U. PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Collaborative knowledge representation processes and techniques to support domain experts in conceptual modeling

Cristóvão Dinis Polido Sousa

Supervisor: Professor Doutor Antonio Lucas Soares

Programa Doutoral em Engenharia Informática

January, 2015

Faculdade de Engenharia da Universidade do Porto

Collaborative knowledge representation processes and techniques to support domain experts in conceptual modeling

Cristóvão Dinis Polido Sousa

Dissertation submitted to Faculdade de Engenharia da Universidade do Porto
to obtain the degree of

Doctor Philosophy in Informatics Engineering

President: Professor Doutor Eugénio de Oliveira
Referee: Professor Doutor Nuno Silva
Referee: Professora Doutora Rute Costa
Referee: Professora Doutora Cristina Ribeiro

January, 2015

Ao meu filho Dinis Ismael.

Abstract

This thesis is focused in designing the appropriate means to help domain experts produce a well-defined account of a shared understanding (i.e., precise and accurate domain representations), in order to promote knowledge management and facilitate knowledge sharing in networked environments. This challenge implies developing, reusing and maintaining common interpretations of domain relevant information in a systematic way. This encloses specific knowledge acquisition and model abstraction problems, here tackled in a twofold perspective, establishing theoretical and practical connections to both terminology and knowledge representations, underpinned by a socio-semantic approach based on semiotic principles.

In particular, here are discussed the challenges related to fostering and supporting the domain experts involvement in collaborative knowledge representation processes, driven by conceptualisation activities. The inherent (inter) subjectivity of the conceptualisation and its outcomes, together with its informality are the reasons for being an overlooked research topic in the literature.

Conceptualisation activities, from the perspective of domain experts, should provide the appropriate artefacts to address conceptual representations as pragmatic artefacts, whose validity and value are time, context and situation dependent. For that, a Design Science Research methodology was followed. It aims towards the design and evaluation of the IT artefacts that could enable this informal and domain specialist oriented approach to the development of conceptual representations. Accordingly, particular attention was given to the following issues:

- How to support conceptual relations elicitation? Relations elicitation revealed to be the major concern when performing conceptualisation activities, in particular if an informal perspective is followed. Moreover, the employed relations in the construction of a conceptual representation influence to which extent it can be reused.

- How to enable the domain experts to bypass the initial difficulties of starting a conceptualisation?

- How to find a tradeoff between the conceptual relations commitment with the real-world and the domain experts vision? This is crucial to endow conceptual representations with an highly reusability degree.

To cope with that, an holistic approach was designed and accommodated into a conceptualisation framework as an abstract artefact supporting conceptual representations construction in a 3-layered formalisation approach. At the end, the goal is to achieve computational and well-formed conceptual representations (i.e., compliant with a well-defined representation schema), without the need for domain experts to commit to any other formalism than those from their own technical field.

The conceptualisation framework and its constructs were implemented in a technological platform mediating the cycle of development and evaluation of the artefacts, following qualitative evaluation methods combined with quantitative measures, built in a comprehensive action-research approach.

Finally, the results showed that the specialists performance in concept and conceptual relations elicitation could be improved with the processes discussed in this thesis, which contributes to a refreshing view for the scientific body of knowledge in the field of ontology engineering and conceptual modeling.

Keywords: Collaborative Conceptualisation Process. Conceptual Graphs. Conceptual Relation Elicitation. Conceptual Representations. Competency Questions. Domain Experts. Knowledge Representation.

Resumo

Esta tese foca-se no desenho dos meios adequados para auxiliar os especialistas de um determinado domínio na definição precisa e rigorosa de representações de um entendimento comum, a fim de promover a gestão do conhecimento e facilitar a partilha de conhecimentos em ambientes de rede. Este desafio implica o desenvolvimento, reutilização e manutenção de interpretações comuns do conteúdo relevante do domínio, numa perspectiva sistemática, o que inclui problemas concretos de aquisição de conhecimento e representação de modelos de abstração aqui abordadas numa dupla perspectiva, estabelecendo ligações teóricas e práticas tanto para a terminologia como para a representação de conhecimento, apoiadas por uma abordagem sócio-semântica baseada em princípios da semiótica.

