

Common Pitfalls in Ontology Development

María Poveda, 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 so-called Ontology Design Patterns (ODPs), which have been defined as solutions to ontological design problems, are of great help to developers when modelling ontologies since these patterns provide a development guide and improve the quality of the resulting ontologies. However, it has been demonstrated that, in many cases, developers encounter difficulties when they have to reuse the correct design patterns and include errors in the modelling. Thus, to avoid pitfalls in ontology modelling, this paper proposes classifying errors into two types: (1) errors related to existing ODPs, called *anti-patterns*, and (2) errors not related to existing ODPs, called *pitfalls*. This classification is the result of analysing a set of ontologies. This paper is focused on the pitfalls identified during the analysis. In addition the paper presents a classification of the pitfalls found and a set of pitfall examples.

Keywords: patterns, anti-patterns, pitfalls, ontology.

1 Introduction

Modelling ontologies has become one of the main topics of research within ontological engineering because of the difficulties it involves. In recent years, the emergence of Ontology Design Patterns (ODPs), which are defined as solutions to design problems [4], has supposed a great help to developers when modelling ontologies.

Some experiments [3] carried out in ontology engineering have demonstrated that design patterns are perceived as an aid to modelling ontologies, a development guide, and a way to improve the quality of the resulting ontologies. However, it is well known that [3, 1], in some cases, ontology developers experience difficulties when reusing the patterns during modelling, and include errors in the modelling. Therefore, in order to understand and use correctly ODPs, we need a better support that prevents the emergence of modelling errors.

Thus, to avoid the appearance of pitfalls in ontology development, we are working on the creation of a new set of methodological guides. These guides, based on the identification and classification of modelling errors, classify errors into two types: (1) errors related to ODPs, called *anti-patterns*; and (2) errors not related to ODPs, called *pitfalls*. In this paper, both types are presented. The paper is focused on the pitfalls

identified during the analysis. In addition, we include a classification of the pitfalls identified and a set of examples of such pitfalls.

The remainder of the article is structured as follows: Section 2 presents the state of the art of patterns and anti-patterns in ontological engineering. Section 3 describes the analysis carried out in 11 ontologies, which shows the presence of ODPs and describes anti-patterns, and pitfalls. Section 4 includes a template where the pitfalls are described as well as the classification of the pitfalls found and a set of examples of pitfalls. Finally, Section 5 includes the conclusions drawn and future lines of work.

2 State of the Art: Patterns and Anti-Patterns

The term “pattern” [4] appeared in the XIV century and derives from the Latin term “patronus”, which, among other meanings, stands for an item that can be imitated. In the 1970s Christopher Alexander introduced the term design pattern [2] to refer to those shared guides that help solve modelling problems.

In ontology engineering, the ontology design patterns can be considered as modelling solutions to problems widely known in the area. These solutions are based on good practices and solve modelling problems.

In the ODPs field, we can distinguish between logical patterns and conceptual patterns [4]. With regard to logical patterns, the W3C work team, known as “*Semantic Web Best Practices and Deployment*”¹, has established that in order to provide support to developers and users of the Semantic Web, a set of good practices is required. To that purpose, this group proposes patterns that solve design problems in the OWL² language, independently of the particular conceptualization, which solve logical problems. Regarding conceptual patterns [4], the author proposes patterns (in OWL or any other logical language) that solve design problems for specific domains, which solve content problems.

In addition to the distinction mentioned above, in [6] the authors propose the classification shown in Fig. 1.

The work described in [6] is focused on content patterns and provides guidelines on how to apply content ODPs using import, specialization, composition and expansion functions. In the same work, the content ontology anti-pattern concept is defined as a design that is different from a content pattern in that the former codes the solution to a problem in a wrong way.

However, we have observed that none of the papers analyzed have carried out a thorough study on the use of any type of ODPs and their corresponding anti-patterns. We have also observed that there is no previous work focused on identifying and preventing to model errors not related to any existing ODP.

