UNIVERSIDADE NOVA DE LISBOA

Faculdade de Ciências e Tecnologia

Departamento de Engenharia Electrotécnica e de Computadores

Methodology for Collaborative Enterprise Reference Ontology Building

By

Tiago Manuel dos Santos Gaspar

MSc. Dissertation presented at Faculdade de Ciências e Tecnologia of Universidade Nova de Lisboa to obtain the Master degree in Electrical Engineering and Computer Science, held under the guidance of

Ricardo Luís Rosa Jardim Gonçalves

João Filipe dos Santos Sarraipa

Lisboa

February 2011

Methodology for Collaborative Enterprise Reference Ontology Building

Aos meus pais,

à minha irmã.

ABSTRACT

In the actual competitive world, doing business globally has become critical to the survival of most enterprises. It is becoming each day more and more difficult for small enterprises to grow by operating alone in the market. Hence, most companies started feeling the need for joining collaborative environments becoming easier to manage their products and services, and where they can offer better products with low production costs. To achieve this, enterprises require the establishment of cooperation agreements among each other with the idea of expanding their business networks. Consequently there is a demand for intelligent solutions capable of reinforcing partnerships and collaborations between enterprises, organised groups or singular people. However, due to the worldwide diversity of communities, a high number of knowledge representation elements, such as ontologies, which are not semantically coincident, have appeared representing the same segment of reality. Even operating in the same domain, enterprises do not understand each other, making the communication among various systems parties more difficult and sometimes impracticable. This dissertation responds to the needs identified above, proposing a collaborative methodology for ontology building, enriched with qualitative information collection methods, to effectively improve the approach to elicit knowledge from business domain experts, towards interoperable intelligent systems. This methodology allows different individuals from enterprises or organisations working on the same field or area, to join a collaborative environment for building a common ontology specific to their "Domain of Discourse". To accomplish this, several steps are taken including terms and definitions gathering, glossary and thesaurus building, and ontology mappings.

KEYWORDS

The keywords of this dissertation are: *Knowledge Representation*, *Ontology*, *Collaborative Ontology Building*, *and Interoperability*.

RESUMO

No tempo actual em que nos encontramos, a globalização do negócio tornou-se uma questão chave para a sobrevivência de um grande número de empresas. Cada vez é mais difícil para as pequenas empresas se desenvolverem e crescerem se continuarem a caminhar sozinhas a procura de mercado para os seus produtos. Tendo em conta isto, a maior parte das empresas começaram a sentir a necessidade de colaborar umas com as outras de forma a aumentar a competitividade dos seus produtos assim como reduzir os custos dos mesmos. Consequentemente há uma exigência por soluções capazes de aumentar significativamente as relações e a colaboração entre empresas, organizações ou simplesmente pessoas. Contudo, devido a grande diversidade de comunidades a nível mundial, um largo numero de elementos de representação de conhecimento, tais como ontologias, apareceu representando uma mesma área de interesse, não sendo elas semanticamente iguais. Isto leva-nos a entender que as empresas, mesmo a trabalhar nas mesmas áreas, não se entendem a 100% quando trocam informações, sendo mesmo por vezes impossível de haver comunicação entre elas. Esta dissertação responde às necessidades anteriormente apresentadas, propondo uma metodologia colaborativa para a construção de ontologias enriquecida com os chamados *Qualitative Information Collection Methods*. Esta metodologia permite que diferentes pessoas, representando empresas ou organizações, que trabalham no mesmo sector possam interagir num ambiente colaborativo para assim construírem uma ontologia de referencia ao seu sector. Para que isto seja alcançado são seguidos vários passos para a construção desta ontologia. Estes passos envolvem a recolha de termos e definições, construção de um glossário, de um thesaurus, e mapeamentos entre ontologias proprietárias e a de referência.

PALAVRAS-CHAVE

As palavras-chave desta dissertação são: Representação de Conhecimento, Ontologia, Construção Colaborativa de Ontologias e Interoperabilidade.

ACKNOWLEDGEMENTS

First of all, I would like to thank all the people who in some way helped me during my studies and later during the realisation of this dissertation.

To my advisor Professor Ricardo Gonçalves for giving me the opportunity to work with him, with his research group, and for giving me the honour of his advices.

To my research group friends, João Sarraipa, Carlos Agostinho, Fernando Ferreira, Maria João and Alexander Todorov for being here every day, helping me with my research, and giving me their expertise for making me grow as a person and student. It was such a great year I spent here with all of you. A special thanks to João Sarraipa for the long talks and for the time you spent with me.

To Mauro Onori, even not contributing directly for this dissertation he was a great teacher and a good friend during the time I lived in Sweden. Everything you said to me helped me a lot for finishing my Master degree.

To my friends Pedro Arruda, David Gonçalves, Filipe Correia, Bruno Alves, Diogo Figueiredo, Fábio Coelho for being here every day and every night helping me always with each obstacle that appeared along the way. The days we spent together in the University were the best days of my life.

To Tiago Ribeiro, a very special friend who was always there for me no matter what. In the good, the bad, or even the worst moments of my life you pushed me to go further, telling me never to give up. There's no one like you!

And finally to the ones who always believed and will always believe in me, my parents, and my sister. You supported me with everything you could, trying to give me the best future. You mean a lot to me and I will never forget what you have done for me. "Para aqueles que sempre acreditaram em mim e irão sempre acreditar, os meus pais e a minha irmã. Vocês estiveram sempre presentes e sempre dispostos a ajudar, dando-me tudo o que podiam para que eu alcançasse o melhor para a minha vida. Vocês são TUDO para mim e nunca esquecerei tudo o que vocês fizeram por mim".

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1. INTRODUCTION

In a constantly changing market, enterprises need to follow the trends and always offer new and innovative products and services. This is a demand for survival. Enterprises started to realise it is important to stop “playing the game” alone and start cooperating with other enterprises to be able to respond to customers’ needs. Collaboration also helps small enterprises to get together and offer products that in multiple cases are making face to products of bigger companies [1].

However, cooperation always brings interoperability problems to enterprises that are trying to combine their knowledge to reach a same goal. These problems are mostly related to the lack of interoperability systems and software applications that manage and progress in their collaborative business [2][3]. To have success in complex environments, enterprises need to be fully interoperable as well as their systems. This means they need to share their competences and information within each member of the network and this information has to be completely understood by all intervenients whether this information comes by a phone call whether it is switched by different software applications. It is then reasonable to say that interoperability between parties in the same domain is still a challenge in progress, but difficult to achieve due to systems using different models and semantics [4].

This leads us to the appearance of semantic interoperability problems between enterprise communications and to find ways for solving this dilemma.

Caused by a big growth of communities worldwide, a high number of ontologies belonging to a same domain but which are not semantically coincident have appeared representing the same segment of reality. Ontologies are a formal and explicit way of describing a domain’s knowledge [5]. Everyone can easily create an ontology for an area of its interest representing its knowledge about that field. It is simple to understand that a large number of ontologies will appear but rarely will we find two completely coincident. This means that we will have lots of different views of the same thing. This can be caused by cultural, chronological, or even educational differences between the people.

An idea that comes from the stated problem is to get all proprietary ontologies of a same domain an together create an unique ontology that describes completely all the ones involved in the process. This way everyone will be satisfied.

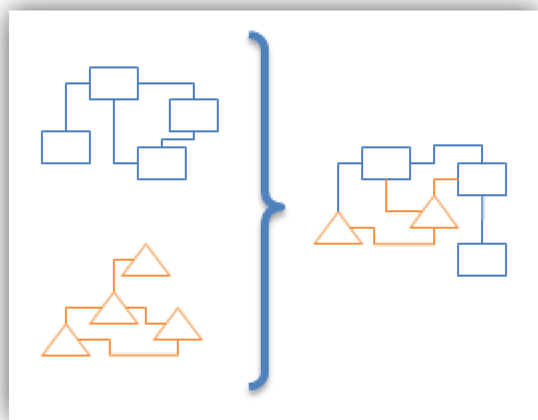


Fig. 1.1: Example of a creation of a unique ontology from existing ones

In Fig. 1.1 it is shown better what was explained in the latest. Of course this is not an easy task and many problems and difficulties may advert from it.

1.1. Background Observation

Since such community reference ontology has to represent all the involved enterprises, the more collaborative this process could be, the easier the community reaches a wider representative ontology. To establish a common understanding in a domain all stakeholders must participate in the construction of the reference ontology.

There are several different methods for creating ontologies depending only on the way the involved organisations want to do it. They can be built from scratch, reengineering, cooperatively, or by merging two or more into a single one.

Since the full automatic machine knowledge acquisition remains in the distant future, human intervention is needed in a reference ontology building process. Thus, the collaborative ontology building process is accomplished by human discussions, which in this case is focused in semantic and knowledge representation agreement.

Based on the study done by the author which is presented in the literature review this dissertation proposes an adaptation of a set of Qualitative Information Collection Methods (QICM) in a collaborative ontology building process, in this case MENTOR methodology, to effectively improve the approach to elicit knowledge from domain experts, towards interoperable intelligent systems.

1.2. Motivation

The main motivation of this dissertation work is the fact that semantic interoperability between enterprises is an actual research challenge. In fact, there is a research community that defined a roadmap on the Enterprise Interoperability (EI) field. Its purpose was to evaluate the EI developments on the past few years, by the time the roadmap was made (2008), and to interpret their impact in order to better specify the EI long-term research goals [6]. Such study envisaged a vision that pursues what was stated in the i2010 communication where its research key is supporting new patterns of business that improve enterprises innovation, and their adaptation to new skill needs [7]. Enterprises also have to remain the main benefactors of such research. This roadmap identified four Grand Challenges (GC) that represent the four global domains for research to reach the overall vision, which are the following:

1. **Interoperability Service Utility**: it extends the previous version by focusing on the key services for the industry, including SME-specific needs and requirements;

2. **Future Internet and Enterprise Systems**: extending the previous version of this GC – Web Technologies for Enterprise Interoperability – it sets the additional target of using and extending Enterprise Systems which are completely open, adaptive and integrated with processes for innovation;

3. **Knowledge-Oriented Collaboration and Semantic Interoperability**: its idea is to carry all key topics forward, paying attention to the semantic needs for organising, managing and exchanging knowledge and information – of both incoming and outgoing nature in the modern SME;

4. **Enterprise Interoperability Science Base**: on this GC the original idea remained the same – empowerment of scientific foundation of EI, new challenges are put forward taken in consideration the previous two years of discussions between the community members.

Therefore, this dissertation intends to contribute to solve semantic interoperability problems by proposing a methodology for Enterprise Reference Ontology Building in a collaborative environment.

1.3. Research Method

The research method took on consideration on this dissertation was based on the classical research method. This method is composed by seven steps, ordered from a more theoretical view to a more practical view of the system. It begins by the study of the problem and definition the area of research and ends up with the proof-of-concept, i.e. the tests and analysis made to its results. This is an iterative methodology. It means that if the results are not what the researcher was expecting for, it is reasonable to go back to the first steps and try a new approach. In Fig. 1.2 are described well the seven different steps imposed by this research method.

Fig. 1.2**:** Phases of the Classical Research Method [8]

A short description of each phase of Fig. 1.2 is given bellow:

1. **Research Question / Problem:** the research question is one of the most important steps for researchers. It is a period of study that intends to define the area of interest of the research. The research question must be clearly defined, making the study practicable of being confirmed or not. Generally a research question never comes alone. It usually comes together with several minor questions to refine the main idea of the research subject. This is presented on section 1.4 – Research Questions.

2. **Background / Observation:** the study of the work done before about the same research area is based in this step. The main ideas previously presented by other

authors are shown here and it takes the readers to the start point of the dissertation. It is also shown what differs from the previous work to the one being developed for this dissertation, as well as the methodologies taken when approaching the solution. The state of the art is at the same time, an important study to be made, reviewing literature and scientific projects bringing up the ideas of was already tested and accomplished. It is important to have a big variety of documents for searching information on the area of interest, since there exists some literature very reliable but does not bring innovation due to the time it exists and on the other hand, some documentation with high level of innovation and newness but not low reliability.

The Background observation is comprehended in sections 2, 3.7, and in the beginning of section 4.5.

3. **Formulate Hypothesis:** is to manage the predictions for the results of the research work. It usually goes straight to what is expected for the project results. Its idea is to make the research problem simpler to understand, stating the ambitions to accomplish at the end of the project. The hypotheses for this research work are presented in section 1.5 of this document.

4. **Design Experiment:** this step works as a preparation for the experimental phase, where the solution design is seen as the system architecture. In addition, it is significant to find a validation plan for the previous step, i.e. the hypothesis. In MENTOR using Qualitative Information Collection Methods (section 4) is done a theoretical design of the ideas for solving the identified research problems and Proof-of-Concept Implementation (section 4.5) is seen as a proof-of-concept for the proposed methodology.

5. **Test Hypothesis:** to test the hypothesis it is needed to get the results from system architecture and evaluate them. A big amount of tests (especially in different scenarios) should be done in order to test effusively the outcomes given by the system. These outcomes are supposed to be collected for later analysis. The section defined as Demonstrator Testing and Hypothesis Validation (section 5) presents a single test made by the author, applying all the necessary steps for building a collaborative domain reference ontology with the intention of evaluating the proof-of-concept addressed in prior sections.

6. **Interpret / Analyse Results:** after the battery of tests have been made to the system it is the time to evaluate and analyse the achieve results. It is, at this point that it is found out the veracity and confidence of the hypothesis placed on the beginning of this journey. Everything is possible to happen, the results can be satisfactory, proving the author was right, or just fail completely the initial idea. If the results point straight to the Hypothesis, then it is reasonable to say that a good prevision was made and it is possible to consider what comes after, making some recommendations for further research. But even the results are not the expected it should not be taken as a failure, but as an opportunity to improve the original approach and go back again to the first steps of Phases of the Classical Research Method, as shown in Fig. 1.2, try a different approach from the taken before. Section stated as Demonstrator Testing and Hypothesis Validation makes the interpretation and analysis of the results of the presented methodology for collaborative ontology building.

7. **Publish Findings:** the final results, if consistent, must end up in valuable contribution to the scientific community as scientific papers. These papers can be then presented in conferences, where the author as the chance to show in person his ideas for the research, presenting the results and answer to questions of other to prove the efficiency of the results.

1.4. Research Questions and Problems

- Is it possible to design a methodology supported by suitable tools able to create a common Knowledge Base to a group of enterprises?
    - How to improve the process of getting the tacit knowledge from the business domain experts?
- Is a reference ontology, relating to a community of enterprises, able to contribute for solving the semantic interoperability difficulty between enterprises communications?

1.5. Hypothesis

- With the development of a collaborative engine allowing several users in representation of enterprises to, together, build a reference ontology related to their domain of interest, it will be possible to reduce semantic interoperability problems;
  - Qualitative Information Collection Methods will improve the approach taken in group discussions to get knowledge from business domain experts and create a reference knowledge base that suits every party involved.

1.6. Dissertation Outline

The first section of this dissertation, the *Introduction*, states the main ideas that conducted to the study for this research project. According to what was done prior to this project work some new ideas and solutions are thought in being tested as a way of giving another step in the right direction for solving the research problem. It also manages the expectations on the chosen approach when it comes to analyse the results.

The next sections, *Knowledge Representation* and *Methods and Methodologies for Ontology Building*, are the topics that talk about the background observation work. In these section state what was done previously to this dissertation study. The *Knowledge Representation* section covers several topics with high significance for this dissertation, covering the main ideas of knowledge and knowledge representation, the challenge for semantic interoperability and ontologies as a key for giving a common language of a domain of interest of a group of enterprises. The *Methods and Methodologies for Ontology Building* section presents several methodologies for building ontologies as well as the different approaches to organise and facilitate group meetings, i.e. the Qualitative Information Collection Methods. This last subsection shows four different approaches, each one with its advantages and drawbacks, that will be used later to improve MENTOR steps.

After comes the *MENTOR using Qualitative Information Collection Methods* section. In this section an adaptation to MENTOR is made after an exhaustive study on QICM. On each step where group discussions are performed, it was decided to apply one from the four QICM presented in the previous section, making these steps easier and fast to execute. *Proof-of-Concept Implementation* subsection, states all the important points that contributed for the practical implementation of the presented methodology enriched with the QICM. It starts by showing the *Used Technologies*, explaining in detail each technology used and why it was chosen, then presents the *Architecture* of the whole system, for implementing the described methodology and in the end the *Detailed Process* is shown as a complementation of the architecture but showing in detail the flow of the system.

The step referred as *Demonstrator Testing and Hypothesis Validation* shows the results of the tests made during the execution of the methodology. It intends to prove that the *Hypothesis* really responds, or not, to the *Research Questions and Problems* made on the very beginning of the research work.

After all, comes the section of *Conclusions and Future Work* where, as stated by the section's caption, the conclusions and future work topics are presented.

2. KNOWLEDGE REPRESENTATION

Knowledge is defined by the Oxford English Dictionary as facts, information, and skills acquired by a person, through experience or education; the theoretical or practical understanding of a subject. It is the awareness or familiarity gained by experience of a fact or situation [9]. Knowledge is also used to mean the confident understanding of a subject with the ability to use it for a specific purpose if appropriate.

Knowledge is the appropriate collection of information, such that it's intent is to be useful. Knowledge is a deterministic process. When someone memorizes information, then they have amassed knowledge. This knowledge has useful meaning to them, but it does not provide for, in and of itself, an integration such as would infer further knowledge [10]. This is not the intention of knowledge. Knowledge pursues the gathering of new knowledge in a kind of not ending cycle. Knowledge acquisition is the action beyond such process. Its main objective is to transform tacit in explicit knowledge, and effectively to improve the approach to elicit knowledge from domain experts, towards interoperable intelligent systems [11].

Tacit knowledge is knowledge that people carry in their minds, which provides context for people, places, ideas, and experiences [12].

Explicit knowledge is knowledge that has been or can be articulated, codified, and stored in certain media [12].

Knowledge representation studies the formalisation of knowledge and its processing within machines. Techniques of automated reasoning allow a computer system to draw conclusions from knowledge represented in a machine-interpretable form.

A Knowledge Representation Element (KRE) makes the formal representation of knowledge in a specific domain become easier. Fig. 2.1 illustrates the KRE's that should be defined in the path to build a domain's knowledge base. It represents the distinct level of conceptualisation that each one has, showing an increase of its presence from Terminology to the Knowledge Base [13].

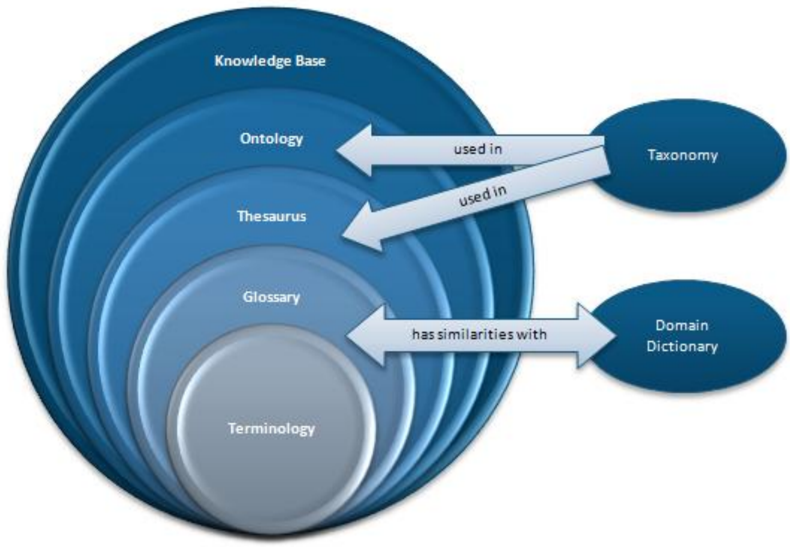


Fig. 2.1: Knowledge Representation Elements [14]

In the following, the knowledge representation elements shown in Fig. 2.1, i.e. Terminology, Domain Dictionary, Glossary, Taxonomy, Thesaurus, Ontology, and Knowledge Base, are discussed [14].

2.1. Terminology

Terminology is the study of terms and their use. It is connected to the creation, description, and naming of concepts in specific fields of knowledge as a key component to the generalized documentation process [15].

Terminology therefore denotes a more formal discipline which systematically studies the labelling or designating of concepts particular to one or more subject fields or domains of human activity, through research and analysis of terms in context, for the purpose of documenting and promoting correct usage. This can be restricted to one idiom or cover more than one at the same time (multilingual terminology, bilingual terminology, etc.) [16].