Em particular, são discutidos desafios relacionados com a promoção a apoio à participação dos especialistas nos processos colaborativos de representação do conhecimento, impulsionados pelas atividades de concetualização. Este é um tópico pouco abordado na literatura, dada a (inter) subjetividade inerente da concetualização e dos seus resultados, juntamente com a abordagem informal na produção dos mesmos.

As atividades de concetualização, a partir da perspectiva de especialistas, devem fornecer os artefatos apropriados para abordar representações conceptuais como artefatos pragmáticos, cuja validade e valor são dependentes do tempo, do contexto e situação. Para isso, foi seguida uma metodologia baseada em *Design Science Research*.

Esta metodologia visa apoiar todo o processo de investigação desde o desenho dos artefatos que visam apoiar os especialistas na representação de representações conceptuais até à sua avaliação. Assim, foi dada especial atenção às seguintes questões:

- Como apoiar a elicitação de relações conceptuais? A elicitação de relações revelou ser a maior preocupação na realização das atividades inerentes à concetualização, em particular quando é seguida uma perspectiva informal. Além disso, as relações empregues na construção de uma representação conceptual influenciam até que ponto esta pode ser reutilizada.

- Como permitir que os especialistas de domínio contornem as dificuldades associadas ao arranque de um processo de concetualização?

- Como encontrar um equilíbrio entre o compromisso das relações conceituais com o mundo real e a visão dos especialistas? Isto é fundamental para dotar as representações conceptuais de um elevado grau de reutilização.

Para fazer face a estes problemas, foi concebida uma abordagem holística, integrada num *framework* de concetualização como um artefato abstrato de suporte à construção de representações conceptuais em 3 camadas de formalização. No final, o objetivo é conseguir representações conceptuais computacionais bem formadas (ou seja, compatíveis com um esquema de representação bem definido), sem a necessidade de os especialistas se comprometerem com quaisquer outros formalismos que não os do seu próprio domínio técnico.

Esta *framework* e os seus elementos (e.g., sub-artefactos) foram implementadas numa ferramenta que serviu de plataforma de suporte o ciclo de implementação e avaliação dos artefatos, seguindo métodos de avaliação qualitativa, combinados com medidas quantitativas, e foram construídas com base numa abordagem de "Investigação-Ação".

Por fim, os resultados mostraram que o desempenho dos especialistas na elicitação de conceitos e relações conceptuais pode, efetivamente, ser melhorado com os processos discutidos nesta tese, contribuindo para o incremento da base de conhecimento científica na área da engenharia de ontologias e modelação conceitual.

Keywords: Processo de Concetualização Colaborativo. Grafos concetuais. Elicitação de Relações Conceptuais. Representações Conceptuais. Questões de Competência. Especialistas de Domínio. Representação de Conhecimento.

Acknowledgments

At the end of this painful route, I saw the light ... and there You were: my beloved family, my son and my wife, my unconditional friend (Carla) and my sweet Angels. You were the fundamental pilar, sustaining my emotions giving me faith towards the completion of this thesis. Indeed, it was the most important support I got.

Additionally I would like to give a special tanks to my advisor for his scientific guidance and support.

I must also include in this so personal part of the thesis a word of appreciation to all those who pushed me forward at INESC TEC and ESTGF, Polytechnic Institute of Porto.

Thank You God!

Cristóvão

“Never cut a tree down in the wintertime. Never make a negative decision in the low time. Never make your most important decisions when you are in your worst moods. Wait. Be patient. The storm will pass. The spring will come."

Robert Schuller

Contents

List of Figures

List of Tables

List of Abbreviations

BG	Basic Conceptual Graphs
CBT	Conceptual Blending Theory
CCP	Collaborative Conceptualisation Process
CG	Conceptual Graphs
CLC	Conceptualisation Life Cycle
CN	Collaborative Network
CP	Conceptualisation Process
CQ	Competency Questions
CR	Conceptual Representation
CRRM	Conceptual Relations Reference Model
CS	Conceptual Structure
DE	Domain Experts
DL	Description Logic
DSR	Desig Science Research
FEUP	Faculdade de Engenharia da Universidade do Porto
FOL	First Order Logic
KBS	Knowledge-Based System
KOCS	knowledge organisation and collaboration systems
KOS	Knowledge Organisation Systems
KR	Knowledge Representation
LO	Lightweight Ontology
OWL	Ontology Web Language
SG	Simple Conceptual Graphs
SWRL	Semantic Web Rule Language
UML	Unified Modeling Language

