A Double Classification of Common Pitfalls in Ontologies

María Poveda-Villalón, Mari Carmen Suárez-Figueroa, Asunción Gómez-Pérez
Ontology Engineering Group. Departamento de Inteligencia Artificial.
Facultad de Informática, Universidad Politécnica de Madrid.
Campus de Montegancedo s/n.
28660 Boadilla del Monte. Madrid. Spain
mpoveda@delicias.dia.fi.upm.es, {mcsuarez, asun}@fi.upm.es

Abstract. The application of methodologies for building ontologies has improved the ontology quality. However, such a quality is not totally guaranteed because of the difficulties involved in ontology modelling. These difficulties are related to the inclusion of anomalies or worst practices in the modelling. In this context, our aim in this paper is twofold: (1) to provide a catalogue of common worst practices, which we call pitfalls, and (2) to present a double classification of such pitfalls. These two products will serve in the ontology development in two ways: (a) to avoid the appearance of pitfalls in the ontology modelling, and (b) to evaluate and correct ontologies to improve their quality.

Keywords: pitfalls, worst practices, ontology evaluation, ontology engineering

1 Introduction

The 1990s and the first years of this new millennium have witnessed the growing interest of many practitioners in methodologies that support the creation of ontologies. All these approaches have supposed a step forward since they have transformed the art of building ontologies into an engineering activity. The correct application of such methodologies benefits the ontology quality. However, such a quality is not totally guaranteed because developers must tackle a wide range of difficulties and handicaps when modelling ontologies [1, 2, 7, 11]. These difficulties can imply the appearance of anomalies or worst practices in ontologies. Therefore, it is important to evaluate the ontologies before using or reusing them in other ontologies and/or semantic applications.

One of the crucial issues in ontology evaluation is the identification of anomalies or worst practices in the ontologies. In this regard, it is worth mentioning that in [11] the authors describe a set of common errors made by developers during the ontology modelling. Moreover, in [4] a classification of errors identified during the evaluation of consistency, completeness, and conciseness of ontology taxonomies is provided. Finally, in [8, 9] authors identify an initial set of common pitfalls.

In this context, our main goal is to provide a catalogue of common worst practices, which we call pitfalls. The catalogue is the result of an empirical analysis of a set of ontologies developed in OWL. The set of ontologies has been obtained from an experiment on ontology development. In addition, we aim to present a double classification of the abovementioned pitfalls. The pitfall catalogue and the double classification will serve as methodological groundings during the ontology development in two senses: (1) to avoid the appearance of worst practices in ontologies and (2) to evaluate and correct ontologies to improve their quality.

The remainder of the paper is structured as follows: Section 2 summarizes the state of the art on ontology evaluation. Section 3 describes the experiment carried out to obtain the ontologies for our analysis, whose aim is to find pitfalls in the ontologies. Section 4 includes the catalogue of pitfalls together with their definitions. Section 5 presents two possible classifications of the pitfalls in our catalogue. Finally, Section 6 concludes and shows some future lines of work.

2 State of the Art

Ontology evaluation is an important activity to be carried out during the whole ontology life cycle. The goal of ontology evaluation is to determine i) what the ontology defines correctly, ii) what it does not define, or iii) what it defines incorrectly.

The first works on ontology content evaluation started in 1994 and the main efforts were made by Gómez-Pérez [4, 5] and by Guarino and colleagues with the OntoClean method [6]. During recent years ontology evaluation has attracted a considerable amount of attention within the research community.

To summarize, the most common evaluation approaches are [2] (a) comparison of the ontology to a "gold standard" (which may itself be an ontology), (b) use of the ontology in an application and evaluation of the results, (c) comparison of the ontology with a source of data about the domain to be covered (e.g., a set of documents), and (d) evaluation by human experts who assess how the ontology meets the requirements. In addition to these existing methods, more recently also methods for pattern-based evaluations have emerged [10]. Pattern-based evaluation is carried out through a checking of the ontology to be evaluated against the issues and solutions represented by the patterns.