2.2. Domain Dictionary

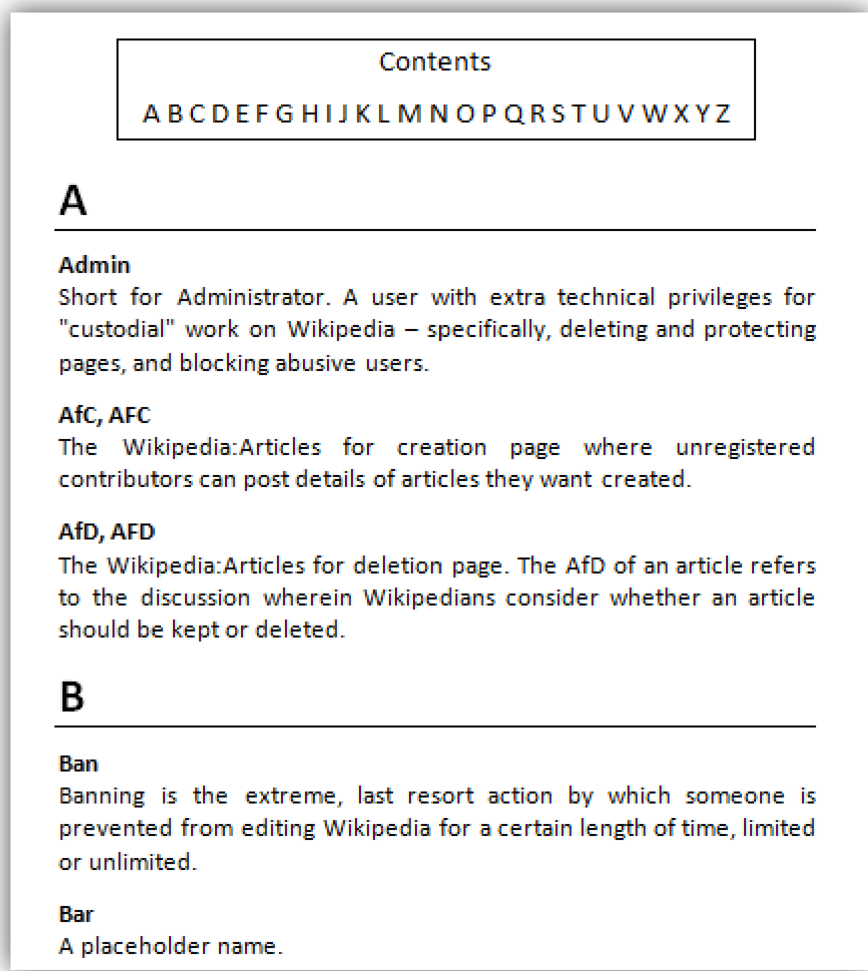
A dictionary is a book that lists the words of a specific language in alphabetical order, and gives their meanings and often other information such as pronunciations, grammatical forms and functions, etymologies, syntactic peculiarities, variant spellings, and antonyms, or that gives the equivalent words in one language – monolingual dictionary – or several different languages - multilingual dictionary [17].

A domain dictionary is considered to be a useful tool for a domain analysis where terms and their definitions are used in a very specific way within a domain [18][19].

2.3. Glossary

A glossary is a list – typically in alphabetical order – of specialized terms sometimes unique to a specific subject, and where each term is composed by a corresponding description, often placed at the back of a book. It includes descriptive comments and notes, such as definitions, synonyms, references, etc.

A glossary can be used when communicating information in order to unify knowledge sharing in such areas as Enterprise Modulation, Architectures, and Ontologies. A good example of a glossary is shown on Fig. 2.2 [15].


Contents

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

A

Admin
Short for Administrator. A user with extra technical privileges for "custodial" work on Wikipedia – specifically, deleting and protecting pages, and blocking abusive users.

AfC, AFC
The Wikipedia:Articles for creation page where unregistered contributors can post details of articles they want created.

AfD, AFD
The Wikipedia:Articles for deletion page. The AfD of an article refers to the discussion wherein Wikipedians consider whether an article should be kept or deleted.

B

Ban
Banning is the extreme, last resort action by which someone is prevented from editing Wikipedia for a certain length of time, limited or unlimited.

Bar
A placeholder name.

Fig. 2.2: Example of a glossary

2.4. Taxonomy

It has become fashionable to use the term *taxonomy* in a wider, more general sense, where it may refer to a classification of things or concepts, as well as to the principles underlying such a classification.

Taxonomy might also be a simple organisation of kinds of things into groups, or even an alphabetical list. However, the term vocabulary is more appropriate for such a list. In current usage within Knowledge Management, taxonomies are considered to contain less information than ontologies since these apply a larger variety of relation types [20].

Mathematically, a hierarchical taxonomy is a tree structure of classifications for a given set of objects. It is also named Containment hierarchy. At the top of this structure is a single classification, the root node that applies to all objects. It represents most general category of all things that the domain is related to. Nodes below this root are more specific classifications that apply to subsets of the total set of classified objects [21]. Each child is a subset of the parent. The intersection of each pair of children, in same level, is empty. Any path from the root to a leaf is called a branch.

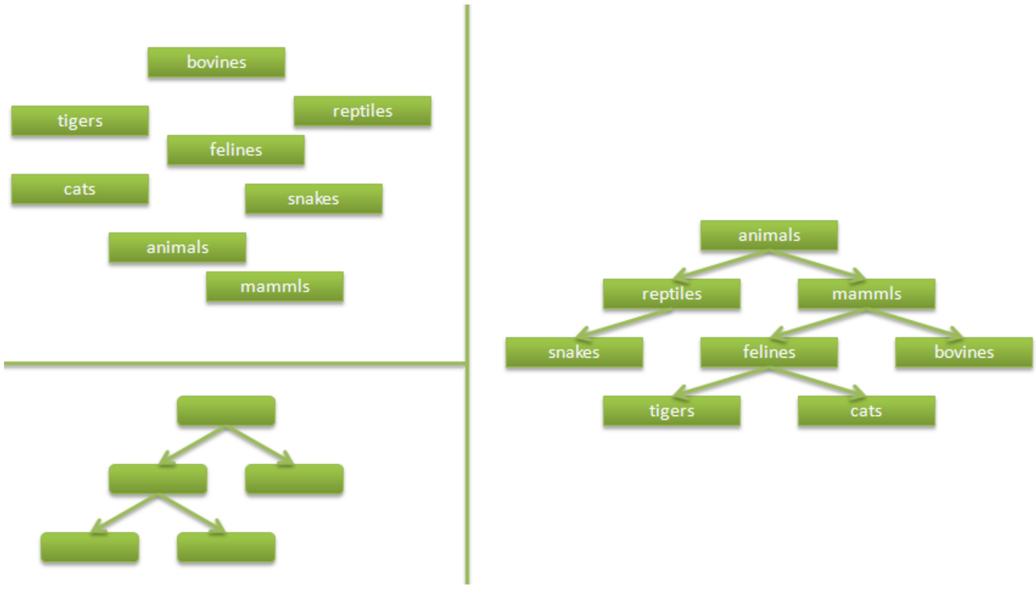


Fig. 2.3: Example of a taxonomy

Fig. 2.3 illustrates an example of a taxonomic structure. The right part of Fig. 2.3 represents the data of the upper left part organised under the structure of the bottom left part if the same figure. The process of organising terms into a taxonomy structure is simple .It starts by getting all terms of a domain. After that, begins a cycle for finding the terms that can aggregate the other ones. The most generic of all terms is chosen to be the root node. After finding the root node, the process is done again to find the next terms seen as the second most generic of the domain. This is an iterative process and is done until have all the terms associated to the overall tree structure.

2.5. Thesaurus

A thesaurus is a structure that manages the complexities of terminology and provides conceptual relationships, ideally through an embedded classification. A classification is a structure that organises concepts into a hierarchy, possibly in a scheme of facets. It can also be seen as a dictionary of synonyms and antonyms [22].

A thesaurus can both represent terms from one language – monolingual thesaurus – as from two or more languages at the same time – multilingual thesaurus.

The thesaurus can be then represented by a set of classes, representing domain reference concepts with associated meanings of a domain in a semantic related structure. Its main objective is to establish a formal lexicon of a specific domain.

A thesaurus is like a taxonomy of domain concepts composed by its reference meanings [23].

2.6. Ontology

An ontology is stated by Thomas Gruber as an explicit specification of a conceptualization. This term originally came from philosophy, where ontology is said to be a systematic account of existence [5].

Therefore, in the context of Artificial Intelligence ontology of a program can be described by defining a set of representational terms. In an ontology, definitions associate names of entities in the universe of discourse – classes, relations, functions, or other objects – with human-readable text describing the meaning of names, and formal axioms that constrain the interpretation and well-formed use of these terms. Formally ontologies are the statement of a logical theory [5].

Ontologies are used to describe concepts while the thesaurus aims to their description. An ontology can be seen as an enriched thesaurus where beyond the relationships and definitions between terms of a same domain it can be represented by a conceptual knowledge, composed by semantic relationships.

Lately it has been seen an explosion of interest in ontologies as artifacts to represent human knowledge and as critical components in knowledge management, the Semantic Web, business-to-business applications, and several other application areas. Various research communities commonly assume that ontologies are the appropriate modeling structure for representing knowledge. However, some discussions have occurred concerning the actual range of knowledge an ontology can successfully represent [24].

2.7. Knowledge Base

A knowledge base is an element of information storage and organisation. Information can be stored in structures such as ontologies, thesaurus, etc., depending on the purpose of such information. Ontologies can be seen as a *Knowledge Base*, and are usually used for describing the essential concepts of a specific domain.

A Knowledge Base also includes the essential information to model and develop a problem, obtain new knowledge, prove theorems, or answer intentional questions about a specific domain. It provides an efficient base for getting information online, allowing the conceptual continuity of learning and preventing the need for relearning from scratch [15].

Obviously not all knowledge can be fully represented in a Knowledge Base. It is not easy to represent certain types of knowledge as skills or distributed knowledge, or transform certain types of representation into ontology-appropriate formats as diagrammatic knowledge. Other types of knowledge are extremely suited to ontological representation, such as taxonomic information [24].

2.8. Semantic Interoperability

Semantic Interoperability is the ability of computer systems of sharing information and having that information properly interpreted by the receiving system in the same sense as intended by the transmitting system. It means that the transmitted information will be used appropriately by a receiving computer system because the logical implications derivable from transmitted information will be the same as those that the sending system would derive.

The interoperability idea presented on this dissertation mainly focused in the interoperation of systems. As a result some interoperability definitions commonly used by researchers are the following:

- It is stated as [25] ”the ability of two or more systems or components to exchange information and to use the information that has been exchanged”;

- Interoperability is also considered as significant if the interactions can take place at least on three different levels: *data*, *services* and *processes*, with a semantics defined in a given business context [26];

- It is also defined [27] as the ability of different types of computers, networks, operating systems, and applications to work together effectively, without prior communication, in order to exchange information in a useful and meaningful manner.

To reach proper semantic interoperability, interoperable systems require sophisticated structures like ontologies to capture a common semantics of the domain. Heterogeneities that have to be dealt in the context of interoperability are usually classified to be of syntactical, structural and semantic nature [28][29]. At the Syntactic level, it is found all forms of heterogeneity that depend on the choice of the formalism used for representing information. Therefore, in what follows, we will assume that different formats can be interoperated at a syntactic level, this is typically achieved through a translation function. At the structural level, we encounter all mismatches related to differences in the arrangement of concepts and their relationships. Finally, at the semantic level, we encounter discrepancies that have to do with the fact that the same real world is represented using different denotations or structures, and so can be interpreted differently [30].

When working in worldwide networked environments, collaboration with other enterprises and/or other people can bring the additional difficulty of natural language. In this case, when the exchanged information is described using native language it requires the other part to understand it. Therefore, the ontology system should be complemented with a multi-language dictionary, where a set of normalized tokens gives the reference to the meanings in many native languages [31].

Multiple organisations operating in the same business domain may have different views of the same “picture”, and they want to describe their knowledge in an electronic way it will probably lead to different domain ontologies. Consequently it might conduct to interoperability problems when systems intend to share information between each other. The need of having complete and fully integrated systems, that seamlessly communicate and understand each other, requires a complete meaning understanding of the data within the several domains involved, and it results in a harmonisation of a common ontology [31].

Ontologies’ harmonization is not an easy task, even when following an established methodology, as it is the case. In particular the harmonization of taxonomies typically conducts to long discussions before getting to an agreement, forcing on several iterations [31].

Common ontologies are used to describe ontological commitments for a set of agents so then they can communicate about a domain of discourse without necessarily operating on a globally shared theory. It is said that an agent “commits” to an ontology if its observable actions are consistent with the definitions in the ontology. Knowledge is attributed these agents by observing their actions; an agent “knows” something if it acts “as if” it had the information and is acting rationally to achieve its goals [5].

A common ontology defines the vocabulary in which queries and assertions are exchanged among agents. Ontological commitments are agreements to use the shared vocabulary in a coherent and consistent way [5].

Forcing stakeholders to adopt a same ontology even if based on standards, it does not work in most of the cases. As a result, a good solution would be to keep the terminologies and classifications used by each one, and use a reference ontology for being the mediator in the communications between them. Additionally, the introduction of a new reference ontology would enrich the community and each enterprise should feel more motivated to be part of the group, with the possibility to keep their own definitions and being offered their own ontologies [32].

In order to support ontology interoperability, it becomes obvious that ontologies and semantic mismatches need to be overcome. Interoperability of ontologies and the approaches to solve it remain a core question, and the interoperation process cannot rely on manual input due to the complexity, size and number of ontologies being developed [33]. It is thus clear that there is a need for automatic ways of interoperating ontologies in order to relieve the inconveniences of manually creating and maintaining ontology mappings.

Three ways in which heterogeneous ontologies can be made interoperable have been recognised and they are identified as:

- Building inclusion relations between ontologies;
- Building mapping relations between ontologies;
- Building a common ontology from local ontologies [34].

Out of those three ways to enable the interoperability of heterogeneous ontologies, the most effective method for solving ontology heterogeneity is ontology mapping. Mapping provides a common layer from which several ontologies can be accessed and hence could exchange information in semantically sound manners [35].

Also, when speaking about interoperability among ontologies, it is important to take in consideration the following operations:

- Ontology mapping and/or matching;
- Ontology alignment;
- Ontology translation;
- Ontology transformation;
- Ontology merging and/or integrating;
- Ontology checking;
- Ontology evolution and/or versioning;
- Ontology management;

Ontology mapping is an activity that attempts to relate the vocabulary of two ontologies sharing the same domain of discourse. The process of defining mappings between ontologies is not an easy task and requires currently human support [1].

Automatic mapping discovery became important for ontology alignment due to the large number of concepts in ontology. However, complex mappings have proven difficult to extract and the mapping discovery procedure certainly requires human feedback [30].

At this moment, if two or more ontologies need to be interoperable, mapping is a process that can help doing so, but it cannot be an automatic process. Consequently this process could take a while and might bring some errors in the end. This is why the manual process could be considered an obstacle of a network who share information between its intervenients. The introduction of tools and methodologies to support knowledge engineering in the process of getting the proper meaning of information and in its exchange between the several systems is crucial for the success of ontology interoperability.

2.8.1. Model Morphisms (MoMo)

In mathematics, “Morphism” is an abstraction of a structure-preserving map between two mathematical structures. It can be seen as a function in set theory, or the connection between domain and co-domain in category theory [36] . Recently, this concept has been increasing its influence when applied to computer science, namely to systems interoperability. This new usage of “morphism” specifies the relations (e.g. mapping, merging, transformation, etc) between two or more information model specifications (*M* as the set of models). An ontology is a type of information model.

In this context, the research community identifies two core classes of MoMo: non-altering and model altering morphisms [36][37].As evidenced in Table 2.1, in the non-altering morphisms, given two models (source A and target B), a mapping is created relating each element of the source with a correspondent element in the target, leaving both models intact. In model altering morphisms, the source model is transformed using a function that applies a mapping to the source model and outputs the target model [38] .Other relations, such as the merge operation, can also be classified as model altering morphisms, however they are not detailed in this dissertation.

MoMo	Formalization	Classification
Mapping: $\theta(A, B)$	$\forall A, B \in M: \theta(A,B) \subseteq Sub(A) \times Sub(B)$	**Non-altering**
Transformation: $\boldsymbol{\tau: A \times \theta \rightarrow B}$	$\boldsymbol{\forall A, B \in M: if\ \exists\theta(A,B)\ then\ \tau(A,\theta) = B}$	**Model altering**

Table 2.1: Model Morphisms Cases

To respond to the constant knowledge and model changes on heterogeneous and dynamic networks, it is required to use a more detailed and traceable mapping format that provides a semantic “link” between two different models and its components. On the following sub-sections, technologies and formalization methods will be analysed concerning their usability towards that goal.

2.8.2. Semantic Mismatches

Mismatches are inconsistencies of information that result from “imperfect” mappings. Due to the differences among the models referred before, almost in every case a MoMo leads to a semantic mismatch, which can either be lossy or lossless of information depending on the nature of the related model elements (see Table 2.2): In lossless cases, the relating element can fully capture the semantics of the related; while in lossy mismatches a semantic preserving mapping to the reference model cannot be built [39].

	Mismatch	Description
Lossless	Naming	Different labels for same concept
	Granularity	Same information decomposed (sub)attributes (see Fig. 2.4)
	Structuring	Different design structures for same information (see Fig. 2.4)
	Subclass-Attribute	An attribute, with a predefined value set (e.g. enumeration) represented by a subclass hierarchy
	Schema-Instance	An attribute value in one model can be a part of the other's model schema (see Fig. 2.4)
	Encoding	Different formats of data or units of measure (e.g. kg and lbs)
Lossy	Content	Different content denoted by the same concept
	Coverage	Absence of information
	Precision	Accuracy of information (see Fig. 2.4)
	Abstraction	Level of specialisation (e.g. "Car" and "Ford")

Table 2.2: Semantic Mismatches (based on [39])

This notion of mismatch can bring a semantic meaning to the type of the relationship being established in the mapping. However, the envisaged semantic "link" between two different models needs to account for more than inference of a meaning. It needs to be represented through a formal expression that is traceable and parseable by an intelligent system that can deduce and recommend mapping readjustments, which might even change the mismatch type.

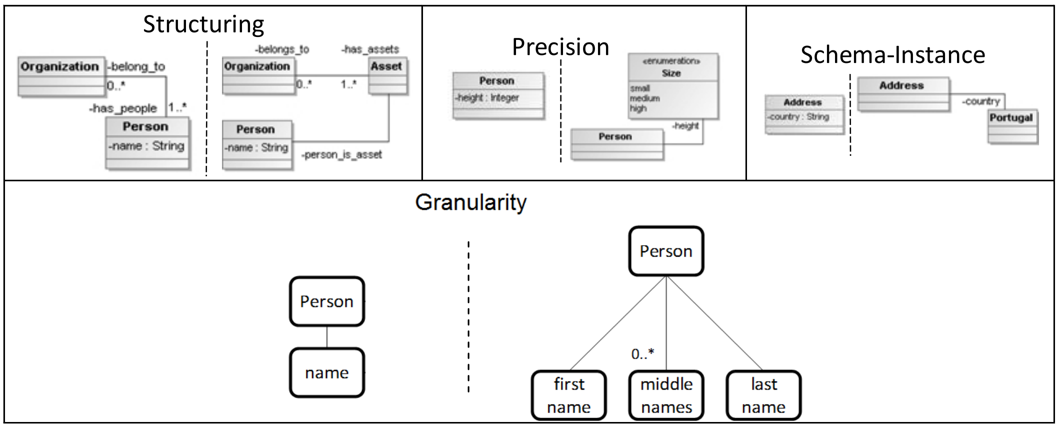


Fig. 2.4: Mismatch Examples

2.9. Ontologies

Gruber stated in 1993 one of the most well known description about what an ontology is: An ontology is a formal, explicit specification of a shared conceptualisation. The term is borrowed from philosophy, where ontology is a systematic account of Existence [5].

In computer and information science, ontology is a technical term denoting an artefact that is designed for a purpose, which is to enable the modelling of knowledge about some domain, real or imagined [40]. It is used to reason about the properties of that domain, and may be used to define the domain. Ontologies provide a shared vocabulary, which can be used to model a domain.

While the terms specification and conceptualization have caused much debate, the essential points of this definition of ontology are:

- definition of concepts, relationships, and other distinctions for modeling a domain;
- the specification takes the form of definitions of representational vocabulary (classes, relations, and so forth), which provide meanings for the vocabulary and formal constraints on its coherent use [40].

Ontologies are used by people, data bases, and software applications that need to share information, where their information domains are related to a particular area of knowledge. Ontologies include definitions of basic concepts used by computers that may contain relationships between them to enable the information organization in several domains [15].