It is worth mentioning that ODPs can be found in on-line libraries that include both the description and the OWL code associated to the patterns as, for example, “the *Ontology Design Pattern Wiki*”³, or they can be obtained from the work team “*Semantic Web Best Practices and Deployment*”. Some other libraries [6, 7] do not

¹ <http://www.w3.org/2001/sw/BestPractices/>

² <http://www.w3.org/TR/2004/REC-owl-guide-20040210/>

³ <http://ontologydesignpatterns.org/>

provide the pattern code, but they store descriptions of a great number of ODPs. These libraries, which follow a software engineering approach, use a template for describing the patterns included in the catalogue.

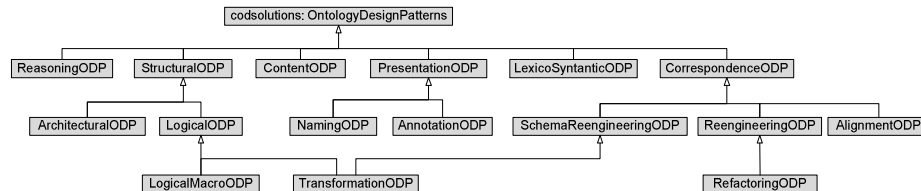


Fig. 1. Ontology Design Pattern types [6].

3 Identification of Patterns, Anti-Patterns and Pitfalls

With the aim of identifying the types of errors normally made when developing ontologies, we have analyzed 11 ontologies that tackle different aspects (art, architecture, geography, tastes and likings, and community services) of the domain of Saint James's Way. These ontologies were developed by Master students as a practical assignment on the "Ontologies and the Semantic Web" subject at the *Universidad Politécnica de Madrid*, during the 2007-2008 academic year. Due to the provenance of the ontologies, most of them were developed in Spanish. The students did not have any previous notion of ODPs; therefore, it did not make sense to focus our study on the correctly or incorrectly reuse of ODPs. Instead, we focused our work on examining whether ODPs, anti-patterns and pitfalls could be detected in the resulting ontologies. The approach adopted in the analysis was based on the manual search of ODPs in ontologies and on the manual identification of two types of modelling errors: (1) those related to ODPs, called *anti-patterns*; and those not related to ODPs, called *pitfalls*.

Taking into account the problem that each ontology intended to solve, the manual analysis of each ontology was centered on the search of ontology modules that corresponded to some of the following scenarios:

- Designs that match solutions proposed in some ODP within those available in libraries [6, 7].
- Design problems that could be solved by reusing some ODP within those available in libraries [6, 7].
- Non appropriate solutions for the design problems under consideration.

As a result of the analysis carried out, we have identified the possibilities shown in the tree of Fig. 2. In that tree, we can distinguish two main branches: the first one, whose nodes are linked by arrows in a continuous line, represents whether an ODP or an anti-pattern has been observed; the second one, whose nodes are linked by arrows in a discontinuous line, represents that no ODP has been observed, but where it is possible to apply some ODPs from the libraries.

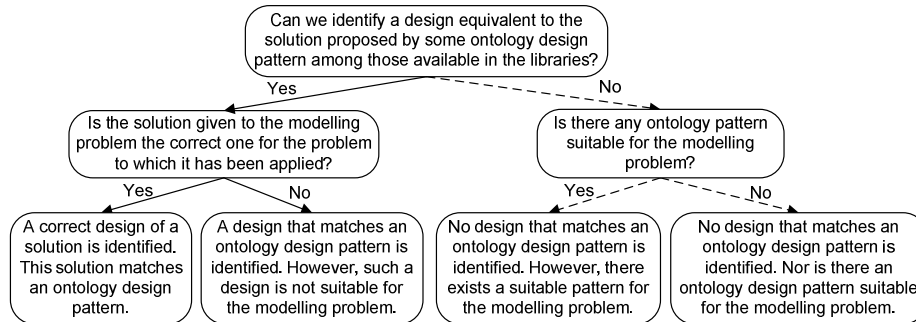


Fig. 2. Decision tree for classifying patterns, anti-patterns, and pitfalls.