Additionally, in [3] a multi-layered approach to ontology evaluation is presented. In this approach there are three layers that are directly related to evaluation: (1) O2, a meta-ontology that allows to treat an ontology as a semiotic object; (2) oQual (for Ontology Quality), a pattern based on O2, which models ontology evaluation as a diagnostic task; and (3) qood (for Quality-Oriented Ontology Description), the component of oQual which describes the desired evaluation criteria. Based on this approach, the quality of an ontology may be measured relative to three main groups of dimensions, which are (a) structural, which refers to syntax and formal semantics; (b) functional, which is related to the intended use of the ontology; and (c) usability-related, which refers to the communication context of an ontology.

The approach in [4] proposes the dimensions of consistency, completeness, conciseness, expandability, and sensitiveness for ontology evaluation. In addition, the author presents a set of potential problems that can appear when ontology developers model taxonomic knowledge in ontologies bearing in mind the consistency, completeness, and conciseness dimensions.

However, as far as we know, there are no ontology evaluation approach that takes into account the abovementioned ontology quality dimensions together with the possible anomalies that can be included in the ontology.

3 Experiment on Ontology Development

In this section we present the setting of the experiment we have carried out on ontology development. Such an experiment was performed with the aim of analyzing the resulting ontologies with an evaluation approach. The main goal of the ontology analysis was to identify a set of pitfalls that developers make when building ontologies.

In Section 3.1 we describe the session setup for the experiment, whose results were the set of ontologies to be analyzed. After that, in Section 3.2, we present the evaluation approach we followed during the analysis performed over the abovementioned ontologies. The analysis allowed us to identify a set of pitfalls that ontology developers, mainly beginners, often make when modelling ontologies.

3.1 Session Setup

We divided the experiment into 3 separate sessions. Each session involved a different set of participants and was held at different periods in time. All the sessions were carried out involving participants with background in computer science and some experience in ontology engineering. Participants, who worked in groups of up to two, were distributed in the three following sessions:

S1. 22 participants attended the "Ontologies and the Semantic Web" course at the Master on Information Technologies (in the following IT Master 07-08) at the *Universidad Politécnica de Madrid (UPM)*, from October 2007 to February 2008.

S2. 10 participants attended the "Ontologies and the Semantic Web" course at the Master on Information Technologies (in the following IT Master 09-10) at UPM, from October 2009 to February 2010.

S3. 19 participants attended the "Ontologies and the Semantic Web" course at the Master of Research on Artificial Intelligence (in the following AI Master 09-10) at UPM, from October 2009 to February 2010.

In all the sessions participants were taught in:

1. the theoretical foundations of ontologies
2. ontologies and terminologies
3. ontology languages, including RDF(S) and OWL
4. methodologies for building ontologies, specifically, the NeOn Methodology framework [12]. In particular, guidelines for specifying requirements, scheduling, reusing, reengineering, and mapping knowledge resources as well as modelling principles.

Additionally, all the sessions incorporated practical trainings[1] in:

PT1. building an initial ontology in RDF(S) and OWL

PT2. specifying the ontology requirements and writing the ontology requirements specification document (ORSD) [12, 13] in one of the domains proposed by the trainers and shown in Table 1

PT3. reusing and reengineering non-ontological resources

PT4. reusing general ontologies

PT5. modelling ontologies

At the end of each session, participants had to perform a global and final practical training, which consisted on the development of an OWL ontology in a certain domain, which was selected at the time when the requirement specification activity was performed (practical training PT2). It is worth mentioning that the trainers reviewed, corrected, and improved all the ORSDs provided by the participants; and thus, participants used the corrected ORSD during the global practical training.

In Table 1 you can observe both the domains proposed by the trainers and the number of ontologies developed by the participants in each session with respect to the domains proposed. As a result of this experiment, we obtained 26 ontologies: 11 ontologies in session S1 developed with Protégé, 5 ontologies in session S2 developed both with Protégé and NeOn Tookit, and 10 ontologies in session S3 built with Protégé and NeOn Toolkit.