Ontologies are now being recognised as important components of information systems and information processing. It is commonly accepted that an ontology is an explicit specification of a conceptualisation [5]. In the areas of knowledge representation and reasoning and conceptual modelling, it has been recognised that conceptualising a domain is a prerequisite for understanding the domain and processing information about the domain, especially in the case of large, non-trivial domains. Nowadays, there is no clear-cut border between large and small domains, simply because information systems are no longer isolated but are parts of the global information *system* and need to be interoperable. Hence, conceptualisations and ontologies are required for all kinds of information systems and information processing. They offer a way to address meaning of terms – concepts, and relations – required for information processing [1].

In the context of the Semantic Web, ontology is a kind of technology enabler – a layer of the enabling infrastructure – for information sharing and manipulation. The approach is simple: parties who have software, or data, or even services to offer identify some common conceptualization of the data; they specify that conceptualisation as clearly they can; they build systems that interoperate on those specifications. This is standard-issue information technology, with the twist that ontologies are specifications of the conceptualizations at a semantic level [41].

Ontologies make easier the process of usage and exchange of data, information, and knowledge between people. They also facilitate the integration of different perspectives from several people. When software tools and applications are involved, ontologies can be used in three levels [15]:

- Design and development of software systems;
  - Specification;
  - Relationship;
  - Reuse;
- Communication;
- Interoperability;
  - Data interoperability;
  - Function interoperability;
  - Process interoperability.

Ontologies can provide contents and meanings based on semantics related to electronic information, and have considerable impact in areas that deal with big amounts of information for computers. The major commercial ontology applications are [15]:

- Electronic commerce;
- Information control.

Ontologies can be used to improve Web based applications providing new solutions such as:

- Web pages;

- Multimedia;
- Design documentation;
- Intelligent agents.

In line with its applications a new concept arose that tries to take the ontologies to a new stage, and this term is folksonomy.

A folksonomy is a classification system derived from the practice and method of collaboratively creating and managing tags to add notes and categorize content, known as collaborative or social tagging.

After a while, it was noticed that regular people who don't write computer programs were happily tagging with keywords the content they read. Of course, the keyword *tagging* is nothing new – tagging is an activity in which you mark some content you create or experience with one or more labels, or tags – but it is interesting to notice when these individuals do their tagging in public areas, the collection of their keywords becomes a useful source of data for further studies [41].

With tag data ontologies, technologies for searching, aggregating, and connecting the people and content they contribute throughout the Web might be enabled. At the same time, the rich data from millions of active, participating people might be a big help for the development of systems that tap the power of collective intelligence [41].

3. Methods and Methodologies for Ontology Building

A typical reason for creating new ontologies is to give a common language for sharing and reusing knowledge of a domain of interest. Common ontologies are also used to describe ontological commitments for a set of agents so that they can communicate about a domain without necessarily operating on a globally shared theory.

Different methodologies for building ontologies exist and it is possible to classify them as follows:

- Methods and methodologies for building ontologies from scratch;
- Methods for reengineering ontologies;
- Methods for cooperative ontology building;
- Ontology merging methods.

For each methodology a brief description will be presented as well as the main research issues [42].

3.1. Methods and Methodologies for building ontologies from scratch

A bunch of approaches have been reported to build ontologies. Lenat and Guha (1990), published the general steps and some interesting points related to the Cyc project development process. Cyc is an artificial intelligence project attempting to assemble an ontology and knowledge base of everyday common sense knowledge, with the goal of enabling AI applications to perform human-like reasoning.

The Cyc methodology consists on the following steps: first of all one have to extract by hand, common sense knowledge that is implicit in different sources. Next, once enough knowledge in the ontology is available, new common sense knowledge can be acquired either using natural language or machine learning tools. Some years later Uschold and King (1995) published the main steps followed in the development of the Enterprise Ontology. The method proposes some general steps to develop ontologies, which are:

- Identify the purpose;
- Capture the concepts and the relationships among concepts and terms used to

denote both of them;

- Codify the ontology [42].

3.2. Methods for ontologies re-engineering

Ontological reengineering is the process of retrieving and mapping a conceptual model of an implemented ontology to a more suitable one, which is re-implemented. The Ontology Group of Artificial Intelligence Laboratory at UPM presented a method for reengineering ontologies that adapts Chikofsky's software reengineering schema to the ontology domain [43]. Three main activities were identified: reverse engineering, restructuring, and forward engineering [42].

3.3. Methods for cooperative ontology construction

Ontologies are a shared and common understanding of a domain. Currently, emphasis has been put on the ontology content consensus, i.e., on the collegial agreement on the formal specification of the concepts, relationships, attributes, and axioms ontologies provide. However, the problem of in cooperation construct an ontology in a distributed environment is still unsolved.

This topic motivated several studies about cooperative environments and some problems have been identified, as the following [44].

- interaction and communication management among people;
- data access control;
- recognition of a moral right about the knowledge (attribution);
- error detection and management;
- concurrent management [42].

Ways for controlling or reducing these problems might be found in order to make the creation of ontologies in cooperative environments a good practice. The most difficult one might be the interaction and communication between people when involved in meeting but there are some studies made on Forecasting Methodologies, presented on Chapter 3, to deal with this.

3.4. Ontology merging methods

Ontology merging is the process of creating a new single ontology from two or more conceptually divergent ontologies related to a specific domain. The idea of this method is to build a new ontology starting from existing ontologies by merging, or extending some of the existing parts. A merging task resembles a construction of a new ontology. This process is similar to database merging [45].

A development starts with defining the domain and a lexicon of a common vocabulary. It forms a base for a hierarchy of concepts – they are divided into classes and proper relations are attached. There are different approaches to hierarchy construction:

- top-down: starts with the most general concepts;
- bottom-up: begins with the most detailed helps to avoid mistakes during modelling large domains.

3.5. Collaborative Ontology Building Methodologies

Until now, several methodologies have been proposed for solving the problem of collaborative ontology building. The following methodologies are related with the methods for cooperative ontology construction mentioned in the previous section.

The Iterative, Collaborative Ontology Construction (ICOC) Scheme [46], supports the online collaborative knowledge contribution. It uses a wiki-like application that allows users to collaboratively integrate their knowledge to build a new ontology. A Delphi-like method is then applied to converge the answers to an automatic generated questionnaire to construct a uniform ontology. This scheme will be processed iteratively until all relations are converged.

Holsapple et al. in “A Collaborative Approach to Ontology Design” [47], used a Delphi-like method to structure collaboration in the direction of consensus. The method used is composed by four phases:

- Preparation phase: the idea is to define the design criteria, determine boundary conditions, and determine evaluation standards;
- Anchoring phase: it is specified the initial ontology that will seed the collaborative effort;
- Iterative Improvement phase: pretends to identify diverse panel of participants who provide their critiques and comments on the ontology to a leader who revises the ontology addressing the feedback from the users. This is an iterative

process that occurs until a consensus is reached;

- Application phase: the idea is to demonstrate the uses of the ontology.

Collaborative Protégé [48], is an extension of the existing Protégé system that supports collaborative ontology editing. It also enables annotation of ontology components and ontology changes. In addition, two types of voting mechanisms have been implemented and can be used for voting of change proposals. It can be used in two different modes:

- Multi-User Mode: allows users to simultaneously edit a same ontology;
- Standalone Mode: where an ontology can be accessed by several users but not at the same time.

Another approach to collaborative environments is the INTEROP project [49], where partners shared documents (and took important information from it), to build a glossary. To build this glossary, firstly it was used the OntoLearn TermExtractor module to extract terms from these documents. Then, the list of terms was reviewed by a set of domain experts to refine the glossary. These terms could be reviewed, rejected, accepted, or ignored by the reviewers in an iterative process until all definitions were approved by the majority of voters.

The OntoWiki [50], tool is other kind of collaborative environment that supports agile Knowledge Engineering. To enable users to edit information the OntoWiki approach supports two complementary edit strategies for the knowledge base:

- Inline editing: The smallest possible information chunks (i.e. statements) presented in the OntoWiki user interface are editable for users;
- View editing: Common combinations of information (such as an instance of a distinct class) are editable in one single step.

Editable *Views* are combinations of widgets to edit a specific view on the knowledge base in a single step. The OntoWiki system provides the following types of editable *Views*:

- Metadata;
- Instances;
- *Views*.

Making means of social interactions as easy as possible furthermore contributes in creating an “architecture of participation” that allows users to add value to the system as they use it. Social collaboration within OntoWiki is in particular supported by:

- Change tracking;
- Commenting;

- Rating;
- Popularity;
- Activity/Provenance.

3.6. MENTOR Methodology

MENTOR – Methodology for Enterprise Reference Ontology Development [51], is a collaborative methodology developed with the idea of helping a group of people, or enterprises, sharing their knowledge with the other in the network, and provides several steps as semantic comparisons, basic lexicon establishment, ontology mappings and some other operations to build a domain's reference ontology. It aims to combine the knowledge described by different formalisms in a semantic interoperable way [57].

This methodology is composed by two phases and each phase has three steps, which can be seen on Fig. 3.1.

The Lexicon Settlement, or Phase 1, represents the knowledge acquisition by getting a collection of terms and related definitions from all participants. This phase is divided into three steps: Terminology Gathering, Glossary Building, and Thesaurus Building. The first step is a very simple one, and it represents the knowledge gathering from all intervenients in the collaborative network in a form of a list of terms. In the Glossary Building step, a glossary is built after serial discussions about the terms that every participant contributed to the network on the previous step. These discussions are followed by a voting process, with all participants deciding which corresponding terms and definitions compose the glossary. Beyond the glossary, the semantic mismatches record is another output that results from this step. The last step of this phase is composed by a cycle where the knowledge engineers define a taxonomic structure with the glossary terms. If there is an agreement in both structure and classified terms, the thesaurus is defined. If not, the cycle starts again for another iteration. In this first phase, it could be valuable to have a multi-language dictionary for situations where a common language is not shared by all participants.

The Reference Ontology Building, or Phase 2, is the phase where the reference ontology is built, and the semantic mappings between participant's ontologies and the reference ontology are established. This phase, just like the first phase, is divided into three steps: Ontologies Gathering, Ontologies Harmonization, and Ontologies mapping. The first step comprehends the acquisition of ontologies in the defined domain. In Ontologies Harmonization step, it is needed to proceed to two harmonization types: taxonomic harmonization and contents harmonization. First, a discussion and voting process about the reference ontology structure takes place where the common classes are defined by unanimity. This process of discussing and voting is then repeated for the contents harmonization. The final step of this phase, the Ontology Mapping, attempts to relate the vocabulary of two ontologies that share the same domain. In this case, the idea is to establish mappings between each participant's ontology and the reference ontology defined on the previous step [51].

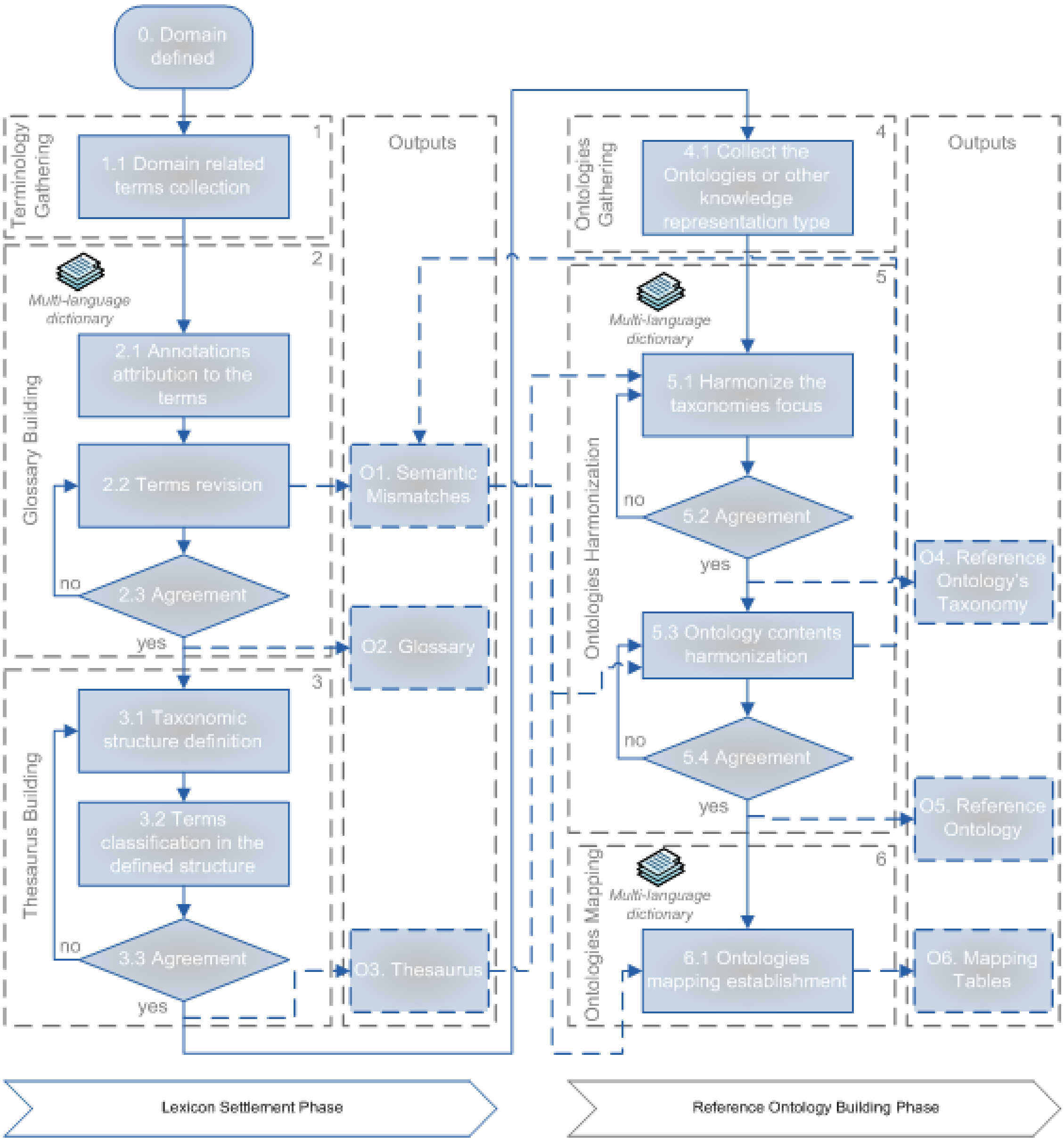


Fig. 3.1: MENTOR Methodology originally proposed [51]

3.6.1. Mediator Ontology

Ontology mapping is an activity that attempts to relate the vocabulary of two ontologies that share the same domain of discourse [53]. The process of defining mappings between ontologies is not an easy task and requires currently human support. The MENTOR uses the Mediator Ontology (MO) as the reference for mediating the mapping establishment and its subsequent "mapping records" reasoning. An example is querying the MO for a correspondence to a reference term in a specific enterprise ontology.

The MO is able to represent ontology semantic operations: the semantic mismatches found in the Glossary Building step; the semantic transformations identified in the Harmonization process; the Ontologies Mapping; and other ontologies operations (e.g. versioning). It was built up as an extension to the Model Traceability Ontology defined by Sarraipa et al [54]. Traceability is ability to chronologically interrelate the uniquely identifiable entities in a way that matters. The mapping relations can be related to a traceability element, in a sense that a specific term defined in the reference ontology has a related one in an organization member ontology, considering ontologies as stages of the desired ontology life-cycle, that is in this case the reference ontology. This makes possible a way to trace ontology elements.

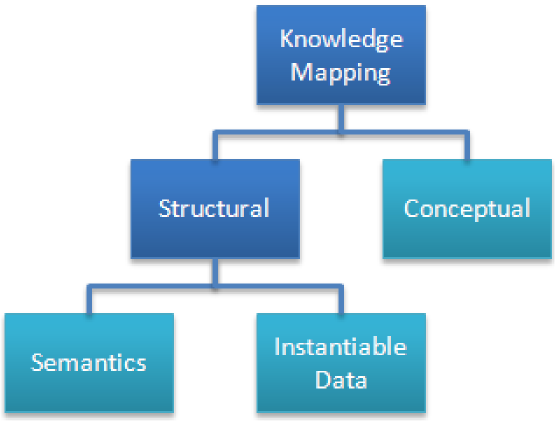


Fig. 3.2: KMType values

For being able to represent such ontologies operations, MO is compliant with a five-tuple mapping expression proposed by Agostinho et al. [55]:

$$Mapping\ Tuple\ (MapT): < ID, MElems, KMType, MatchClass, Exp >$$

- ID is the unique identifier of the MapT and can be directly associated with the a's vertex number: $IDi.j\_x: 1 \leq i \ \leq |V(A)|\ and\ 1 \leq j \leq |V(Sub(B))| and\ x \in \mathbb{N}$. The depth of the sub-graph detail used in the mapping is not limited, and x is a counter for multiple tuples associated with the same concept;
- MElems is the pair (a, b) that indicates the mapped elements. If the ID specifies a mapping at the n-the depth level of the graph, a should be at the same level, i.e. a.ai (for i =1..n);
- KMType stands for Knowledge Mapping Type, and can be classified as: "Conceptual" if mapping concepts and terms; "Semantics" if mapping model schemas; and "InstantiableData" if the mapping is specifying instantiation rules.
    - $KMType = \{Conceptual, Semantics, InstantiableData\}$;
- MatchClass stands for Match/Mismatch Classification and depends on KMType, such as $\forall(a, b) \in MElems$:
    - $\forall KMType$, if a=b, the mapping is absolute and $MatchClass = Equal$;
    - if $KMType = Conceptual$ the mapping is relating terms/concepts:
        - and $MatchClass \in \left\{ \begin{matrix} Equal, Naming, Coverage, \\ MoreGeneral, LessGeneral, Disjoint \end{matrix} \right\}$ depending on the coverage of the relationship;
    - if $KMType = Semantics$ the mapping is structural, $MatchClass \in \{\text{Naming, Granularity, Structuring, SubClass} - \text{Attribute, Schema} - \text{Instance, Encoding, Content, Coverage, Precision, Abstraction, Equal, Disjoint}\}$;
    - if $KMType = InstantiableData$ the mapping is structural;
- Exp stands for the mapping expression that translates and further specifies the previous tuple components. It can be written using a finite set of binary operators derived from the mathematical symbols associated with the mapping types and classes (e.g. "$=, \sim, \subseteq, \supseteq, \perp, +, -, \times, \div, concatenate, split$").

This mapping tuple which represents $\theta(a, b)$, can also be used to generate a transformation function τ, where $\tau(a, \theta) = b$, being $(a, b) \in MElems$. Therefore, when used by intelligent systems such as CAS-like IS, the tuple's information enables automatic data transformations and exchange between two organizations working with/on different information models, thus achieving an interoperable state among them and supporting the recovery from any harmonization breaking situation.