Chapter 1

Introduction

1.1 Background and motivation

This thesis addresses the creation and reuse[1] of conceptual representations[2], as mediator artefacts of a common domain understanding. The pragmatic nature and the heterogeneity of conceptual representations (models), as well as the social process of developing them are debated in this thesis. Typically, conceptual representations can be used for information management and knowledge sharing purposes through the collaborative creation of knowledge organisation and collaboration systems (KOCS) (Pereira et al., 2013).

The process of developing such artefacts (conceptual representations) encloses semantic modeling activities. In fact, conceptual modelling, as a core activity of the information systems (IS) discipline (Frank, 1999) has been used to define what domain experts have in mind, in a twofold perspective: as a social function for understanding and communication purposes (Mylopoulos, 1992) and; as a content function, describing explicitly the structure of specific domain concepts. The development of high-quality conceptual modeling is of utmost importance (Wang and Weber, 2002) because it facilitates early detection of possible conceptual misalignment among all parties involved in what regards to the view of the domain. However and "*despite the importance of conceptual modeling, (...) research evidence suggest that it is not done well. Practitioners report that conceptual modeling is difficult and that it often falls into disuse within their organizations*" (Wand and Weber, 2002). Moreover, (Roussopoulos and Karagiannis, 2009) claimed that a number of problems remain to be solved (e.g., the real use and reutilisation of conceptual models). Some research points to an ontology-based approach for conceptual modeling, arguing that it could improve the quality of information systems (Guarino and Guizzardi, 2006). Gulla (2007), argues that a conceptual model should meet the needs of the conflict between understanding and representation, in which understanding refers to the ability of the stakeholders to understand the

[1]To reuse encloses itself a semantic process, which implies the ability to find, select and adapt

[2] Conceptual representation is a semantic artefact, representing a semi-formal view, shared by a group of domain experts, about the understanding of the relations between concepts and its resulting conceptual structures, within a particular context.

modeling language and the developed models, and representation refers to the capability of a modeling language to represent the most important aspects of a system.

The process of conceptual modeling often requires negotiation involving strong interaction between people and heavily mediated by representational artefacts. It is a complex and time-consuming process. Accordingly, in this thesis we put emphasis on knowledge acquisition and model abstraction in conceptual modeling, with particular focus on knowledge representation and organization, enhancing the development of knowledge representation artifacts for human understanding. This is the understanding of conceptual knowledge representation followed in this thesis. In order to the domain experts (absent on knowledge representation skills) to be able to conceptualise their view about specific technical domains (for knowledge organization, domain modeling, etc.), they must be supported in the process of gathering and storing information about a specific reality - whose utility depends on how efficiently the knowledge could be retrieved and reused. Additionally, storing and retrieving information depends on the representation and organisation of the information. These knowledge-driven activities must be configured into collaborative conceptualisation processes with focus on human understanding and on informal representation schemas to be used in building semi-formal semantic artifacts (e.g., ontologies) - for information managements purposes. This semi-formal semantic artifacts, also called conceptual ontologies or lightweight ontologies, beyond a means to organize information and promote knowledge management, it intends to support the management of knowledge-related activities (e.g. structuring, storing and retrieving information) more efficiently and effectively, allowing exchanging and sharing information semantics. Such artifacts (available as conceptual representations) provide organised interpretations of knowledge structures semantically enriched by means of collaborative activities, carried out according to a socio-semantic perspective in the development of tools for collaborative knowledge representation.

The construction of these modelling structures in a consistent way is a challenge, especially when it intends to follow an informal approach to represent conceptual structures (e.g, concept maps) and its later semi-automatic formalisation. As stated by (Kharatmal and Nagarjuna, 2010), an informal representation of conceptual structures is harder than it may look like and it is in this context that the importance of the conceptual relations arises.