Table 1. Domains proposed and number of ontologies by domains

Domain	S1	S2	S3	Total
Architecture	2			**2**
Art	2			**2**
Biological organisms			1	**1**
Community services	4			**4**
Electronic devices		1		**1**
Food			2	**2**
Football		1	2	**3**
Geography	2			**2**

Domain	S1	S2	S3	Total
Judo			1	**1**
Movies		1		**1**
Music			1	**1**
Personality	1			**1**
Religion			1	**1**
Theatre			1	**1**
Travelling		1		**1**
Vehicles		1	1	**2**

3.2 Ontology Analysis with an Evaluation Approach

The focus of the ontology analysis performed with the 26 ontologies was on the following ontology evaluation dimensions: structural, functional, and usability-profiling [3]. The analysis was carried out by manually inspecting each ontology with respect to the three abovementioned dimensions. For each dimension we propose a set of aspects that were analyzed in the ontologies.

Structural dimension: In this context we focus on syntax and formal semantics. In this regard we propose the following aspects:

[1] It is worth mentioning that these practical trainings were performed by participants after the corresponding theoretical training. The results of the practical trainings performed for each participant group were evaluated and corrected by the trainers, who provided participants with the corresponding feedback and comments.

- Modelling decisions: this aspect refers to evaluate if developers use the primitives provided by ontology implementation languages in a correct way, or if there are modelling decision that could be improved.
- Real world modelling or common sense: this aspect deals with well-known knowledge that has not been represented in the ontology.
- No inference: this aspect refers to check if desirable or useful knowledge is not inferred.
- Wrong inference: this aspect refers to the evaluation of the inference of erroneous or non valid knowledge.

Functional dimension: This dimension is related to the intended use of a given ontology; thus, the focus is on the ontology conceptualization. In this sense we observed if all the features that were expected to be found in the ontology were in place. In this regard we propose the following aspect:

- Requirement completeness: this aspect refers to the coverage of the requirements specified in the ORSD by the ontology.

Usability-profiling dimension: This dimension refers to the communication context of an ontology. In this sense we wanted to find out if the ontology provides information that facilitates its understanding. For this aim we propose the following two aspects:

- Ontology understanding: this aspect refers to the evaluation of any kind of information that can help the user to understand the ontology.
- Ontology clarity: this aspect refers to the properties of ontology elements of being easily recognizable and understood by the user.

As a result of this analysis, we obtained a set of shortcomings in the ontologies that we identified as pitfalls. We want to note that we understand pitfalls as a clue to find possible mistakes that often appear when modelling ontologies.

The results of this analysis are shown in the next sections. Section 4 presents a catalogue of pitfalls that include not only those pitfalls discovered in this experiment but also others identified in [4, 7, 11]. Section 5 proposes a double categorization of the pitfalls based on the one hand on the abovementioned dimensions [3] and on the aspects proposed, and on the other hand on the evaluation criteria defined in [4].

4 Catalogue of Common Pitfalls

The manual inspection of the 26 ontologies implemented in OWL that were obtained in the global practical training explained in Section 3.1 allowed us to demonstrate that there are common pitfalls that occur when developers build ontologies (a) being aware of modelling techniques and ontological foundations, but (b) without knowing in advance the set of possible anomalies in ontology modelling. As a result of this empirical analysis, in this section we present the catalogue of 24 pitfalls[2] with examples of all of them extracted from the 26 ontologies analyzed. The current catalogue includes the pitfall definitions, that is, what is the anomaly, and some related examples. It is worth mentioning that we are working on the extension of the

[2] http://delicias.dia.fi.upm.es/wiki/index.php/Catalogue-of-Pitfalls

pitfall catalogue by means of including the methodological guidelines to avoid the pitfalls, that is, the solution to these ontology anomalies.

P1. **Creating polysemous elements:** an ontology element whose name has different meanings is included in the ontology to represent more than one conceptual idea. For example, the class "Theatre" is used to represent both the artistic discipline and the place in which a play is performed.

P2. **Creating synonyms as classes:** several classes whose identifiers are synonyms are created and defined as equivalent. As an example we could define "Car", "Motorcar" and "Automobile" as equivalent classes. Another example is to define the classes "Waterfall" and "Cascade" as equivalents. This pitfall is related to the guidelines presented in [7] which explain that synonyms for the same concept do not represent different classes.

