of time from a general ontology.

Suppose now that we have to develop an ontology about pharmaceutical products in which we directly define an object property as *isPartOf*. In this case, a 20-line Java⁷ program would be necessary to answer a competency question (CQ) such as 'Which drug contains iron?'. However, if we know the formal properties (e.g., transitivity and anti-symmetry relation) of the *part_of* relation and we include such formal notions (e.g., transitivity) in the ontology, then the above CQ could be solved with a five-line SPARQL⁸ query.

Thus, we can save time and effort if we reuse general ontologies that explicitly define 'what things are' in the ontology, instead of implementing a Java program that refers to 'how things are done'. Reusing general ontologies in ontology development (a) stops the repeated definition of general notions (e.g., *part_of*); (b) can take advantage of the formal characteristics of these notions (properties and definitions); and (c) helps us to avoid conflicting interpretations of general terms inside the same domain ontology. For example, Smith and Rosse [5], and Schulz et al. [6] show that a lack of a formal definition of the *part-of* notion leads to conflicting uses of this relation in biomedicine ontologies.

The main aim of this paper is to bridge the gap between people with a solid theoretical background working on formal specifications and people working on more technical issues. To achieve this objective, we have specified our knowledge on the reuse of general ontologies (time ontology and implemented mereologies, mainly) through detailed and prescriptive methodological guidelines.

Researchers such as Gangemi et al. [7] and Masolo et al. [8] propose the use of a mereology in ontology construction. Borst [9] puts this idea into practice by building and using a mereology ontology in an engineering application for modeling, simulating and designing physical systems. The main difference between the research reported in this paper and Borst's work is that we establish how to analyze, reuse and customize already existing general ontologies instead of explaining how to develop such ontologies from scratch.

Experiences in the reuse of general ontologies (e.g., [10]) have also been documented. But these have not yet been abstracted as detailed prescriptive methodological guidelines; their authors present their experience only in a particular case.

Before deciding which general ontology should be reused, we propose to conduct a comparative study. The difference between this study and that presented by Farrar and Bateman [11] is that, whereas they perform a domain independent comparison, we propose to make a comparison taking into account the requirement specification of the domain ontology in which the general ontology will be reused.

This paper describes an evolution of [12,13] and Section 10.3 of [14], and is organized as follows. Section 2 provides useful background for understanding this paper. Section 3 presents the methodological guidelines we propose for reusing general ontologies. These guidelines are intended to apply formal concept analysis (FCA) to decision-making on what general ontology should be reused. Section 4 gives an illustrative example of how to use the guidelines here presented in the general ontology reuse process. Section 5 describes a controlled experiment conducted to check whether the methodological guidelines are precise, understandable, non-redundant and comprehensive. Section 6 provides the conclusions and, finally, Section 7, the future trends.

2. Background

Examining formal ontologies such as DOLCE-Lite,⁹ SUMO-OWL,¹⁰ OpenCyc,¹¹ BFO¹² and Sowa's [15],^{13,14} we have observed that time and space play an important role in them. Space can be expressed by combining mereology and topology [16]. However, we do not deal with the *connected_to* relation and other domain-independent relations here. Our purpose is to use mereology and time modeling as examples for showing how general ontologies can be reused. In fact, our guidelines can lead to the study of other notions, possibly with the participation of specialists in such concepts.

A special case, not addressed here, is the subsumption relation. Although this relation is already a primitive in languages like OWL [17], other possible interpretations to the one provided in OWL [18] are interesting. One example is the development of an ontology that requires either non-monotonic [19] or imprecise reasoning [20].

In this section, we address ontological distinctions (Section 2.1) and some basic concepts (Section 2.2). Our purpose is to avoid misinterpretations of terms such as *ontology*, and *formal competency question*. We also make a brief introduction of formal concept analysis (FCA) (Section 2.3), mereology and time modeling, and view some of the key notions formalized in mereology and time theories and examples (Sections 2.4 and 2.5).

2.1. Ontological distinctions

As the concepts presented in Section 2.2 are subsumed by DOLCE concepts [8], we first present the ontological distinctions assumed in DOLCE. This will make Section 2.2 more comprehensible.

⁷ The Java language specification is available in [69].

⁸ The W3C Recommendation for SPARQL is available in [70].

⁹ <http://www.loa-cnr.it/ontologies/DOLCE-Lite.owl>

¹⁰ <http://www.ontologyportal.org/translations/SUMO.owl.txt>

¹¹ <http://suo.ieee.org/SUO/ontologies/Cyc-upper.txt>

¹² <http://www.ifomis.org/bfo>

¹³ <http://www.jfsowa.com/ontology/>

¹⁴ <http://suo.ieee.org/SUO/ontologies/Sowa.txt>

Universal versus particular:

“The fundamental ontological distinction between universals and particulars can be informally understood by taking the relation of instantiation as a primitive: particulars are entities which have no instances¹⁵; universals are entities that can have instances.” ([8], page 9).

For example, *Company* is a universal, whereas *Pharmacia Iberia®* is a particular that instantiates *Company*.

Endurant versus perdurant:

Inside particulars “classically, *endurants* (also called *continuants*) are characterized as entities that are ‘in time’, they are ‘wholly’ present (all their proper parts are present) at any time of their existence. On the other hand, *perdurants* (also called *occurrents*) are entities that ‘happen in time’, they extend in time by accumulating different ‘temporal parts’, so that, at any time *t* at which they exist, only their temporal parts at *t* are present.” ([8], page 11).

Example 1. A leaflet that is being read by a patient can be considered an *endurant* because it is wholly present (now), whereas the patient’s reading of this leaflet is a *perdurant* because his (her) reading of the previous paragraph is not present now.

Physical and non-physical endurants:

“Within *endurants*, we distinguish between *physical* and *non-physical* endurants, according to whether they have direct spatial qualities.” ([8], page 22).

Example 2. A small bottle, a copy of a leaflet, a spot of ink that represents a letter on a copy of a leaflet, etc., are *physical* endurants, whereas a letter, a leaflet, etc., are *non-physical* endurants. Note that a leaflet can be printed on paper; thus printed papers are *physical* endurants.

On top of the above, *non-physical* endurants have another distinction adapted from [21]:

2.1.1. Information object and description system

An *information object* is a piece of information (e.g., a leaflet) irrespective of how it is actually realized (e.g., the leaflet printed on paper). A *description system* is a *non-physical* *endurant* (e.g., a particular natural language) that is used to express an *information object* (see page 103 of [8]). It should be noted that both *information objects* and *description systems* may change a long time. Therefore, we have not considered them *abstract* entities (page 10 of [8] shows the distinction between *abstract* and *concrete* entities).

Example 3. OWL, Prolog,¹⁶ Spanish, among others are *description systems*.

2.2. Basic concepts

Fig. 1 shows an UML class diagram containing the basic concepts of this paper, of which *Physical* *endurant*, *Non-physical* *endurant*, *Description system* and *Information object* have been described in Section 2.1 The rest of the concepts are described in semantic order (the most foundational concepts are listed first) as follows:

- *Social agent* (DOLCE category): any individual granted to exist merely on the grounds of social communicability and capability of action.¹⁷ Examples of *social agent* are the *Pharmacia Iberia®* company, or a particular stockholder of *Pharmacia Iberia®*.
- *Natural language*: language used by humans in their everyday communication. *Natural language* is a *description system*.
- *Implementation language*: a language endowed with an inference engine working on an electronic computer. An *implementation language* is a *description system*.
- *Ontology*: “An ontology is a formal, explicit specification of a shared conceptualisation. A ‘conceptualisation’ refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. ‘Explicit’ means that the type of concepts used, and the constraints on their use are explicitly defined. For example, in medical domains, the concepts are diseases and symptoms, the relations between them are causal, and a constraint is that a disease cannot cause itself. ‘Formal’ refers to the fact

¹⁵ More exactly, we should say that they *can’t* have instances. This coincides with the saying that they have no instances, since we include *possibilia* (possible instances) in our domain (note from original).

¹⁶ The general core of the Prolog specification is published in [71].

¹⁷ <http://www.ontologydesignpatterns.org/ont/dul/DUL.owl>.

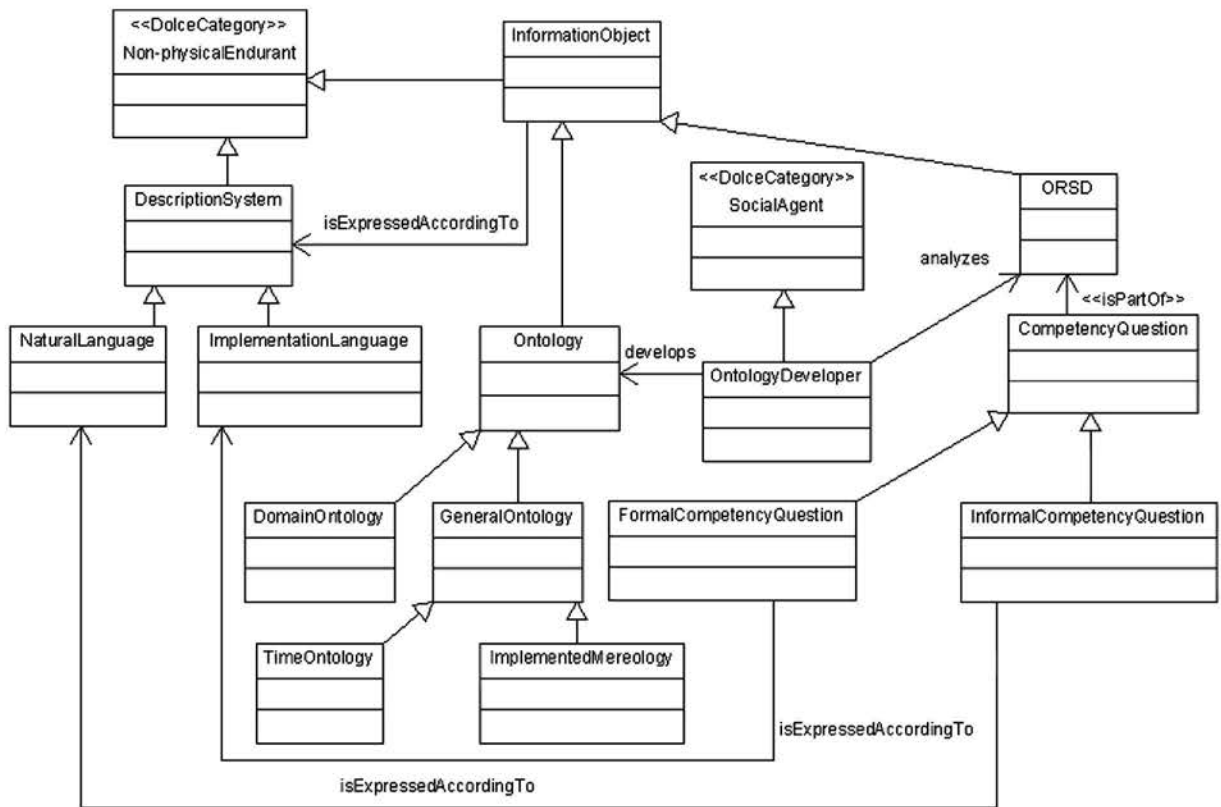


Fig. 1. UML class diagram with the basic concepts.⁴¹

⁴¹ All UML class diagrams were built using ArgoUML.

that the ontology should be machine readable, which excludes natural language. ‘Shared’ reflects the notion that an ontology captures consensual knowledge, that is, it is not private to some individual, but accepted by a group.” ([22], page 185).

We construe this definition to mean that an ontology is a set of formulas written in an implementation language.

Ontology is also a philosophical discipline, namely the branch of philosophy that deals with the nature and the organization of reality [23]. Let’s note that if nobody was devoted to ontology, it would disappear as a discipline. However, an information object can continue to exist although no one uses it. As a consequence, the meaning of ‘ontology’ illustrated in the UML diagram does not match with the definition of ‘ontology’ as a discipline.

- *Domain ontology*: an ontology that is reusable in a given specific domain (for example medical, pharmaceutical, engineering, law, enterprise, and automobile ([24], page 33).
- *General ontology*: (see the first paragraph of the Introduction section).
- *Implemented mereology*: a mereology that is written in an implementation language. The term mereology is introduced in Section 2.4.
- *Time ontology*: an ontology that represents temporal entities (*instant, temporal interval, duration, etc.*), their relations (*before, after, etc.*) and/or their representations (*second, minute, hour, time zone, etc.*).
- *Ontology developer*: a social agent who is in charge of ontology development.
- *Ontology requirements specification document (ORSD)*: a document that includes the purpose, the level of formality and scope of the ontology, the target group and the intended uses of the ontology, plus a set of requirements specifying the needs that the ontology to be built should cover ([14], page 38). An ORSD is an information object.
- *Competency question*: a requirement in the form of a question that an ontology must be able to answer ([25], page 6.1).
- *Formal competency question*: a competency question that is expressed in an implementation language (adapted from [25]).
- *Informal competency question*: a competency question that is expressed in natural language. Informal competency questions should be part of the ontology requirements specification document (ORSD).

2.3. Formal concept analysis

FCA is a subfield of *applied mathematics* based on the mathematization of concepts and concept hierarchies [26]. According to Wille ([26], page 447), “concepts can be philosophically understood as the basic units of thought formed in dynamic processes within

social and cultural environments". FCA formalizes the notion of *concept* relative to a *formal context* (FC). Next, we provide the FCA definitions useful for our work.

Definition 1. A *one-valued FC* is a tuple $K = (G, M, I)$, where G is a set whose elements are called objects, M is a set whose elements are called attributes, and I is a binary relation between G and M (that is, $I \subseteq G \times M$). $(g, m) \in I$ indicates that object $g \in G$ has the attribute $m \in M$ ([27], page 179).

Example 4. In the FC of Table 1, $G = \{\text{paracetamol, acetylsalicylic acid, dihydroergotamine, metoclopramide, morphine, pentazocine}\}$, $M = \{\text{interacts with ethyl alcohol, contraindicated in patients with liver failure, opiate}\}$, and I is the relation represented in Table 1. Consequently, the tuples of I are (paracetamol, interacts with ethyl alcohol), (paracetamol, contraindicated in patients with liver failure), (acetylsalicylic acid, interacts with ethyl alcohol), among others.

To define the formal concepts of the FC (G, M, I) , we need the following *derivation operators* defined for an arbitrary subset $X \subseteq G$ and $Y \subseteq M$ as follows [26]:

- $X^d \equiv \{m \in M \mid (g, m) \in I \forall g \in X\}$. For example, $\{\text{morphine, pentazocine}\}^d = \{\text{interacts with ethyl alcohol, opiate}\}$, since the two features that morphine and pentazocine have in common are that they interact with ethyl alcohol and they are opiate.
- $Y^d \equiv \{g \in G \mid (g, m) \in I \forall m \in Y\}$. For instance, $\{\text{interacts with ethyl alcohol, opiate}\}^d = \{\text{morphine, pentazocine}\}$ because the drugs that have the common properties of interacting with ethyl alcohol and of being opiate are morphine and pentazocine.

Definition 2. A *formal concept* of a FC $K = (G, M, I)$ is defined as a pair (A, B) with $A \subseteq G, B \subseteq M, A = B^d$, and $B = A^d$; A and B are called the *extent* and the *intent* of the formal concept (A, B) , respectively.

Example 5. The pair $(\{\text{morphine, pentazocine}\}, \{\text{interacts with ethyl alcohol, opiate}\})$ is the formal concept with extension $\{\text{morphine, pentazocine}\}$ and intension $\{\text{interacts with ethyl alcohol, opiate}\}$. It represents the concept *opiate* in the FC of Example 4.

Definition 3. We say that concept (A_1, B_1) is a *subconcept* of (A_2, B_2) if and only if $A_1 \subseteq A_2$ and $B_2 \subseteq B_1$.

Example 6. In the FC of Example 4, the concept $C_1 = (\{\text{morphine, pentazocine}\}, \{\text{interacts with ethyl alcohol, opiate}\})$ is a *subconcept* of $C_2 = (\{\text{morphine, pentazocine, metoclopramide, paracetamol, acetylsalicylic acid}\}, \{\text{interacts with ethyl alcohol}\})$ because the set of objects of C_1 is a subset of the set of objects of C_2 , and the set of attributes of C_2 is a subset of the set of attributes of C_1 .