The development of knowledge representation artefacts, such as conceptual representations, carries the challenge related to the proper definition, in short-term, of the concepts and relations, which are the basis of the information and knowledge architecture to be common to the collaborative network (CN). In fact, the most difficult problem in a conceptualisation process is the elicitation of conceptual relations (Auger and Barriere, 2010; Elsayed, 2009). The problem has to be tackled from the double perspective of terminology and knowledge representation (with focus on ontology development). However, knowledge representation and organisation activities enclose two main obstacles: (1) KR dynamicity - the represented knowledge structures are not static. A certain knowledge structure is only valid within a context and during a certain period of time; for these KRs should be easy to reuse, modify and update (Seufert and Seufert, 2000; Bhatt, 2000) and; (2) KR expressivity - typically, the more expressive a KR is, the easier and the more compact it is to express a fact. However, more complex and ambiguous it could be (e.g. natural language). A tradeoff must be defined/achieved between KR expressiveness and the understanding about what it is intended to represent (Levesque and Brachman, 1987)

This research follows the vision that knowledge representation, while an applied discipline, has to address conceptual representations as designed, pragmatic and socially constructed artifacts whose validity and value are time, context and situation dependent. Yet, researching solutions (tools/processes) for these problems have been surprisingly scarce in the knowledge representation literature, more specifically in the ontology engineering area. This thesis, is based on collaborative conceptualisation tasks, focusing on a particular approach about the symbiosis between terminology and ontology engineering to build shared conceptual knowledge representations, addressed at the conceptual level (understanding relations between terms) to foster a collective learning of the domain and reaching agreements about its representation.

1.2 Application context and research implications

Current research work applies to collaborative learning contexts, where groups of social actors (project teams, cross-functional teams, etc.), are structuring knowledge and information for creating other semantic tools (thesaurus, taxonomies, ontologies, etc.) that can help the specification, maintenance and evolution of the structure of a knowledge management platform. This can be, for instance, a community of practice, a RDI project, etc. Additionally, it provides means to support the conceptual exploration and debate towards a common (shared) conceptualisation in collaborative networks.

The research results coming from this thesis, could streamline the process of achieving semantic agreements about different world-views, since it provides means to support the semi-automatically discovering of conceptual misalignments (e.g., match and merge operation) of the involved partners and aligning them to a common conceptual reference. The described 'build-to-reuse' approach could be used during requirements elicitation and problem conceptualisation in the software engineering domain. This work may also contribute to assist domain experts to provide accurate inputs for the definition of formal constraints, during ontology development activities. Moreover, there can also be found applicability of the results of our research, in the joint conceptual modeling of enterprise architectures, allowing the generation and maintenance of updated organisational blueprints.

Beyond the scenarios where our research efforts could be welcome, with this work we intent to provide specific research contributions, namely:

1. to improve collaborative semantic processes (e.g. collaborative conceptualisations) introducing new approaches to informally represent knowledge within groups of domain experts;

2. to provide inputs to improve storing and retrieval of graphical conceptual representations and, additionally;

3. to provide new approaches for the development of mechanisms to ease the negotiation phase, especially in the detection of conceptual misalignments and;
4. to provide means on how formal and informal representations could cohabit in IT artefacts, connecting the domain experts mental models to interoperable conceptual representations

1.3 Thesis statement

1.3.1 Research question

As mentioned earlier, this work is particularly focused on knowledge representation forms and activities grounded on conceptual modeling and tailored to collaborative environments, aiming at expressing the contents of the shared knowledge models (conceptual models). Those activities, compounded into semantic collaborative processes, are a key conceptualisation add-on for the collaborative construction of knowledge organisation systems. Given the focus of this work, the research question is posed as follows:

> How to provide the adequate support for a group of actors collaboratively building conceptual representation (e.g. a semi-formal ontology) in terms of terminological and knowledge representation aids for identifying the relevant conceptual structures (concepts and conceptual relations) of the domains involved?

This research will also address the following research sub-questions:

> (1) How to achieve a set of basic conceptual structures and identify its main components, in such a way that it can be used as a guideline (or schema) to represent and organise a domain knowledge? (2) And, how to systematize its reuse?

