1 Introduction

Requirements engineering plays a decisive role in the software industry. The main difficulty in applying reuse to this area is that requirements are generally expressed in natural language. For that reason, automatic analysis and reuse processes will need textual treatment techniques and natural language processing.

Some researchers like Barry Boehm [3][4] show that the strategy of software artifacts reuse involve a reduction of 47% of the invested effort. The benefits of reuse are bigger as far as the level of abstraction increase and not only code is reused, but also designs and specifications, because these are guides for the next stages of development [23].

In a specific domain, requirements about restrictions, organization data and policies are potentially reusable, and may represent more than 50% of the requirements in a new project [14]. This kind of reuse has quantitative benefits: if we start from the point where a set of requirements have been specified for other projects, then the time of producing these specifications may be reduced [21]. Moreover, we can reuse other artifacts linked to these requirements, accelerating in consequence the process in later stages of the software development. Therefore, the productivity may be optimized because of the improvement of development times (Time to Market).
On the other hand, reuse could provide us with qualitative enhancement too, because previous used requirements are supposed to have passed quality filters. In addition, the information of previous projects related to reused requirements (e.g. risks, given solutions, problems, etc.) could offer knowledge that brings improvements in control and planning activities, so we could reduce potential deviations on later developments.

Our proposal deals with technology and tools for requirements reuse, and it also considers the process that must be applied. However, the core of the proposal is the creation of an ontology that holds the Requirement Engineering Domain Knowledge. The remainder of this article is structured as follows: in section 2 some adopted ideas in the requirements reuse shows the problems to deal with. Section 3 shows the proposal and advantages, and Section 4 the methodology used to develop the new ontology. Results are shown in section 5. And finally, in Section 6 and 7 conclusions and future works ends the paper.

2 Requirement Reuse Approach

The methods and techniques developed in the Software Engineering field in order to reuse requirements are diverse. Some of the most used are requirement patterns [6], parameterization [17] and domain models [19][20]. These approaches are based on previous classification and modeling of potential artifacts that could be useful in the reuse process. These techniques are effective if previous and new projects are similar, but they offer low flexibility in case of changes; this inconvenience is often called Oracle Hypothesis [7]. Besides, in domain modeling, the high level of abstraction and semantic content of requirements specifications makes very difficult to specify modeling languages. For example, the UML modeling language provides use cases diagrams in order to specify requirements, but this kind of models are only useful for functional requirements. Furthermore, the use of these languages implies an additional need of training in the client side, which is against one of the most important advantages of requirements as contracts that are easy to understand by the client.

Other approaches confer more value to the retrieval process (such as identification, evaluation and search of software assets [2]), than to the characterization of artifacts to retrieve. They use searching and retrieval techniques, but also require pre-modeled knowledge, sometimes combined with Natural Language Processing (NLP). With these approaches, the flexibility of the pre-modeling knowledge systems is increased, although the precision rate in the search is usually reduced. Approaches like the Domain Mapping Approach [5] use NLP techniques, as well as knowledge of specific domains represented as a thesaurus. The main inconvenience, notified by the own authors, is the need of a manual classification of terms in facets for each requirement, and the problems shown in [18] related to the faceted classification.

Other approaches use ontologies with the aim of improve the searching queries. In reuse, they have been used for mapping XML schemas and software artifacts structure [10]. Another proposals use ontologies just to facilitate the manual specification of requirements and to remove ambiguities [25].
The analysis of the diverse approaches shows that they share common needs and have not solved yet the main problems. The analyzed proposals require:

- Domain Modeling
- Requirements Modeling
- Flexibility in the modeling updates
- NLP systems for the analysis and retrieval of similar requirements
- Requirement management tools

Until now, integrated proposals that can deal with all the problems have not been found. However it is needed to solve the problems in order to achieve an effective reuse system that could be considered by enterprise environments.