The set of all formal concepts of a FC K together with the *subconcept* relation is always a lattice, called the *concept lattice* of K (see [28]). Fig. 2 shows the lattice of the FC of Example 4. The nodes in Fig. 2 are the formal concepts of the FC. If the concept C_1 is a *subconcept* of C_2 , then they are related by a line, and C_1 is positioned lower down.

Definition 4. A *many-valued FC* is a tuple $K = (G, M, (W_m)_{m \in M}, I)$, where G is a set whose elements are called objects; M is a set whose elements are called attributes; W_m is the set of possible values for the attribute $m \in M$; and I is a relation $I \subseteq G \times M \times W_m$ with the constraint $(g, m, w_1) \wedge (g, m, w_2) \Rightarrow w_1 = w_2$. $(g, m, w) \in I$ indicates that object $g \in G$ has value $w \in W_m$ for attribute $m \in M$ ([27], page 184).

Example 7. Table 2 shows a FC where the objects are different options (e.g. general ontologies to be reused, or drugs to prescribe one of them). The attributes of this FC are the different features that the options can have. If the options are ontologies, an example of feature is the *adaptation to the reasoner*, that is, whether the definitions and axioms need to be adapted to satisfy the existing reasoner constraints. If the options are drugs to be prescribed, an example of feature is *good price*.

- $G = \{\text{option 1, ..., option 4}\}$;
- $M = \{\text{feature 1, ..., feature 3}\}$;
- $(W_m)_{m \in M} = W_{\text{feature 1}} \cup \dots \cup W_{\text{feature 3}} = \{\text{high, low}\} \cup \dots \cup \{\text{yes, poorly}\}$;

Table 1
Formal context of drugs.^a

Drugs	Interacts with ethyl alcohol	Contraindicated in patients with liver failure	Opiate
Paracetamol	X	X	0
Acetylsalicylic acid	X	0	0
Dihydroergotamine	0	X	0
Metoclopramide	X	0	0
Morphine	X	0	X
Pentazocine	X	0	X

^a The Galicia FCA tool was used to draw all FCA figures [72].

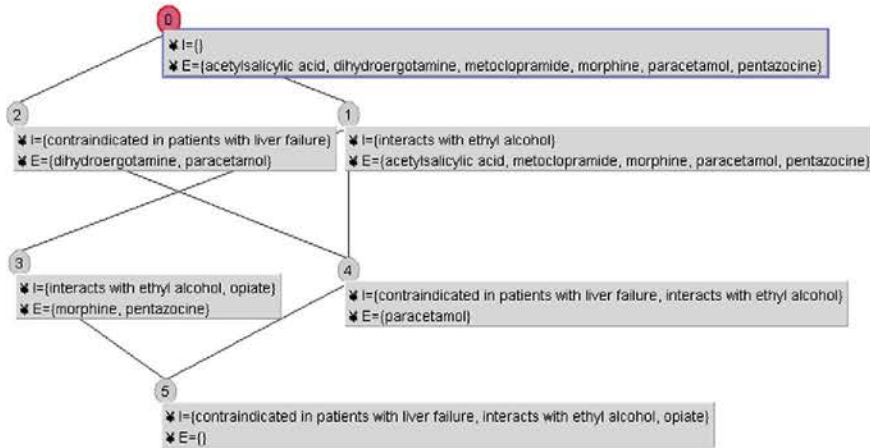


Fig. 2. Lattice for Example 5.

- an example of tuple of I is (option 1, feature 1, high), which means that the option 1 has the feature 1 with a high level; and
- given that each object for each attribute has at most one value, the constraint stated in the definition is satisfied.

Definition 5. A conceptual scale for a subset $B \subseteq M$ of attributes is a (one-valued) FC $S_B = (G_B, M_B, I_B)$ with $G_B \subseteq \times_{m \in B} W_m$ ([27]).

Example 8. Table 3.a shows a scale, valid for all the attributes of the set $B = \{\text{feature 1, feature 2}\}$, that represents $\text{high} \geq \text{low}$. Therefore, $S_B = (G_B, M_B, I_B)$, where $G_B = \{\text{high, low}\}$ is the set of objects; $M_B = \{\geq \text{high}, \geq \text{low}\}$ is the set of attributes; and I_B is the relation shown in Table 3.a. As we can observe, the object high has the properties of being greater or equal to high, and of being greater or equal to low.

Example 9. Table 3.b shows a scale, valid for all the attributes of set $B = \{\text{poorly, yes}\}$, that represents $\text{yes} \geq \text{poorly}$. Consequently, $S_B = (G_B, M_B, I_B)$, where $G_B = \{\text{yes, poorly}\}$ is the set of objects, $M_B = \{\geq \text{poorly}, \geq \text{yes}\}$ is the set of attributes, and I_B is the relation shown in Table 3.b.

Definition 6. Let S be the set of conceptual scales for the many-valued FC $K = (G, M, (W_m)_{m \in M}, I)$. The derived FC K_S is defined by $K_S = (G, M_S, I_S)$, where

- $M_S = \{(m, w) \mid m \in M, w \in W_m\}$; and
- $(g, (m, w)) \in I_S$ if and only if there exists $v \in W_m$, a scale $S_B \in S$ and $m \in B$ such that $(g, m, v) \in I$ and $(v, w) \in I_B$.

Example 10. If K is the FC of Example 7, then

- $S = \{S_{\{\text{feature 1, feature 2}\}}, S_{\{\text{feature 3}\}}\}$,
- $M_S = \{\text{feature 1} \geq \text{low}, \text{feature 1} \geq \text{high}, \dots, \text{feature 3} \geq \text{yes}\}$.
- I_S is the relation shown in Table 4.
- $(\text{option 1, feature 1} \geq \text{low}) \in I_S$ because $(\text{option 1, feature 1, high}) \in I$, and $\text{high} \geq \text{low}$ according to the scale $S_{\{\text{feature 1}, \dots, \text{feature 2}\}}$

As we can see in Table 4, which option is the best depends on the features that have more importance for the decider. Thus, for example, option 1 is better than option 2 if feature 3 is more important than feature 2, whereas option 2 is better than option 1 if feature 2 is more important.

2.4. Meronymy, mereology and mereologies

Meronymy studies part-whole relations from a linguistic and cognitive science perspective [29]. The term 'mereology' (without the indefinite article) refers to a sub-discipline of philosophy concerning the formal ontological investigation of the

Table 2
Formal context of options.

Option choosing	Feature 1	Feature 2	Feature 3
Option 1	[High]	[Low]	[Yes]
Option 2	[High]	[High]	[Poorly]
Option 3	[Low]	[High]	[Yes]
Option 4	[High]	[Low]	[Yes]

Table 3.a
Scale for features 1 and 2.

Scale f-1-2	> = High	> = Low
High	X	X
Low	0	X

part-whole relation [29]. The term 'a mereology' (with the indefinite article) refers to a particular object. A *mereology* is a formal theory¹⁸ of parts and associated concepts [9,3]. We can say 'a mereology' instead of 'the mereology' because different assumptions can be taken into account in the formalization of *parthood*. Therefore, different mereologies can be proposed.

In the following paragraphs we try to explain Varzi's minimal mereology. A wider set of mereologies is discussed in [16].

Most authors agree on the following core of axioms and definitions [16].

Axiom 1 (reflexivity). Every object of the universe of the discourse is a part of itself.

Axiom 2 (antisymmetry). If an object x is part of y , and y is part of x , then x and y are the same object.

Axiom 3 (transitivity). If x is part of y , and y is part of z , then x is part of z .

According to Keet and Artale [29], there are relations studied by meronymy that do not satisfy these axioms. Thus, there are usages of 'part of' in natural language that do not share the same properties as the *part-of* relation in mereology. The main semantic difference revolves around the transitivity of the part-whole relation. For example, Winston et al. [30] categorized the *member-collection* relation (tree-forest) as a part-whole relation. On the other hand, for Guizzardi [31], this relation is not transitive. For more information on the question of part-whole relation transitivity, some of the recommended readings are [30,32-34,29,31],

A number of additional mereological predicates can be introduced by their definition:

Definition 7. A *proper part* is a part that is not the individual itself.

Definition 8. X is a *direct part* of y if and only if x is a proper part of y and there is no part between x and y .¹⁹

Definition 9. x and y overlap if and only if there is z such that z is part of x and part of y , that is, the *overlap* relation is defined as a common part.

Definition 10. x and y underlap if and only if there is z such that x and y are parts of z , that is, the *underlap* relation is defined as a common whole.

Definition 11. The *disjoint* relation is the logical negation of *overlaps*.

Most authors stress that the axiom below should be also incorporated as a further fundamental principle of the meaning of *part-of* ([16] page 56). Other authors provide scenarios that could be counterexamples of this principle ([16] pages 56 and 57). So far, however, it has not been demonstrated that such supposed counterexamples have implications for computer applications.

Axiom 4 (weak supplementation principle). Every object x with a proper part y has another part z that is disjoint from y .

Example 11. The drug composition domain, for example, satisfies this principle. For instance, given that paracetamol is a proper part of Apiretal®, then Apiretal® has other parts that are disjoint from paracetamol, for example, glycerol and benzoic acid.

2.5. Time modeling

This section presents an intuitive introduction to some notions of time theories that will help non-experts to select time ontologies. These notions are based on [12]

- *Time points.* As a first intuitive approximation, time points can be viewed as points on the time line. For a study of the different ways of defining the concept of time point, see [35].
- *Time intervals.* Also as a first intuitive approximation, a time interval can be considered as the time between two time points. Hayes [58] also studies different ways of defining the concept of time interval.
- *Absolute and relative time.* On the one hand, time can be said to be represented in absolute terms when it is related to an event.²⁰

¹⁸ *Theory*: a set of formulas closed under logical implication ([73], page 54). For example, if the formulas F_1 and $F_1 \rightarrow F_2$ belong to a theory, then F_2 , $F_1 \wedge F_2$, $F_2 \wedge F_1$, etc., do so too. Informally, we can say that an ontology implements the axioms of a theory. However, given that the term *theory* belongs to the meta-language instead of the language, the formal inclusion of a relation between *theory* and *ontology* could lead to circular or unpredictable definitions.

¹⁹ <http://hes.science.uva.nl/projects/NewKACTUS/library/lib/mereology.html>.

²⁰ <http://www.cs.auc.dk/~csj/Glossary/>.

Table 3.b
Scale for feature 3.

Scale f-3	>= Poorly	>= Yes
Poorly	X	0
Yes	X	X

Example 12. The 1st of May of 1986 can be associated with the registration date of a particular drug.

On the other hand, we say that the valid time of a fact is represented in relative terms when it is related to the valid time of another fact.

Example 13. Drug 2 was registered later than drug 1.

- *Relations between time entities.* These relations can be established between time points (e.g., the date 8 May 2010 comes before 29 December 2010), between time intervals (e.g., [2008, 2009] is before [2010, 2011]) or even between points and intervals (e.g., 2010 is before [2011, 2012]). Allen [36] identified the most well-known relations between intervals. Some are shown in Fig. 3. Note that I_1 is inside I_2 if and only if I_1 is part of I_2 . I_1 is equal to I_2 if and only if I_1 is part of I_2 and I_2 is part of I_1 . *Overlap* is the mereological relation defined in Definition 9 and applied to time intervals. Sowa [15] shows how to define the *before* relation in terms of mereology; that is, relations in Fig. 3 can be represented by a mereology. In fact, it is habitual to use the *part_of* relation as equivalent to *inside or equal_to* in time modeling (see, for example, [37,38]).
- *Convex and non-convex intervals.* These notions can identify periodic intervals separated by gaps (e.g., “every Wednesday”). They are known as non-convex intervals. Conversely, convex intervals are continuous intervals, that is, they do not have “separate pieces” (e.g., from 14 April 2010 to 29 December 2010).
- *Open and closed intervals.* Interval endpoints may or may not be inside the interval.

Example 14. The interval [1985, 1986] is closed on the left and open on the right.

- *Proper intervals.* An interval with different endpoints is called a proper interval [39].

Example 15. [1985, 1986] is a proper interval, but [1985, 1985] is not

- *Temporal granularities* refer to the detail level used to represent temporal entities.

Example 16. The duration of an event in days can be converted into minutes.

This is an easy operation to perform. However, the conversion is not always straightforward. Consider, for instance, converting an interval between two different calendar days into working days [40].

- *Total ordering.* Total ordering means that, for every pair of time points t_1 and t_2 , necessarily $t_1 < t_2$ or $t_2 < t_1$. It is useful to make this assumption in many applications. However, this assumption may not be always right, especially in distributed systems.

Example 17. The time point t_1 at which process P_1 is started on machine M_1 and the point t_2 when process P_2 is started on machine M_2 may not be linked by either relation $t_1 < t_2$ or $t_2 < t_1$. Lamport [41], for instance, shows the phenomenon of partial ordering between events in distributed systems, and provides an algorithm to solve synchronization problems in the access to shared resources.

- *Infinity.* An infinite interval is one that is not limited in the past or in the future. There are two common ways of allowing infinitely long intervals.²¹ In the first approach (see Fig. 4), an infinite interval in the past and in the future is modeled by one point in the negative infinite and another in the positive infinite. In the second approach, an infinite interval in the past and in the future is modeled by an interval with no beginning and an interval with no ending. As we cannot consider points in the infinite using the second approach, the two approaches are incompatible with the same ontology. This is why some ontologies leave this part of the modeling open.
- *Density.* Density is used to represent that there is a third distinct point between any two distinct points. Assuming that there is no other second between second s and second $s + 1$, then time is viewed as an ordered set of seconds and density is ruled out. The same applies if time is represented as an ordered set of minutes, or hours, etc. However, we can always assume that

²¹ <http://www.cs.rochester.edu/~ferguson/daml/>

Table 4

Derived formal context for the poll example.

Option choosing-derived	Feature 1 >= high	Feature 1 >= low	Feature 2 >= high	Feature 2 >= low	Feature 3 >= poorly	Feature 3 >= yes
Option 1	X	X	0	X	X	X
Option 2	X	X	X	X	X	0
Option 3	0	X	X	X	X	X
Option 4	X	X	0	X	X	X

a second establishes the interval containing a time point. According to this approach, if we say that point P is at second s , we are specifying the interval of time containing P . In this case, density can be assumed.

- *Isomorphism to the real numbers.* The set of real numbers is very often the model of the time theory. Note that if we take this model exactly as it is, then we assume the properties of density, convexity, and total ordering; we also assume that there are no points in the infinite.

3. Methodological guidelines for reusing general ontologies

The goal of reusing general ontologies is to find, select and then, to integrate general ontologies into the domain ontology being developed. Fig. 5 is a diagram illustrating the top-level use cases of our methodological guidelines. According to Cockburn [42] (page 1), “a use case describes the system's behaviour under various conditions as the system responds to a request from one of the stakeholders, called the primary actor”. If we particularize this definition to our methodological guidelines, then (1) the ontology developer plays the role of stakeholder; and (2) the members of the ontology development team responsible for the reuse of general ontologies, aided by software tools, play the role of the system. The arrows of the figure relate each use case with its inputs and outputs. The ORSD conditions the whole process. As the diagram shows, the *reuse of general ontologies* includes the following use cases.

1. *Conduct a comparative study of reusable general ontologies.* The purpose of this use case is to find and compare general ontologies. We propose to build a table to represent the comparative study. This table is essentially a FC. The contextual objects are the ontologies to be compared, and the attributes are both the definitions and the axioms that these ontologies include or features of the models that they describe (e.g., the definition of time points, the assumption of different granularities).
2. *Make a decision on which general ontology is going to be reused.* The table output by the comparative study is used to select the general ontology that best fits the requirements of the ontology to be developed. FCA will be used to make this decision by structuring the different options (candidate general ontologies) in lattices. Such lattices are obtained and used during this step of decision making. In the following sections the reader will better understand why we are using several lattices.
3. *Assemble the whole ontology.* This use case specifies that the ontology development team has to customize and integrate the most appropriate general ontology into the ontology to be developed.

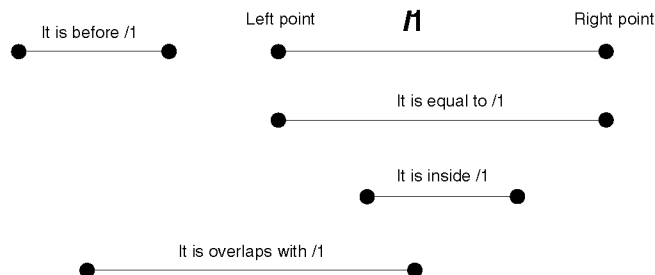
Note that if there is a previous comparative study, then Use Case (1) does not have to be carried out, and thus the reuse process should start directly from Use Case (2).