1.3.2 Hypothesis

With the current research work our intention is to prove that:

1. The use of semantic artifacts in (inter) organisational contexts follows some patterns[3] that can be organized and structured towards its reusability, in order to support the creation of either individual or shared conceptualisation proposals.
2. Eliciting, interpreting and organizing expert knowledge (model conceptualisation) is an iterative process of the conceptual structures under development requiring informal representation schemas and mediated by the components specified in 1.
3. There is an identifiable tradeoff between informal and formal knowledge representation needs, which could be systematized in order to allow domain experts to efficiently store and retrieve basic structures of knowledge.

[3]This view of pattern encloses the successful approach and series of actions used for expressing basic structures of knowledge. It refers to the recurrence of some basic features during the (conceptualisation) process lifecycle

1.3.3 Objectives

Aiming at expressing the contents of the shared knowledge models (conceptual models), involving the identification of relevant domain conceptual structures (concepts and their relationships), the specific objective surrounding this work, is:

1. To define artefacts (semantic processes and tools) to foster the joint construction of explicit representations of common conceptualisations, expressing concepts and conceptual relations using the appropriate terms, in an environment that brings together experts and users, and without a particular commitment to a specific KR formalism.

More specifically, we aim at developing "conceptualization assets library" (CAL), as a repository of artefacts necessary to support the creation, re(use) and retrieval of shared conceptual representations. Domain experts could use such repository, hosted in a collaborative modeling tool under development, during their conceptualisation activities. Thus, the intention is to:

1. Define a catalog of basic conceptual relations and based on this, develop a set of templates to assist the conceptual relations elicitation during conceptualisation activities.

2. Provide methods and tools to help domain specialists to continuously align the knowledge models under construction and the texts on a certain domain. (Defining a set of terminologically based mechanisms to assist users with the creation of conceptual representations in conceptual modeling tasks introduces some precision to informal knowledge representation approaches, such as concept maps.)

3. Develop a model of mapping informal conceptual representations and formal KR schemas.

4. Develop a process for systematic analysis of the conceptual relations well- formedness in order the evaluate a conceptual representation reusabilit

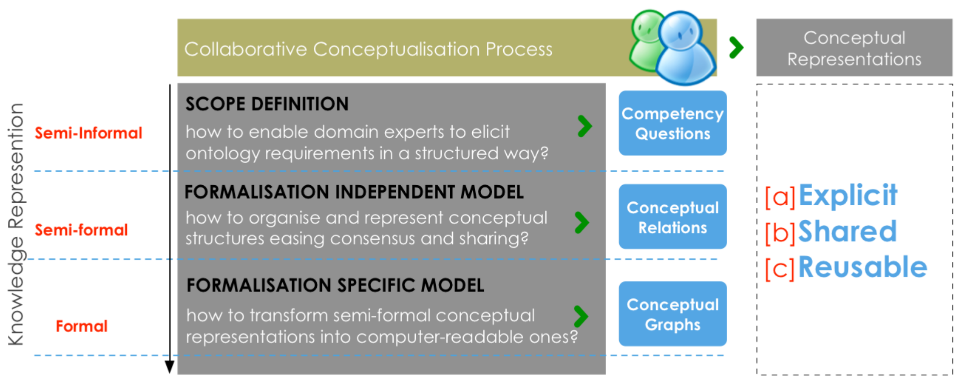


Figure 1.1: Thesis frame of concerns

Figure 1.1 synthesizes the aims (concerns) reported in this section and envisages how they could be accomplished. This "frame of concerns" will guide all this thesis development. So, considering that the final conceptualisation result should be developed by domain experts and characterized as explicit, shared and reusable conceptual representations, a 3-layered knowledge representation approach should be considered. In the top layer, domain experts define the conceptualization scope in a semi-informal way using competency questions. The middle layer is concerned about the organization of the concepts and relations between concepts in a semi-formal manner. The bottom layer is concerned to the semi-automatic transformation of the developed models in formal conceptual graphs.

1.3.4 Thesis organisation

This thesis was structured according to Design Science Research (DSR) (Hevner et al., 2004). DSR is often presented as a relatively new approach within the Information Systems discipline, which offers a general process and specific guidelines for evaluation and iteration within research projects. The five steps outlined in the methodology from Takeda et al. (1990), adapted to DSR form a natural way of presenting the structure of the development of this research. Accordingly, Figure 1.2 depicts the structure os this thesis according to a DSR view.