P3. **Creating the relationship "is" instead of using "subclassOf", "instanceOf" or "sameIndividual":** the "is" relationship is created in the ontology instead of using OWL primitives for representing the subclass relationship ("subclassOf"), the membership to a class ("instanceOf"), or the equality between instances ("sameAs"). An example of this type of pitfall is to define the class "Actor" in the following way 'Actor ≡ Person ⊓ ∃interprets.Actuation ⊓ ∃is.Man'. This pitfall is related to the guidelines for understanding the "is-a" relation provided in [7].

P4. **Creating unconnected ontology elements:** ontology elements (classes, relationships or attributes) are created with no relation to the rest of the ontology. An example of this type of pitfall is to create the relationship "memberOfTeam" and to miss the class representing teams; thus, the relationship created is isolated in the ontology.

P5. **Defining wrong inverse relationships:** two relationships are defined as inverse relations when actually they are not. For example, something is sold or something is bought; in this case, the relationships "isSoldIn" and "isBoughtIn" are not inverse.

P6. **Including cycles in the hierarchy** [4, 7]**:** a cycle between two classes in the hierarchy is included in the ontology, although it is not intended to have such classes as equivalent. That is, some class A has a subclass B and at the same time B is a superclass of A. An example of this type of pitfall is represented by the class "Professor" as subclass of "Person", and the class "Person" as subclass of "Professor".

P7. **Merging different concepts in the same class:** a class is created whose identifier is referring to two or more different concepts. An example of this type of pitfall is to create the class "StyleAndPeriod", or "ProductOrService".

P8. **Missing annotations:** ontology terms lack annotations properties. This kind of properties improves the ontology understanding and usability from a user point of view.

P9. **Missing basic information:** needed information is not included in the ontology. Sometimes this pitfall is related with the requirements in the ORSD [12, 13] that are not covered by the ontology. Other times it is related with knowledge that could be added to the ontology in order to make it more complete. An example of this type of pitfall is to create the relationship "startsIn" to represent that the routes have a starting point in a particular location; and to miss the relationship "endsIn" to show that a route has an end point. Another example is to create the relationship "follows" when modelling order relations; and do not create its inverse relationship "precedes".

P10. **Missing disjointness** [4, 7, 11]**:** the ontology lacks disjoint axioms between classes or between properties that should be defined as disjoint. For example, we can create the classes "Odd" and "Even" (or the classes "Prime" and "Composite") without being disjoint; such representation is not correct based on the definition of these types of numbers.

P11. **Missing domain or range in properties:** relationships and/or attributes without domain or range (or none of them) are included in the ontology. There are situations in which the relation is very general and the range should be the most general concept "Thing". However, in other cases, the relations are more specific and it could be a good practice to specify its domain and/or range. An example of this type of pitfall is to create the relationship "hasWritten" in an ontology about art in which the relationship domain should be "Writer" and the relationship range should be "LiteraryWork". This pitfall is related to the common error when defining ranges and domains described in [11].

P12. **Missing equivalent properties:** when an ontology is imported into another, classes that are duplicated in both ontologies are normally defined as equivalent classes. However, the ontology developer misses the definition of equivalent properties in those cases of duplicated relationships and attributes. For example, the classes "CITY" and "City" in two different ontologies are defined as equivalent classes; however, relationships "hasMember" and "has-Member" in two different ontologies are not defined as equivalent relations.

P13. **Missing inverse relationships:** there are two relationships in the ontology that should be defined as inverse relations. For example, the case in which the ontology developer omits the inverse definition between the relations "hasLanguageCode" and "isCodeOf", or between "hasReferee" and "isRefereeOf".

P14. **Misusing "allValuesFrom"** [11]**:** this pitfall can appear in two different ways. In the first, the anomaly is to use the universal restriction ("allValuesFrom") as the default qualifier instead of using the existential restriction ("someValuesFrom"). This means that the developer thinks that "allValuesFrom" implies "someValuesFrom". In the second, the mistake is to include "allValuesFrom" to close off the possibility of further additions for a given property. An example of this type of pitfall is to define the class "Book" in the following way 'Book $\equiv \exists$producedBy.Writer $\sqcap\ \forall$uses.Paper' and closing the possibility of adding "Ink" as an element used in the writing.