3 Requirements Engineering Ontology

The domain modeling could be done by using an ontology. The use of ontologies as knowledge models brings enormous benefits in front of some other approximations because they can be used to model any domain, they can be understood by humans and developers because of the intelligible representations, and it is possible the use of formal languages that makes feasible the interoperability between tools. As counterpart, ontology creation is translated into a manual task, slow and with a high cost [1][8]. It means big efforts for its construction before setting up the reuse system, with an initial investment but unknown expectations of benefits at medium-long time.

This work offers a whole reuse system where the knowledge core will be an ontology for requirements reuse. This ontology will contain common knowledge of Requirements Engineering but not of any other particular domain knowledge related to the software specified by the requirements. The ontology will be called Requirements Engineering Ontology (REOntology) and will work as a knowledge base for the textual process in a reuse system. The proposed reuse system is structured in three modules: requirements description, indexing and retrieval (Fig. 1).

Requirements Description: the user must introduce the textual requirements into the system. Most tools allow introducing and storing the requirements concerned with each project, and also the edition, classification and in short, the management of them. Some additional tasks, as the existence of patterns to introduce them, or the links with other related artifacts, could help in the indexing and retrieval activities and therefore in the reuse process. We will just use a tool to introduce the textual descriptions of the requirements into the system and relate them with the projects they belong to.

Indexing: The later analysis of requirements is done by an NLP system for the Spanish language. The most important characteristic of the NLP system is the identification and normalization of nominal and verbal linguistic units (simple and complex ones). The semantic content of each requirement is captured by means of the generic knowledge about the Spanish language (grammatical and morphological rules), and especially by means of the specific knowledge of the Requirements

---

1 The tools used in this work have been developed by The Reuse Company (www.reusecompany.com)
Engineering Domain. This specific knowledge is included in the REOntology and it is articulated in four points:

- Controlled vocabulary of the Requirements Engineering Domain
- Types of relationships
- Explicit relationships between the controlled vocabulary
- Lexical-syntactic patterns

We can use controlled vocabulary and their relationships to extract the meaning of the requirements. This way, we have important static knowledge that helps the identification of important concepts and so, the later search process of any requirement artifact. But, what about the relationships we can find dynamically in a requirement? How can we capture and model this knowledge? The use of lexical-syntactic patterns like the proposed by [11][12][13] plays an important role in the process, as let us capture this knowledge in a flexible way: the related verbal units are associated to a kind of phrase pattern and some grammatical restrictions are set in the category of the participants elements of the relationship. Besides, some flexibility is present in the order of appearance of elements in the phrase.

At last, the indexing process captures vocabulary and relationships in an automatic way, and as result the knowledge of each requirement is modeled conformed with a metamodel called RSHP [16], which makes possible the representation of concepts, their relationships and its types.

Retrieval: Once the requirements have been indexed and stored as RSHP models, we need a retrieval tool for finding reusable requirements. The retrieval tool finds similar requirements, considering as similar those requirements that have equivalent concepts and relationships. Depending on the number of equivalent elements and the semantic distance between these elements, the requirements are shown in relevance order by similarity. Moreover, the retrieval tool we used has an additional task that let us search among the concepts and types of relationships stored, and retrieve those requirements which contain them.

In short, in the system described the REOntology is integrated as support of additional knowledge for requirements retrieval: it expands the queries thanks to the relationships between terms in the ontology, and it includes lexical-syntactical
patterns as meta-knowledge that makes possible the automatic identification of relationships in a sentence.

Besides, this approach can be enhanced by means of a continuous improvement in the stored knowledge. The REOntology might become a support for the extension to specific domains, what would be easier than starting from scratch an ontology of each domain independently. Specifically, the REOntology can be expanded thanks to the use of the Incremental Reuse Method (IRM) [15], so the reuse would turn into an incremental process (Fig. 2). This process is possible thanks to the use of the same metamodel for the requirements modeling as well as for the REOntology modeling, and thanks to the knowledge of the REOntology itself, that aids semi-automatic extraction of new knowledge coming from the new requirements incorporated to the reuse system (an exhaustive study of these techniques could be studied in [24][22]).