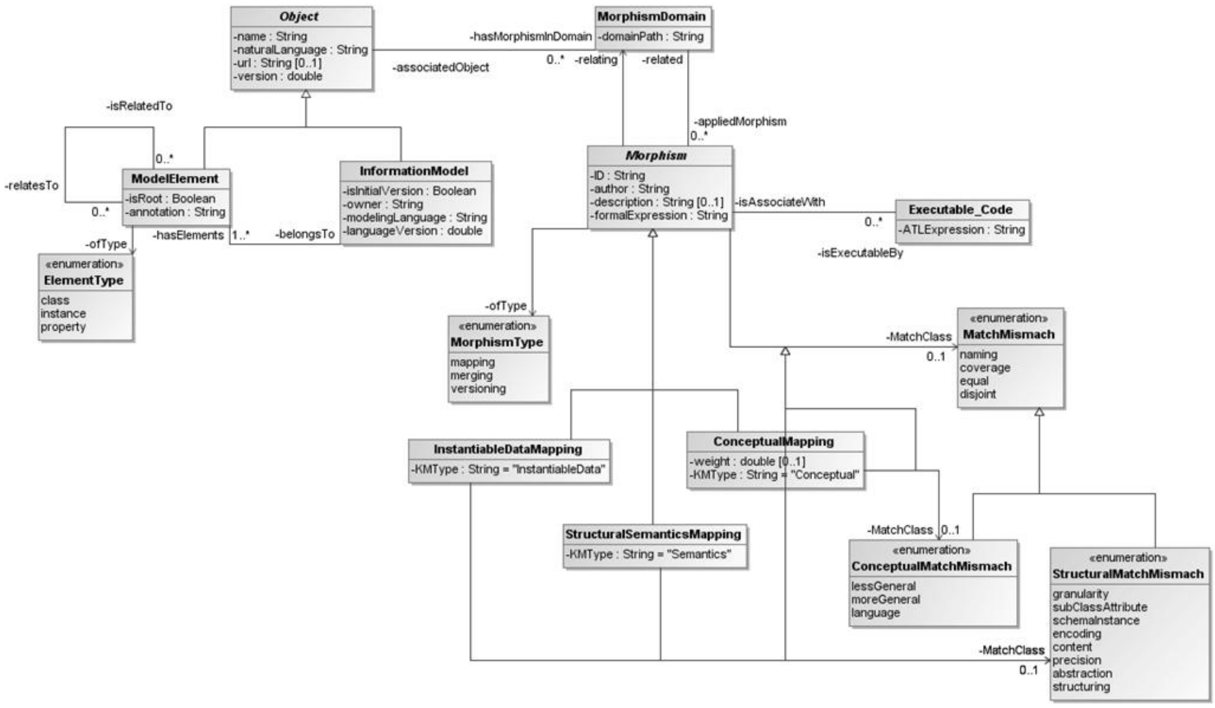


Fig. 3.3: Structure of Mediator

The structure of the MO is presented in Fig. 3.3. It has two main classes: Object and Morphism. The Object represents any InformationModel (IM) which is the model/ontology itself and ModelElements (also belonging to the IM) that can either be classes, properties or instances. The Morphism basically associates a pair of Objects (related and relating – Melems in MapT), and classifies their relationship with a MorphismType, KnowledgeMappingType (if the morphism is a mapping), and Match/Mismatch class (MatchClass in MapT). The Morphism is also prepared to store transformation oriented ExecutableCode that will be written in the *Atlas Transformation Language* (ATL) and can be used by several organizations to automatically execute the mapping, transforming and exchanging data with their business partners as envisaged in [56].

With the MapT stored in a knowledge base (KB) as the MO, to support communications intelligence, all information concerning the mappings between models or ontologies of business partners can be accessed by their local systems (Fig. 3.4). This allows communities to build intelligent systems with reasoning capabilities able to understand each others' representation format, without having to change their data and communication functions [57].

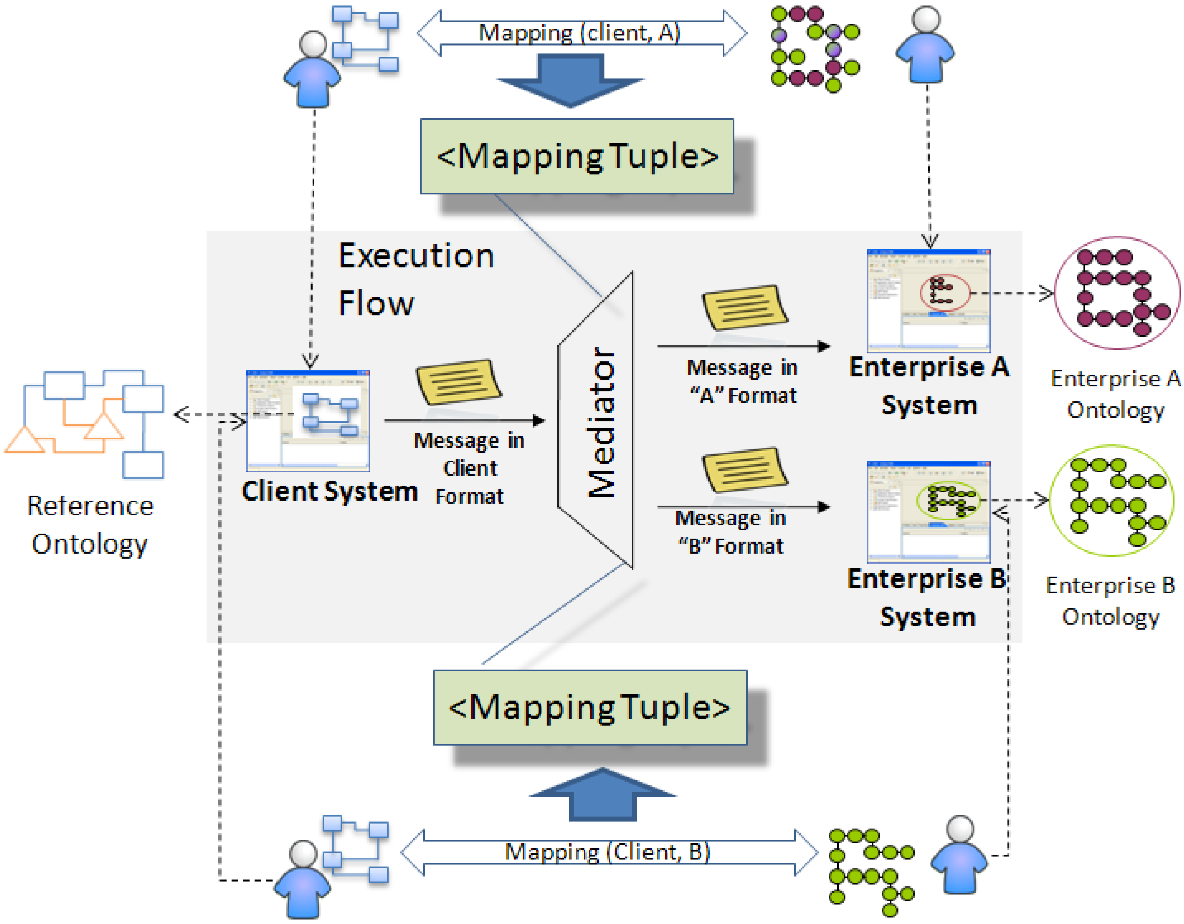


Fig. 3.4: Mapping design and execution flow in data exchange

3.7. Qualitative Information Collection Methods

In this section are presented four different approaches that appeared to structure people interaction, more precisely to structure group meetings. Each of this methods has particular properties, and have some advantages as well as drawback that may be interesting to read. They are presented in this section and later some of them are adapted to MENTOR steps.

3.7.1. Nominal Groups

A Nominal Group session requires several participants to discuss about a topic to generate a list of ideas. This approach is an alternative to brain storming, because it is a more structured discussing method. In this method are given time to participants to think and write down their ideas before telling them to the group [58]. This technique is a structured variation of small group discussion methods. The process prevents the domination of discussion by a single person, encourages the more passive group members to participate, and results in a set of prioritized solutions or recommendations. The steps to follow this technique are the following [59]:

- Silent Generation of Ideas in Writing;
- Round Robin Recording of Ideas;
- Serial Discussion for Clarification (for each idea);
- Voting on the Priority Strategies;
- Discussion of Preliminary Voting;
- Final Voting.

This method has many benefits and advantages that include:

- Motivates all participants to get involved;
- Generates many ideas in a short period of time.
- Obtain ideas from people of different backgrounds and experiences;
- Stimulates creative thinking and effective dialogue;
- Allows clarification of ideas.

The most relevant disadvantages can be listed as the following:

- Requires a skilled leader;
- Can take too much time if the group is not properly controlled and is allowed to run for too long;
- Assertive personalities may dominate unless leadership skills are exercised.

3.7.2. Metaplan

The Metaplan technique is a learning method especially for groups. It is a collaborative and moderated technique with focus on solving group decisions or problems [60]. It combines individual and collective contributions and is used to organize concrete ideas into more general conclusion leading to recommendations. It is mostly useful to explore an issue and dig out what is the key on it. The Metaplan process can be described in five steps:

- Agreement on principles;

- Exploratory discussions;
- Development of the 'dramaturgy';
- Meeting / Workshop;
- Post meeting follow-up.

The advantages of this method are the following:

- Reach common points of view and take actions to support participants' convictions;
- Lead the group into joint actions;
- Discuss and clarify the objectives within the allotted time.

The disadvantages of this method are:

- Requires a skilled leader;
- There may be ideas overlapping due to unclear or inadequate group discussion;
- "Knowledgeable" individuals selected to participate may not represent all community subgroups.

3.7.3. Surveys

Surveys are a widely used method to gather scientific information about how people feel about a particular issue [61]. They are useful in determining correlations between sets of beliefs and perhaps giving hints to cause and effect relationships.

Surveys are based on information collected from a sample of a community or population. On the other hand, surveys can be administered to all people in a community or organization to provide everyone with an equal opportunity to express themselves. The most commonly used survey methods are person-to-person interviews, drop-off and pick-up questionnaires, mail questionnaires, and telephone interviews. While each approach is somewhat different, the format is similar. Each one asks an individual to supply attitudes, beliefs, behaviours, and attributes in response to specific questions.

The major advantages of this method are:

- Can be inexpensive, especially if volunteers are available to conduct the survey, or if data records about the particular issue already exist and can be reused;
- A statistical sample can provide accurate information about a population.

The disadvantages of surveys are the following:

- To assure statistical meaning, random samples must be carefully selected;
- Results may not be valid if surveys are not designed correctly;
- May require time and expertise to develop the survey, train interviewers, conduct interviews, and analyze results.

3.7.4. Delphi

Delphi is a method for structuring a group communication process so that it is effective in allowing a group of individuals, as a whole, to deal with a complex problem [62]. It allows free discussion of views without the influence of personal status; allows the iteration of personal views with controlled feedback and involves groups of people interested in a specific subject. This methodology is specialized in technology evaluation and can be used either for qualitative as well as for quantitative methods. The Delphi process can be described in three steps:

- Selection of the “expert panel”: An expert may be considered to be an individual who has recognized expertise in a particular subject or may be anyone who can provide a worthwhile opinion on the subject in question. These groups of people with expertise in some specific areas of the study are addressed to give input to the formulation of the questionnaires in the next step;
- The Delphi rounds: It has three or more rounds, depending on the situation:
    - Round 1: In “classical” Delphi, the first round is completely unstructured, asking members to express any opinions about the current topic. The first round contains a synopsis of the issue in question together with the source and validity of the information upon which it is based;
    - Round 2: With the results of the first round a questionnaire is constructed containing a series of statements or questions that respondents are invited to express an opinion on;

  - Round 3: In the round 3, participants re-rank their initial statements in the light of the results of round two. Their own answers from the second round are fed back to the panellists so they may see their own answers in the light of the group's overall response;

- Results and conclusion: The results of the process are disseminated to the group as a well researched guideline to best practice. This may then be used as a benchmark document for subsequent audit.

The Delphi method has the following advantages:

- Allows participants to remain anonymous;
- It is free of social pressure, personality influence and individual dominance;
- The questionnaires will be posted and answered using Internet, making the process inexpensive.

The disadvantages of this methodology are:

- More time-consuming than the group process method;
- Should not be seen as a total solution to forecasting;
- Requires skills in written communication;
- Requires adequate time and participant commitment.

4. MENTOR USING QUALITATIVE INFORMATION COLLECTION METHODS

After the research and further study of each QICM, it is proposed an adaptation to MENTOR where in each step involving collaboration, some methods where introduce. The steps where these QICM were introduced are:

- Glossary Building;
- Thesaurus Building Cycle;
- Ontologies Harmonization;
  - Ontology's taxonomy harmonization Cycle;
  - Ontology's contents harmonization Cycle.

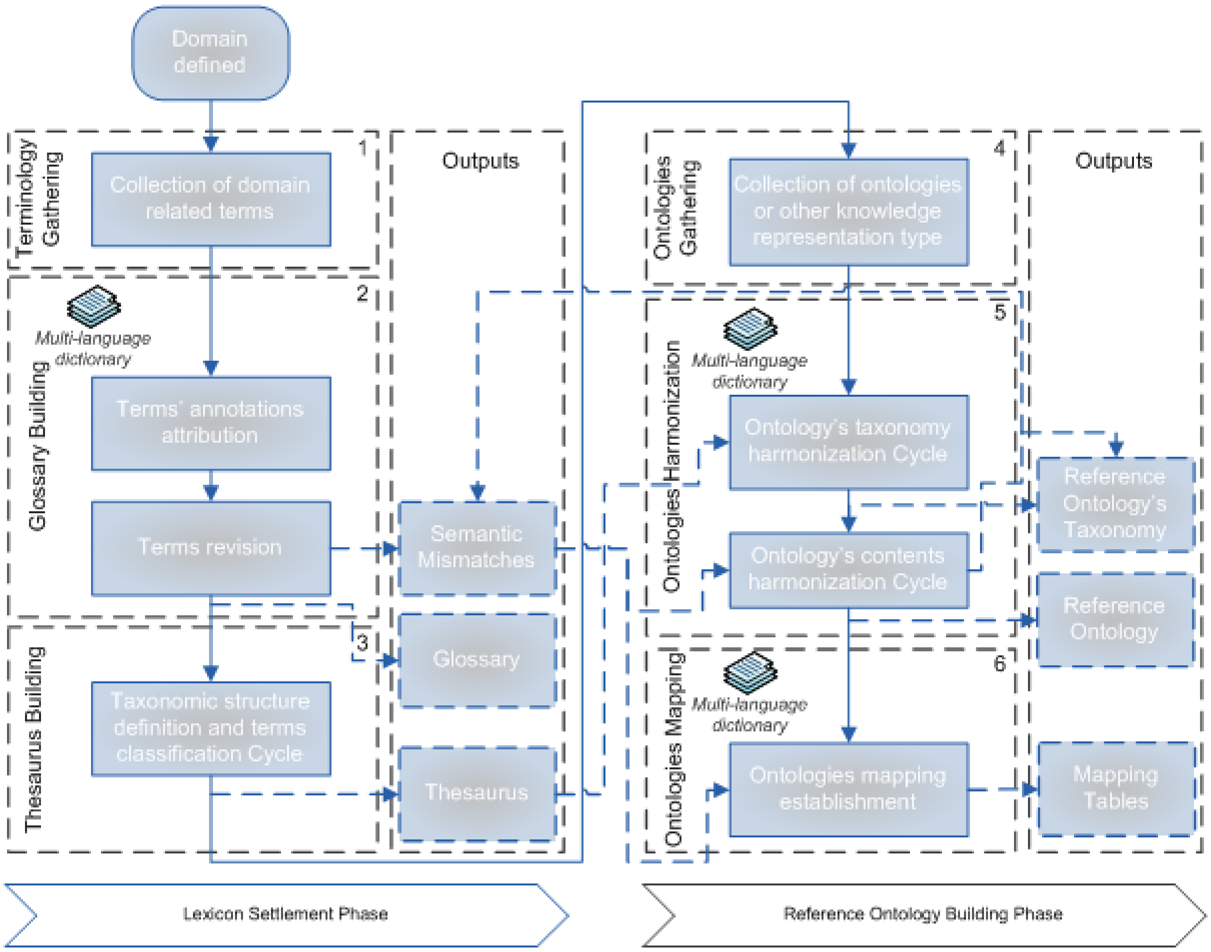


Fig. 4.1: Changes to MENTOR Methodology

4.1. Glossary Building

After gathering all the terms with proprietary (own enterprise) definitions, it is time to establish a group of reference terms with reference definitions – the glossary definition. The Glossary Building step is enriched with the Nominal Groups approach.

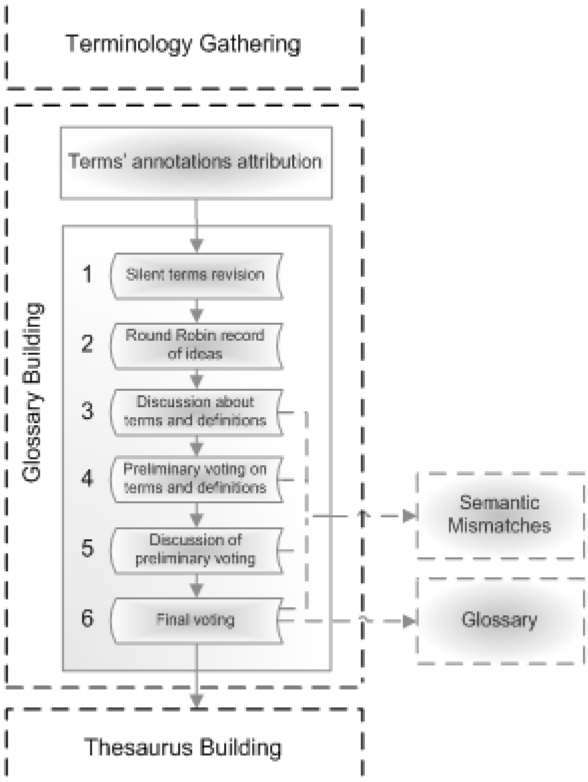


Fig. 4.2: Terms revision

This choice is based on the sequential nature of the adopted method, where a glossary building consists on a list of terms to be defined, aiming to discuss these terms one by one, from the beginning to the end without cycles. Since the implementation of the collaborative platform for MENTOR is planned to use web services, Metaplan method does not seem to be the appropriate solution as it proposes a meeting for discussions which makes such implementation difficult to achieve. Surveys are also discharged because of its absence in discussions. Therefore, the Glossary Building step is accomplished following the six Nominal Groups' steps as depicted in Fig. 4.2:

Step 1: Participants review alone the overall list of terms rating and commenting them;

Step 2: Participants write down some ideas about each term and corresponding definition (equivalent terms identification);

Step 3: The system manager starts and leads a group discussion about the terms and their definitions. The objective of this group discussion is finding which terms should compose the glossary being the reference ones and find a proper definition for each term;

Step 4: Preliminary voting process is to check if there was an agreement between participants. Members vote if they agree, or not, with each term and definition;

Step 5: New discussion takes place to clear up any doubts about some specific terms or definitions;

Step 6: Final voting session to define the output of this step, the glossary. This glossary contains all terms and their definition decided by majority.

From the last steps of this approach it is established a list of the semantic mismatches records related to each term linked, between proprietary and reference ones. The semantic mismatches record in the Mediator Ontology.

4.2. Thesaurus Building Cycle

After having the glossary built, it is performed the thesaurus definition. In this process it is discussed the taxonomic structure of the thesaurus.

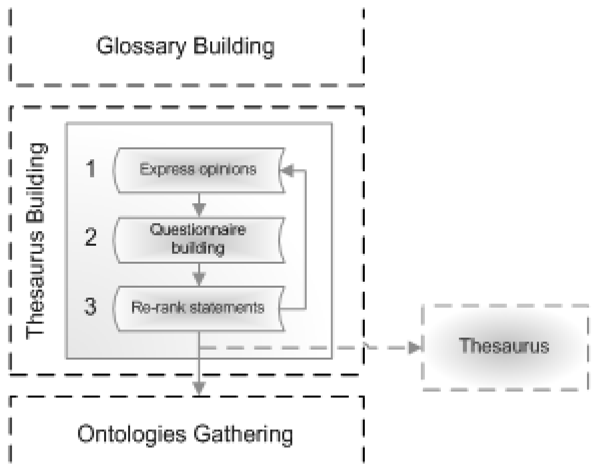


Fig. 4.3: Taxonomic structure definition and terms classification

Discussions around a taxonomic structure definition are more complex than for a simple glossary definition. It requires higher conceptual discussions than in the semantic attribution to the terms. For instance, it is needed to know the entire domain, to be able to relate and classify the existing terms in a common structure. Thus, as described in the previous chapter, when faced with complex discussions Delphi is a good method to follow. Additionally, Delphi facilitates a kind of re-rank answers enabling sometimes to reach a faster solution, avoiding repetition of extra discussions. For these reasons, Delphi method (using three rounds), is the best solution for realisation the thesaurus building, and according to Fig. 4.3 the steps for its development are:

Round 1: Express opinions – Every group member is asked to express its opinions about the structure of the thesaurus. This discussion is used to identify questions, for example, "which classes are able to classify a specific business domain?";

Round 2: Members answer a questionnaire built with the results of round 1 (this could be open to other people from the involved companies);

Round 3: Participants re-rank their initial statements with the results of the round 2.

These rounds are performed at least once for each taxonomic level definition (Fig. 4.4). However for reaching a general consensus in each level, all the process could be accomplished more than one time.

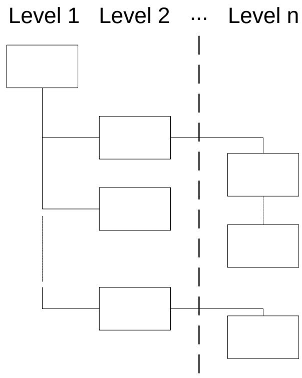


Fig. 4.4: Taxonomic levels

Once the thesaurus is defined, it is time to proceed to the reference ontology building. It starts by gathering the various proprietary ontologies from the involved enterprises, plus the thesaurus defined previously. Then, the Ontology's Harmonization step starts, through two cycles:

Ontology's Taxonomy Harmonization Cycle;

Ontology's Contents Harmonization Cycle.