Use Cases (1), (2) and (3) are explained in detail below. For the sake of clarity, we consider the reuse of just one general ontology since the reuse of more general ontologies implies the iterative performance of the above process.

The heuristics shown in the following sections are coherent with the following high level criteria:

Criterion 1 An analysis, at least, at the following levels of the CQs provides valuable information to guide the reuse of general ontologies: lexical (its collection of words), syntactic (the relations between its words) ([43], page 6), and semantic (the relations of its words to the entities to which they are applicable) ([43], page 6).

Criterion 2 The ORSD may contain CQs that refer to entities that can be described by means of axioms and definitions included in general ontologies.

**Fig. 3.** Relations between intervals.

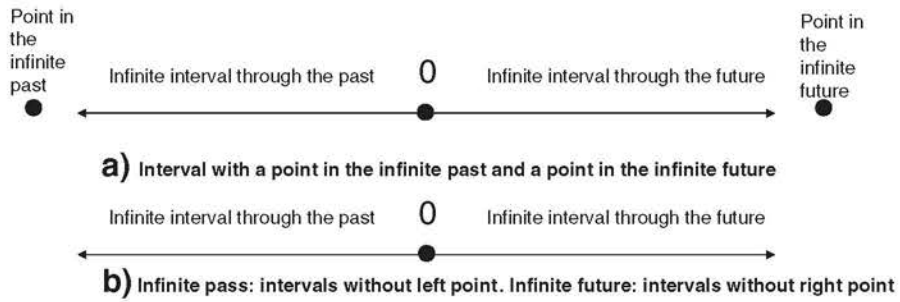


Fig. 4. The two usual representations of infinite temporal intervals.⁴²
⁴² <http://www.cs.rochester.edu/~ferguson/dam/>.

The way of obtaining the CQs is outside of the scope of this paper. Here we just provide some of the guidelines that we follow, such as (1) the ontology developer should try to cover the maximum part of the domain with the minimum quantity of CQs; (2) the domain should be divided into sub-domains to ease the elaboration of CQs (e.g. drug products, active ingredients, laboratories); (3) the future uses of the ontology should be anticipated (for example, by checking contraindications, searching for a drug with some particular features, or searching for laboratories that sell drugs with a particular active ingredient), and (4) “the CQs should be defined in a stratified manner with higher level questions requiring the solution of lower level questions” ([25], page 6.3) (e.g. *is paracetamol appropriate for patient X?* requires the solution of *what are the paracetamol contraindications?*, *what are patient X’s illnesses?*). It can be added that the process of obtaining CQs is not automatic, but creative.

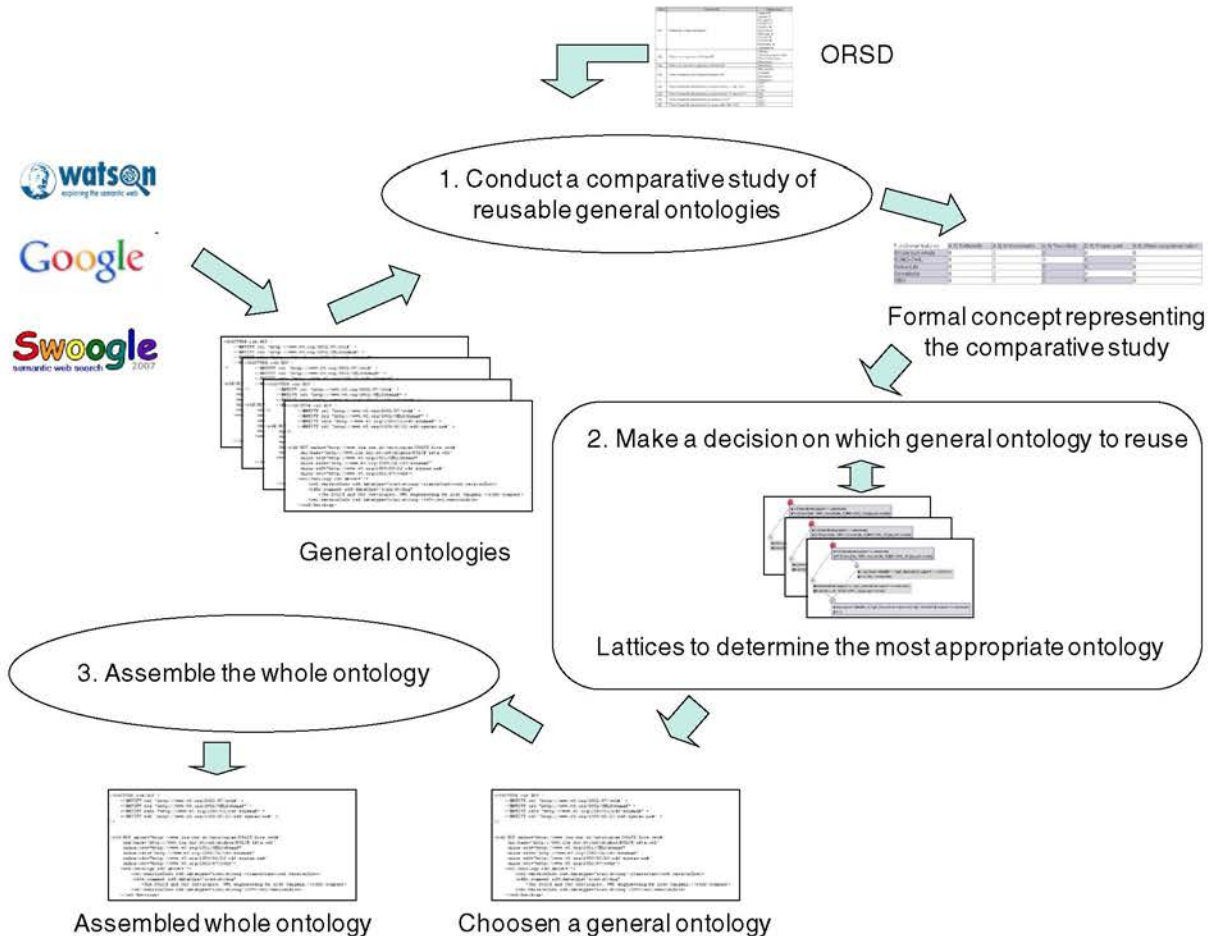


Fig. 5. Level 1 use cases.

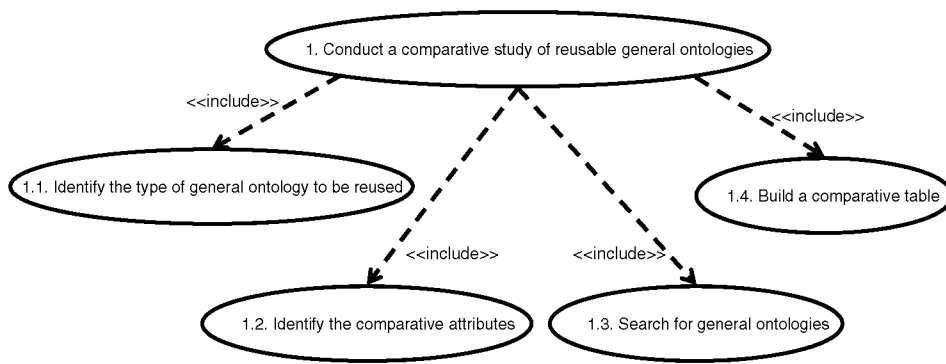


Fig. 6. Use case diagram for conduct a comparative study of reusable general ontologies.

3.1. Use Case 1. Conducting a comparative study of reusable general ontologies

Fig. 6 illustrates the cases included in Use Case 1 (*Conduct a comparative study of reusable general ontologies*) following the UML notation. The result of this comparative study is represented as a table that can be interpreted as a one-valued FC.

3.1.1. Use Case 1.1. Identifying the type of general ontology to be reused

The goal of this use case is to determine whether general ontologies should be reused, and, if so, what type of general ontology should be reused. This section reports some heuristics that are useful for deciding the type of general ontology to be reused. Such heuristics are applied to all CQs for the ontology being built.

Heuristic 1 (mereology reuse I). We should *reuse an implemented mereology* if the conjunction of the following conditions is satisfied: (CM₁)²² the CQ refers to a relation *R* that establishes an order; and (CM₂) *R* fulfills the weak supplementation principle. Fig. 7 illustrates relations taken from [29] that satisfy this heuristic. Fig. 7 assumes that a *space region* is an ordinary space, i.e., it can be a geographical, cosmological, anatomical and topographical space⁷ among others. A distinction is made between 2D regions and 3D regions.

1) The *is_located_in* relation *R*, which relates two 2D regions.

Example 18. Sant Cugat del Vallès (the location of *Pharmacia Iberia @ headquarters*) is (located) in the province of Barcelona. We can check that the two conditions are met:

- Satisfaction of CM₁: a place *p*₁ can be located at place *p*₂, *p*₂ to *p*₃, etc., such that an order *p*₁ <::: *p*₂ <::: *p*₃ is established, where '<:::' means *is located in*. For instance, Sant Cugat del Vallès <::: the province of Barcelona <::: Spain.
- Satisfaction of CM₂: if *x* ≠ *y* and *x* is located in *y*, then *y* necessarily has another place *z* such that *z* does not overlap with *x*. For example, given that Sant Cugat del Vallès is located in the province of Barcelona and Sant Cugat del Vallès is not equal to the province of Barcelona, then the province of Barcelona has other places.

2) The *is_contained_in* relation *R*, which relates two 3D endurants.

Example 19. A small bottle is (contained) in its box.

3) The *is_structural_part_of* relation *R*, which relates two endurants.

Example 20. The hydroxyl group is (structural) part of paracetamol.

4) The *is_involved_in* relation *R*, which relates two perdurants.

Example 21. Nitrating phenol is (involved) in synthesizing paracetamol.

Another heuristic derives from Heuristic 1.

Heuristic 2 (mereology reuse II). If the CQ refers to a relation *S* that is a subrelation of an *R* that meets conditions CM₁ and CM₂, then an implemented mereology should be reused.

²² CM means *condition for the reuse of a mereology*.

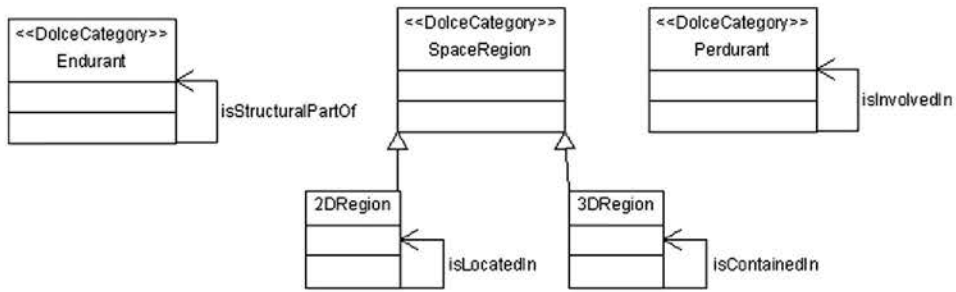


Fig. 7. Cases that conform to Heuristic 1.

Example 22. *Main_component_of* is a subrelation of *structural_part_of*.

The heuristic used to identify the reuse of time modeling is based on the words that appear in the CQ. It has been taken from Section 2.1 of “*Instruction Manual for the Annotation of Temporal Expressions*”, by Lisa Ferro [44], and Section 2.3 of “*TimeML Annotation Guidelines Version 1.2.1*” [45].

Heuristic 3 (time modeling reuse). We should *reuse a time ontology* if some of the following words appear in the CQ:

- Nouns: ‘day’, ‘minute’, ‘weekend’, ‘midnight’, ‘millennium’, ‘era’, ‘semester’, ‘summer’, [the] ‘future’, [the] ‘past’, ‘month’.
- Proper names: ‘Monday’, ‘January’, ‘New Year’s Eve’, ‘Washington’s Birthday’.
- Specialized time patterns: 8.00, 12/2/00, 1994, 1960s.
- Adverb: ‘currently’, ‘hourly’, ‘daily’, ‘monthly’.
- Noun/adverb of time: ‘today’, ‘yesterday’, ‘now’.
- Prepositions of time: ‘on’, ‘in’, ‘at’, ‘from’, ‘to’, ‘before’, ‘after’, ‘during’.
- Conjunctions of time: ‘before’, ‘after’, ‘while’, ‘when’.

We have omitted the *number* item proposed in [44] because we consider it to be too imprecise. Ferro’s example for such an item is 3 (as in “*He arrived at 3.*”), *three*.

Given that a mereology can be used to represent time (see Section 2.5), we also have the following heuristic.

Heuristic 4 (mereology reuse III). If the CQ refers to a time relation (using the words ‘before’, ‘after’, ‘in’, ‘during’, etc.), then an implemented mereology should be reused.

3.1.2. Use Case 1.2. Identifying comparative attributes

The ontology development team should study the general theories identified in Use Case 1.1, and extract attributes to compare general ontologies.

In this paper, we provide some guidelines for mereologies and theories of time (Sections 2.4 and 2.5, respectively). To elaborate a similar description of other general theories (e.g., mereotopologies) to those appearing in Sections 2.4 and 2.5, readers are referred to the literature on the topic. In this case, the identification of comparative attributes may require the help of any ontologist familiar with general theories.

Note that this use case can be time and resource consuming, but its result can be exploited in future projects.

3.1.3. Use Case 1.3. Searching for general ontologies

The ontology development team should search for comparable ontologies using the attributes identified in Use Case 1.2.

The use case output is a set of candidate general ontologies. To obtain such candidate ontologies, any general-purpose search engine (e.g., Google²³), ontology search engines (e.g., Swoogle²⁴ and Watson¹), repositories (e.g., Protégé ontology library²⁵ and

²³ <http://www.google.es/>.

²⁴ <http://swoogle.umbc.edu/>.

²⁵ <http://protege.cim3.net/cgi-bin/wiki.pl?ProtegeOntologiesLibrary>.

Table 5
Time ontologies and mereologies (Use Case 1.3).

Identified ontologies	Project or institution	General theory
Single part whole ^a	W3C	Mereology
SUMO-OWL ^b	IEEE Standard Upper Ontology working group	Mereology time
DOLCE-Lite ^c	Italian Research Council (CNR)	Mereology time
Oswebsite ^d	OS Open data	Mereology
OBO ^e	Open Biological and Biomedical Ontologies ^f	Mereology
OWL-Time ^g	W3C	Time
AKT-Time ^h	Advanced Knowledge Technologies (AKT)	Time

^a <http://www.w3.org/2001/sw/BestPractices/OEP/SimplePartWhole/part.owl>.

^b <http://www.ontologyportal.org/translations/SUMO.owl.txt>.

^c <http://www.loa-cnr.it/ontologies/DOLCE-Lite.owl>.

^d <http://www.ordnancesurvey.co.uk/oswebsite/ontology/Mereology.owl>.

^e <http://www.obofoundry.org/>.

^f <http://www.berkeleybop.org/ontologies/obo-all/relationship/relationship.owl>.

^g <http://www.w3.org/2006/time>.

^h <http://www.aktors.org/ontology/support>.

the Open Biological and Biomedical Ontologies²⁶), or well-known general ontologies (e.g., DOLCE-Lite²⁷ and SUMO-OWL²⁸) can be used.

Example 23. Table 5 lists a representative set of time ontologies and mereologies. Unfortunately, we were unable to load OpenCyc in OWL²⁹ in Protégé,³⁰ in NeOn Toolkit³¹ or in the free version of TopBraid composer.³² On the other hand, BFO³³ provides light representations for time point and time interval. Finally, we did not find any OWL implementation for Sowa's ontology³⁴ (see also [15]).

3.1.4. Use Case 1.4. Building a comparative table

The goal here is to compare the candidate general ontologies output in Use Case 1.3 using the attributes output in Use Case 1.2. For usability, we propose to represent this comparative study as a table. The table columns list the attributes, and the rows, the candidate ontologies, forming a one-valued FC.

Example 24. Tables 6 and 7 are examples of a comparative table of ontologies implementing mereologies and time theories.

3.2. Use Case 2. Making a decision on which general ontology is to be reused

Fig. 8 shows the decomposition of the use case addressed in this section. Each use case will be described below.