P15. **Misusing "not some" and "some not"** [11]**:** to mistake the representation of "some not" for "not some", or the other way round. An example of this type of pitfall is to define a vegetarian pizza as any pizza which both has *some* topping which is *not* meat and also has *some* topping which is *not* fish. This example is explained in more detail in [11].

P16. **Misusing primitive and defined classes** [11]**:** to fail to make the definition 'complete' rather than 'partial' (or 'necessary and sufficient' rather than just 'necessary). It is critical to understand that, in general, nothing will be inferred to be subsumed under a primitive class by the classifier. This pitfall implies that the developer does not understand the open world assumption. A more detailed explanation and examples can be found in [11].

P17. **Specializing too much a hierarchy:** the hierarchy in the ontology is specialized in such a way that the final leaves cannot have instances, because they are actually instances and should have been created in this way instead of being created as classes. Authors in [7] provide guidelines for distinguishing between a class and an instance when modelling hierarchies. An example of this type of pitfall is to create the class “RatingOfRestaurants” and the classes “1fork”, “2forks”, and so on, as subclasses instead of as instances. Another example is to create the classes “Madrid”, “Barcelona”, “Sevilla”, and so on as subclasses of “Place”. This pitfall could be also named “Individuals are not Classes”.

P18. **Specifying too much the domain or the range** [7, 11]**:** not to find a domain or a range that is general enough. An example of this type of pitfall is to restrict the domain of the relationship “isOfficialLanguage” to the class “City”, instead of allowing also the class “Country” to have official language or a more general concept such as “GeopoliticalObject”.

P19. **Swapping intersection and union:** the ranges and/or domains of the properties (relationships and attributes) are defined by intersecting several classes in cases in which the ranges and/or domains should be the union of such classes. An example of this type of pitfall is to create the relationship “takesPlaceIn” with domain “OlympicGames” and with range the intersection of the classes “City” and “Nation”. Another example can be to create the attribute “Name” for the classes “City” and “Drink” and to define its domain as the intersection of both classes. This pitfall is related to the common error that appears when defining ranges and domains described in [11] and also related to the guidelines for defining these elements provided in [7].

P20. **Swapping Label and Comment:** the contents of the Label and Comment annotation properties are swapped. An example of this type of pitfall is to include in the Label annotation of the class “Crossroads” the following sentence ’the place of intersection of two or more roads’; and to include in the Comment annotation the word ‘Crossroads’.

P21. **Using a miscellaneous class:** to create in a hierarchy a class that contains the instances that do not belong to the sibling classes instead of classifying such instances as instances of the class in the upper level of the hierarchy. This class is normally named “Other” or “Miscellaneous”. An example of this type of pitfall is to create the class “HydrographicalResource”, and the subclasses “Stream”, “Waterfall”, etc., and also the subclass “OtherRiverElement”.

P22. **Using different naming criteria in the ontology:** no naming convention is used in the identifiers of the ontology elements. Some notions about naming conventions are provided in [7]. For example, we can name a class by starting with upper case, e.g. “Ingredient”, and its subclasses by starting with lower case, e.g. “animalorigin”, “drink”, etc.

P23. **Using incorrectly ontology elements:** an ontology element (class, relationship or attribute) is used to model a part of the ontology that should be modelled with a different element. A particular case of this pitfall regarding to the misuse of classes and property values is addressed in [7]. An example of this type of pitfall is to create the relationship “isEcological” between an instance of “Car” and the instance “Yes” or “No”, instead of creating the attribute “isEcological” whose range is Boolean.

P24. **Using recursive definition:** an ontology element is used in its own definition. For example, it is used to create the relationship “hasFork” and to establish as its range the following ’the set of restaurants that have at least one value for the relationship “hasFork”.

In addition, we include in Table 2 the figures of the appearance of the 24 pitfalls identified in the 26 ontologies obtain in Section 3.1. During the analysis, we have observed that some pitfalls can appear almost in all the elements of the ontology, for example, the P8 “Missing annotations”. In contrast, other pitfalls only appear in specific parts of the ontology for example, the P13 “Missing inverse relationships”. Because of this, the numbers in Table 2 show how many ontologies contain the pitfall instead of how many times appears the pitfall along the 26 ontologies. It is worth also mentioning that in Table 2 those pitfalls in bold come out from our experiment whereas the rest were reported in other works.