The tree in Fig.2 classifies each of the scenarios found in some of the four leaf nodes and determines if such scenario matches a pattern identification, an anti-pattern identification, or a pitfall identification in the following way:

1. *Identification of an ontology design pattern.* (1st leaf from the left) We have identified some cases that have a correct design for the modelling problem; such a design matches an ODP of those in the available libraries.
2. *Identification of an anti-pattern.* (2nd and 3rd leaf) In some cases we have identified a design that matches an ODP but this design is not a suitable solution to the modelling problem. In some other cases we have not identified a design that could match an ODP in a suitable solution to the modelling problem; however, there is a suitable pattern that could have been applied if we had known the ODPs.
3. *Identification of a pitfall.* (4th leaf) In some other cases we have observed that there is an unsuitable solution to the modelling problem whereas there is not an ODP suitable for the modeling problem. Most of these cases have been treated as pitfalls; however, a few set of cases could have been solved by means of adapting or combining existing ODPs.

As a result of the analysis performed with 11 ontologies, we have found 208 cases in which we have identified a correct design of a solution that matches an ODP. We have also found 117 cases of anti-patterns, of which 83 correspond to situations in which the solution matches the modelling proposed in some ODP. However, such a solution is not suitable for the design problem intended to solve. Besides, we have found 34 cases in which we were unable to identify a design solution matching an ODP though a suitable ODP exists. Finally, we have come across a set of pitfalls, that is, solutions unsuitable for the design problem in the domain of the ontologies studied for such design problem; such pitfalls do not have an ODP associated. These pitfalls are described in Section 4.

4 Pitfalls: Description, Classification and Examples

As can be seen in Section 3, we have identified a set of pitfalls for solving a modelling problem for which there is no an available ODP.

To describe the pitfalls found, we propose the template shown in Table 1 that includes, among other fields, a series of recommendations to avoid pitfalls.

Table 1. Template describing possible pitfalls during ontology development.

Name	Representative name of the pitfall intended to avoid.	
Description		
This field can include a general description of the pitfall. It can also include the possible inconveniences that its use may imply and the different cases in which the pitfall can appear.		
Recommendation		
This field provides some guidelines or recommendations to avoid the use of the pitfall.		
Example⁴		
Description of a representative example of the application of the pitfall. In the “Not Recommended” field we can see a graphical representation of the example where the pitfall has been identified. In the “Recommended” field we can see a possible design for the problem proposed in the example; such a design tries to avoid the pitfall.		
Not Recommended	Recommended	
Graphic representation: Protégé screenshots.	Graphic representation: Protégé screenshots.	
Comments		
This optional field includes possible comments on the pitfall or the associate recommendation.		

In addition, we have classified the pitfalls identified in the ontologies studied bearing in mind the types of ODPs proposed in Fig.1. Each pitfall is associated to the type(s) of ODP to which a pattern created to avoid such a pitfall could belong.

As can be observed in the pitfalls classification appearing in Fig.3, the types of pitfalls identified are the following:

- *Annotation pitfall.* It refers to the ontology usability from the user’s point of view. Therefore, it means additional information in the form of annotations included in the ontology. These annotations should improve the user’s understanding of the ontology and its elements.
- *Reasoning pitfall.* It refers to the implicit knowledge derived from the ontology when reasoning procedures are applied to such an ontology.
- *Naming pitfall.* It refers to the ontology usability from the user’s point of view and, specifically, to the naming of the ontology elements. It is important to note that the naming of the elements should provide the users with a better understanding of the ontology.
- *Logical pitfall.* It refers to the solution to design problems in which the primitives of the representation language used do not provide support.
- *Content pitfall.* It refers to the solution to design problems related to the ontology domain.

⁴ Some examples found in Spanish have been translated into English to facilitate the reading.

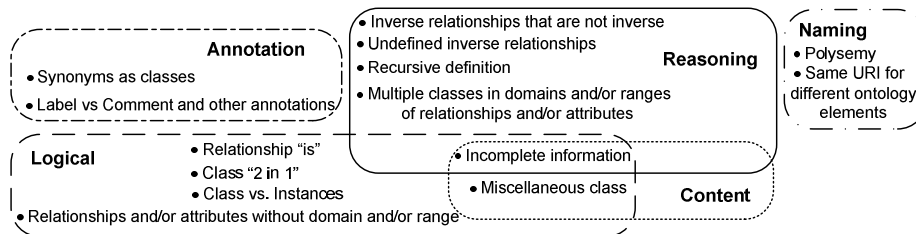


Fig. 3. Classification of the pitfalls identified.