3.2.1. Use Case 2.1. Reformulating the CQs and add linking axioms

The main goal of this use case is to reformulate the CQs included in the ORSD of the domain ontology that is being developed with typical vocabulary of the general theories identified in Use Case 1.1.

Example 25. If an implemented mereology is going to be reused in a pharmaceutical product ontology (PPO), then each CQ of the ontology should, wherever possible, be formulated by means of mereology vocabulary.

Another goal is to identify axioms that link CQ terms to general ontology terms.

For this use case, we propose a series of heuristics both to transform CQs in order to reuse a mereology and to identify linking axioms.

Heuristic 5 (overlap). If we want to know if two objects have common parts, then we should reformulate the CQ to include the term *overlap*.

²⁶ <http://www.obofoundry.org/>.

²⁷ <http://www.loa-cnr.it/ontologies/DOLCE-Lite.owl>.

²⁸ <http://www.ontologyportal.org/translations/SUMO.owl.txt>.

²⁹ <http://sw.opencyc.org/>.

³⁰ <http://protege.stanford.edu/>.

³¹ http://neon-toolkit.org/wiki/Main_Page.

³² http://www.topquadrant.com/products/TB_Composer.html.

³³ <http://www.ifomis.org/bfo>.

³⁴ <http://www.jfsowa.com/ontology/>.

Table 6

A partial view of a comparative study of mereology ontologies (Use Case 1.4).

Functional features	A.1) Reflexivity	A.2) Antisymmetry	A.3) Transitivity	D.1) Proper part	A.4) Weak supplementation
Single part whole	0	0	X	0	0
SUMO-OWL	0	0	0	X	0
Dolce-Lite	0	0	X	X	0
Oswebsite	0	0	X	0	0
OBO	0	0	X	X	0

Heuristic 6 (underlap). If we want to know if two objects have common wholes, then we should reformulate the CQ to include the term *underlap*.

Heuristic 7 (proper part). If we want to know the parts of an object not including the object itself, then we should reformulate the CQ to include the term *proper part of*.

Heuristic 8 (direct part). If we want to know the first level of decomposition of an object, then we should reformulate the CQ to include the term *is direct part of*.

Notice that it is not always easy to know if object o_1 is really a direct part of o_2 . For example, we might know that drug d_1 is composed of substances s_{11} and s_{12} , but there may be an s_1 such that s_{11} is part of s_1 , whereas s_1 is, in turn, part of d_1 , even if this is not provided in the information about d_1 . Consequently, we have to be very careful when reusing *direct part of*.

Heuristic 9 (part of). If we want to know the parts of an object, including the object itself, then we should reformulate the CQ to include the term *is part of*. A typical case is that in which the mereological relation appears in a composition of relations. This is shown more clearly in Example 26.

Heuristic 10 (disjoint). If we want to know which parts of object o_1 are not in object o_2 , then we should reformulate the CQ to include the term *are disjoint*.

Heuristic 11 (subrelations of *part_of*). If we applied Heuristic 2, then we should introduce a linking axiom establishing that S is a subrelation of *part_of*.

Example 26. Fig. 9 shows some applications of the above heuristics applied to (metabolic) pathways, reactions, structure of compounds and locations, among others, (“*In biochemistry, metabolic pathways are series of chemical reactions occurring within a cell*”).³⁵ We have used the KEGG code for reactions.³⁶ Concerning the Pharmacia Iberia ® example, given that its headquarters are located in a particular place of Spain, they are located in Spain. Note that the system may only have the information ‘Pharmacia Iberia ® is located in Spain’, but no information of the specific place. To answer this CQ, Spain has, in this case, to be considered as a part of itself.

Below, we list heuristics for adding linking axioms to reuse time modeling; Fig. 10 shows examples of these heuristics.

Heuristic 12 (time points). If we are not interested in the endpoints of the temporal entity returned by the CQ, then we should introduce a linking axiom in the general ontology to relate a domain ontology concept to time points.

Heuristic 13 (time intervals). If we are interested in the endpoints of the temporal entity returned by the CQ, then we should introduce a linking axiom in the general ontology to relate a domain ontology concept to time intervals.

3.2.2. Use Case 2.2. Identifying the attributes of the general ontology to be reused

The goal here is to identify which typical definitions, axioms and other properties of the general theories identified in Use Case 1.1 are needed in the domain ontology to be developed. For example, the domain ontology may need *part_of* transitivity but not *underlap*.

³⁵ We have shown here the Wikipedia definition of the term *metabolic pathway* because we think that it is intuitive enough for non-specialists in biochemistry.

³⁶ <http://www.genome.jp/kegg/>.

Table 7

Partial view of a comparative study of time ontologies (Use Case 1.4).

Functional features	Time points	Time intervals	Absolute time	Relations between temporal entities	Modeling of conv...	Modeling of...
OWL-Time	X	X	X	X	X	0
SUMO-OWL	X	X	0	X	0	0
AKT-Time	X	X	X	0	X	0

There follows a series of heuristics for making a decision on mereology reuse in this use case.

Heuristic 14 (axioms always to be reused). If they can be implemented, we recommend reusing both the reflexivity and the antisymmetry of *part_of*, and the weak supplementation principle.

This heuristic has the purpose of ensuring the right meaning of *part_of* and *proper_part_of*.

Example 27. The CQ on Pharmacia Iberia® in Example 26 works thanks to the reflexivity of *part_of*.

Example 28. Suppose we have the following incorrect facts: (1) compound 1 is part of compound 2; (2) compound 2 is part of compound 1; and (3) compound 1 and compound 2 are different. The inference engine would detect the mistake by means of the antisymmetry of the *part_of* relation.

Example 29. It is wrong to say that paracetamol is the only proper part of Apiretal®, although it is its main active ingredient. The weak supplementation principle would be used by the inference engine to detect the incorrect fact.

Heuristic 15 (transitivity). Suppose that the ontology should model X that has parts X_1, X_2, \dots, X_n , and some of these parts, e.g., X_i has parts $X_{i1}, X_{i2}, \dots, X_{im}$, that is, X has several levels of parts. Besides, we want to know all the levels when we ask, *which are the parts of X ?* In such a case, the transitivity axiom should be reused.

Example 30. Suppose we formulate the CQ *Does Apiretal® have any alcohol group?* The inference engine can answer this question using transitivity, because Apiretal® has paracetamol, and paracetamol has an alcohol group.

Heuristic 16 (reuse of definitions). The new terms appearing in the CQs after the reformulation proposed in Use Case 2.1 should be included in the general ontology.

Example 31. If the CQ has been reworded to include the term ‘overlaps’, then the relation *overlaps* should be reused.

The heuristics for identifying time modeling attributes are as follows:

Heuristic 17. Heuristic 16 is also valid for time.

Example 32. If the CQ has been reworded to include the term ‘time point’, then the concept *time point* should be reused.

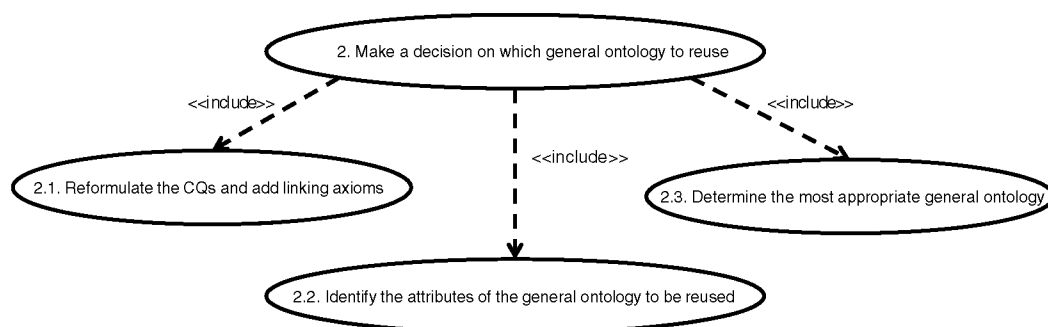


Fig. 8. Use case diagram. How to make a decision on which general ontology is to be reused.

EXAMPLES

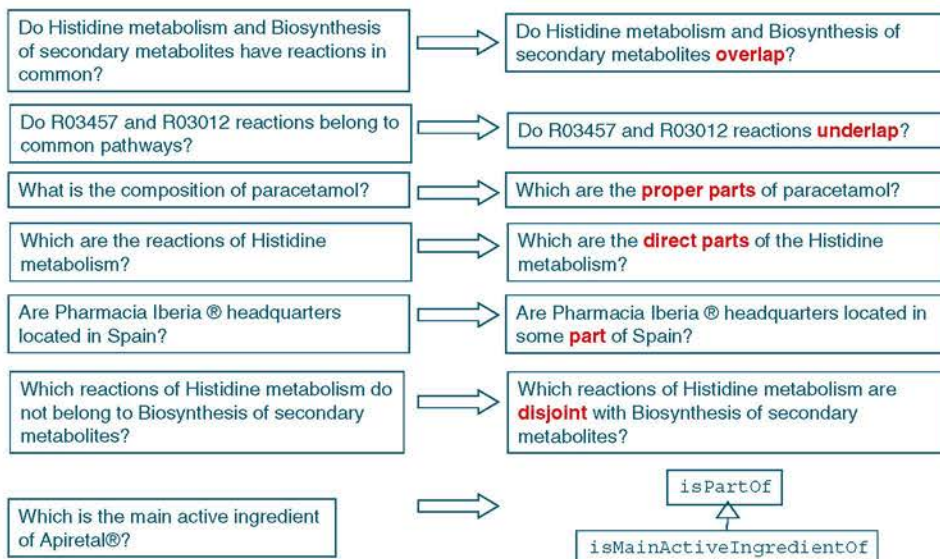


Fig. 9. Examples of the use of heuristics (a) to transform CQs to reuse a mereology and (b) to identify the linking axioms (Use Case 2.1).

Heuristic 18 (absolute time). If the CQ can be expressed as *When does X happen?*, then the ontology should model absolute time (by means of time units).

Example 33. Suppose we formulate the CQ, *When was Apiretal® registered?* The answer to this CQ associates a date with the registration of the drug, that is, it associates a time with an event.

Heuristic 19 (relations between time entities). If the CQ can be expressed such as *Does X happen before Y?*, *Does X happen after Y?*, *Does X happen at the same time as Y?* among others, where X and Y are time entities or events, then the ontology should model relations between time entities.

Example 34. Is the Fungusol® registration date *before* Apiretal®'s?

Example 35. Is the Fungusol® registration date *before* 1980?

Heuristic 20 (relative time). If the CQ can be expressed according to the pattern of Heuristic 19, where, X and Y are events, then the ontology should model relative time.

Example 36. (See Example 34).

The other heuristics for the reuse of time modeling are listed in Annex I.

The axioms and definitions generated by the heuristics have been implemented in at least one of the following languages: OWL, OWL + SWRL [46] and Prolog.³⁷ It should be remembered that Prolog works under the closed world assumption (CWA). Therefore, if a formula *F* is neither implemented nor deduced, then we assume that $\neg F$. This assumption changes the meaning of the terms defined in Section 2.4.

Example 37. If we represent the *disjoint* relation through the following Prolog rule:

`disjoint(X, Y) :- \+ overlaps(X, Y)`

when the common parts of *x* and *y* are unknown, then *x* and *y* are disjoint. Consequently, the implementation of this term in Prolog does not fit Definition 11.

³⁷ We have carried out our experiments with SWI Prolog (<http://www.swi-prolog.org/>).

EXAMPLES

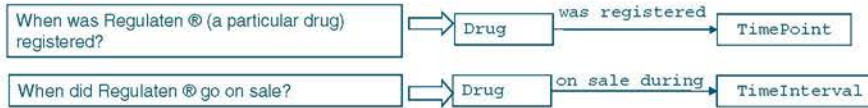


Fig. 10. Examples of the use of Use Case 2.1 heuristics.

We have not studied the impact that higher order theories that use an intermediate position between CWA and Open World Assumption (OWA) (e.g. [47,48]) have in the meaning of the terms. A possible solution is to introduce the notion of specification which is a pair describing a data structure together with the axioms that the structure should satisfy (see e.g., [49] for more details). Nevertheless, to facilitate the execution of Use Case 2.3, the ontology developer should mark the identified attributes in the table containing the results of the comparative study output of Use Case 1.4.

3.2.3. Use Case 2.3. Determining the most appropriate general ontology

The goal of this use case is to determine which of the candidate general ontologies identified in Use Case 1.3 is the best to be reused in the ontology being developed. Besides the comparative study output in Use Case 1.4, the following four dimensions should be analyzed. They have been gleaned from our own experience, although some have also been provided by other authors.

- *Reuse cost.* This item refers to the estimated financial and temporal cost of reusing the candidate ontology. In this case, the following attributes should be analyzed:
 - *Reuse financial cost.* It refers to the estimated financial cost of accessing and using the candidate ontology. The cost of acquisition and/or exploitation of the candidate ontology should be taken into account if it is subject to licensing terms [50].
 - *Required reuse time.* It refers to the estimated time taken to access the candidate ontology. Access time should be considered if the candidate ontology is available on slow servers or servers with bad connectivity.
- *Understandability effort.* It refers to the estimated effort of understanding the candidate ontology. In this case, the following attributes should be analyzed:
 - *Documentation quality.* It refers to whether there is any communicable material that describes or explains different aspects of the candidate ontology (e.g., modeling decisions). The documentation should describe the knowledge pieces represented in the ontology in such a manner that a non-expert could understand the knowledge that it represents [51].
 - *External knowledge source availability.* It refers to whether the candidate ontology makes references to documentation sources, and whether experts are available.
 - *Code clarity.* This attribute refers to whether the code is easy to understand and modify, that is, if the knowledge entities follow unified patterns and if patterns are clear [51]. It is handy to use the same pattern to make sibling definitions, thus increasing ontology understanding and making it easier to add new definitions [52]. All of which would improve the clarity of the ontology and its monotonic extensibility. *Code clarity* also refers to whether the code is documented, that is, if it includes clear and consistent definitions and comments for the knowledge entities represented in the candidate ontology.

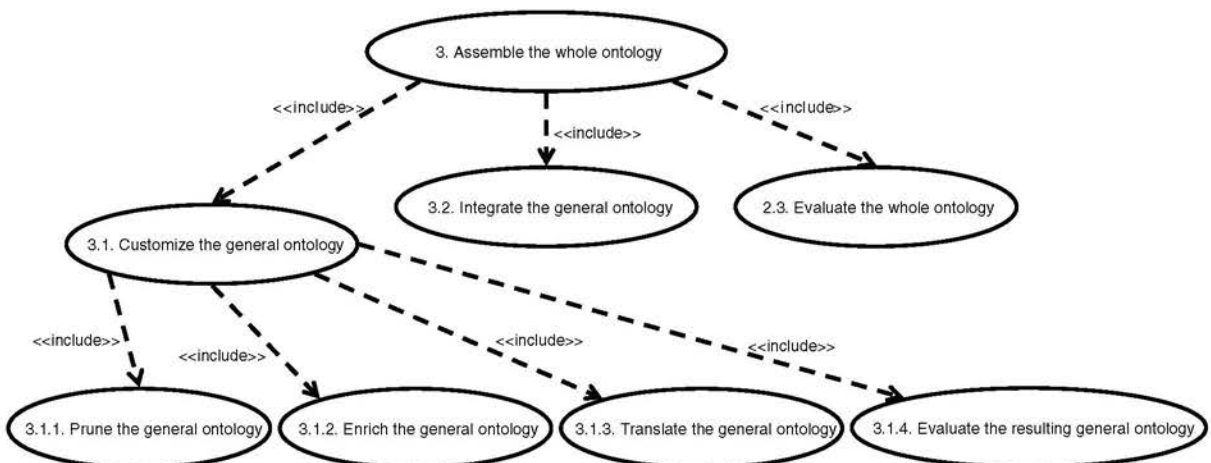


Fig. 11. Use case diagram to assemble the whole ontology.

Table 8

Informal host competency questions.