Table 2. Appearance of existing and new pitfalls

Pitfall	S1	S2	S3	Total
P1	3	0	0	**3**
P2	2	0	0	**2**
P3	1	0	1	**2**
P4	4	2	5	**11**
P5	2	0	1	**3**
P6	0	0	0	**0**
P7	2	2	2	**6**
P8	7	5	8	**20**
P9	8	3	5	**16**
P10	10	3	10	**23**
P11	8	4	13	**25**
P12	1	1	2	**4**

Pitfall	S1	S2	S3	Total
P13	4	1	1	**6**
P14	2	1	2	**5**
P15	0	0	0	**0**
P16	4	0	4	**8**
P17	7	3	5	**15**
P18	4	2	3	**9**
P19	6	1	6	**13**
P20	9	0	4	**13**
P21	2	0	1	**3**
P22	6	3	6	**15**
P23	2	2	4	**8**
P24	3	0	2	**5**

5 Classifications of the Common Pitfalls

This section presents two different classifications of the common pitfalls presented in the catalogue in Section 4. The first categorization is organized based on the dimensions defined in [3] and on the aspects proposed in Section 3.2. The second classification is focused on categorizing, if possible, the pitfalls, according to the evaluation criteria defined in [4] (consistency, completeness, and conciseness). We want to point out that in both classifications those pitfalls in bold come out from our experiment whereas the rest were reported in other works.

On the one hand, Fig. 1 shows the classification of the common pitfalls according to the following aspects related to the dimensions defined in [3]: (a) modelling decisions, real world modelling or common sense, no inference, and wrong inference related to the structural dimension; (b) requirement completeness related to the functional dimension; and (c) ontology understanding and ontology clarity related to the usability-profiling dimension. As we can observe in Fig. 1 each pitfall is classified at least with respect to one of the abovementioned aspects.

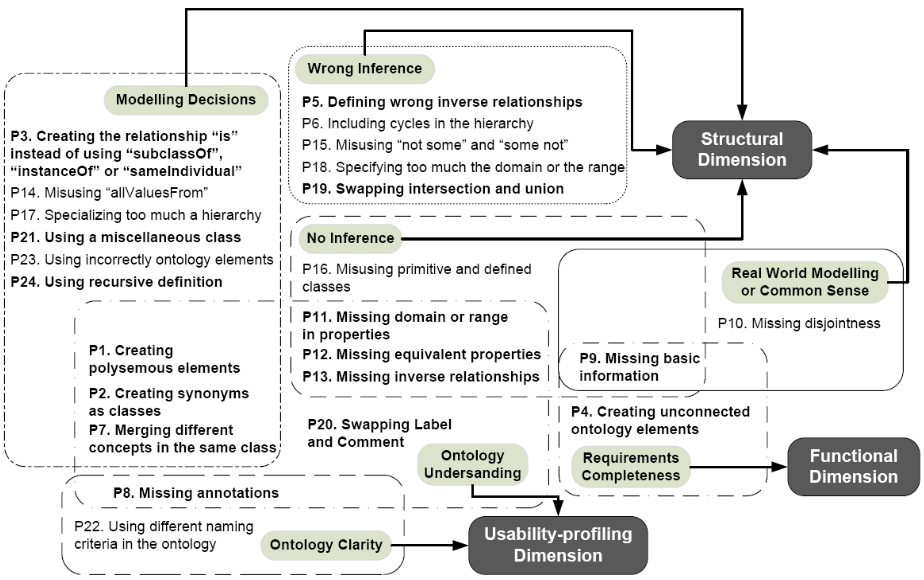

Fig. 1. Classification of the pitfalls according to the dimensions defined in [3]

On the other hand, we have performed a second classification of the pitfalls defined in Section 4 based on the evaluation criteria defined in [4]. As Fig. 2 shows, most of the common pitfalls identified match with the criteria defined in [4].