![Fig. 2. Incremental Reuse Method](image)

With this approach we get a base ontology which is useful in different specific domains of each organization. Moreover, the REOntology makes possible the creation of a common structure between these domains, working as a requirements top ontology and giving support to the interoperability between them. The advantages of it are not only in the maintenance of the knowledge, but also in the development of applications in diverse domains or even in the modeling of domains with fuzzy boundaries.

4 Methodology for the development of the Requirements Engineering Ontology

The construction of the REOntology is based on two points: concepts and relationships. In order to extract and identify this knowledge, a significant corpus of the Requirements Engineering field must be analyzed. In general terms, the main steps followed are:

1. Retrieval of a textual corpus of requirements specifications
2. Semi-automatic analysis of the corpus for the extraction of terms
3. Semi-automatic analysis of the corpus for the extraction of relationships
4.1 Retrieval of a textual corpus of requirements specifications

In the ontology development, it has been used a plain text corpus of approximately 500 requirements with different lengths, which takes up around 0.6 Mb. It includes requirements specifications of companies of diverse areas: Telecommunication, Finance, and Consulting. The criteria for the selection of the corpus are:

- Written in Spanish, avoiding orthographic errors and typos.
- From companies of different business sectors.
- Real documents containing relevant requirements (atomic and not atomic requirements).

The criteria aided the authors in the identification of the knowledge shared by the three business sectors, distinguishing it from the knowledge related to the own business area of each company.

4.2 Semi-automatic analysis of the corpus for the extraction of terms

A set of the most used terms in the Requirements Engineering domain has been identified as a result of applying filtering strategies for stop words, normalization processes and frequencies analysis of the terms in the corpus. Syntax analyses were done so only nominal, verbal and adjectival units were considered as candidates to be part of the REOntology.

It was also noticed that in the requirements domain the actions are very important, don’t mind if they are represented by substantives or verbal forms. In general, this schema is valid for the case of functional requirements. As a consequence, we must identify this kind of terms and establish relationships between the different grammatical categories. It was possible to achieve in an automatic mode taking as referent the root of the terms by stemming techniques.

4.3 Semi-automatic analysis of the corpus for the extraction of relationships

In the REOntology two kinds of relationships were modeled: paradigmatic and phrasal ones. Paradigmatic relationships are those that come from the meaning of the words, so they are used to link a set of words related semantically (e.g. synonymies, hierarchy, association, etc.). On the other hand, phrasal relationships are those that come from the structure of a sentence. The first kind of relationship allows query expansions through concepts related to the searched terms. The second kind of relationship allows concepts to be linked dynamically by matching lexical-syntactic patterns from the text (Fig. 3). So they make possible an automatic catch of relationships between concepts as well as the identification of the semantics associated to the relationship (in fact, the use of this kind of techniques could lead to the identification of atomic requirements inside non atomic ones).

---

2 Atomic requirements are the minimal well defined requirement, and non atomic requirements are a composition of them.
In our approach, phrasal and paradigmatic relationships were identified by the analysis of the resulting relationships after the indexing process of the corpus. These relationships respond to generic patterns, that are capable to catch valid semantic relationships in any domain (e.g. A is equal to B: identity relationship between A and B) or linguistic patterns that could fetch co-occurrences between terms (e.g. relationship between subject and direct object in a sentence). It must be noticed also that the process of identify relationships helped in the identification of useful vocabulary that could not be identified in preliminary stages.

![Fig. 3](image-url) Syntactic pattern with the associated semantic dar de baja (remove). NS is a nominal phrase and PS is a prepositional phrase. In this example, just the verb (remove) and NS2 (the identifiers) are mandatory.

### 5 Results

As result, the REOntology contains 493 terms, which are structured in groups of 7 families, with 1’25 average of relationships by term and 0’37 synonyms in average per term. The paradigmatic relationship types that are part of the ontology are hierarchy, synonymy and association. Regarding phrasal relationships, the ontology has 90 lexical-syntactic patterns, grouped in 28 semantics.

An example of the utility of the REOntology in the retrieval process is shown in Table 1 and 2. Both of them shows the same query in the same retrieval system described, but Table 2 shows the results obtained with the use of the REOntology, and Table 1 shows the results without this knowledge support.