4.3. Ontology's Taxonomy Harmonization Cycle

The Ontology's Taxonomy Harmonization Cycle has a similar objective as the thesaurus building step. Consequently, the method to follow is Delphi (Fig. 4.5), although in this case the Delphi rounds are applied to the whole structure and not level by level as in the thesaurus building step (Fig. 4.6). The difference is justified by the fact that in this case there is a taxonomic structure already agreed by all, which is the thesaurus. Feedback is important for the sustainability of the harmonization cycle.

The Ontology's Taxonomy Harmonization Cycle comprehends:

An Expert panel composed by the involved participants;

Round 1: Express opinions – Every member of the discussion group is asked to express its opinions about the reference ontology taxonomy. For instance, a usual discussion could be related with considering, or not, that the taxonomic structure of the thesaurus has higher weight in comparison to the enterprise ontologies. With this discussions are identified a set of questions;

Round 2: Members answer to a questionnaire built with the results of round 1;

Round 3: Participants review all answers and consequently if needed reformulate the reference ontology taxonomy.

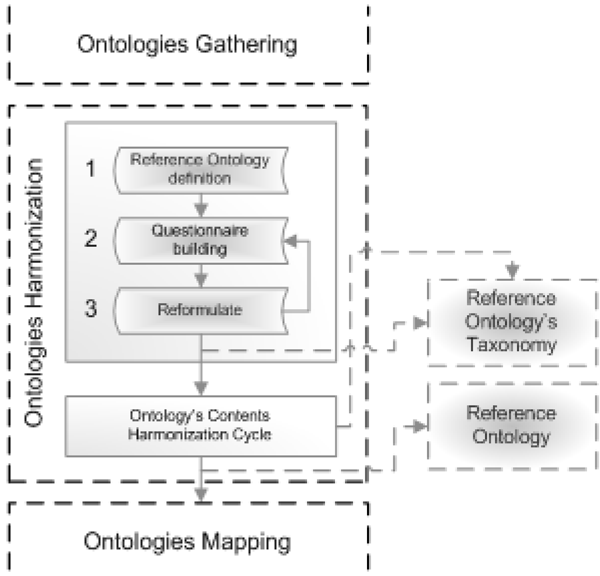


Fig. 4.5: Ontology's Taxonomy Harmonization Cycle

The construction of the reference ontology's taxonomy is based on the previously defined thesaurus and the user's private ontologies. It works like a sum of all components and the result is an ontology that meets the ideas of all the involved participants. This process is represented on Fig. 4.6.

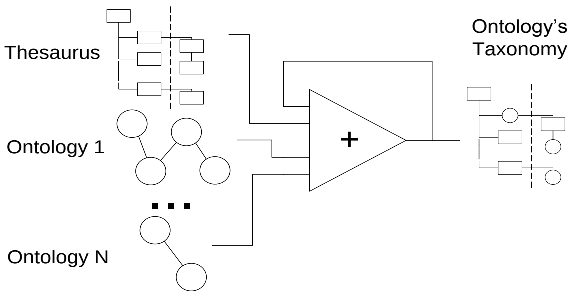


Fig. 4.6: Ontology's Taxonomy Building Process

4.4. Ontology's Contents Harmonization Cycle

With the Reference Ontology Taxonomy defined, it is needed to establish its properties and rules. The process applied here is similar to the previous one. Thus, it follows the Delphi method but by this time it is applied to the properties and rules (Fig. 4.7), comprehending:

An Expert panel composed by the involved participants;

Round 1: Express opinions – Every member of the discussion group is asked to express its opinions about the properties and rules of the new reference ontology;

Round 2: Members answer to a questionnaire built with the results of round 1

Round 3: Participants re-rank their initial statements with the results of the round 2.

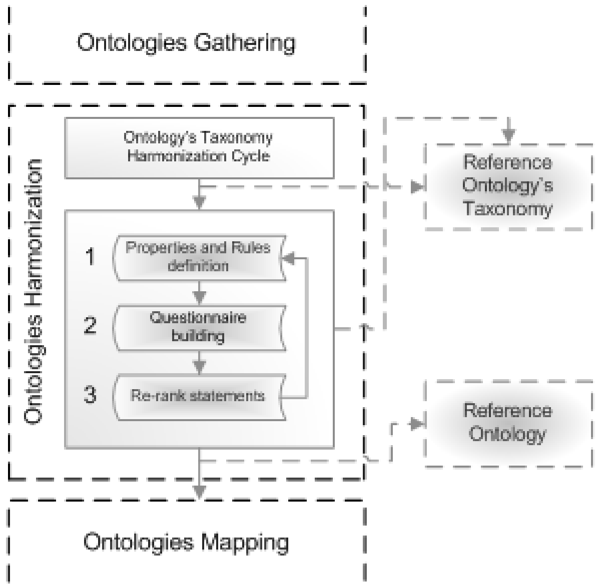


Fig. 4.7: Ontology's contents harmonization cycle

As it can be seen on Fig. 4.8, for the construction of the reference ontology's contents it is important to take into account the reference ontology's taxonomy and the properties and rules given by each project participant. This process is repeated until a final solution is found to feat every participant's interests.

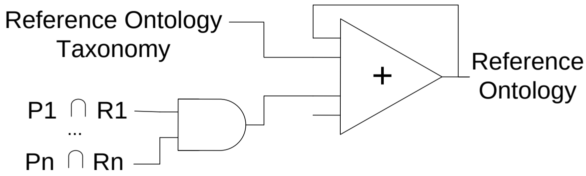


Fig. 4.8: Ontology's Contents Building Process

4.5. Proof-of-Concept Implementation

The objective of this project was to show that it is possible to create an ontology in a collaborative environment using MENTOR methodology with the help of qualitative information collection methods for the situations where it is needed to structure a group talk.

Although the idea of this work is to have a solution that does not depend on other tools, it was possible to do so only for the first and second steps of the first phase of MENTOR. For the rest, the other 4 steps, it was used Collaborative Protégé, an existing tool for creating, and editing ontologies, but it must follow a structure that fulfils MENTOR requirements.

A study on the required technologies has been made before starting the implementation of the project. The chosen technologies and their descriptions are shown on the following sub-section.

4.5.1. Used Technologies

4.5.1.1. Java

Java is a flexible technology that allows developers to write software to be run on practically all platforms (platform-independent), to create multi-threaded programs that run in Web browsers and Web services, and to combine existing Java objects in their programming solutions to quickly server-side applications [63].

Java is, at the same time, a high-level object oriented programming language, and the software developed must run on every hardware that has installed a Java Virtual Machine (JVM). For this, Java is compiled to bytecode instead of directly to platform-specific machine code.

4.5.1.2. Web Services

Web services, typically convert a normal application into Web applications. They have been created for giving the possibility to share a service with the world. If you want to share your application functions on the internet, you just need to publish your services and they are ready to be consumed by anyone [64].

Web services are also important when there is a need to change data between different network platforms, programming languages, operating systems, or different hardware systems. They help on the interoperability problem by letting applications made in different programming languages or different operating systems to link their data.

4.5.1.3. JavaServer Pages

JavaServer Pages (JSP) technology provides a fast and simplified way for creating dynamic web content [65]. A JSP file is generally composed by HTML or XML code mixed with tags and scriptlets written in the Java programming language.

JSP files allow a Web server, such as Apache Tomcat, to add content dynamically to HTML pages before they are sent to a requesting browser.

```
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
          <% out.println("JSP Example"); %>
    </body>
</html>
```

Fig. 4.9: JSP Example

The first time a JSP is run in the by the server, it compiles the file and converts it to a simple HTML file. This process may take a few seconds but happens only one time. The next time the page is requested it is automatically shown because there is no need to recompile the JSP file.

4.5.1.4. Protégé

Protégé is a free, open source ontology editor and knowledge-base framework that supports two main ways of modelling ontologies via the Protégé-Frames and Protégé-OWL

[66].

Protégé also provides a set of libraries that enable a user to work with ontologies in Java language [67]. Consequently, it is possible to create an application or a set of functions in Java to open or create an ontology, and do operations such as read, edit, create and delete classes, properties and their instances.

4.5.1.5. Collaborative Protégé

Collaborative Protégé is an extension of the existing Protégé tool to support collaborative ontology building [68]. With Collaborative Protégé users are able to join a project through a server, and if they have permission to do so, they can edit ontology classes and properties through desktop or web Protégé clients. All changes made on the ontology will be immediately reflected on the associated annotations and changes ontology and all users will be able to see these changes.

Collaborative Protégé has a collaboration panel and users are able to see entity notes, track changes, all notes, ontology notes, search notes or changes on the ontology and chat with the other users. The collaboration panel can be seen on the figure below.

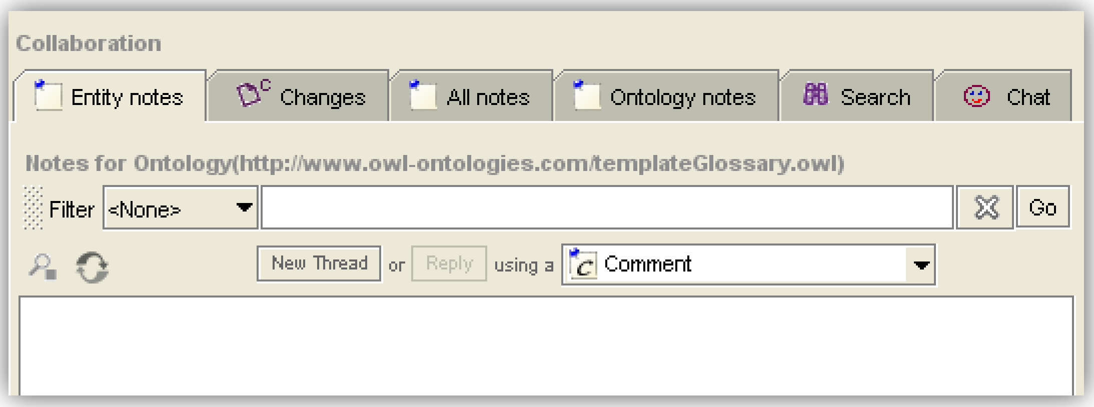


Fig. 4.10: Collaborative Protégé panel

4.5.2. Architecture

To put MENTOR methodology into action, a simple architecture has been developed using the tools and technologies mentioned before. The architecture can be seen on Fig. 4.11 as well as the interactions between the different elements that compose it.

The architecture is composed by five different parts:

- a metaproject file;

- a set of OWL files;
- two Java classes (working as libraries);
- a Web service with a collection of services;
- two user interfaces to consume the Web Services.

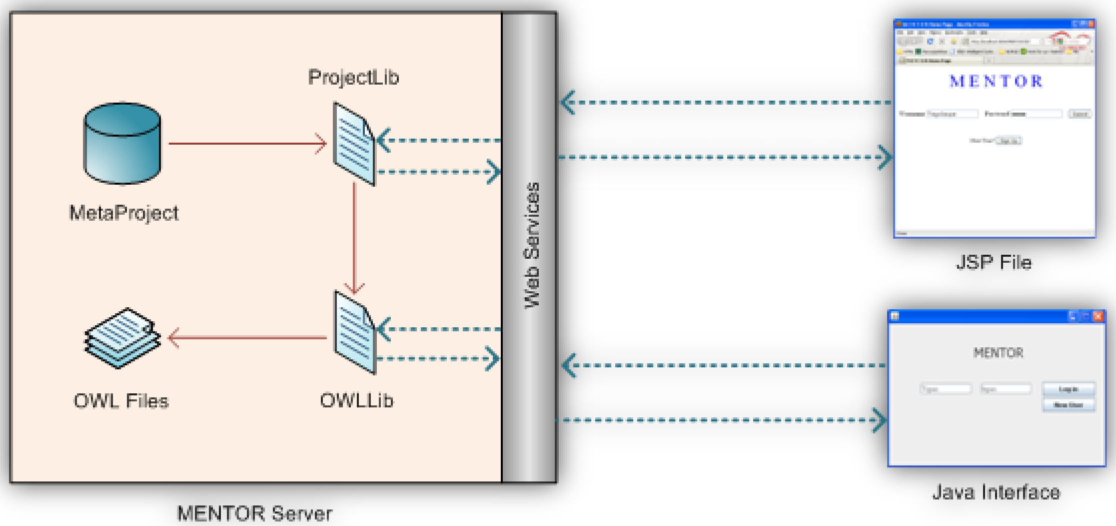


Fig. 4.11: Project Architecture

With this architecture users are able to create projects or connect to existing ones, and run all the six MENTOR steps to build domain reference ontologies. Other operations have been also developed and users are allowed to see a whole set of information of a given project. These operations are represented as a set of Java functions that compose a group of two Libraries. The Java Libraries are connected to the metaproject – that keeps the relevant data about users, projects, groups, and operations – and to the OWL files – to save the glossary, thesaurus, and the reference ontology. Web Services are useful in this architecture as they are responsible for giving worldwide users the opportunity to use these services (that are connected to the Java Libraries), and build themselves a user interface to execute MENTOR steps.

4.5.2.1. Metaproject

The Metaproject Project is a frame-based ontology, and is one of the most important elements that compose the Project Architecture. For being able to use Collaborative Protégé on MENTOR Methodology originally proposed steps *three*, *four* and *five* it is needed to run a Protégé server that uses the Metaproject for getting the information about the hosted projects, which users are defined, and which projects these users have permission to access [69].

A default Metaproject comes with Protégé, but one is allowed to edit its structure and change its elements – classes, properties, slots, etc – to make the best use of it. The original structure of the Metaproject project is shown on Fig. 4.12, and it is composed by classes like Project, User, Group, GroupOperation, and all this classes save instances for the related objects.

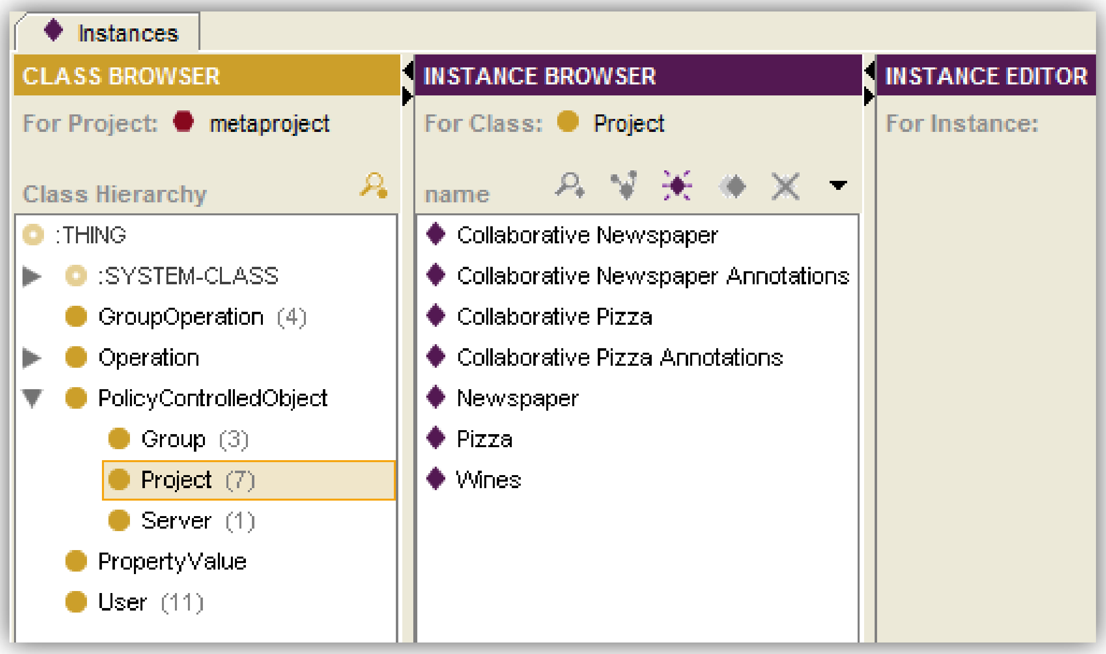


Fig. 4.12: Default Metaproject

These instances save the important data with the help of slots. For example, an instance of the *User* class saves the relevant information for a user, i.e., user name, email, password (encoded), user description, and the groups where he belongs, etc. An example of what was described here can be seen on Fig. 4.13.

After doing a study on what was necessary from the Metaproject to accomplish the requirements of MENTOR, it could be concluded that the default Metaproject – as it comes Protégé – was prepared for MENTOR, but some modifications had to be done to fully meet its needs.

Therefore, a new Metaproject structure that best suited MENTOR requirements was created by adding new classes that keep track of the MENTOR steps and the step status (if it started or finished), new instances, and new slots for saving crucial information. For example, in the class *Project* has been added a slot for storing information of MENTOR current step, a slot for storing the status of the current step, and a slot to store the name of the users that finished the operations of the current step. These and some other modifications have been made to Metaproject so it could become just right for MENTOR needs.

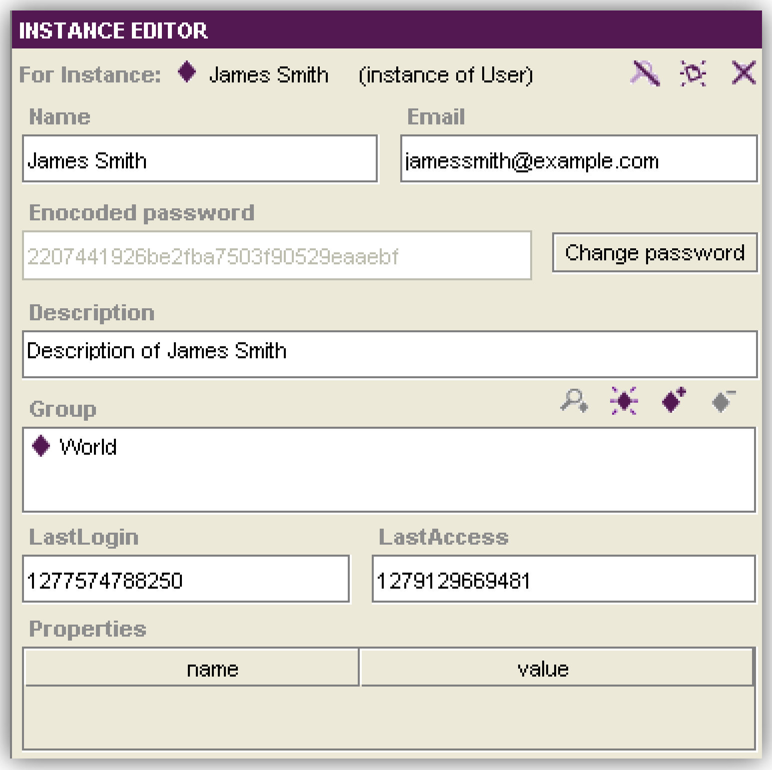


Fig. 4.13: User's instance

4.5.2.2. Java

The two existing libraries were developed in Java programming language due to three simple reasons: its runtime performance, for being an open source software, and because Protégé provides an Application Programming Interface (API) – the Protégé-OWL API – for the Web Ontology Language and RDF(S) [70]. This API is an open source Java library that provides classes and methods to create, load and save OWL files, and to query and manipulate OWL data models.

The reason for having two different libraries is mainly related to the organization of the Java code. The idea was to separate the Java functions that deal with the Metaproject file from the other functions that deal with the OWL files created for and by users as own glossaries, the domain glossary, the thesaurus, and the reference ontology.

These libraries are responsible for managing the execution of MENTOR steps. They guarantee the steps run in the right order and all the milestones of each of these steps are met properly.

4.5.2.3. Web Services

The decision of using this technology was purely to turn MENTOR into a Web Application. Therefore, instead of creating a software application to execute MENTOR and running in a local area, a step forward was taken to put MENTOR in a higher level, giving the possibility to join it wherever a person is.

A big advantage of using Web services is the fact that there is no need to install extra software or even a specific one to be able to use MENTOR. Everyone can consume the available services and give them the proper use – always respecting the defined Methodology – independently the operating system or programming language in use. It is also possible to create a Web solution and run MENTOR on a Web browser.

On the server side, the Web Services connect with the Java libraries to publish their services on the Web. They do not have logic inside, just receive parameters, call a library function and return a value given by the function. On the client side, it is even easier to understand, it works like a normal function that receives parameters and returns the desired value.

4.5.2.4. JavaServer Pages

The JavaServer Pages programming language (JSP) – as well as the application form made in Java represented in the Project Architecture – was developed to show how the available Web Services can be consumed and how to make the best use of them. A good advantage of using JSP to create a user interface to run MENTOR is that users can access the interface remotely, independently their location, and if there is the need to update the user interface they don't need to reinstall it over and over again since this update is made on the server side [71].