CQ id	Informal CQ	Sample answer
CQ ₁	Which drugs contain paracetamol?	Algidol ® ^a Apiretal ® Bisolgrip ® Cortafriol ® Dolgesic ® Dolostop ® Efferalgan ® Frenadol ® Gelocatil ® Pharmagrip ® Termalgin ®
CQ ₂	What is the composition of Frenadol®?	Caffeine Chlorpheniramine citrate Dextromethorphan Paracetamol
CQ ₃	What is the main active ingredient of Frenadol®?	Paracetamol
CQ ₄	With which substances does Frenadol® interact?	Ethyl alcohol Isoniazid Propranolol Rifampicin
CQ ₅	Which Frenadol® manufacturing lot expires before 11 May 2012?	C63 ^b C125 C243
CQ ₆	Which Frenadol® manufacturing lot expires before 30 April 2010?	C63
CQ ₇	Which Frenadol® manufacturing lot expires in 2010?	C63 C125
CQ ₈	Which Frenadol® manufacturing lot expires after May 2012?	C243

^a All the answers to CQ₁ are drug names.^b The manufacturer assigns an identifier to each set of particular specimens of a drug product so that it is traceable from manufacture to consumption.

□ *Integration effort.* It refers to the estimated effort needed to integrate the candidate ontology into the ontology being developed. In this case, the following attributes should be analyzed:

- *Knowledge extraction adequacy.* It refers to whether it is easy to identify and extract parts of the candidate ontology to be reused. For example, it is very hard to extract the part of the knowledge of interest from large and non-modularized ontologies with SWRL rules.
- *Naming conventions adequacy.* It refers to whether both ontologies (the candidate ontology and the ontology under development) follow the same rules for naming the different ontology components (e.g., concept names should start with capital letters, relation names should start with non-capital letters) [53].
- *Implementation language adequacy.* It refers to whether both languages (the language of the candidate ontology and the ontology under development) are the same or are, at least, able to represent similar knowledge with the same granularity.
- *Knowledge clash.* It refers to whether the candidate ontology and the ontology being developed contain contradictory bits of knowledge.

Table 9

Identification of the types of general ontologies to be reused according to the heuristics listed in Use Case 1.1.

CQ identifier	Application of heuristics	Type of general ontology for reuse
CQ ₁ and CQ ₂	Heuristic 1.CM ₁) A substance s_1 can be part of a substance s_2, s_2 of s_3 , etc., such that an order $s_1 <:: s_2 <:: s_3$ is established, where '<::' means <i>structural_part_of</i> . For example, alcohol <:: paracetamol <:: Frenadol. Heuristic 1.CM ₂) if $s_1 \perp s_2$ and $s_1 <:: s_2$, then s_2 necessarily has a component s_3 such that s_3 does not overlap s_1 . For example, given that paracetamol \perp Frenadol® and paracetamol <:: Frenadol®, then there are other substances (e.g., dextromethorphan and chlorpheniramine) that are structural parts of Frenadol®.	Mereology
CQ ₃	Heuristic 2. Every active ingredient of a drug is a structural part of the drug. Therefore, <i>main_active_ingredient_of</i> is a sub-relation of <i>part_of</i> .	Mereology
CQ ₄	Heuristic 1. To find out whether a drug interacts with other substances, it is necessary to know the interaction of each of its structural parts with other substances.	Mereology
CQ ₅ , CQ ₆ , CQ ₇ and CQ ₈	Heuristic 3. <i>There are specialized time patterns: 11 May 2012, 30 April 2010</i> <i>There is a preposition of time: before.</i>	Time modeling

Table 10

Analysis and transformation of the CQs, and addition of linking axioms according to the heuristics for mereology reuse listed in Use Case 2.1.

Heuristic	Competency question	Action	Result of the action
Heuristic 7. We want to know the parts of an object not including the actual object.	CQ ₂) What is the composition of Frenadol®?	Reformulate the CQ to include the term <i>is proper part of</i> .	Which are the proper parts of Frenadol®?
Heuristic 9. We want to know the parts of an object, including the actual object.	CQ ₁) Which drugs contain paracetamol? (The actual substance is included because the user could ask for a drug directly)	Reformulate the CQ to include the term <i>is part of</i> .	Which drugs is paracetamol part of?
Heuristic 11. The CQ refers to a relation S that is a sub-relation of R that satisfies conditions CM ₁ and CM ₂ .	CQ ₄) With which substances does Frenadol® interact?	With which substances do the parts of Frenadol® interact?	Add the following axiom: <i>Main active ingredient of</i> is a sub-relation of <i>part_of</i> .
Heuristic 12. We want to know which are the endpoints of the temporal entity returned by the CQ	CQ ₃) Which is the main active ingredient of Frenadol®? CQ ₅) Which drug manufacturing lot expires before 11 May 2012?	Add a linking axiom establishing that S is a subrelation of <i>part_of</i> . Add a linking axiom to time points.	Add the following axiom: The range of <i>has expiration date</i> in the concept <i>drug</i> is <i>time point</i> .

Example 38. An example of a knowledge clash is the time ontology for reuse that has density, whereas the ontology under development does not require density.

- *Adaptation to the reasoner.* It refers to whether the definitions and axioms need to be adapted in order to satisfy the existing reasoner constraints.

Example 39. OWL ontologies may include explicit definitions; however, ontologies written in Prolog cannot include this kind of definitions.

- *Need for bridge terms.* It refers to whether it is necessary to create new linking axioms and/or relations to integrate the candidate ontology to be reused into the ontology being developed.
- *Reliability.* In this case, the following attributes should be analyzed:
 - *Design criteria.* It refers to whether the ontology has been built according to the design criteria assumed by the team developing the domain ontology.
 - *Test availability.* It refers to whether tests are available for the candidate ontology to be reused.
 - *Former evaluation.* It refers to whether the ontology has been properly evaluated, which here means that the ontology has passed a set of unit tests.
 - *Theoretical support.* It refers to whether the candidate ontology implements a tested theory.
 - *Development team reputation.* It refers to whether the development team of the candidate ontology is reliable.
 - *Fitness for purpose.* It refers to whether the candidate ontology has been developed as a simple academic example.
 - *Practical support.* It refers to whether there are well-known projects or ontologies reusing the candidate ontology [54].

For each of the four dimensions, we have a *many-valued FC*, where the objects are the candidate ontologies. We use the scale {unknown, low, medium, high} for all attributes. The value equal to high corresponds to the best attribute value, whereas a low value is assigned to the least preferred one. Thus, for example, the best value for both financial cost and documentation quality is

Table 11

Identification of definitions and axioms from a mereology for reuse according to the heuristics listed in Use Case 2.2.

Axioms and definitions	Conditions established by heuristics	Are conditions met?
A.1) Reflexivity	Heuristic 14 condition: implementation is possible.	Yes
A.2) Antisymmetry	Heuristic 14 condition: implementation is possible.	Yes
A.3) Transitivity	Heuristic 15 condition: X has parts X_1, X_2, \dots, X_n , and some X_i has parts $X_{i1}, X_{i2}, \dots, X_{in}$, that is, X has several levels of parts. Besides, we want to know all the levels when we ask, <i>Which are the parts of X?</i>	Yes
D.1) Proper part	Heuristic 16 condition for <i>proper part</i> is met.	Yes
D.2) Direct part	Heuristic 16 condition for <i>direct part</i> is met.	No
D.3) Overlap	Heuristic 16 condition for <i>overlap</i> is met.	No
D.4) Underlap	Heuristic 16 condition for <i>underlap</i> is met.	No
D.5) Disjoint	Heuristic 16 condition for <i>disjoint</i> is met.	No
A.4) Weak supplementation	Heuristic 14 condition: implementation is possible.	Yes

Table 12

Partial view of the integration effort many-valued formal context.

Integration effort	Knowledge extraction adequacy	Naming conventions adequacy	Implementation language adequacy
Single part whole	[High]	[Low]	[High]
SUMO-OWL	[High]	[High]	[High]
Dolce-Lite	[High]	[Low]	[High]
Oswebsite	[High]	[High]	[High]
OBO	[High]	[Low]	[High]

high. Since we want to penalize ontologies about which little is known, we have placed unknown and low at the same level on the scale. The decision-making steps are

1. *Assigning values to all the attributes of all the ontologies.* The result of this step can vary from one engineer to another. Therefore, a group decision technique, e.g., Delphi [55], may be useful in this step.
2. *Removing the attributes that have the same value for all the candidate ontologies.* They are not useful for decision making.
3. *Outputting the derived FC from each many-valued FC.*
4. *Outputting the lattice for each derived FC.* This helps us to determine, for each dimension, what advantages each ontology has over the other candidate ontologies.
5. *Deciding which candidate ontology is the best.* To do this, we have to analyze the lattices generated. If an ontology dominates the other candidate ontologies in all dimensions, then the decision is easy. However, different ontologies may dominate in different dimensions. In this case, the final decision depends on the importance that the ontology development team attaches to each attribute. This step may also involve discussion to reach a consensus.

Not only do these steps have to consider the above four dimensions, but they also have to take into account the comparative study output in Use Case 1.4 filtered using the attributes identified in Use Case 2.2.

3.3. Use Case 3. Assembling the whole ontology

The goal of this use case is to output a domain ontology that includes the general ontology selected in Use Case 2. Use Case 3 is decomposed as follows (see Fig. 11).

3.3.1. Use Case 3.1. Customizing the general ontology

The general ontology is adapted for integration into the domain ontology. This includes the following use cases:

- *Use Case 3.1.1. Pruning the general ontology according to the required attributes.* The goal here is to prune the selected general ontology taking into account the attributes required in the domain ontology being developed.

Example 40. If the attribute *underlap* is not needed in the mereology ontology, its definition and dependent definitions should be removed.

- *Use Case 3.1.2. Enriching the general ontology.* The goal here is to add the new conceptual structures needed in the domain ontology being developed to the selected general ontology.

Example 41. If the general ontology does not include the reflexivity axiom of *part of* and this is required, it should be added.

- *Use Case 3.1.3. Translating the general ontology into the implementation language of the domain ontology being developed.* The goal here is to translate the general ontology selected into the implementation language of the domain ontology being developed, if the two ontologies are written in different languages.

Table 13

Integration effort many-valued formal context where attributes that have the same value for all the candidate ontologies have been omitted.

Integration effort	Naming conventions adequacy
Single part whole	[Low]
SUMO-OWL	[High]
Dolce-Lite	[Low]
Oswebsite	[High]
OBO	[Low]

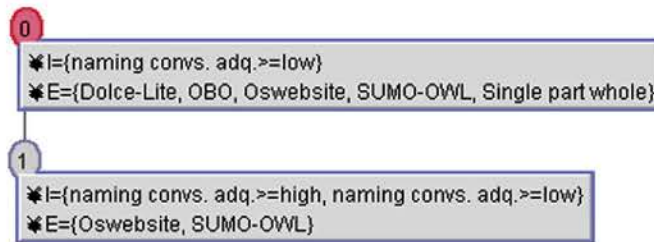


Fig. 12. Integration effort lattice.

The general ontology can be translated automatically or manually. However, it should be noted that the complete translation into different languages is not always possible.

- *Use Case 3.1.4. Evaluating the resulting general ontology.* The goal here is to check that the resulting ontology has the attribute output in Use Case 2.2 and to remedy any deficiencies detected in Use Case 2.3.

Example 42. If the general ontology does not have the *naming convention adequacy* attribute, names should be changed.

How hard it is to customize an ontology generally varies depending mainly on the rules (e.g., in SWRL) that it contains. At one end of the scale, we have an ontology that contains no rules. In this case, the customization process takes less than 1 h. At the other end of the scale, we have an ontology with 10 or more rules. In this case, customization may take four or more hours. Therefore, the engineer should decide whether to customize the general ontology or reuse the irrelevant axioms and rules.

3.3.2. Use Case 3.2. Integrating the general ontology in the domain ontology being developed

The goal here is to integrate the general ontology output of Use Case 2.3 in the domain ontology being developed.

Looking at different general ontologies, we find that they assume different ontological distinctions and different top-level concepts. These differences may also occur between the general ontology to be reused and the domain ontology being developed. We have identified the following scenarios:

- *Scenario 1.* Heterogeneity between the general ontology and the domain ontology is insoluble or particularly difficult to solve. In this case, the decision reached in Use Case 2 should be revised.
- *Scenario 2.* Once the general ontology has been pruned, there is no longer any heterogeneity or the heterogeneity that can be easily solved.

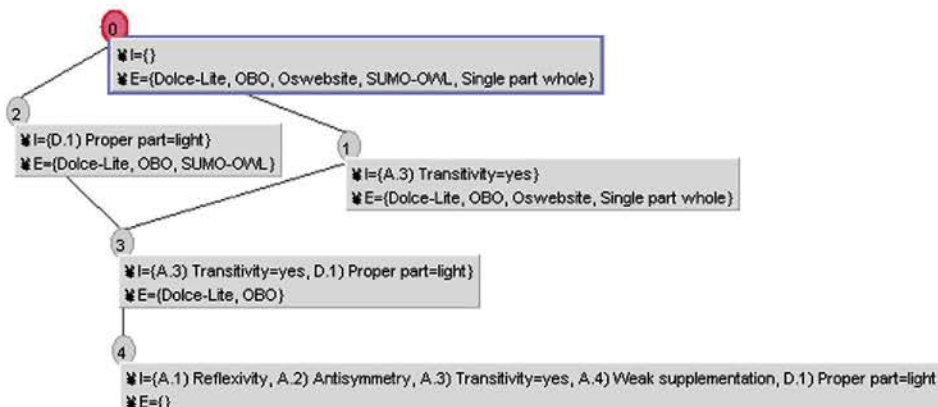


Fig. 13. Lattice for axioms and definitions.

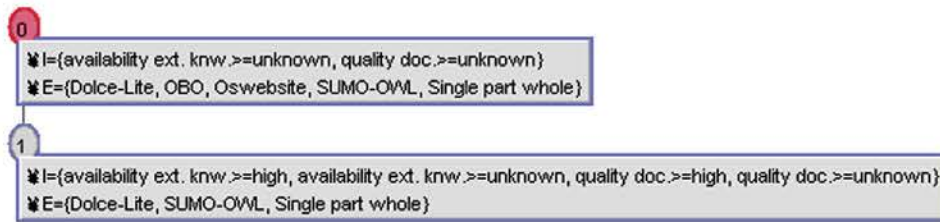


Fig. 14. Understandability effort lattice.

Example 43. The *part* relation specified in SUMO-OWL is applicable to (physical) objects. Given that this ontology makes a distinction between objects and processes, this relation may not be applicable to processes. Nevertheless, if the *process* concept is not necessary in the domain ontology, then the *part* relation of SUMO-OWL can be reused as it is.

3.3.3. Use Case 3.3. Evaluating the whole ontology

Most of the ontology evaluation effort is spent on examining how the ontology answers the CQs executing the inference engine. Thus, we check that (1) all the expected answers are there (completeness criterion); (2) no unexpected answers (accuracy criterion) emerge; (3) the terms are expressed according to the naming conventions established in the ORSD (vocabulary aspect); and (4) the response time is within the interval allowed by the ORSD (computational efficiency). Nevertheless, the analysis of CQ answers is not enough. For instance, we will not be able to directly determine whether there are irrelevant axioms and whether the ontology anticipates its uses. One of the evaluation methods that go beyond CQs is OntoClean [56]. Its purpose is to validate the adequacy and logical consistency of taxonomic relationships.

“It is based on highly general ontological notions drawn from philosophy, like essence, identity, and unity, which are used to elicit and characterize the intended meaning of properties, classes and relations making up an ontology. These aspects are represented by formal metaproperties, which impose several constraints on the taxonomic relationships between concepts. The analysis of these constraints helps in evaluating and validating the choices made.” ([56], page 201).

The development environment automatically checks syntactical correctness. For more information on ontology evaluation, see [57,58].