Below we provide a brief description and an example of each pitfall classified. Some examples are provided both in Spanish and English.

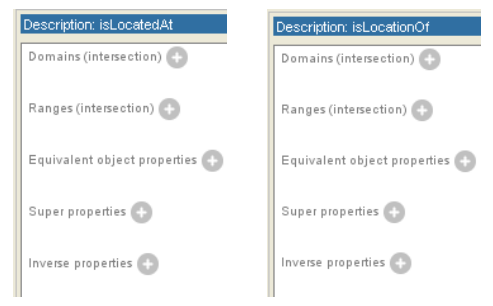

- **Synonyms as classes.** This pitfall consists both in creating several classes whose identifiers are synonyms and in defining them as equivalent.
- **Label vs. Comment and other annotations.** This pitfall consists in interchanging the contents of the annotations of the types "label" and "comment" and in not including any annotation of the type "label" and "comment".
- **Inverse relationships that are not inverse.** This pitfall consists in defining two relationships as inverse when, in fact, they are not.
- **Undefined inverse relationships.** This pitfall consists in having inverse relationships in the ontology, but they are not defined as such.
- **Recursive definition.** This pitfall entails using an ontology element in its own definition.
- **Multiple classes in domains and/or ranges of relationships and/or attributes.** This pitfall consists in defining the ranges and/or domains of the relationships and/or attributes by intersecting several classes in cases in which they should be the union of such classes.
- **Polysemy.** This pitfall entails using an ontology element to represent concepts different from the domain under consideration.
- **Same URI for different ontology elements.** This pitfall entails assigning the same URI to two different ontology elements.
- **Relationship "is".** This pitfall entails confusing the subclass relationship (*subclassOf*), the membership to a class (*instanceOf*), or the equality between instances (*sameIndividual*) with an ad hoc relation called "is".
- **Class 2 in 1.** This pitfall entails creating a class whose name is "Class1AndClass2".
- **Classes vs. Instances.** This pitfall consists in deepening into a hierarchy so that the more specific classes do not have instances since such classes become class instances of the upper level of the hierarchy.
- **Relationship and/or attributes without domain or range.** This pitfall consists in not specifying the domain or range in the relationships/attributes.
- **Incomplete information.** This pitfall entails not representing all the knowledge that could be included in the ontology.
- **Miscellaneous class.** This pitfall consists in creating an artificial miscellaneous class to classify in a certain level the instances not belonging to any of the sibling classes of this level.

In section 4.1 we present a set of pitfalls, related among them, that have appeared very frequently in the ontologies analyzed. These pitfalls are related among them in Section 4.2.

4.1 Pitfalls in Relationship Modelling

This section presents three examples of pitfalls, identified in the ontologies we have analyzed, that would affect the relationship modelling. The two first ones are related to reasoning patterns (Undefined inverse⁵ relationships (Table 2) and Inverse relationships that are not inverse (Table 3)), and the last one is related to logical patterns (Relationships and/or attributes without domain and/or range (Table 4)).

Table 2. Pitfall referring to “Undefined inverse relationships”.

Name	Undefined inverse relationships	
Description		
The pitfall consists in having inverse relationships in the ontology but not defining them as such. This implies that when reasoning we obtain less information than that we could infer.		
Recommendation		
<ul style="list-style-type: none"> When creating a new relationship, verify whether by putting the verb into the passive voice its inverse relationship can also be created; verify also whether this new relationship makes sense. Define the relationships by establishing, if possible, the domains and ranges. If two relationships have the domains and ranges inverted, that is, the domain of one is the range of the other and vice versa, it is probable that both are in inverse relation. Then, analyze their meanings. 		
Example		
<p>In the non-recommended scenario, within the ontology on architecture, we can observe how the relationships “isLocatedAt” and “isLocationOf” have been created without defining them as inverse. With this modelling, if we have that “building1 isLocatedAt site3”, then, when we activate a reasoner, we will not obtain that in “site3 isLocationOf building1”.</p> <p>In the recommendation we have added the domain and range that the relationship “isLocatedAt” could have.</p>		
Not Recommended		Recommended
		

⁵ If a relationship P1 is defined as inverse to other relationship P2, then for all x and y is satisfied that $P1(x,y) \leftrightarrow P2(y,x)$.