The whole process of creating a reference ontology (MENTOR methodology), can be controlled by the JSP page or other user interface that consume MENTOR Web Services. For every existing project on the "database" (Metaproject), it is possible access to its information during the whole process. Consequently, it is possible to check for a particular project which step is running at that moment, the step status, the users enrolled on the project and to which group they belong, and also accept or deny user requests, and change the step and the current step status.

The first step of MENTOR (Terminology Gathering), as well as the beginning of the second step (Glossary Building), is executed by consuming MENTOR Web Services, in this case by using the JSP page. After joining a project, users start by inserting their own terms and related definitions into the system. When all users have finished the terms and definitions revision, the project admin closes the step one and advances to step two. Step two begins by combining all the individual glossaries into a single file – the project Glossary. During the glossary building process, some semantic comparisons between user own glossaries are performed, and whenever a semantic collision between two terms or descriptions happen, an alert is shown on the final glossary.

For the other steps, the ones that have the need for real time interaction between project users are executed with Collaborative Protégé.

4.5.3. Detailed Process

The execution of the methodology is detailed in this subsection to better understand how it flows and how the information is processed on each step and the way it is passed from step to step. In the following are presented two figures with both sequence diagrams of the first and second steps of MENTOR Methodology.

4.5.3.1. Terminology Gathering Step

Every project starts by being created and the person responsible for its creation in the system is registered as the project admin. This user can do everything on the project, such as:

- Accept users;
- Remove users;
- Edit project information;
- Participate in the project contribution with knowledge (like a normal user);
- Start and finish a step from run;
- Moderate discussions.

Then, when all users are already connected to the project and ready to start contribution with their terms the project leader activates the first step. The generic representation of how the Terminology Gathering step is performed can be seen on Fig. 4.14

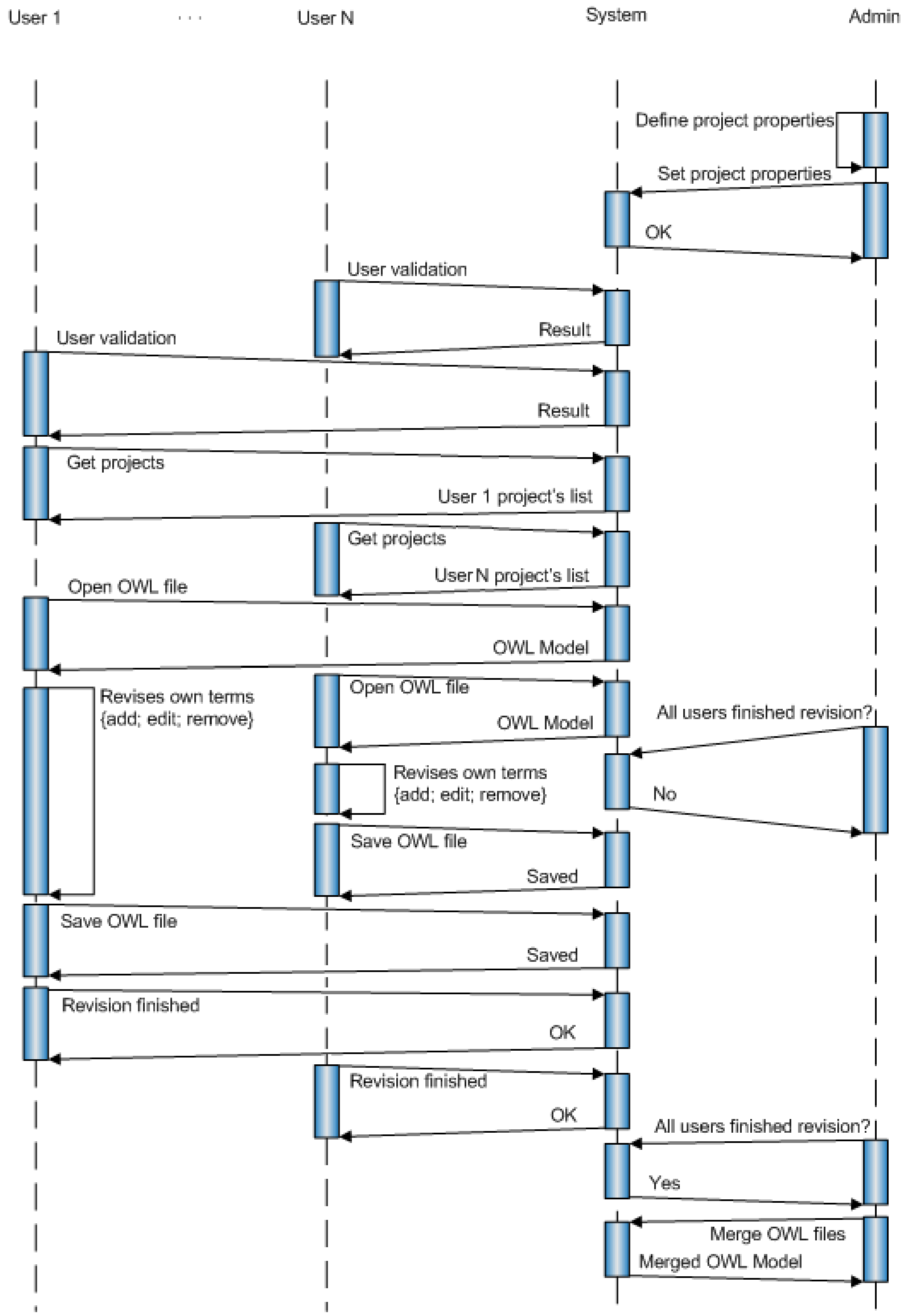


Fig. 4.14: Sequence Diagram - Terms Revision Step

The number of users or the amount of their contribution is not a relevant point for the system administration. Even with just one user it would be possible to build a domain ontology, the system permits from one to several users for doing so.

In this step there is no need of any specific QICM approach since there is no interaction between the project participants. All users revise their terms and definitions for themselves and when they finish they send a confirmation to the admin that proceeds to the next step when every member had finished the revision.

During the whole execution of the presented methodology the admin has the control on his hands of the project. He decides when to move on to the next steps and keeps track of everything that is happening. At the end of Terminology Gathering, OWL files from each user containing its terms and associated definitions are merged into a single file making at the same time an operation for finding collisions between terms and definitions. This helps the next step, where it is needed to find the best set of terms and definitions based on all the actors' inputs.

4.5.3.2. Glossary Building Step

Glossary Building step is one step that uses a QICM approach based on the Nominal Groups. Thus, it serves as an example for other steps that use QICM approaches. A representation of the progress of this step can be seen of Fig. 4.15.

The Glossary Building step starts by importing the whole set of terms that were introduced to the project in the previous step. By that moment, the admin presses the "start" button and then users can connect to Collaborative Protégé server and join the project that they are involved. The system administrator is also the responsible for controlling the proper execution of the current step. He/she must guarantee that everyone follows the Nominal Groups approach as stated on Fig. 4.2.

In this step, all the interactions between users pass through the system and only then reach the other users. It is not explicit on the diagram but when user 1 sends an idea (on a discussion period) or votes the other users get this information and are able to reply to him in the same way he did (sending a message to the system that then broadcasts for everyone). When the admin closes the step and analyses the results of it, he saves an OWL file with the glossary as a result of the process. In additional, automatically a set of mappings are saved on the MO about the mismatches found on the glossary building process. This glossary has all the terms and definitions approved by all the members of the project, representing in this way, the reference list of terms and definition of the addressed business domain.

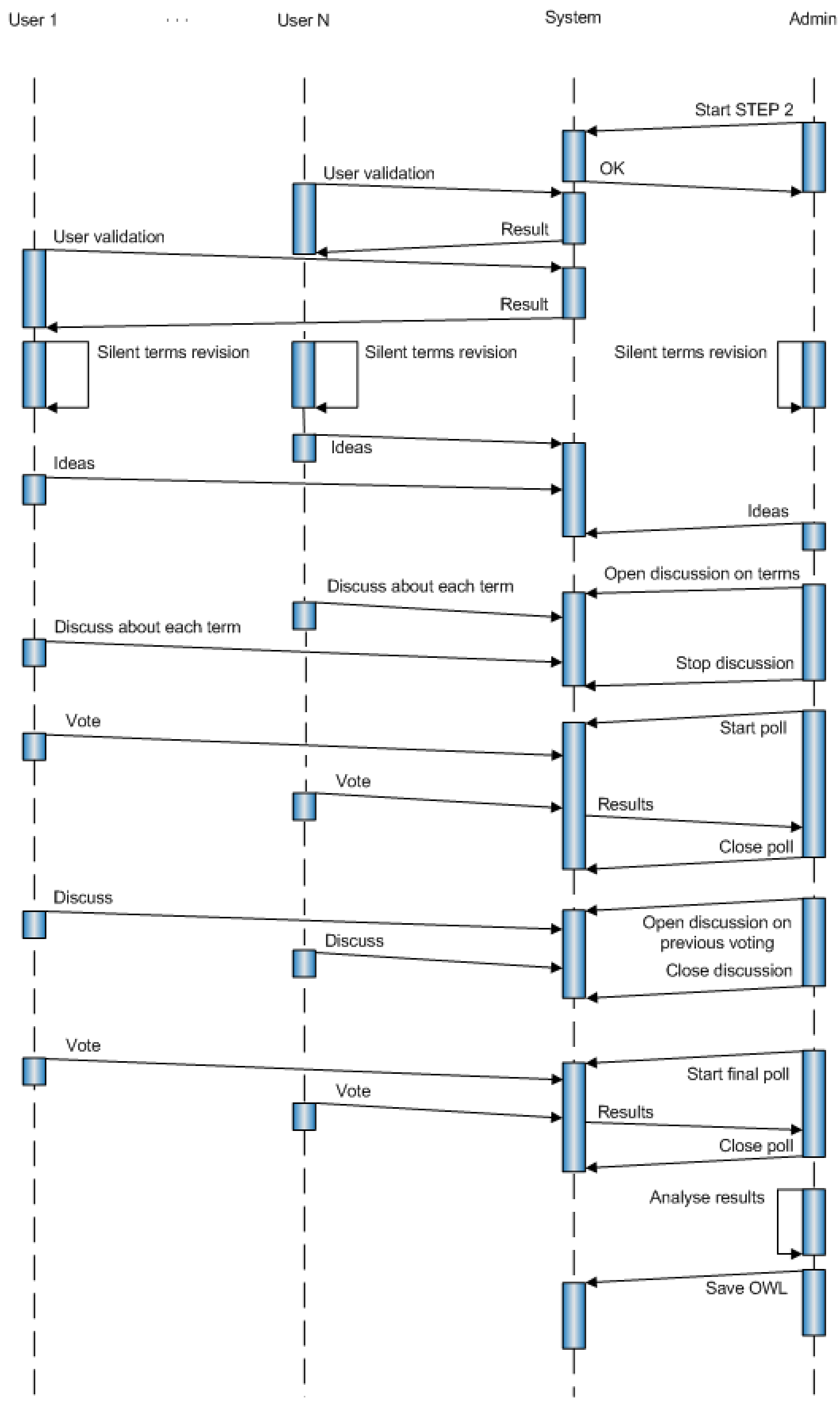


Fig. 4.15: Sequence Diagram - Glossary Building Step

4.5.3.3. All the other steps

For the next steps, it was adopted the Collaborative Protégé for helping in the collaborative part of the work. The implementation for this part is only related to the preparation of a protégé server with the “collab tab” active and adapted to follow the various QICM approaches proposed to follow on these steps (as presented in chapter 4).

The first step of the second phase, i.e. Ontologies Gathering is a similar step to the starting step of the current methodology. But instead of collecting the terms and definitions it gets the collection of ontologies, one per member, in their raw format. It is a step that, as the Terminology Gathering, where no interactions are needed between members since all the work is done alone and refers to each one’s ontologies.

The further steps, specifically Thesaurus Building Cycle and Ontology Harmonization (Ontology’s Taxonomy Harmonization Cycle and Ontology’s Contents Harmonization Cycle), are similar to the Glossary Building step in a way that both of them involve interaction (group discussions and voting sessions) between the project members. All of them are executed with the help of Collaborative Protégé, just like the second step.

The Thesaurus Building Cycle step begins when the administrator launches the server with the terms from the glossary. The process is to create a hierarchical structure among these terms to offer a better understanding of the domain based only on the semantics side. The final product obtained here is a thesaurus, accorded by everyone including the project administrator.

The last steps in which discussions and voting processes are involved are the ones that define the taxonomy and contents harmonization for the final ontology. They are based on the Delphi method, which means several rounds will be performed until a fine result in found. The output of this step is the reference ontology.

At the end, and if everything went just like specified in the methodology, the project admin could close the project and analyses the final results of the project. Subsequently the desired domain ontology is created and it represents the domain initially defined. Every member is supposed to feel identified with the reference ontology.

5. Demonstrator Testing and Hypothesis Validation

The architecture presented on the previous chapter (Fig. 4.11) was implemented according to all established parameters, and its results are shown in this chapter. First, for better understanding of the methodology for collaborative enterprise reference ontology building, it is presented an example about bolt suppliers. Afterwards, the dissemination made about the topic of this dissertation is explained as well as how the hypothesis defined in section 1.5 is validated.

5.1. Methodology Developing Demonstration

The process of choosing a “bolt” supplier by a mechanical engineer or designer, very often brings interoperability issues. Suppliers usually define proprietary nomenclatures for their products and its associated knowledge representation. Problems persist although standardization bodies developed and proposed several standards focused in bolt specifications. Thus, the need to align product data and knowledge emerged as a priority to solve the dilemma. By using MENTOR these interoperability problems will disappear.

The data used on this demonstration was based on the data used in the MENTOR – Methodology for Enterprise Reference Ontology Development publication [32].

When users connect to MENTOR, assuming they are using a browser to consume the Web services, they will find a *Login* page where, if already registered, they can access to their private area. If they are not registered, it is possible to create a new user by adding all the necessary information to the system. The *Log In* page can be seen on Fig. 5.1.

Fig. 5.1: Log In page

When logged in, users have access to their private area. In this area they can see their own information about the projects they are enrolled, and the ones they are not.. Here is also possible to create a new project by simply entering the project's name.

Assuming that users Henry and George, bolt suppliers, are new in the system and want to start a new project for having a common understanding of their business, with the name "Bolt Suppliers". Both users made already their registration and are now ready to create the project. Henry is chosen to be the project administrator but also contributes with terms and definitions for it, and George is a simple user that uploads his own terms and definitions. To create the project, Henry just needs to type *Bolt_Suppliers* on the *Project Name* textbox, press the *Create button* and the new project is created on the Metaproject file. In Fig. 5.2 is shown the moment of *Bolt_Suppliers project* creation by the defined administrator.

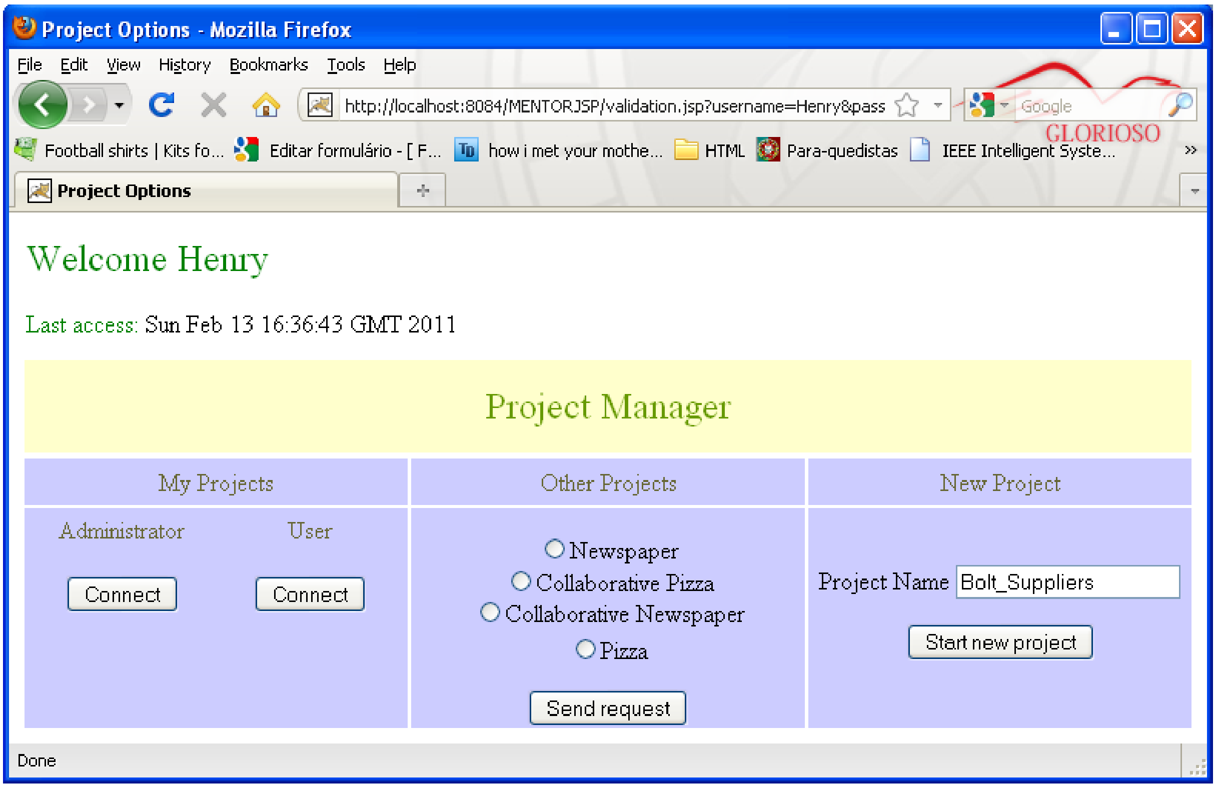


Fig. 5.2: Creation of a new Project

Project *Bolt_Suppliers* is now available for other users, and it is possible to see it appears on George's private area, marked as *1* on Fig. 5.3. To connect to the project, George has to send first a request to *Bolt_Suppliers* administrator, and if the request is accepted George is then ready to participate in the project. As it is shown on Fig. 5.3, represented by *2*, it is seen that Henry accepted the request for participating in *Bolt_Suppliers* project.

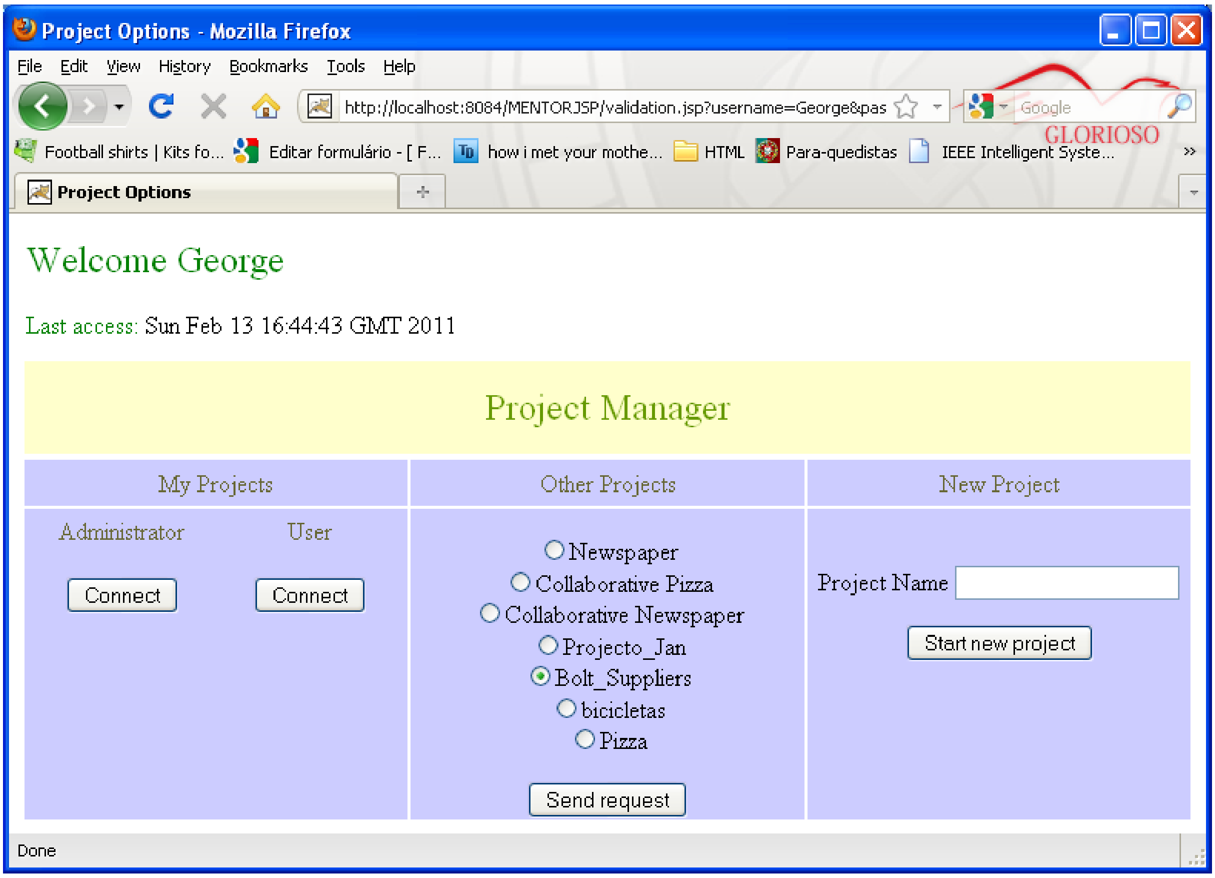


Fig. 5.3: 1 - Project send request; 2 - User projects

Both users are now ready to start contributing with their own terms and definitions. For the current example, Henry and George will contribute with the following terms and definitions. To start the project, the administrator has to go to the administration panel of *Bolt_Suppliers* and change the status from *Not_Started* to *Started* and the Terminology Gathering step just had started. Done this, it is now possible to start importing their terms and definitions. The used terms with their definitions is presented on Table 5.1 for each user.