4. Case study: Reusing general ontologies to build a pharmaceutical product ontology

We present an example of general ontology reuse for the pharmaceutical product ontology (PPO) that represents pharmaceutical products. The PPO will be used as a bridge between proprietary financial management systems and proprietary product knowledge systems in pharmaceutical laboratories, companies and distributors in Spain. The general ontologies (specifically, a time ontology and a mereology) have been reused in the PPO following the methodological guidelines described in

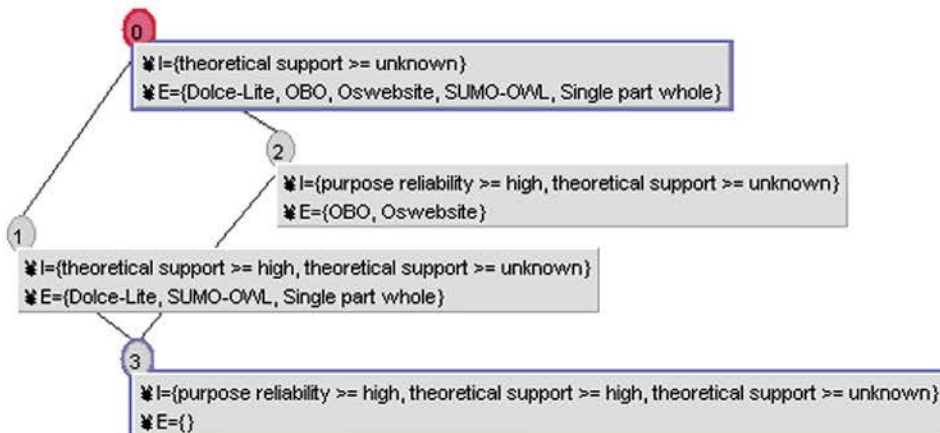


Fig. 15. Reliability lattice.

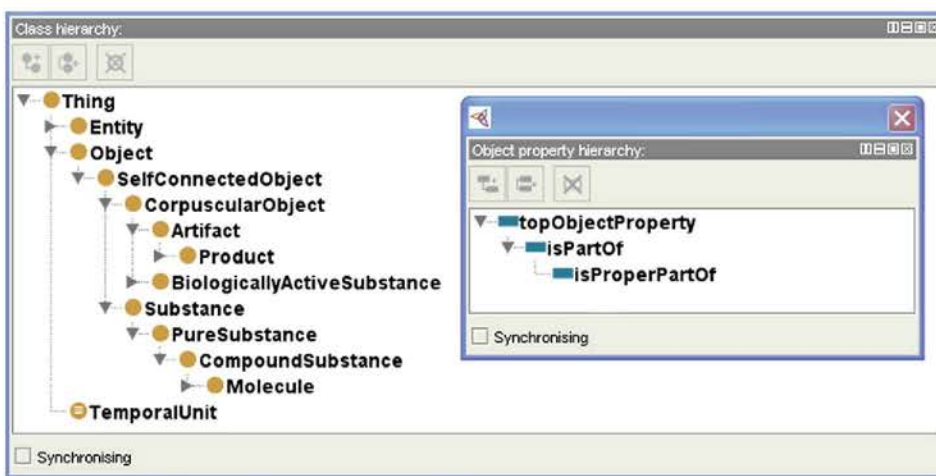


Fig. 16. Concepts and object properties saved from SUMO-OWL.⁴³
⁴³ Snapshot taken from Protégé (<http://protege.stanford.edu/>).

this paper. For reuse, we took into account the eight competency questions (CQs) shown in Table 8. Table 9 shows the result of applying the heuristics to identify the types of general ontology to be reused (see Use Case 1.1).

The identification of the most significant definitions and axioms that characterize each type of general theory are shown in Sections 2.4 and 2.5 (Use Case 1.2). Table 5 shows the search for general ontologies (Use Case 1.3). The comparative study is illustrated in Tables 6 and 7 (Use Case 1.4). For decision-making on what general ontologies are going to be reused (Use Case 2), we had to carry out the following use cases: Use Case 2.1. Reformulating CQs and adding linking axioms (see Table 10); Use Case 2.2. Identifying the attributes of the implemented mereology to be reused (see Table 11); and Use Case 2.3. Determining the best implemented mereology. It should be explained here that we analyzed the OWL versions of SUO-OWL and DOLCE-Lite. For Use Case 2.3, we followed the steps shown in Section 3.2.3, that is:

1. *Assigning values to all attributes of all ontologies.* Table 12 shows the assignment for the integration effort dimension.
2. *Removing attributes with the same value for all the candidate ontologies.* The extreme case was reuse cost. It was completely omitted because it is the same for all the ontologies. Table 13 shows the filtered integration effort FC.
3. *Outputting the derived FC from each many-valued FC and outputting the lattice for each derived FC.* Fig. 12 shows the lattice for the FC of Table 13.
4. *Deciding which candidate ontology is the best.* The lattices provide the following information:
 - DOLCE-Lite and OBO dominate over the other ontologies with regards to the axioms and definitions that they formalize (see Fig. 13). They are the only ones that formalize transitivity and include the *proper_part_of* relation.
 - DOLCE-Lite, SUMO-OWL and Simple part whole dominate over with regards to the understandability effort dimension (see Fig. 14).
 - SUMO-OWL dominates over with regards to integration effort (Fig. 12). Oswebsite and SUMO-OWL have the advantage of having a better naming convention adequacy. Furthermore, in SUMO we have found top-level concepts, like *biologically active substance* and *molecule* (and their ancestors), and the knowledge extraction adequacy criterion was rated high for this ontology.
 - No ontology dominates any of the others with regards to reliability (see Fig. 15). The purpose reliability is rated high for Oswebsite and OBO, whereas SUMO-OWL and DOLCE-Lite score high on theoretical support.

Note that the value unknown here actually means that we do not know the real value of the attribute.

We believe that it is more costly to change the naming conventions than to add the transitive property of the *part_of* relation to SUMO-OWL; therefore, we selected this ontology rather than DOLCE-Lite.

Although we are used to work with a lattice for each dimension, it is also possible to reassemble the above lattices in a wider one containing all sub-lattices as part. The main advantage of our approach is that we analyze each dimension without the interference of the others. On the other hand, the advantage of the global lattice is that it is easier to simultaneously work with attributes belonging to different dimensions.

Table 14

Formal host CQs that require *part-of* modeling (for simplicity's sake, both the prefixes and the value data types are omitted in the answers).

Informal CQ	Formal CQ	Sample answer
Which drugs are composed of paracetamol?	<pre># CQ1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX ub: <http://www.oeg-upm.net/ontologies/ppo.owl#> SELECT ?X WHERE { ?X rdf:type ub:DrugSubstance . ub:Paracetamol ub:isProperPartOf ?X . }</pre>	<pre>----- X ===== FrenadolSubstance BisolgripSubstance CortafriolSubstance DolgesicSubstance TermalginSubstance AlgidolSubstance EfferalganSubstance DolostopSubstance GelocatilSubstance ApiretalSubstance PharmagripSubstance -----</pre>
What is the composition of Frenadol®?	<pre># CQ2 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX ub: <http://www.oeg-upm.net/ontologies/ppo.owl#> SELECT ?X WHERE { ?X ub:isProperPartOf ub:FrenadolSubstance . }</pre>	<pre>----- X ===== Dextromethorphan Chlorpheniramine Citrate Caffeine Paracetamol -----</pre>
What is the main active ingredient of Frenadol®?	<pre># CQ3 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX ub: <http://www.oeg-upm.net/ontologies/ppo.owl#> SELECT ?X WHERE { ?X ub:isMainActiveIngredientOf ub:FrenadolSubstance . }</pre>	<pre>----- X ===== Paracetamol -----</pre>
Which substances does Frenadol® interact with?	<pre># CQ4 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX ub: <http://www.oeg-upm.net/ontologies/ppo.owl#> SELECT ?X WHERE { ub:FrenadolSubstance ub:interactsWith ?X . }</pre>	<pre>----- X ===== Rifampicin Propranolol Isoniazid EthylAlcohol -----</pre>

The selected ontology is used to assemble the whole ontology as follows (Use Case 3.1):

Use Case 3.1.1. Pruning the general ontology according to the required attributes. We removed all the SUMO-OWL concepts, except the concepts and object properties shown in Fig. 16.

Use Case 3.1.2. Enriching the general ontology. We added (a) transitivity to the part and properPart object properties; (b) reflexivity to part; (c) asymmetry to properPart; and (d) antisymmetry to part.

Use Case 3.1.4. Evaluating the resulting general ontology. We checked that the resulting ontology has the attributes obtained as output in Use Case 2.2. No deficiency was detected in Use Case 2.3; consequently, no change was made to the ontology during the execution of this use case.

Use Case 3.2. Integrating the general ontology into the host ontology. We established that the concept *DrugSubstance* is a subclass of *BiologicalActiveSubstance*; the concepts *AlgidolSubstance*, *ApiretalSubstance*, *BisolgripSubstance*, etc., are subclasses of *DrugSubstance*; concepts such as *AlgidolProduct*, *ApiretalProduct* and *BisolgripProduct*, are subclasses of *Product* (“an artifact that is manufactured and intended to be sold”); whereas *Dextromethorphan*, *EthylAlcohol* and

Table 15

Analysis and transformation of CQs, and addition of linking axioms according to the heuristics listed for time modeling reuse in Use Case 2.1.

Heuristic applied	Competency question	Action	Result of the action
Heuristic 12. We do not know which are the endpoints of the temporal entity returned by the CQ, hence we reuse time points	Which Frenadol® manufacturing lot expires before May 11, 2012? Which Frenadol® manufacturing lot expires on April 30, 2010? Which Frenadol® manufacturing lot expires in 2010? Which Frenadol® manufacturing lot expires after May 2012?	Introduce a linking axiom to time points.	The range of <i>has expiration date</i> in the concept <i>drug product</i> is <i>time point</i> .

Isoniazid, among others, are subclasses of `Molecule` and `BiologicallyActiveSubstance` (“a substance that is capable of inducing a change in the structure or functioning of an organism”). In other words, we have taken advantage of the possibilities that SUMO-OWL offers to easily represent different perspectives of the notion of drug. For example, from one perspective a drug is a substance that acts in our organism and, from another, it is a product that can be sold. Given that transitivity, antisymmetry, etc., involve individuals, we have added an individual for each type of substance and product. Thus, for example, the system can infer that `Caffeine` is part of `Frenadol` because there is an individual of `Caffeine` (also with tag ‘Caffeine’) that is part of an object `Frenadol`, which is an individual of `Frenadol`. This is a means to obtain a part-whole relation between classes from a part-whole relation between individuals. Let’s note that, according to OWL semantics [18], the universes of discourse where individuals and classes are interpreted are disjoint. Therefore, individuals `Caffeine` and `Frenadol` are interpreted in a different universe of discourse to the one of classes `Caffeine` and `Frenadol`. Hence, it is allowed, for example, that there are different kinds of `Frenadol®`. However, given that inheritance cannot be applied to individuals, the fact of that `Caffeine` is part of every specific kind of `Frenadol` is not inferred from the fact that `Caffeine` is part of `Frenadol`. To deep in the use of the same name both for a class and for an instance, the reader can consult [59]. Smith and Rosse [5] solve the problem of the obtaining of a part-whole relation between classes from a part-whole relation between individuals using full first order logic.

We have also adapted the names to the convention used in PPO. Thus, for example, part was changed to `isPartOf`.

Use Case 3.3. Evaluating the whole ontology. Part of the evaluation consisted in executing the CQs (see Table 14) and analyzing their results as shown in Section 3.3.3. To answer CQ₄ we have added this rule to the ontology:

```
interactsWith(?x, ?y), isPartOf(?x, ?z) -> interactsWith(?z, ?y)
```

That is, if a substance `?x` interacts with another substance `?y`, then it interacts with every whole of `?y`. For example, given that paracetamol interacts with ethyl alcohol, `Frenadol®` also interacts with ethyl alcohol.

Tables 15 and 16 show the result of Use Case 2.1 and Use Case 2.2 applied to CQs that cause time modeling reuse. The results of decision-making on which time ontologies should be reused (Use Case 2.3. (Determining the best time modeling implementation)) is, again, very tight. In this case, the selected ontology was OWL-Time. Its design criteria scored better than the other ontologies because OWL-Time separates time entities (`TemporalEntity`) from their description (`DateTimeDescription` and `DurationDescription`).

The selected ontology is assembled in the whole ontology according to the following use cases:

Use Case 3.1.1. Pruning the general ontology according to the required attributes. The OWL-Time terms that remain after pruning are (a) the concepts `DateTimeDescription`, `TemporalEntity`, `Instant` and `TemporalUnit`; (b) the object properties `inDateTime` (between `TemporalEntity` and `DateTimeDescription`) and `unitType` (between `DateTimeDescription` and `TemporalUnit`); and (c) the data properties `day`, `month` and `year`.

Use Case 3.1.2. Enriching the general ontology. Given that only simple absolute time queries are needed, no rules or definitions are required.

Use Case 3.1.4. Evaluating the resulting general ontology. Each of the properties has been tested.

Use Case 3.2. Integrating the general ontology in the host ontology. We have reinserted the term `Abstract` from SUMO-OWL into the ontology to be reused as a superclass of both `TemporalEntity` and `DateTimeDescription`. It should be noted that the meaning of `Abstract` in SUMO is different from the meaning of `Abstract` in Dolce. In fact, Dolce non-physical endurants would be SUMO abstract entities. As a consequence, although a date time description may change a long time, it is a SUMO abstract entity (but not a Dolce abstract particular).

Use Case 3.3. Evaluating the whole ontology. Part of the evaluation consisted in executing the CQs (see Table 17).

A partial view of the resulting ontology is shown in Fig. 17.

Table 16

Identification of time modeling definitions and axioms for reuse according to the heuristics listed in Use Case 2.2. For brevity's sake, most of the attributes whose conditions are not satisfied are omitted.

Attributes	Conditions established by the heuristics	Are conditions met?
Modeling of time points	Heuristic 12 condition: We are not interested in the endpoints of the temporal entities.	Yes
Modeling of time intervals	Heuristic 13 condition: We are interested in the endpoints of the temporal entities.	No
Modeling of absolute time (by means of time units)	Heuristic 18 condition: Some CQs that can be expressed as: <i>When does X happen?</i>	Yes
Modeling of relations between time entities	Heuristic 19 condition: Some CQs can be expressed such as <i>Does X happen before Y?</i> <i>Does X happen after Y?</i> <i>Does X happen at the same time as Y?</i> , , where X and Y can be either dates or events.	Yes

5. Controlled experiment

This section describes the experiment carried out to measure some important features (e.g. intelligibility) of the methodological guidelines proposed in this paper. First, we present the experiment setup (Section 5.1), with its context, objectives and procedure. Then, we describe the statistical processing (Section 5.2) and data analysis (Section 5.3).

5.1. Experiment setup

5.1.1. Context

The experiment started on 25 October 2011 and the participants were graduated students attending the *Ontologies and Semantic Web* course as part of the Master in Artificial Intelligence program at UPM.³⁸ The main purpose of the experiment was to test the methodological guidelines proposed in this paper.

5.1.2. Objectives

The experiment aimed at measuring the precision, intelligibility, completeness and necessity of each part of the methodology.

5.1.3. Steps

The experiment consisted of the following steps:

1. Explaining the methodological guidelines proposed for the reuse of general ontologies (this was carried out in a one-and-a-half-hour lecture). Given the time constraints, lecturers did not provide formal definitions for FCA but intuitive explanations, speaking of tables rather than FCs and graphs rather than lattices.
2. Providing (to all the students) (a) the wording of the assignment to be completed; (b) the description of the methodology; and (c) a questionnaire available on the Web.
3. Timing the experiment. Students had 10 days to complete the assignment, which they were allowed to carry out individually or with a partner of their choice. Nine groups were formed.
4. Collecting the complete questionnaire. The answers were analyzed as described in Section 5.2.
5. Analyzing data (see Section 5.3).

5.1.4. Task

Each group had to develop a cultural event ontology that answers, at least, the CQs³⁹ shown in Table 18.

Students had to apply the guidelines for reusing general ontologies and were asked to rate every section of them. On a scale from 1 to 5, they had to rate, the following questions: (1) *Is it precise?*; (2) *Is it understandable?*; (3) *Is it complete?*; and (4) *Are all its explanations necessary?* A score of 5 was the best, and a score of 1 was the worst.

5.2. Statistical processing

As Table 19 shows, we have estimated the mean and standard deviation of the score for each answer. Table 19 aggregates the scores of **seven** groups, since both the group that assigned the best scores in the questionnaire (with a mean of 4.65) and the one that assigned the worst scores (with a mean of 2.30) have been ruled out.

³⁸ <http://www.fi.upm.es/?id=masterinteligenciaartificial&idioma=english>.

³⁹ Information on the London Symphony Orchestra can be found at <http://lso.co.uk/whatson/programmes> for the program and <http://lso.co.uk/page/3128/> Players for the structure of the orchestra.

Table 17

Formal host CQs requiring time modeling (for simplicity's sake, both the prefixes and the value data types are omitted in the answers).