Table 3. Pitfall referring to “Inverse relationships that are not inverse”.

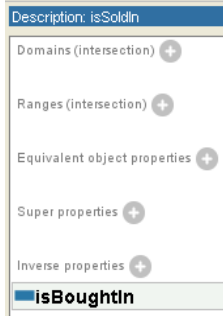
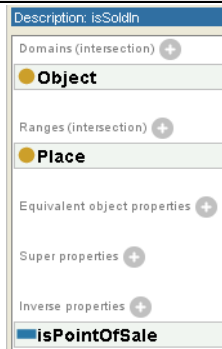
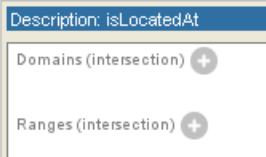

Name	Inverse relationships that are not inverse
Description	
<p>This pitfall consists in defining two relationships as inverse when, in fact, they are not.</p> <p>As a consequence of this pitfall, undesired knowledge is probably obtained when a reasoner is applied to the ontology. This is because if we have one of the instanced relationships, then the reasoner will conclude the inverse relationship, which could be incorrect in the domain.</p> <p>Some possible situations where these mistake can appear</p> <ul style="list-style-type: none"> ▪ Confuse two complementary relationships with inverse relationships. ▪ Do not check the bidirectional implication of the definition of the inverse relationships in both directions. That is, we want define the relationships A and B as inverse, and we know that whenever A occurs B occurs, but we do not check if whenever B occurs, A occurs necessarily. 	
Recommendation	
<ul style="list-style-type: none"> ▪ Check that if the verb from a relationship is put in passive the verb of the other relationship is obtained. ▪ Check that the domain of one of the relationships matches the range of the other, and vice versa. This is one of the reasons why we recommend to set the domain and range of the relationships whenever possible. ▪ Check that whenever a relationship between two individuals exists, the inverse relationship necessarily exists. 	
Example	
<p>In this example we can see the result of considering two complementary relationships, “toSell” and “toBuy”, as inverse.</p> <p>In the non-recommended scenario and within the ontology on art, we can observe how the relationships “isSoldIn” and “isBoughtIn” are defined as inverse. This fact could cause an error at the semantic level, since if we have that “object1 isSoldIn place3”, then the reasoner will infer that “place3 isBoughtIn object1”.</p> <p>In the recommended scenario, the relationship “isBoughtIn” has been changed to “isPointOfSale”. In this way, the semantic error above commented disappears, since if “object1 isSoldIn place3”, then the reasoner will infer that “place3 isPointOfSale object1”.</p> <p>It has been added in the recommendation the domain and range that could have the relationship “isSoldIn”.</p>	
Not Recommended	Recommended
	
Comments	
<p>It is possible that sold and buy are inverse relationships; for example, if the relationship refers to the people involved in buying and selling, then it can be said that “buyTo” (buy something to) is inverse of “sellTo” (sell something to). That is, if “company1 buysTo company2”, it will be correct to infer that “company2 sellsTo company1”.</p>	

Table 4. Pitfall referring to “Relationships and/or attributes without domain and/or range”.

Name	Relationships and/or attributes without domain and/or range	
Description		
This pitfall entails not specifying the domain and/or range in the relationships/attributes.		
Recommendation		
It is recommended to specify, if possible, the domains and/or ranges in the properties and/or attributes. Additionally, annotations in the “comment” fields should be added. These annotations should describe what we wish to represent with the Property and/or attribute; for instance, if the property or attribute forms part of an n-ary pattern.		
Example		
We can observe in the not recommended scenario, how the relationship “isLocatedAt” has been defined without specifying its domain or range, whereas in the recommended scenario, the relationship “isLocatedAt” has been defined by specifying its domain and range.		
	Not Recommended	Recommended
	 <p>The screenshot shows a property definition for 'isLocatedAt'. It has a description field and two expandable sections for 'Domains (intersection)' and 'Ranges (intersection)', both of which are currently empty.</p>	 <p>The screenshot shows the same property definition for 'isLocatedAt', but with specific domains and ranges defined. The 'Domains (intersection)' section contains the class 'Building OR ArchitecturalElement', and the 'Ranges (intersection)' section contains the class 'Place'.</p>

4.2 Relationships among Pitfalls

During the analysis of the ontologies we have observed that the pitfalls can be related through different types of relationships. For instance, some pitfalls can be specific cases of a more general pitfall, or a pitfall can occur as a consequence of other pitfall that has taken place previously.