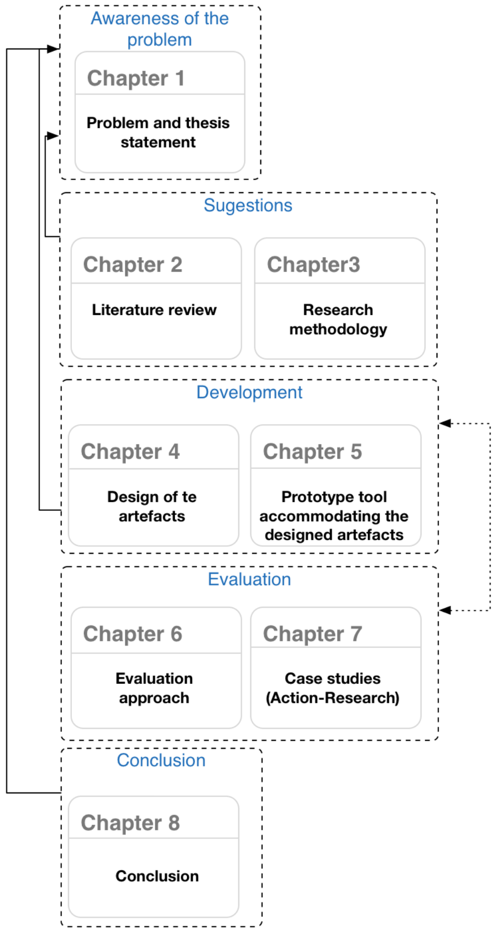


Figure 1.2: Thesis structure according to DSR

Chapter 1 Describes the context and the relevance of the research and the thesis statement. In the thesis statement the reader find the research question, the underlying hypothesis and the main research objectives.

Chapter 2 Encloses the literature review focused on the main knowledge areas evidenced by the research problem. The main discipline is "Collaborative Knowledge Representation", whereas the main concept is the "Conceptual Relations" concept. Thus, the mutidisciplinary challenge of collaborative conceptual representation is discussed as well as the conceptual relations elicitation phenomena.

Chapter 3 Presents the research design approach including the research process description, in particular, how the development of the research artefacts will be conducted and how to validate them (i.e., which cases, experiences and methods).

Chapter 4 Describes the designed artefacts as the means to address the challenges posed in chapter 1. A comprehensive framework is presented to carry out the conceptualisation process towards the representation of "explicit shared conceptual representations". The framework has a double theoretical perspective combining terminological and knowledge representations methods in order to foster the commitment of domain experts as the main knowledge modelers. Still in this chapter, a functional overview of the prototype tool is given, expecting the reader to better inform the reader on how the design artefacts will be used in practice.

Chapter 5 Describes the conceptME architecture and main features. conceptME is a technological platform built to accomodate the designed artefacts.

Chapter 6 Defines the evaluation approach followed in this thesis. to evaluate the artefacts, as the means to address the challenges posed in research problem, a qualitative approach was followed carrying out action-research studies. In addition, qualitative measures were defined to asses the results of the case studies, i.e., the conceptual representations.

Chapter 7 Describes two case studies and discusses the results.

Chapter 8 Concludes this thesis with some final remarks. Additionally, further research is for the near future is presented.

Part I

Theoretical Background

Chapter 2

Knowledge organization and collaboration

The current thesis proposal follows, by definition, a multidisciplinary approach and the theoretical background focus the construction of conceptual representations in different scientific areas. Through this chapter it is intended to frame the research efforts in the field of collaborative and knowledge representation. Furthermore, and taking as assumption the fact that the activities underlying the development of conceptual representations are tied to conceptualization processes, the notion of conceptualization is explored and the process decomposed into its basic components. Moreover, it is argued about the different dimensions inherent to a collaborative conceptualization process. By its turn, the result of a collaborative conceptualisation process is semi-formal conceptual representations, whose major challenge is related to the way concepts are interlinked. Following this, a study about available approaches on conceptual relations definition was carried out. The study was elaborated considering several scientific areas. Again, the option on exploring different domains was important to understand the possibility of accommodating informal and formal approaches in the creation of shared conceptual representations, gathering clues on how to provide the best support to the reuse and retrieval of such artefacts.