Term		Definition
Henry	Bolt	Headed fasteners having external threads that meet an exacting, uniform bolt thread specification (such as M, MJ, UN, UNR, and UNJ) such that they can accept a no tapered nut
	s	Dimension across flats in a hexagonal head
	Pitch	The axial distance between a point on a thread flank and the equivalent point on the immediately adjacent and corresponding flank
	Tolerance	Permissible limits of variation that a measure can fall within; determined by the inspection phase after manufacture of the component
George	Bolt	Term used for a threaded fastener, with a head, designed to be used in conjunction with a nut
	Flat with	Diameter across the flats of the bolt's head. It is also the size of wrench to use
	Pitch	Distance between similar points on adjacent threads, property of a thread that distinguishes and enables the interconnection with complementary surfaces
	Tolerance	Allowable deviation from a nominal or specified dimension, determining maximum and minimum material condition

Table 5.1: Terms and Definitions

Each user introduce in the system the terms and definitions they use. The terms shown on Table 5.1 are just a small example. The list of terms could be much bigger than the represented one. But the set of terms presented here is enough for the demonstration.

When users finish introducing the terms, they must select the *Finished Revision* option to alert the administrator they are ready to go to the next step. On Fig. 5.4 it is possible to see the terms introduced by the user Henry.

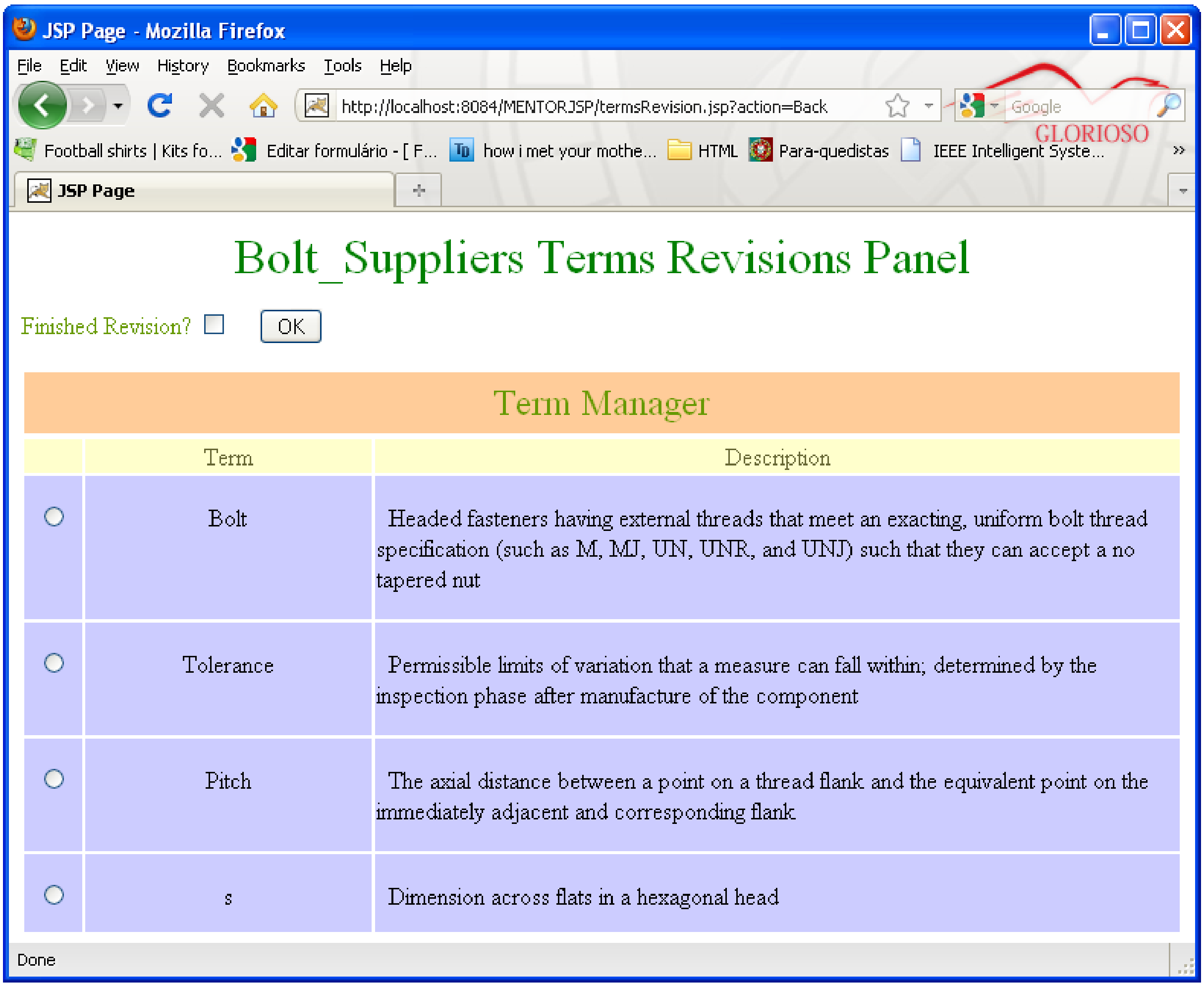


Fig. 5.4: Terms Revision panel

When a user finishes revising its terms and corresponding definitions, an OWL file is stored in the server side for keeping the information for further steps.

Then when Henry confirms that everyone finished revising terms, he can set the status of the *Terminology Gathering* to *Finished*. It is now time to proceed to the next step, the *Glossary Building*. But before passing to the next step it is needed to merge all OWL files crested on the previous step into a single OWL file, relating each term to its creator. In this single OWL file are stated the collisions that exist between terms of different users. For example terms with the same name or terms where the definition is equal. On the presented case the terms *Bolt* and *Major diameter* from Henry will collide with the terms *Bolt* and *Major diameter* from George, respectively. To prove this, Fig. 5.5 shows the OWL file created in the end of Terminology Gathering step and for each term it is saved the information of: *Name*, *Description*, *Creation Date*, *Author*, and *Collisions*. On the field *hasCollisions* it can be seen a collision between Henry's term Bolt and George's term Bolt.

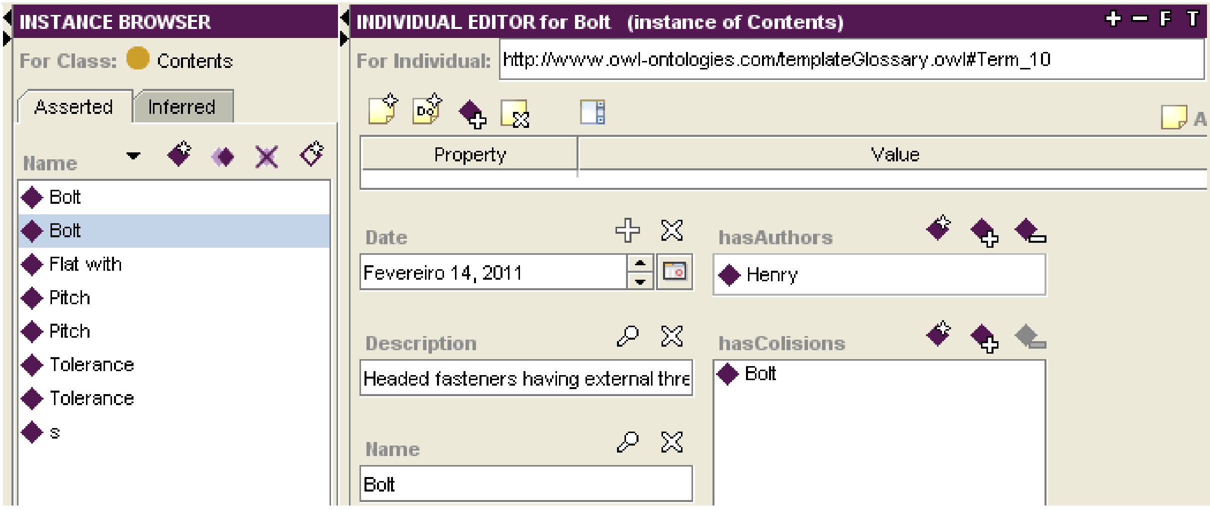


Fig. 5.5: OWL file generated in the end of Terminology Gathering

The Glossary Building step starts the same way as the previous one, i.e. the administrator goes to the project administration page and sets the status to *Started*. Then, as it was said before, all the steps that come after the first step use Collaborative Protégé toll. For this purpose, on the *Bolt_Suppliers* project must be registered in the Protégé server running on the server side.

Both users connect then to the server and join the project they are working on, in this case the *Bolt_Suppliers* project is available for Henry and George. Fig. 5.6 states the moment where Henry joined the project as a normal user (since he has privileges for joining it as Admin to keep track of the project flow as well as guiding the intervenients in the approach to follow) and starts searching the functionalities he can use with Collaborative Protégé. Both Henry and George have access to a set of functions in this tool. They can see the terms they have introduced in the previous step with all its information, not only their own terms but also the terms the others introduced, they are able to use a Collaboration panel with options like Chat, Notes, Track Changes, and others. In the chat option Henry can see that George is already connected and, when Henry (with the administrator privileges) puts the step status on *Started*, they can start building the glossary.

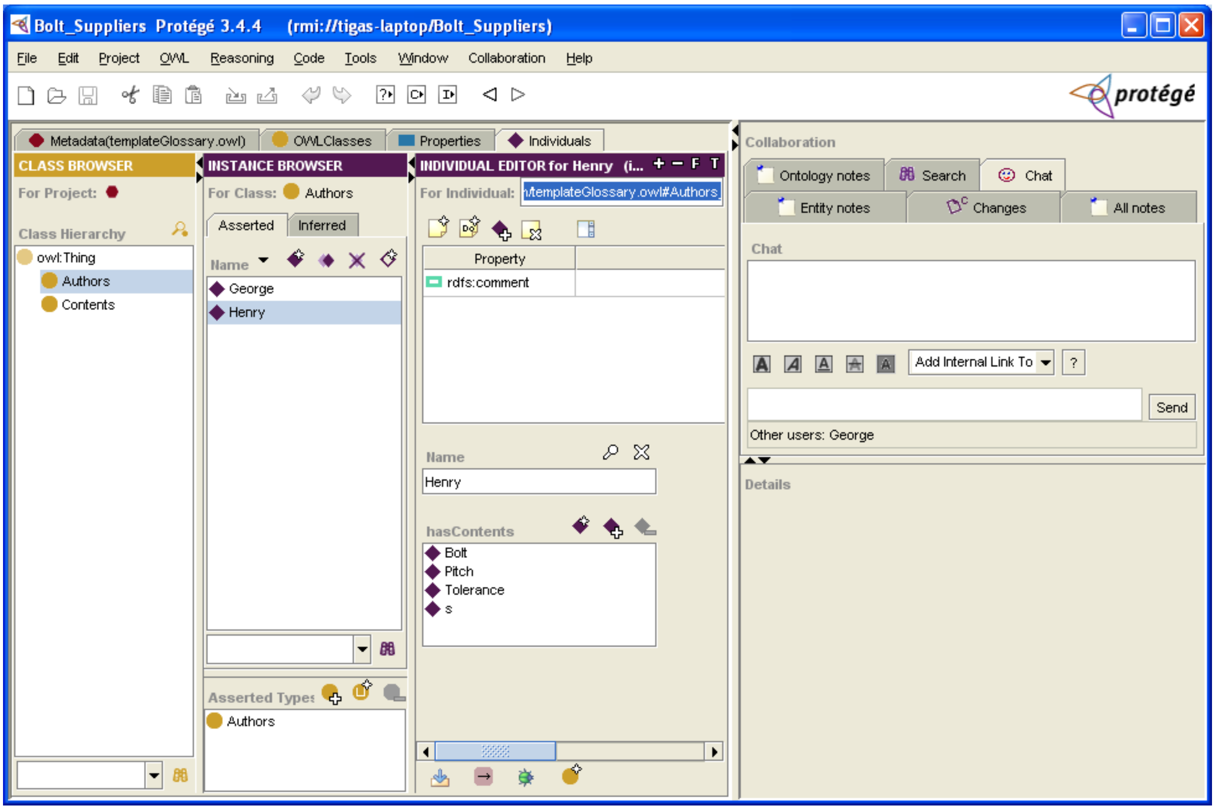


Fig. 5.6: Beginning of Glossary Building step

The first step of Nominal Groups tells the users to silently revise all the terms. This is what Henry and George start by doing. They revise the six existing terms and keep the comments for them. When they finish the revision, they can proceed to the record of ideas and so they use the chat to send everyone's ideas and all ideas are saved for clarification. Next a discussion takes place between the members to clarify what they think about the ideas recorded before. After the discussion period it is time for a voting session, where everyone votes if the changes to make. Changes like changing a term name or definition, when two terms collide it can be decided to keep just one of them or even create a new one and delete the other two, etc. When they finished voting they start discussing about the results of it to take new ideas about the preliminary glossary. At last, users vote again on the list of terms for finally close this step and get the glossary.

In the end only four terms resisted, since users noticed that from the six there were three pairs of two terms that were similar to each other, even in the name or in the description. The final result can be seen on Table 5.2.

	Term (Henry)	Term (George)	Term Adopted	Definition
Glossary	Bolt	Bolt	**Bolt**	Headed fasteners having external threads that meet an exacting, uniform bolt thread specification (such as M, MJ, UN, UNR, and UNJ) such that they can accept a no tapered nut
	s	Flat with	**Nominal diameter**	Dimension across flats in a hexagonal head
	Pitch	Pitch	**Pitch**	The axial distance between a point on a thread flank and the equivalent point on the immediately adjacent and corresponding flank. This property enables the interconnection with complementary surfaces
	Tolerance	Tolerance	**Tolerance**	Interval of values of allowable deviation from a nominal or specified dimension

Table 5.2: Glossary Building result

The final Glossary can be seen on Fig. 5.7.

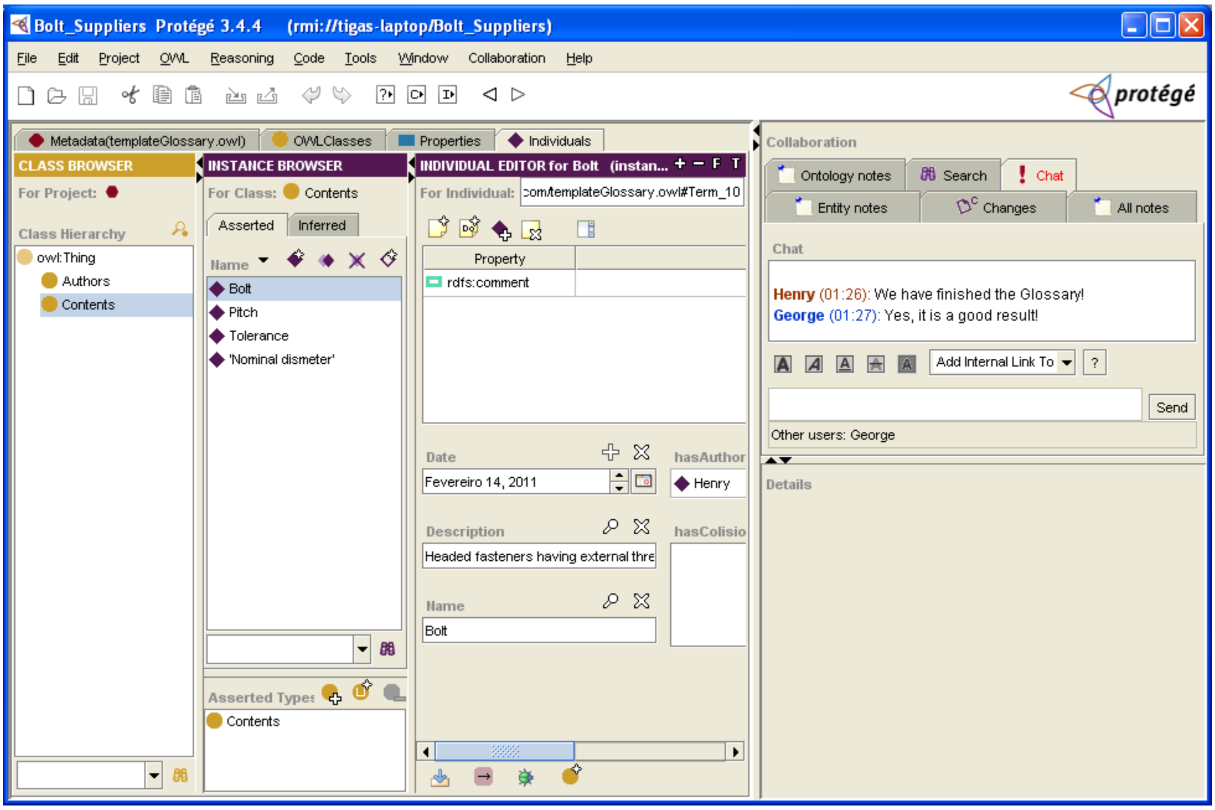


Fig. 5.7: Glossary obtained in the end of the second step

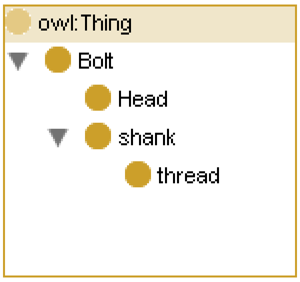
After all done on the Glossary Building step, the project administrator puts the current status to *Finished* and the Glossary is finally saved. With the glossary finished, the semantic mismatches found on the project are also saved in the Mediator Ontology.

So then Henry can proceed to the Thesaurus Building Cycle step with the terms defined in the glossary. In this step, the Thesaurus Building Cycle, the terms will pass from a list with no relations between each other to a hierarchical structure relating each term with the others.

Like in the other steps, the process is the same for all the steps that come next. Project administrator changes the step status from *Not_Started* to *Started* and later then when all users have finished their actions on that step Henry will change from it again to *Finished* so it is possible to extract the results and proceed to the next step.

The Thesaurus Building Cycle step uses the Delphi approach and so it ends up in a simple process to build the thesaurus. This method is composed by three rounds, as it was shown in section 3.7.4. With all the terms defined from the previous step, Henry and George start by expressing their opinions about their initial ideas for the structure. Then with these ideas a questionnaire is built and both users answer to it. The third step is to re-rank their initial statements with the results of the questionnaire. This is an iterative process and so it must run until a structure that satisfied everyone is met. With Collaborative Protégé users use again the collaborative panel for sharing ideas and meeting the satisfactory result. When the structure is found, the administrator closes the step and a thesaurus is save as an output of this step. No figure is shown since it uses again the Collaborative Protégé tool and it was already shown in the Glossary Building step how to use it, as well as the interaction among users.

And the first phase of the MENTOR Methodology is finished. The terms organisation and structuring is was the focus of the first phase and now it is time to start the second phase that is more focused on the ontology side. When Henry decides to give the *Start* status on the Ontologies Gathering step, both Henry and George start by uploading to the system their own ontologies. Like in the first step on the first phase, here they can revise their ontologies and make changes if necessary. The proprietary ontologies taxonomy from each user is found on Fig. 5.8.