Informal CQ	Formal CQ	Sample answer
<p>Which Frenadol® manufacturing lot expires before 11 May 2012?</p>	<pre># CQ5 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX ub: <http://www.oeg-upm.net/ontologies/ppo.owl#> PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> SELECT ?DrugProduct WHERE { ?DrugProduct rdf:type ub:FrenadolProduct . ?DrugProduct ub:hasExpirationDate ?ExpirationDate . ?ExpirationDate ub:isInDateTime ?DateDescription . ?DateDescription ub:year ?Year . ?DateDescription ub:month ?Month . ?DateDescription ub:day ?Day . FILTER(?Year < 2012 ?Year = 2012 && ?Month < 5 ?Year = 2012 && ?Month = 5 && ?Day < 11) . }</pre>	<pre>----- DrugProduct ===== frenadolC63 frenadolC125 frenadolC243 ----- </pre>
<p>Which Frenadol® manufacturing lot expires on 30 April 2010</p>	<pre># CQ6 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX ub: <http://www.oeg-upm.net/ontologies/ppo.owl#> PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> SELECT ?ManufacturingLot WHERE { ?DrugProduct rdf:type ub:FrenadolProduct . ?DrugProduct ub:hasExpirationDate ?ExpirationDate . ?ExpirationDate ub:isInDateTime ?DateDescription . ?DateDescription ub:year ?Year . ?DateDescription ub:month ?Month . ?DateDescription ub:day ?Day . ?DrugProduct ub:hasManufacturingLot ?ManufacturingLot . FILTER(?Year = 2010 && ?Month = 4 && ?Day = 30) . }</pre>	<pre>----- ManufacturingLot ===== "C63" ----- </pre>
<p>Which Frenadol® manufacturing lot expires in 2010?</p>	<pre># CQ7 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX ub: <http://www.oeg-upm.net/ontologies/ppo.owl#> PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> SELECT ?ManufacturingLot WHERE { ?DrugProduct rdf:type ub:FrenadolProduct . ?DrugProduct ub:hasExpirationDate ?ExpirationDate . ?ExpirationDate ub:isInDateTime ?DateDescription . ?DateDescription ub:year ?Year . ?DrugProduct ub:hasManufacturingLot ?ManufacturingLot . FILTER(?Year = 2010) . }</pre>	<pre>----- ManufacturingLot ===== "C63" "C125" ----- </pre>
<p>Which Frenadol® manufacturing lot expires after May 2010?</p>	<pre># CQ8 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX ub: <http://www.oeg-upm.net/ontologies/ppo.owl#> PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> SELECT ?ManufacturingLot WHERE { ?DrugProduct rdf:type ub:FrenadolProduct . ?DrugProduct ub:hasExpirationDate ?ExpirationDate . ?ExpirationDate ub:isInDateTime ?DateDescription . ?DateDescription ub:year ?Year . ?DateDescription ub:month ?Month . ?DrugProduct ub:hasManufacturingLot ?ManufacturingLot . FILTER(?Year > 2010 ?Year = 2010 && ?Month > 5) . }</pre>	<pre>----- ManufacturingLot ===== "C243" ----- </pre>



Fig. 17. Partial view of the concept and object property hierarchies.

5.3. Data analysis

The estimation of the mean of each variable is greater than 3.00; therefore, the experiment outcomes have yielded positive evidence of the quality of the set of methodological guidelines that we provide. The three lowest values of the mean are the precision, the completeness and the understandability of the guidelines for assembling the whole ontology. Therefore, Use Case 3 is a candidate for future research.

6. Conclusions

Experience (see Section 1) has shown that people working on technical issues do not have the time or inclination to understand and apply formal ontology development theories. Bearing this fact in mind, we have built, with the help of the research reported in [12–14], some methodological guidelines. These heuristic-based guidelines show how (a) to reuse general ontologies to develop domain ontologies; and (b) to take advantage of the sound work completed by philosophical researchers, particularly by formal ontology researchers.

We have tried to familiarize ontology development engineers with time modeling and mereology. We have also provided heuristics to identify what type of general theories should be reused and what axioms and definitions of such theories should be extracted from the CQs of the domain ontology to be developed. Likewise, we have proposed a procedure to identify the best general ontologies to be pruned, extended and integrated into the domain ontology under construction.

At the beginning of our research, we thought that the most important contribution of general ontology reuse would be the reuse of formulas originally expressed in a KIF-like [60] or CML-like [61] language. However, the further our work progressed, the more we came to realize the importance of taxonomic patterns. Thus, for example, the distinction between duration and temporal intervals, or the separation between temporal entities and their description, are extremely valuable patterns. Hence, we now directly reuse OWL general ontologies supplemented, if need be, with axioms from other sources,

Table 18

CQs to be answered by the experiment ontology.

CQ id	Informal CQ	Answer
CQ ₁	When is the London Symphony Orchestra to play in October 2011?	1st, 2nd, 4th, 6th, 7th, 9th, 11th, 12th, 13th, 14th, 15th, 16th, 19th, 20th, 21st, 23rd, 27th, 30th, 31st October 2011
CQ ₂	What works did the London Symphony Orchestra interpret at the concert on 4 October 2011?	Haydn Symphony No 92 ('The Oxford') Nielsen Symphony No 1 Beethoven Piano Concerto No 3
CQ ₃	Which works is the London Symphony Orchestra to interpret during the 2010/11 season?	Beethoven Piano Concerto No 3 Beethoven Piano Sonata No 10 in G major Beethoven Piano Sonata No 22 in F major Beethoven Piano Sonata No 15 in D major ('Pastoral') Haydn Symphony No 92 ('The Oxford') Nielsen Symphony No 1
CQ ₄	Which works did the London Symphony Orchestra play at more than one concert in the 2010/11 season?	... Britten War Requiem ...
CQ ₅	What groups of instruments does the London Symphony Orchestra have?	First violins Second violins Violas Cellos ...
CQ ₆	Who are London Symphony Orchestra's first violins?	Gordan Nikolitch Roman Simovic Ginette Decuyper Jörg Hammann ...
CQ ₇	Who are the leaders of the London Symphony Orchestra?	... Gordan Nikolitch Roman Simovic

such as those described in Sections 2.4 and 2.5, for instance. Other facts that support the direct reuse of OWL general ontologies are (1) the comparison of general ontologies written in very expressive languages is more difficult for people working with domain ontologies than the comparison of OWL ontologies; and (2) the expressiveness of some formal

Table 19

Results of the experiment (figures have been rounded to two decimal points).

QUESTION	Mean	Standard deviation	Minimum	Median	Maximum
<i>Mereology modeling description (Use Case 1)</i>					
1. Is it precise enough?	4.14	0.69	3.00	4.00	5.00
2. Is it understandable enough?	4.29	0.76	3.00	4.00	5.00
3. Is it complete enough?	3.86	0.90	3.00	4.00	5.00
4. Are all its explanations necessary?	4.29	0.76	3.00	4.00	5.00
<i>Time modeling description (Use Case 1)</i>					
5. Is it precise enough?	4.29	0.95	3.00	5.00	5.00
6. Is it understandable enough?	4.43	0.79	3.00	5.00	5.00
7. Is it complete enough?	4.00	0.82	3.00	4.00	5.00
8. Are all its explanations necessary?	4.14	0.69	3.00	4.00	5.00
<i>Guidelines for the comparative study (Use Case 1)</i>					
9. Are they precise enough?	4.14	1.07	2.00	4.00	5.00
10. Are they understandable enough?	4.14	1.21	2.00	5.00	5.00
11. Are they complete enough?	3.86	1.07	2.00	4.00	5.00
12. Are all their explanations necessary?	4.00	0.00	4.00	4.00	4.00
<i>Guidelines for decision-making on what general ontologies should be reused (Use Case 2)</i>					
13. Are they precise enough?	4.00	1.00	2.00	4.00	5.00
14. Are they understandable enough?	4.00	1.15	2.00	4.00	5.00
15. Are they complete enough?	3.57	1.27	2.00	4.00	5.00
16. Are all their explanations necessary?	4.00	0.58	3.00	4.00	5.00
<i>Guidelines for assembling the whole ontology (Use Case 3)</i>					
17. Are they precise enough?	3.14	0.69	2.00	3.00	4.00
18. Are they understandable enough?	3.50	1.05	2.00	3.50	5.00
19. Are they complete enough?	3.43	0.98	2.00	4.00	4.00
20. Are all their explanations necessary?	3.86	0.69	3.00	4.00	5.00

languages⁴⁰ is a disadvantage when practical implementation and use of general ontologies for computational modeling (adapted from [11], page 10).

Concerning our experience in reusing general ontologies in PPO, it has already been explained that we have reused a time ontology and a mereology, both of them in OWL. In the first case, reusing took less effort than modeling time from the scratch. Unfortunately, not much time was saved when we reused the mereology in OWL. Most of the components implemented were almost directly taken from Section 2.4. Nevertheless, we expect that the effort of reusing OWL code will pay off when new, more detailed OWL mereologies appear.

We have shown how to perform the decision-making use case by means of FCA. It should be added that we also have previous experience in decision making using a numerical process [14] based on an objective tree (not reported in this paper). The numerical process has the advantage of directly reaching the decision. However, the weights of the objectives have to be properly tuned for it to be effective. Such a tuning should be changed if either the decision maker's viewpoint or the circumstances change.

The controlled experiment confirmed that when we have to describe other types of general theories (e.g. topologies [16]) in future extensions of the methodology, the description should be detailed, albeit not overly formal. In this respect, we found that our students felt more comfortable with heuristics and informal and easy-to-understand explanations than with very formal descriptions. This confirms our approach of intuitively introducing engineers to formal notions so that they can understand the formal definitions of the general ontologies that they have to reuse.

We assume that complex technical decisions have a subjective component that cannot be completely removed. Therefore engineers should be allowed to work with their own variations of the proposed methodology.

7. Future work

We will distinguish in this section between the future directions of the methodological guidelines themselves, the software that supports them and experimentation.

With regards to the first, we propose to further research on ontology integration. We also propose to thoroughly study the reuse of ontologies with concepts with an intermediate level of abstraction, for example, `BiologicalActiveSubstance`, and `CompoundSubstance`. Concerning the identification of the reuse of time modeling, it could be improved by taking ideas from works on extraction of meaning from texts (see for example [62]).

With respect to the integration of our guidelines in more general methodologies, we should explain that we have used them inside NeOn Methodology [63]. Such adaptation is specially challenging for those methodologies that follow other means than raising CQs to analyze and structure requirements. For example, AKEM [64] starts from user stories: use cases that describe a unit of knowledge or business logic, identified in knowledge elicitation (e.g. the entity *drug definitions*, with their sources and episodes related to such definitions.). The possibility of combining our guidelines with collaborative approaches (e.g. the DILIGENT one [65]) can be also explored. It would be also interesting to study the integration of our guidelines with methodologies and methods for ontology reuse that are mainly focused on the reuse of domain ontologies (see, for example, [66,67]).

With respect to the technology that supports the methodological guidelines, we have used Galicia [72], for FCA although a tool to analyze CQs at the lexical and the syntactic level would be also useful. This tool would make, at least, recommendations about the type of general ontologies to be reused and the way of transforming CQs, among others. Given the large amount of information provided by search engines, the automatic coordination of the analyzer of CQs with search engines to limit the quantity of candidate ontologies is also an interesting field of research for the future. But even without these tools an ontology developer trained in the methodology needs just few seconds to determine if a CQ is interesting enough to extract information from it in order to reuse general ontologies. In the case of PPO, for example, the CQs that interest us represent approximately 25% of the total CQs. In other ontologies, such percentage is even less. Therefore, we can follow the process proposed in this paper without additional software support even for ORSDs with hundreds of CQs.

It would be appealing to perform other controlled experiments requiring the reuse of attributes other than those required in Section 5, even though some of these attributes are not very useful in contexts where OWL ontologies are used. In fact, as we have already mentioned, some examples in Section 3 have been checked in Prolog, assuming the CWA.

Acknowledgments

This work has been partially supported by the Spanish *BabelData: Multilingüismo en Ontologías y Linked Data* (TIN2010-17550) project. We are also grateful to the referees of the magazine for their valuable comments, and to Luis Fernando Alguacil-Merino for providing us some of the examples that we have used in Section 2.3.

⁴⁰ A formal language is that in which the notations of meaningful expressions, axioms and inference rules are precisely described [74]. Although, according to this definition, an ontology is written in a formal language, we have reserved the term *formal language* for theories and *implementation language* (defined in Section 2.2) for ontologies to avoid the problems mentioned in note 18.

Annex I. Other Section 3.2.2 heuristics applied to time modeling

Heuristic 21 (convex intervals). If we want to know about time intervals without “separate” pieces, then convex intervals should be reused.

Example 44. The CQ *Is the Medinfo 2013 call for submissions open on 13 November 2012?* can be answered using convex intervals, since a call is open during a gapless time interval.

Heuristic 22 (non-convex intervals). If we want to know about time intervals with “separate” pieces (e.g. every Wednesday, every 8 hours, every working day), then non-convex intervals should be reused.

Example 45. If the information available is that meetings are held every Wednesday, the system can answer the CQ *Is there a meeting on 13 February 2012?* by reasoning with non-convex intervals.

Heuristic 23 (open intervals). If we want to know about time intervals expressed (explicit or implicitly) in the form *from X to Y (X and/or Y not included)*, then the system can answer the question by reasoning with open intervals.

Example 46. Supposing that we have the fact *The registry of the Ministry of Health closes at 14.00 (Madrid time)*, the system can solve the CQ *Are we in time to register a document at the Ministry of Health?* by reasoning with open intervals.

Heuristic 24 (closed intervals). If we want to know about time intervals expressed (explicit or explicitly) as *from X to Y (both included)*, then the system can answer the question by reasoning with closed intervals.

Example 47. If the CQ is *When is the Medinfo 2013 call for submissions open?*, then the answer will be a closed interval.

Heuristic 25 (different granularities). If the CQs or/and their answers can be expressed in different units of measurement, then we are dealing with different granularities.

Example 48. Example 46 is described with a *granularity* of minutes, whereas Example 45 is specified with granularity of days.

Heuristic 26 (different time zones). If the CQs or/and their answers can be expressed in different zones of measurement, then we are dealing with different time zones.

Example 49. If the CQ in Example 46 was not formulated in Madrid time, it would have to be transformed.

Time zones are a complex consideration. For example, the Northern and Southern hemispheres adjust Daylight Saving Time at opposite times of the year (according to seasonal differences in the two hemispheres) [68]. Because of these difficulties, we have not yet worked with or tested this attribute in our ontologies.

Heuristic 27 (total ordering). If we disregard time with branching future in our ontology, then we assume total ordering. Until now, all our ontologies assume total ordering.

Heuristic 28 (modeling of infinity). If the ontology has to reason with intervals where at least one endpoint is not limited, then it has to model infinity.

Example 50. This is useful in formal models in disciplines such as physics, biology and economics. For instance, suppose we have (Fact 1), *three individuals are born every minute in community 1*; (Fact 2) *one individual dies every minute in community 1*; and (CQ) *When will community 1 be wiped out?* Note that, as the number of births is greater than the number of deaths, the time point of community extinction is infinity.

Heuristic 29 (density). If time is assumed to be continuous, then density should be represented.

Heuristic 30 (isomorphism to the real numbers). If total ordering, infinity and density are assumed, then isomorphism to real numbers is also assumed.

We have not observed so far the utility of the explicit **modeling** of proper intervals, although we do not rule out the possibility of its application in the future.