With regard to the pitfalls presented in Section 4.1, it should be noted that the “Inverse relationships that are not inverse” and “Not defined inverse relationships” pitfalls are inverse. That is, in the first case, not valid knowledge is represented, whereas in the second case knowledge is omitted although is valid. In addition, these two pitfalls can be a consequence of the “Relationships and/or attributes without domain and/or range” pitfall, since in the case of “Not defined inverse relationships” pitfall, two inverse relationships may not be defined as such because their ranges and domains have not been defined.

We think that the relationships between the pitfalls can be very useful during the development of the methodological guides to avoid pitfalls. Therefore, we are now analyzing the existence and types of relationships.

5 Conclusions and Future Work

This paper presents the analysis carried out on 11 ontologies with the aim of identifying a series of patterns and anti-patterns. During the analysis we have also found a set of pitfalls that may appear during the ontology development.

The pitfalls identified have been classified according to a subset of the types of ODPs found in the literature with which they could be related. The paper also provides a template in which the pitfalls and some of their examples are described.

However, even though it is possible to find relationships among the different pitfalls, we have not studied all the possible relationships. Therefore, as a future line of work we propose the identification of the relationships between the different pitfalls. It would be interesting to identify the groups of pitfalls that usually appear simultaneously so that, once the concurrence of a pitfall is identified, the other pitfalls that could appear could also be identified.

We also propose to analyze the existence of pitfalls and their relationships in domains other than those represented in the ontologies studied in this paper.

Another interesting work could be to carry out the analysis described in this paper but modifying the ontology development process followed by the students. In this new process we propose to include notions about ODPs in order to compare the resulting ontologies with and without notions about ODPs and analyze to what extent the pitfalls do not appear.

Finally, as we have already mentioned, the pitfalls identified are not related to any ODP present in the current libraries; therefore, we are investigating the creation of new design patterns with the objective of avoiding the use of such pitfalls.

Acknowledgments. This work has been partially supported by the European Commission project NeOn (FP6-027595) and the Spanish project GeoBuddies (TSI2007-65677-C02). We are very grateful to Rosario Plaza for her English revisions and comments.

References

1. Aguado De Cea, G., Gómez-Pérez, A., Montiel-Ponsoda, E., and Suárez-Figueroa, M.C. *Natural Language-Based Approach for Helping in the Reuse of Ontology Design Patterns*. In *Proceedings of the 16th International Conference on Knowledge Engineering (EKAW)*, pp. 32-47. (2008)
2. Alexander, C. *The timeless way of building*. Oxford University Press, New York (1979)
3. Blomqvist, E.; Gangemi, A.; Presutti, V. *Experiments on Pattern-based Ontology Design*. In *Proceedings of the Fifth International Conference on Knowledge Capture (K-CAP)*, pp. 41-48. (2009)
4. Gangemi, A. *Ontology Design Patterns for Semantic Web Content*. Musen *et al.* (eds.): *Proceedings of the Fourth International Semantic Web Conference (ISWC 2005)*, Galway, Ireland. LNCS, vol. 3729, pp. 262–276. Springer, Heidelberg (2005)
5. Gangemi, A.; Presutti, V. *Ontology Design Patterns*. *Handbook on Ontologies* (Second Edition). S. Staab and R. Studer Editors. Springer. *International Handbooks on Information Systems*. (2009)
6. Presutti, V.; Gangemi, A.; David S.; Aguado de Cea, G.; Suárez-Figueroa, M.C.; 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*. <http://www.neon-project.org>. (2008)
7. Suárez-Figueroa, M.C.; Brockmans, S.; Gangemi, A.; Gómez-Pérez, A.; Lehmann, J.; Lewen, H.; Presutti, V.; Sabou, M. *NeOn D5.1.1: NeOn Modelling Components*. *NeOn project*. <http://www.neon-project.org>. (2007)