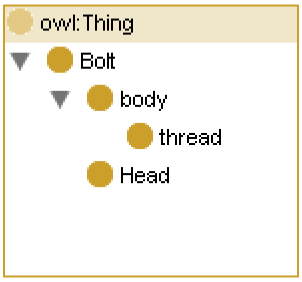


Fig. 5.8: Left-side - Henry's Ontology taxonomy; Right-side - George's Ontology taxonomy

When they confirm they finished revising their ontologies the administrator one closes the step and takes the system to the next step, Ontologies Harmonization.

This step, as stated on Fig. 4.1, is divided into two mini-steps. These steps are run both the same way, just the focus of each is different. One focuses more on ontologies taxonomy side and the other on the contents side.

After starting the Ontologies Harmonisations steps, either for taxonomy or contents way, users will start doing a cyclic process for defining the reference ontology structure. They start by expressing their opinions about the reference ontology (structure or contents). Then with the results of round 1 they build a questionnaire and answer to it. A third round implies that every user re-rank the initial statements and if needed reformulate the reference ontology. When this step is finished a Reference Ontology is found and it is the output of the current step. The output for the *Bolt_Suppliers* project from Henry and George is shown below, on Fig. 5.9.

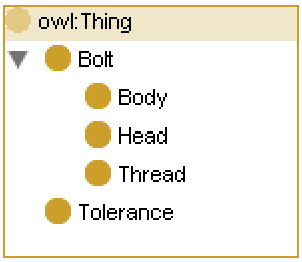


Fig. 5.9: *Bolt_Suppliers* Reference Ontology taxonomy

Bold_Suppliers project finished just like it was expected, i.e. with the construction of a reference ontology common to a community, in this case a community of bolt suppliers. In the last step it is supposed to build the mappings between the reference ontology and the proprietary ontologies from each participant. So every participant can contribute to a common understanding without the need to change the semantics of his “local” business. These can also be extended to international enterprise communities to contribute to a network where no one needs to lose its culture or change its natural language. The mappings take care of it.

5.2. Dissemination Executed and Hypothesis Validation

Regarding the hypothesis validation, has demonstrated in section 5.1, by designing a methodology that creates a common knowledge base to a group of enterprises it was possible to allow the communications within enterprises for building a reference ontology. The methodology was complemented with qualitative information collection methods to facilitate the organizations of group discussions. With these methodology users shared their terms and definitions building a single glossary that suited each of them, built thesaurus which is a structured way of representing the terms, and built a reference ontology beginning from their own ontologies.

Qualitative information collection methods, more precisely the Nominal Groups and Delphi approach which were used in some steps, improved the efficiency of interactions between the intervenients in the project. They represent a structured way for managing the opinions expressing, serial discussions, and final voting where participants were involved.

It is possible to create a common understanding between enterprises who originally do not share the same concepts or ideas of each concept for the same domain by applying this methodology.

For intentional purposes of the research results of this dissertation, a scientific publication was accepted in the International Conference on Intelligent Systems, IEEE, from 7th to 9th of July 2010 in London – United Kingdom, and it was published on the proceedings of the conference:

- Sarraipa J., Jardim-Gonçalves R., Gaspar T., Steiger-Garção A., Collaborative Ontology Building using Qualitative Information Collection Methods, Accepted In: International Conference on Intelligent Systems, IEEE. Jul 7-9, London, United Kingdom, (2010).

6. CONCLUSIONS AND FUTURE WORK

To better respond to the requirements of the market enterprises are changing the way they do business in order to survive in such a demanding world. SMEs started to realize if that the small markets do not bring big benefits for them and do not allow them to grow as fast as they wish. To compete with large enterprises SMEs must seek for collaboration between each other to act as a bigger one. But collaboration does not seem a easy thing to do and some enterprises are afraid of doing so by the possibility of needing to change the way they are used to work. Interoperability is the keyword for this enterprise collaboration but it seems to be a big barrier instead.

When working together, enterprises need to communicate to make each other understand each one’s ideas. But most of the time this communications are not well succeeded due to semantic interoperability problems.

MENTOR methodology appears with the idea of creating a common understanding between enterprises that operate in the same domain. The methodology is composed by six steps allowing enterprises to keep their information models while they create a common semantic model that feats every member of the network. MENTOR was already prototyped and tested by the *funStep* initiative under the *INNOVAfun* project (www.funstep.org) in their furniture reference ontology building. The thesaurus and the reference ontology built in such process have been used for testing and consolidation of semantic enrichment for the ISO 10303-236 (Product data representation and exchange standard) model. With such work, a carefully validation based on independent reviews, followed by a consequent improvement was carried out. Additionally, to these tests in furniture area, Sarraipa et al. in [53], also tested MENTOR through a small case-study related to the choice of a bolt in the mechanical area. All of these studies helped authors to identify that tacit knowledge acquisition process isn’t easy and clear.

Sometimes discussions were too long compromising the effectiveness of MENTOR. Thus, this dissertation came with the proposal for using MENTOR as a collaborative methodology for ontology building enriched with QICM. The implementation of such enrichment process has proven to be a good evolution for MENTOR solving the problematic of human interaction. It reduces the time of group meeting and improves the results that come out of it.

In this dissertation a solution was implemented to take advantage of this methodology. The used technologies helped building a platform where enterprises can connect, create a project and start building a reference ontology by passing through by six steps. Some steps were implemented with the use of an external toll, Collaborative Protégé, for helping in the steps which require interaction between enterprises.

The presented methodology with QICM enrichment has proven to be a good approach for enterprise reference ontology building.

6.1. Future Work

As for future work, a few things regarding the implementation of the solution can be refined and others can be introduced. The idea is to have a single solution that executes MENTOR methodology from the beginning to the end, without using external tools such as Collaborative Protégé. This way, users do not have the need of signing in and out every time the system goes from one step to the next one. In this dissertation the system started to grow from step one, and in the future it is supposed to have all six steps working together in the same solution. It means that it this solution must offer a complete set of services that allow all the operations used with Collaborative Protégé, like the collaboration panel which users can use for communication, for making comments, tracking changes, and some other options.

It is good to create a platform able to fully operate with OWL files, i.e. capable of showing the classes, and their properties organising them into a structure like it is possible to see on Protégé. Then when enterprises want to import and revise their ontologies, and later create the reference one, to see its structure.

7. REFERENCES

[1] Silva, J.; Monteiro, A.; Sarraipa, J.; Gonçalves, R.: Towards Ontology Harmonization of Mechanical Manufacture Constraints Through PLC. In: 20th International Congress of Mechanical Engineering (COBEM 2009), pp 15 - 20, (2009).

[2] Jardim-Goncalves, R., Grilo, A., & Steiger, A. (2006). Challenging the interoperability between computers in industry with MDA and SOA. Computers in Industry, 57(8–9), 679–689.

[3] Panetto, H., Jardim-Gonçalves, R., & Pereira, C. (2006). EManufacturing and web-based technology for intelligent manufacturing and networked enterprise. Journal of Intelligent Manufacturing,17(6), 639–640.

[4] White W, O'Connor A, Rowe B, "Economic Impact of Inadequate Infrastructure for Supply Chain Integration", NIST Planning Report 04-2, Gaithersburg, MD: National Institute of Standards and Technology, 2004.

[5] Gruber, T.: Toward Principles for the Design of Ontologies Used for Knowledge Sharing. Originally in N. Guarino and R. Poli, (Eds.), International Workshop on Formal Ontology, Padova, Italy. Revised August 1993. Published in International Journal of Human-Computer Studies, Volume 43, Issue 5-6 Nov/Dec. 1995, pp 907-928, (1993).

[6] Charalabidis, Y; Gionis, G; Hermann, K; Martinez, K.: Enterprise Interoperability: Research Roadmap. Update Version 5.0 (2008).

[7] Commission of the European Communities. Communication from the Commission to the Council, The European Parliament, The European Economic and Social Committee and the Committee of the Regions "i2010 – A European Information Society for growth and employment" - {SEC(2005) 717}; Brussels. (2005)

[8] Camarinha-Matos L.: Scientific Research Methodologies and Techniques - Unit 2: Scientific Method, PhD Program in Electrical and Cumputer Engineering (2010).

[9] Oxford Dictionaries: http://oxforddictionaries.com/view/entry/m_en_gb0447820#m_en_gb0447820.

[10] Bellinger, G.; Castro, D.; Mills, A.: Data, Information, Knowledge and Wisdom. Retrieved from the web in January 2010: http://systems-thinking.org/dikw/dikw.htm.

[11] Sarraipa, J.; Jardim-Goncalves, R. and Steiger-Garcao, A. (2010). MENTOR: an enabler for interoperable intelligent systems, International Journal of General Systems, 39: 5, 557 — 573, First published on: 13 May 2010 (iFirst)

[12] Nonaka, I.; Konno, N.; Toyama, R.: Emergence of "Ba". In: Knowledge Emergence: Social, Technical, and Evolutionary Dimensions of Knowledge Creation, pp 13 - 29, Oxford University Press US, (2001).

[13] Auer, S.; Dietzold, S.; Riechert, T.: OntoWiki – A Tool for Social, Semantic Collaboration. In: The Semantic Web - ISWC 2006, ISBN: 978-3-540-49029-6, pp 736 - 749, (2006).

[14] Sarraipa, J.; Jardim-Gonçalves, R.: Semantic Adaptability for Systems Interoperability.

[15] Sarraipa, J.: Uma solução para a Interoperabilidade Semantica em ambientes globais de negócios. Master Thesis, (2004).

[16] *Wikipedia:* http://en.wikipedia.org/wiki/Terminology#Typology, retrieved on January 2011.

[17] *Britannica:* http://www.britannica.com/EBchecked/topic/162272/dictionary, retrieved on December 2010.

[18] Jardim-Goncalves, R.; Sarraipa, J.; Agostinho, C.; and Panetto, H. (2009). Knowledge Framework for Intelligent Manufacturing Systems, International Journal of Intelligent Manufacturing, 2009.

[19] CMU (Carnegie Mellon University)-Domain Dictionary. (2007). Accessed 30 Sep 2008. Available on: www.sei.cmu.edu/domainengineering/domain_dict.html

[20] Suryanto, Hendra and Paul Compton. "Learning classification taxonomies from a classification knowledge based system." University of Karlsruhe; "Defining 'Taxonomy'," Straights Knowledge website.

[21] Malone, Joseph L. (1988). *The Science of Linguistics in the Art of Translation: Some Tools from Linguistics for the Analysis and Practice of Translation,* pp. 112.

[22] Thesaurus.com: http://thesaurus.com/browse/thesaurus, retrieved on February 2011.

[23] Soergel. Thesauri in Digital Libraries, JCDL 2002, Portland, OR, 2002-7-14 (2002).

[24] Brewster, C.; O'Hara, K.; Fuller, S.; Wilks, Y.; Franconi, E.; Musen, M.; Ellman, J.; Shum, S.: Knowledge Representation with Ontologies: The Present and Future. In: IEEE Intelligent Systems, vol. 19, no. 1, pp. 72-81, doi:10.1109/MIS.2004.1265889, (2004).

[25] IEEE Standard Computer Dictionary, In: A Compilation of IEEE Standard Computer Glossaries. (1990)
[26] Chen, D.; Doumeingts, G.; Vernadat, F.; Architectures for enterprise integration and interoperability: Past, present and future, In: Computers in Industry, v.59 n.7, p.647-659 (2008).
[27] Panetto, H.: Towards a classification framework for interoperability of enterprise applications. In: International Journal of Computer Integrated Manufacturing, Vol. 20, No. 8, December 2007, 727 – 740 (2007).
[28] Paolo Bouquet, J´erˆome Euzenat, Enrico Franconi, Luciano Serafini, Giorgos Stamou, and Sergio Tessaris.D2.2.1 specification of a common framework for characterizing alignment. Technical report, University of Trento, 2004.
[29] Leo Obrst. Ontologies for semantically interoperable systems. In CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management, pages 366–369, New York, NY,USA, 2003. ACM Press.
[30] Ayadi, N.; Ahmed, M.; Pollet, Y.: Ontology-Based Meta-Model for Semantically Interoperable Systems. In: Proceeding of the 8th International Conference on Information Integration and Web-based Application & Services (iiWAS 2006), pp 413 - 422, (2006).
[31] Gonçalves, R.; Olavo, R.; Sarraipa, J.; Steiger-Garção, A.: Ontology-Based Framework for Enhanced Interoperability in Networked Industrial Environments. In: Proceedings of the 11th IFAC Symposium on Information Control Problems in Manufacturing (INCOM 2004), (2004).
[32] Sarraipa, J.; Silva, J. P.; Jardim-Gonçalves, R.; Monteiro, A.: MENTOR – Methodology for Enterprise Reference Ontology Development. In: Intelligent Systems, 2008. (IS '08). 4th International IEEE Conference, pp 6-32 - 6-40, (2008)
[33] Ehrig, M., Sure, Y.: Ontology Mapping: An Integrated Approach. In: 1st European Semantic Web Symposium. LNCS , vol. 3053, pp. 76--91. Springer, Heraklion, Greece (2004)
[34] Liping, Z., Guangyao, L., Yongquan, L., Jing, S.: Design of Ontology Mapping Framework and Improvement of Similarity Computation. Journal of Systems Engineering and Electronics 18(3), 641--645 (2007).
[35] Chungoora, N.; Young, R.I.M.: Ontology Mapping to Support Semantic Interoperability in Product Design and Manufacture. In: Proceedings of the First International Workshop on Model Driven Interoperability for Sustainable Information Systems (MDISIS'08), pp 1 - 15, (2008).
[36] INTEROP NoE. Deliverable MoMo.2 - TG MoMo Roadmap. InterOP, 2005.
[37] Agostinho C, Sarraipa J, D'Antonio F, et al. Enhancing STEP-based interoperabity using model morphisms. In: 3rd International Conference on Interoperability for Enterprise Software and Applications (I-ESA 2007).
[38] Delgado M., Agostinho C., Malo P., Jardim-Gonçalves R., (2006) A framework for STEP-based harmonization of conceptual models. 3rd International IEEE Conference on Intelligent Systems (IEEE-IS'06), Wetsminster, Jul 4-6, 2006.
[39] INTEROP NoE. Deliverable D.A3.3 - Semantic Annotation language and tool for Information and Business Processes. InterOP, 2006.
[40] Gruber, T.: Ontology. In: Encyclopedia of Database Systems, Ling Liu and M. Tamer Özsu (Eds.), (2009).
[41] Gruber, T.: Ontology of Fonksonomy: A Mash-up of Apples and Oranges. In: Int'l Journal on Semantic Web & Information Systems, 3(2), 2007, (2005)
[42] Missikoff, M.; Boudjlida, N.; Lenzerini, M.; Jeusfeld, M.; Johannesson, P.; et al. D8.1 - State of the art and state of the practice including initial possible research orientations, Interoperability Research for Networked Enterprises Applications and Software Network of Excellence, n° IST 508 - 011, (2004).
[43] Gomez-Perez, A.; Rojas, D.: Ontological Reengineering for Reuse. In : 11th European Workshop on Knowledge Acquisition, Modelling and Management. ISBN:3-540-66044-5. (1999).
[44] Euzenat, J. (1996). Corporative memory through cooperative creation of knowledge bases and hyper-documents. Proceedings 10th KAW, Banff (Canada).
Rene Robin, C. R.; Uma, G. V.: A Novel Algorithm for Fully Automated Ontology Merging
[45] Using Hybrid Strategy. In: European Journal of Scientific Research, ISSN 1450-216X Vol.47 No.1 (2010), pp.074-081. (2010).
[46] Lin, HN.; Tseng, SS.; Weng, JF.; Lin, HY.; Su, JM.: An Iterative, Collaborative Ontology Construction Scheme. In: Second International Conference on Innovative Computing, Information and Control (ICICIC 2007), ISBN: 0-7695-2882-1, pp. 150, (2007).
[47] Holsapple, C.W.; Joshi, K.D.: A Collaborative Approach to Ontology Design. In: Communications of the ACM, ISSN: 0001-0782, Vol. 45, pp. 42 - 47, (2002).
[48] Collaborative Protégé, http://protegewiki.stanford.edu/index.php/Collaborative_Protege, (2009).

[49] Poler, R.; Tomás, J.; Velardi, P.; et al. D10.1 - Interoperability Glossary, Interoperability Research for Networked Enterprises Applications and Software Network of Excellence, n° IST 508-011, (2004).
[50] Auer, S.; Dietzold, S.; Riechert, T.: OntoWiki – A Tool for Social, Semantic Collaboration. In: The Semantic Web - ISWC 2006, ISBN: 978-3-540-49029-6, pp 736 - 749, (2006).
[51] Sarraipa, J.; Silva, J. P.; Jardim-Gonçalves, R.; Monteiro, A.: MENTOR – Methodology for Enterprise Reference Ontology Development. In: Intelligent Systems, 2008. (IS '08). 4th International IEEE Conference, pp 6-32 - 6-40, (2008).
[52] [11:48:46] João Sarraipa: Sarraipa, J.; Jardim-Goncalves, R. and Steiger-Garcao, A. (2010). MENTOR: an enabler for interoperable intelligent systems, International Journal of General Systems, 39: 5, 557 — 573, First published on: 13 May 2010 (iFirst).
[53] Jardim-Goncalves, R.; Silva, J. P.M.A.; Steiger-Garcao, A.; Monteiro, A. (2007). Framework for Enhanced Interoperability Through Ontological Harmonization of Enterprise Product Models in book Ontologies: A Handbook of Principles, Concepts and Applications. Integrated Series in Information Systems , Vol. 14-2007, XXIV, 915 p., 245 illus. ISBN: 978-0-387-37019-4.
[54] Sarraipa, J.; Zouggar, N.; Chen, D; Jardim-Goncalves, R. (2007). Annotation for Enterprise Information Management Traceability. In Proceedings of IDETC/CIE ASME (2007).
[55] Agostinho, C.; Sarraipa, J.; Goncalves, D.; Jardim-Goncalves, R. (2011). Tuple-based semantic and structural mapping for a sustainable interoperability. In the proceedings of DoCEIS'11 – 2nd Doctoral Conference on Computing Electrical and Industrial Systems, 21-23 February 2011 – Costa da Caparica, Lisbon – Portugal.
[56] Agostinho, C. Correia, F. and Jardim-Goncalves, R. (2010) Accepted In: 17th ISPE International Conference on Concurrent Engineering (CE 2010). Sep 6-10, Cracow, Poland.
[57] Sarraipa, J., Jardim-Goncalves, R. and Steiger-Garcao, A. (2010). MENTOR: an enabler for interoperable intelligent systems. International Journal of General Systems, Volume 39, Issue 5 July 2010 , pages 557 – 573.
[58] Borysowich, C; (2006). Using the Nominal Group Method to Drive a Team Decision. Retrieved from the web, at March 2010: http://it.toolbox.com/blogs/enterprise-solutions/using-the-nominal-group-method-to-drive-a-team-decision-11774.
[59] Sample, J. A.; (1984). Nominal Group Technique: An Alternative to Brainstorming, in: Journal of Extension, Volume 22, Number 2, ISSN 1077-5315, March 1984.
[60] Metaplan, in http://elearn.jku.at/wiki/index.php/Metaplan, read on March 2010.
[61] McBurney, D. H.; White, T. L.; (2007). "Nonexperimental Research, Part 2: Survey Research", in Resarch Methods (8th edition), Chapter 10, pp. 245-246, (2007).
[62] Linstone, H.; Turoff, M.; (1975). The Delphi Method: Techniques and Applications, Addison-Wesley, Reading, MA, (1975).
[63] About Java Technology, available from http://www.sun.com/java/about/, accessed on 19th July 2010.
[64] Introduction to Web Services, available from http://www.w3schools.com/webservices/ /ws_intro.asp, accessed on 21st July 2010.
[65] JavaServer Pages Technology, available from http://java.sun.com/products/jsp/, accessed on 22nd July 2010.
[66] Protégé, available from http://protege.stanford.edu/, accessed on 22nd July 2010.
[67] Protégé-OWL API Programmer's Guide http://protegewiki.stanford.edu/wiki/ProtegeOWL_API_ Programmers_Guide, accessed on 7th August 2010.
[68] Collaborative Protégé, from http://protegewiki.stanford.edu/index.php/Collaborative_Protege, accessed on December 2009.
[69] Configuring Protégé Metaproject, from http://www.enterprise-architecture.org/documentation/doc-administration/71-configure-metaproject, accessed on 25th July 2010.
[70] Protégé-OWL API, from http://protege.stanford.edu/plugins/owl/api/, accessed on 3rd August 2010.
[71] JavaServer Pages[tm] Technology – White Paper, from http://java.sun.com/products/jsp/ whitepaper.html, accessed on 6th August 2010.