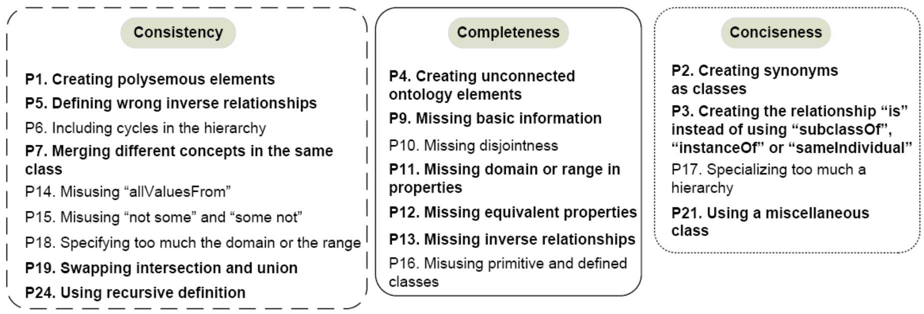

Fig. 2. Classification of the pitfalls according to the evaluation criteria defined in [4]

However, there are some common pitfalls that cannot be classified by these criteria, owing to the fact that such criteria were defined in [4] with respect to the ontology content (such as definitions and axioms). The pitfalls that could not be classified were "P8. Missing annotations", "P20. Swapping Label and Comment", and "P22. Using different naming criteria in the ontology", because they are not related to the ontology content. Finally, we have also observed that the "P23. Using incorrectly ontology elements" seems not to match with any criterion defined in [4]. Bearing this pitfall in mind and the aforementioned cases of P8, P20, and P22, we are working on analyzing the possibility of extending the criteria defined in [4] with the aim of being able to classify all the pitfalls included in our catalogue. Fig. 2 shows the classification of pitfalls following the evaluation criteria defined in [4]. This classification only contains those pitfalls that could be clearly classified for the time being; that is, pitfalls P8, P20, P22, and P23 do not appear in the classification due to the abovementioned reasons.

In addition, we realized that there are some correspondences between both classifications presented in this section. In this sense we can establish that pitfalls belonging to "No inference", "Requirement completeness", and "Real world modelling or common sense" groups[3] from Fig. 1 are examples of incompleteness, that is, they contradict the completeness criterion. Also, pitfalls belonging to the "Wrong inference" group[4] from Fig. 1 are examples of pitfalls that do not meet the criterion about consistency. The pitfalls classified in the "Modelling decisions" group do not match only one criterion from [4]; on the contrary, some of them (P2, P3, P17, and P21) are related to the conciseness criterion and others (P1, P7, P14, and P24) are related to the consistency of the ontology.

6 Conclusions and Future Lines of Work

This paper presents a catalogue of common pitfalls in ontology modelling. Such a catalogue has been identified by analysing 26 ontologies in different domains (e.g. art, architecture, football, vehicles, etc.) developed by participants who have attended one semester course on ontologies and semantic web foundations. We have identified a set of 24 pitfalls that occur when developers build ontologies having been taught in the theoretical foundations of ontologies, ontologies and terminologies, ontology languages (RDF(S) and OWL), and methodologies for building ontologies, but without knowing this set of pitfalls in advance.

We have also classified these common pitfalls according to two different approaches. The first one involves the structural, functional, and usability-related dimensions presented in [3] and some related aspects proposed and described in this paper. Such related aspects are respectively (a) modelling decisions, real world modelling or common sense, no inference and wrong inference, (b) requirement completeness, and (c) ontology understanding and ontology clarity. The second approach deals with a classification of the pitfalls using the evaluation criteria defined in [4]. In addition, it is worth mentioning that we have identified some correspondences between both classifications.

Both the catalogue of pitfalls and the double classification will serve as methodological groundings in ontology development in two different ways: (a) to avoid the appearance of pitfalls in ontology modelling and (b) to evaluate and correct ontologies, with the aim of improving their quality. In this sense, we are working on providing some methodological guidelines for helping ontology developers both in the modelling and in the evaluation of ontologies. To that end, our next steps aim to

[3] The list of pitfalls involved in such groups is P4, P9, P10, P11, P12, P13, and P16.