Table 1 shows the gestor requirement as the result of the search. It contains the terms gestor (manager) and usuarios (users) interrelated by a semantic relationship administrar (to manage). The relevant terms, highlighted in cursive style, are the same terms that the ones found in the query, but it is not the case of the semantic relationship between them. Thanks to the use of the REOntology, the system is able to identify two additional requirements which are even more relevant. These requirements, baja de usuarios and usuarios, are in fact more relevant because they relate equivalent terms (gestor -manager- is synonym of administrador -administrator-), and the action eliminar (delete) and dar de baja (remove) are both part of the same semantic dar de baja (remove). In order to detect this similarity, it is needed to take care of lexical-syntactical patterns and its associated semantics, as well as term expansion thanks to the paradigmatic relationships. All these semantic knowledge were found thanks to the use of the REOntology.
With the aim of measuring the results of the ontology developed, a set of queries has been designed, and a set of requirements specifications has been stored for testing (the test application is available at http://www.reusecompany.com/demo.aspx?id=4). For that purpose 23 real requirements of diverse companies were considered, with different subjects and extensions, and 13 queries were made. In order to measure how effective is the use of the REOntology in the proposed reuse system, precision and recall rates were analyzed for the queries with and without the use of this ontology. The experiments could be repeated by anyone downloading the application from the site mentioned above.

The resulting values (Fig. 4) show that without the REOntology it is achieved 72% of precision and 70% of recall. It means that 72% of the retrieved requirements are relevant, and 70% of relevant requirements are retrieved. If the REOntology is used, the values for precision and recall grow to reach 78% and 100% correspondingly. Even the precision rates improves, it is remarkable how recall improves. It means that the use of REOntology aids mainly in the retrieval of relevant requirements which could not be retrieved without the use of the ontology.

In addition, one of the more relevant factors for a retrieval system is the order of the results. In case of having a results set with many similar requirements, how they are ordered by the system is essential for the user to detect the most relevant ones. In our previous test set, if an analysis is done over the most relevant requirements of each query, the results show that 92% of these requirements are ranked first when we use the REOntology, while only 51% of these requirements are ranked first when the REOntology is not used.

3 The queries were done in Spanish. The text in English and Spanish remains in the tables to make it more clear.
An undersized analysis of data shows that the use of the REOntology improves the retrieval mainly in two ways:

- Allows locating requirements impossible to be found without the use of the ontology.
- Assigns more accurate levels of relevance to the resulting requirements, avoiding them to be camouflaged between those that are not really so interesting for the query.

In short, we can observe that the REOntology facilitates the search and retrieve processes, improving precision, recall and improving also the order of the resulting requirements of a query. However, it is important to say that these tests are not exhaustive enough to demonstrate the goodness of the ontology made. Our intention is just to show the usefulness and feasibility of the retrieval system proposed. As a future work, a more representative test set will be defined and implemented.

6 Conclusions

The applications of natural processing language and ontologies are diverse in the Software Engineering field. In this work, the benefits offered to Requirements Engineering area from the reuse viewpoint have been shown.

The requirement specifications contemplate a common knowledge, independently of the application domain of the software they belong. This knowledge, modeled in an ontology called Requirements Engineering Ontology (REOntology), and integrated in a complete retrieval system, brings the support needed for an incremental reuse process. An ontology with these characteristics acts like a knowledge base for retrieving requirements that are potentially reusable, and moreover it helps in the progressive and semi-automatic development of specific ontologies for improving the retrieval results. The advantages increase as the REOntology could be used in any organization dedicated to software development and because of the interoperability offered between diverse specific domains.

The different approaches for integrating Software Engineering and Knowledge Engineering tend to be academic, stepping aside some aspects like applied research...
[99]. The approach showed in this article pretends to overcome this tendency including this development in a whole reuse methodology, and in particular with the support of tools for professional development and software reuse.

At present, the reuse system described is being used in the company SAGE SP (http://www.sagesp.com/web/web/index.asp). Adequate parameters must be determined in order to realize metrics that support that this reuse system makes companies to save money, and it is proposed as a future work.

References