References

- [1] G. van Heijst, A. Schreiber, B. Wielinga, Using explicit ontologies in kbs development, *International Journal of Human Computer Studies* 45 (1997) 183–292.
- [2] R. Mizoguchi, J. Vanwelkenhuysen, M. Ikeda, Towards very large knowledge bases: knowledge building and knowledge sharing (KBKS'95), Ch. Task Ontology for reuse of problem solving knowledge, IOS Press, University of Twente, Enschede, The Netherlands, 1995, pp. 46–57.
- [3] L. Schneider, How to build a foundational ontology: the object-centered high-level reference ontology ochre, <http://www.ifomis.unisaarland.de/Research/Publications/forthcoming/ki2003epaper.pdf> 2004.
- [4] P. Grenon, Spatio-temporality in basic formal ontology snap and span, upper-level ontology, and framework for formalization part i, Tech. rep, Leipzig University, 2003. (http://www.ifomis.org/Research/IFOMISReports/IFOMIS%20Report%2005_2003.pdf).
- [5] B. Smith, C. Rosse, The role of foundational relations in the alignment of biomedical ontologies, in: M. Fieschi, E. Coiera, Y.-C. Li (Eds.), 11th World Congress on Medical Informatics (MEDINFO 2004), IOS Press, Amsterdam, San Francisco, California, 2004.
- [6] S. Schulz, A. Kumar, T. Bittner, Biomedical ontologies: what part-of is and isn't, *Journal of Biomedical Informatics* 39 (2006) 350–361.
- [7] A. Gangemi, N. Guarino, C. Massolo, A. Oltramari, Understanding top-level ontological distinctions, in: A. Gómez-Pérez, M. Grüninger, H. Stuckenschmidt, M. Uschold (Eds.), IJCAI'01 Workshop on Ontologies and Information Sharing, CEUR Workshop Proceedings 47:26–33, 2001, pp. 26–33, (Amsterdam, The Netherlands (<http://ceur-ws.org/Vol-47/>), Seattle, Washington).
- [8] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, A. Oltramari, The wonderweb library of foundational ontologies, WonderWeb Deliverable D18, Ontology Library (final), 2003. (<http://wonderweb.semanticweb.org/deliverables/documents/D18.pdf>).
- [9] P. Borst, Construction of engineering ontologies, (Ph.D. thesis) Centre for Telematica and Information Technology, University of Twente, Enschede, The Netherlands, 1997.
- [10] D. Peralta, H. Pinto, N. Mamede, Enterprise Information Systems V, Ch. Reusing a Time Ontology, Springer, Dordrecht, The Netherlands, 2005, pp. 241–248.
- [11] S. Farrar, J. Bateman, General ontology baseline, OntoSpace Project Report Deliverable 1, University of Bremen, 2005.
- [12] M. Fernández-López, A. Gómez-Pérez, Searching for a time ontology for semantic web applications, in: A. Vari, L. Vieu (Eds.), 3th International Conference on Formal Ontology in Information Systems (FOIS'04), IOS Press, Amsterdam, Turin, Italy, 2004, pp. 331–441.
- [13] M. Fernández-López, A. Gómez-Pérez, M. Suárez-Figueroa, Selecting and customizing a mereology ontology for its reuse in a pharmaceutical product ontology, in: M. Grüninger, C. Eschenbach (Eds.), Formal Ontology in Information Systems. Fifth International Conference (FOIS'08), IOS Press, Amsterdam, The Netherlands, Saarbrücken, Germany, 2008, pp. 181–194.
- [14] M. Suárez-Figueroa, Neon methodology for Building Ontology Networks: Specification, Scheduling and Reuse (Ph.D. thesis) Facultad de Informática, Universidad Politécnica de Madrid, Spain, 2010.
- [15] J. Sowa, Knowledge Representation, Brooks/Cole, 2000.
- [16] A. Varzi, Handbook of Spatial Logics, Ch. Spatial Reasoning and Ontology: Parts, Wholes, and Locations, Springer-Verlag, 2007, pp. 945–1038.
- [17] P. Patel-Schneider, P. Hayes, I. Horrocks, Owl web ontology language semantics and abstract syntax, W3C Recommendation, <http://www.w3.org/TR/owl-semantics/> 2004.
- [18] I. Horrocks, B. Parsia, U. Sattler, Owl 2 web ontology language direct semantics, W3C Recommendation, <http://www.w3.org/TR/owl2-direct-semantics/> 2009.
- [19] G. Wang, Z. Lin, An argumentation framework for non-monotonic reasoning in description logic, in: T. Li, H. Nguyen, G. Wang, J. Grzymala-Busse, R. Janicki, A. Hassaniin, H. Yu (Eds.), 7th International Conference on Rough Sets and Knowledge Technology (RSKT'2012), Chengdu, China, Lecture Notes in Computer Science, vol. 7414, Springer Verlag, Berlin, Germany, 2012, pp. 200–206.
- [20] J. Iglesias, J. Lehmann, Towards integrating fuzzy logic capabilities into an ontology-based inductive logic programming framework, in: S. Ventura, A. Abraham, K. Cios, C. Romero, F. Marcelloni, J. Benítez, E. Gibaja (Eds.), 11th International Conference on Intelligent Systems Design and Applications (ISDA'2011), Córdoba, Spain, IEEE Press, Piscataway, New Jersey, USA, 2011, pp. 1323–1328.
- [21] A. Gangemi, P. Mika, Understanding the semantic web through descriptions and situations, in: R. Meersman, Z. Tari (Eds.), DOA/CoopIS/ODBASE 2003 Confederated International Conferences DOA, CoopIS and ODBASE, Vol. LNCS 2519, 2003.
- [22] R. Studer, V. Benjamins, D. Fensel, Knowledge engineering: principles and methods, *IEEE Transactions on Data and Knowledge Engineering* 25 (1–2) (1998) 161–197.
- [23] N. Guarino, P. Garetta, Ontologies and knowledge bases: Towards a terminological clarification, in: N. Mars (Ed.), Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing (KBKS'95), University of Twente, Enschede, The Netherlands, IOS Press, Amsterdam, The Netherlands, 1995, p. 25–32.
- [24] A. Gómez-Pérez, M. Fernández-López, O. Corcho, Ontological Engineering, Springer Verlag, 2003.
- [25] M. Grüninger, M. Fox, Methodology for the design and evaluation of ontologies, in: D. Skuce (Ed.), IJCAI95 Workshop on Basic Ontological Issues in Knowledge Sharing, 1995, pp. 6.1–6.10.
- [26] R. Wille, Formal concept analysis, Vol. LNAI 3626, Ch. Formal Concept Analysis as Mathematical Theory of Concepts and Concept Hierarchies, Springer-Verlag, Berlin Heidelberg, 2005, pp. 1–33.
- [27] G. Stumme, Handbook on ontologies, Ch. Formal Concept Analysis, Springer-Verlag, Berlin Heidelberg, 2009, pp. 177–199.
- [28] R. Wille, Ordered sets, Ch. Restructuring Lattice Theory: an Approach Based on Hierarchies of Concepts, Reidel, Dordrecht-Boston, 1982, pp. 445–470.
- [29] C. Keet, A. Artale, Representing and reasoning over a taxonomy of part-whole relations, *Applied Ontology* 3 (2008) 91–110.
- [30] M. Winston, R. Chaffin, D. Herrmann, A taxonomy of part-whole relations, *Cognitive Science* 11 (4) (1987) 417–444.
- [31] G. Guizzardi, The problem of transitivity of part-whole relations in conceptual modeling revisited, in: C. Wieringa (Ed.), 21st International Conference on Advanced Information Systems Engineering (CAISE'09), Amsterdam, The Netherlands, 2009.
- [32] I. Johansson, Relations and predicates, Ch. On the Transitivity of the Parthood Relation, Ontos-Verlag, Frankfurt, Germany, 2004, pp. 161–181.
- [33] A. Artale, E. Franconi, N. Guarino, L. Pazzi, Part-whole relations in object-centered systems: an overview, *Data & Knowledge Engineering* 20 (3) (1996) 347–383.
- [34] L. Vieu, M. Aurnague, Categorization of spatial entities in language and cognition, Ch. Part-of relations, functionality and dependence, John Benjamins, 2006, pp. 307–336.
- [35] P. Hayes, A catalog of temporal theories, Tech. Rep. UIUC-BI-Al-96-01, the Beckman Institute and Departments of Philosophy and Computer Science University of Illinois, 1995. (<http://www.ihmc.us/users/phayes/timecatalog.pdf>).
- [36] J. Allen, Towards a general theory of actions and time, *Artificial Intelligence* 23 (1984) 123–154.
- [37] M. Linehana, E. Barkmeyerb, S. Hendryxc, The date-time vocabulary, in: M. Donnelly, G. Guizzardi (Eds.), Formal Ontology in Information Systems (FOIS'12), Graz, Austria, IOS Press, Amsterdam, The Netherlands, 2012, pp. 265–278.
- [38] R. Baumann, F. Loebe, H. Herre, Ontology of time in gfo, in: M. Donnelly, G. Guizzardi (Eds.), Formal Ontology in Information Systems (FOIS'12), Graz, Austria, IOS Press, Amsterdam, The Netherlands, 2012, pp. 293–306.
- [39] J. Hobbs, F. Pan, Time ontology in owl, W3C Working Draft, <http://www.w3.org/TR/owl-time/2006>.
- [40] C. Bettini, X. Wang, S. Jajodia, A general framework and reasoning models for time granularity, in: L. Chittaro, S. Goodwin, H. Hamilton, A. Montanari (Eds.), Proceedings of the Third International Workshop on Temporal Representation and Reasoning (TIME'96), Key West, Florida, 1996, pp. 104–111.
- [41] L. Lamport, Time, clocks, and the ordering of events in a distributed system, *Communications of the ACM* 21 (7) (1978) 558–565.
- [42] A. Cockburn, Writing Effective Use Cases, 15th print, Addison-Wesley, London, United Kingdom, 2006.
- [43] C. Morris, Foundations of the theory of signs, in: C.M.O. Neurath, R. Carnap (Eds.), International Encyclopedia of Unified Science, vol. I, 1938.
- [44] L. Ferro, Manual for the Annotation of Temporal Expressions, Tech. Rep ftp://jaguar.ncsl.nist.gov/ace/phase2/docs/mtrinstructionmanual_v1_32.pdf instruction The MITRE Corporation, 2001. (ftp://jaguar.ncsl.nist.gov/ace/phase2/docs/mtrinstructionmanual_v1_32.pdf).

- [45] R. Saurí, J. Littman, B. Knippen, R. Gaizauskas, A. Setzer, J. Pustejovsky, Timeml annotation guidelines version 1.2.1, Tech. rep., TimeML, http://timeml.org/site/publications/timeMLdocs/annguide_1.2.1.pdf2006.
- [46] I. Horrocks, P. Patel-Schneider, H. Boley, S. Tabet, B. Grosz, M. Dean, Swrl: a semantic web rule language combining owl and ruleml, W3C Member Submission, <http://www.w3.org/Submission/SWRL/2004>.
- [47] C. Schürmann, Recursion for higher-order encodings, in: L. Fribourg (Ed.), Proceedings of the 15th International Workshop on Computer Science Logic, Springer, London, United Kingdom, 2001.
- [48] P. Barlatier, R. Dapoigny, A type-theoretical approach for ontologies: the case of roles, *Applied Ontology* 7 (3) (2012) 311–356.
- [49] R. Dapoigny, R. Barlatier, Towards ontological correctness of part-whole relations with dependent types, in: A. Galton, R. Mizoguchi (Eds.), Formal Ontology in Information Systems (FOIS'10), Toronto, Canada, IOS Press, Amsterdam, The Netherlands, 2010, pp. 45–58.
- [50] A. Gómez-Pérez, A. Lozano-Tello, Business systems analysis with ontologies, Ch. Applying ONTOMETRIC Method to Measure the Suitability of Ontologies, Idea Group Publishing, 2005, pp. 249–269.
- [51] H. Pinto, J. Martins, A methodology for ontology integration, in: Y. Gil, M. Musen, J. Shavlik (Eds.), First International Conference on Knowledge Capture (KCAP'01), ACM Press, New York, Victoria, Canada, 2001, pp. 131–138.
- [52] A. Gómez-Pérez, M. Rojas, Ontological reengineering and reuse, in: D. Fensel, R. Studer (Eds.), 11th European Workshop on Knowledge Acquisition, Modeling and Management (EKAW'99), Lecture Notes in Artificial Intelligence LNAI 1621, Springer-Verlag, Berlin, Germany, Dagstuhl Castle, Germany, 1999, pp. 139–156.
- [53] J. Arpírez, A. Gómez-Pérez, A. Lozano, H. Pinto, (onto)2agent: an ontology-based www broker to select ontologies, in: A. Gómez-Pérez, R. Benjamins (Eds.), ECAI'98 Workshop on Applications of Ontologies and Problem-Solving Methods, Brighton, United Kingdom, 1998, pp. 16–24.
- [54] A. Lozano-Tello, Métrica de idoneidad de ontologías, (Ph.D. thesis) Universidad de Extremadura, Cáceres, Spain, 2002.
- [55] H. Linstone, M. Turoff, The Delphi Method: Techniques and Applications, Addison-Wesley, 1975.
- [56] N. Guarino, C. Welty, An overview of ontoclean, in: S. Staab, R. Studer (Eds.), Handbook on Ontologies, Springer, 2009.
- [57] D. Vrandečić, Handbook on ontologies, Ch. Ontology Evaluation, 2nd edition, Springer-Ver, Berlin Heidelberg, Germany, 2009, pp. 293–313.
- [58] M. Sabou, M. Fernández, Ontology engineering in a networked world, Ch. Ontology (Network) Evaluation, Springer, Berlin, Germany, 2012, pp. 193–212.
- [59] C. Welty, D. Ferrucci, What's in an instance?, Tech. rep, Department of Computer Science – Rensselaer Polytechnic Institute, Troy, New York, 1994. (<http://www.cs.vassar.edu/~weltyc/papers/instances/instances.pdf>).
- [60] M. Genesereth, R. Fikes, Knowledge interchange format. version 3.0. reference manual, Tech. Rep. Logic-92-1, Computer Science Department. Stanford University, California, 1992. (<http://meta2.stanford.edu/kif/Hypertext/kif-manual.html>).
- [61] G. Schreiber, B. Wielinga, H. Akkermans, W. van de Velde, A. Anjewierden, Cml: the commonkads conceptual modelling language, in: L. Steels, A. Schreiber, W. van de Velde (Eds.), A Future for Knowledge Acquisition. Proceedings of the 8th European Knowledge Acquisition Workshop (EKAW'94), Lecture Notes in Artificial Intelligence LNAI 867, Springer-Verlag, Berlin/Heidelberg, Hoegaarden, Belgium, 1994, pp. 1–25.
- [62] E. Kuzey, G. Weikum, Extraction of temporal facts and events from Wikipedia, in: J. Masanès, M. Spaniol, R. Baeza-Yates (Eds.), Proceedings TempWeb'12 Proceedings of the 2nd Temporal Web Analytics Workshop, ACM, Lyon, France, 2012, pp. 25–32.
- [63] In: M. Suárez-Figueroa, A. Gómez-Pérez, E. Motta, A. Gangemi (Eds.), Ontology Engineering in a Networked World, Springer, Berlin, Germany, 2012.
- [64] G. Zhao, Akem: an ontology engineering methodology in ff poirot, Tech. rep, The Semantics Technology and Applications Research Laboratory, Free University of Brussels, 2005.
- [65] H. Pinto, C. Tempich, S. Staab, Ontology engineering and evolution in a distributed world, in: S. Staab, R. Studer (Eds.), Handbook on ontologies, Springer, Berlin, Germany, 2009.
- [66] E. Simperl, Reusing ontologies on the semantic web: a feasibility study, *Data & Knowledge Engineering* 68 (10) (2009) 905–925.
- [67] D. Lonsdale, D. Ebley, Y. Ding, L. Xu, M. Hepp, Reusing ontologies and language components for ontology generation, *Data & Knowledge Engineering* 69 (4) (2010) 318–330.
- [68] A. Phillips, F. Sasaki, M. Davis, M. Dürst, Working with time zones, W3C Working Group Note, <http://www.w3.org/TR/2005/NOTE-timezone-20051013/2005>.
- [69] J. Gosling, B. Joy, G. Steele, The Java Language Specification, Sun Developer Network, <http://java.sun.com/docs/books/jls/1996>.
- [70] E. Prud'hommeaux, A. Seaborne, Sparql query language for rdf, W3C Recommendation, <http://www.w3.org/TR/rdf-sparql-query/2008>.
- [71] Information technology – programming languages – prolog part 1: general core, 1995.
- [72] P. Valtchev, D. Grosser, C. Roume, M. Hacene, Galicia: an open platform for lattices, in: B. Ganter, A. de Moor (Eds.), Using Conceptual Structures: Contributions to the 11th International Conference on Conceptual Structures (ICCS'2003), Shaker Verlag, Aachen, Germany, 2003, pp. 241–254.
- [73] M. Genesereth, N. Nilsson, Logical Foundations of Artificial Intelligence, Morgan Kaufmann Publishers, Inc., Los Altos, California, 1987.
- [74] E. Mendelson, Introduction to Mathematical Logic, 4th edition Chapman & Hall/CRC, 2001.