[4] The list of pitfalls involved in this group is P5, P6, P15, P18, and P19.

create pitfalls descriptions including methodological guidelines to avoid and/or repair the pitfalls. In addition, we have in mind to compare in depth the two classifications provided in this paper with the aim of establishing correspondences between them.

Another interesting point with respect to the pitfalls themselves is to analyze whether the ontology editor GUI could have an influence on the appearance of the pitfalls in our catalogue.

Acknowledgments. This work has been partially supported by the Spanish projects *mIO!* (CENIT-2008-1019) and *GeoBuddies* (TSI 2007-65677 C02). We would like to thank Elena Montiel-Ponsoda for her comments and revisions.

References

1. Aguado de Cea, G., Gómez-Pérez, A., Montiel-Ponsoda, E., Suárez-Figueroa, M.C. *Natural language-based approach for helping in the reuse of ontology design patterns*. In Knowledge Engineering: Practice and Patterns, Proceedings of EKAW 2008, LNCS 5268, pp. 32–47, 2008.
2. Blomqvist, E., Gangemi, A., Presutti, V. *Experiments on Pattern-based Ontology Design*. In Proceedings of K-CAP 2009, pp. 41-48. 2009.
3. Brank, J., Grobelnik, M., Mladenic, D. *A survey of ontology evaluation techniques*. In Proceedings of SiKDD 2005, Ljubljana, Slovenia. 2005.
4. Gangemi, A., Catenacci, C., Ciaramita, M., Lehmann J. *Modelling Ontology Evaluation* and *Validation.* Proceedings of ESWC2006, number 4011 in LNCS, Budva. 2006.
5. Gómez-Pérez, A. *Ontology Evaluation*. Handbook on Ontologies. S. Staab and R. Studer Editors. Springer. International Handbooks on Information Systems. Pp: 251-274. 2004.
6. Gómez-Pérez, A. *Some Ideas and Examples to Evaluate Ontologies*. Technical Report KSL-94-65. Knowledge System Lab. Stanford Univ. Also in Proceedings of CAIA94. 1994.
7. Guarino. N., Welty. C. *A Formal Ontology of Properties*. In Proc. of EKAW2000, LNAI 1937. Springer Verlag: Pp. 97-112. 2000.
8. Noy, N.F., McGuinness. D. L. *Ontology development 101: A guide to creating your first ontology.*Technical Report SMI-2001-0880, Standford Medical Informatics. 2001.
9. Poveda, M., Suárez-Figueroa, M.C., Gómez-Pérez, A. *Common Pitfalls in Ontology Development*. In "Current Topics in Artificial Intelligence, CAEPIA 2009 Selected Papers". Springer LNAI 5988. Pp: 91-100. 2010.
10. Poveda, M., Suárez-Figueroa, M.C., Gómez-Pérez, A. *Ontology Analysis Based on Ontology Design Patterns*. Proceedings of the WOP 2009 - Workshop on Ontology Patterns at ISWC 2009. Pp: 155-162. Washington. 2009.
11. Presutti, V, Gangemi, A, David S, Aguado, G, Suárez-Figueroa, MC, Montiel-Ponsoda, E, Poveda, M. *NeOn D2.5.1: A Library of Ontology Design Patterns: reusable solutions for collaborative design of networked ontologies.* NeOn project. (FP6-27595). 2008.
12. Rector, A., Drummond, N., Horridge, M., Rogers, J., Knublauch, H., Stevens, R.,; Wang, H., Wroe, C. *Owl pizzas: Practical experience of teaching owl-dl: Common errors and common patterns*. In Proc. of EKAW 2004, pp: 63–81. Springer. 2004.
13. Suárez-Figueroa, M.C. *PhD Thesis: NeOn Methodology for Building Ontology Networks: Specification, Scheduling and Reuse*. Spain. Universidad Politécnica de Madrid. June 2010.
14. Suárez-Figueroa, M.C., Gómez-Pérez, A., Villazón-Terrazas, B. *How to write and use the Ontology Requirements Specification Document*. In Proceedings of ODBASE 2009. On the OTM 2009. LNCS 5871. Pp: 966-982. Vilamoura, Algarve-Portugal. November 2009.