2.1 Collaboration in Knowledge Representation

At its basis, knowledge representation, apart from its foundations - whether philosophical or computational - aims at explicitly represent (organise) the objects, its attributes and the relationships between objects from a specific domain, in a transparent, accessible and (some) computable way. The concerns about KR, independently from its foundation, have been occupied researchers from several areas, either from AI, cognitive science or information science. Despite of the purposes or employed methods, the issues related to expressing knowledge representations are similar. Within AI, the following characteristics of knowledge representation are commonly accepted: i) representational adequacy; ii) inferential adequacy; iii) inferential efficiency and; iv) acquisitional adequacy. Identically, cognitive science points out the following issues on KR artefacts (Siau and

Wang, 2007): i) ease of encoding; ii) expressive adequacy; iii) acquisitional adequacy; iv) inferential adequacy and; v) the schema itself and the interpreter. Information science, by its turn, is more focused on how KR artefacts fit the purposes of information storing and retrieval and on how KR artefacts can reproduce commonly the real world concepts, allowing an easily connection/mapping between the user's concept and its representation (Hodge, 2000).

However, it is not trivial to assure that the characteristics mentioned above are embraced when building knowledge representation artefacts. KR concerns are in two levels:

1. At the content level, and

2. At the level of the form.

The content (the knowledge itself) is: (1) dynamic, since it is created in social interactions amongst individuals and organisations; (2) context-specific, as it depends on a particular time and space; and (3) it depends on the epistemological framework of the individuals (Seufert and Seufert, 2000). The form, by its turn, is related to KR formalism, which determines, considerably, the KR expressiveness (Levesque and Brachman, 1987; Martin et al., 2002; Martin, 2003), and, consequently, the understanding about what is intended to represent.

Considering that, independently from the method used for knowledge representation, three main phases for knowledge representation could be identified, such as: (i) acquisition; (ii) organisation/structuring and; (iii) share and reuse; a cross check could be made regarding the phases, characteristics and the KR levels.

Activity / **Level**	Acquisition	Organisation/structuring	Share and reuse
Content level	Acquisitional adequacy	- Expressive adequacy - Representational adequacy - Commonality - The schema itself and the interpreter	-Inferential adequacy; -The schema itself and the interpreter
Level of the form	Acquisitional efficiency	- Ease of encoding - Commonality - Expressive adequacy	-Inferential efficiency

Figure 2.1: Characterictics of KR artefacts according to different phases and levels

As shown in the above figure 2.1, the characteristics inherent to KR artefacts relate to different phases and levels. Although this research work is particularly concerned about the possible methods or models to support the creation of KR artefacts according to a collaborative KR approach, the way knowledge is organised, influences the extent to which it is reusable. It is, after all, an iterative process.

It is our view that Knowledge representation, while applied discipline, has to address conceptual representations as designed, pragmatic artefacts whose validity and value is time, context and situation dependent. In fact, in several application areas (e.g., domain engineering or terminology work), conceptual representations (or conceptual modelling) need to be created and recreated, used and reused, decomposed and synthesised and eventually disposed of, according to specific needs.

As for FOLDOC[1] in which: "Choosing a conceptualisation is the first stage of knowledge representation." and following (Gomez-Perez, 2004) for whom conceptualisation is "the first stage of the knowledge engineering cycle", we consider conceptualisation as the cornerstone of knowledge representation. We follow the stance that conceptualisation phenomenon constitutes a collaborative instrument to support the knowledge representation by means of conceptual models.

2.1.1 Conceptualisation and the conceptualisation process

2.1.1.1 The notion of conceptualisation

The process of identification of a problem or an idea and its explanation or formulation is an old issue, which has gained a new outline mainly driven by the increased complexity of the system of human activities. Roughly, the decomposition of one or more ideas into its concepts to draw up a clarified user representation over a subject, is generically called conceptualisation. Currently, the new challenges related to the new work environments, increasingly complex and demanding, led to revisiting conceptualisation. The activities performed within organisations are knowledge-driven, seeking for better answers for problems and easy ways to identify new opportunities, enclosing “non-routine” problem solving tasks requiring convergent, divergent and creative thinking (Reinhardt et al., 2011). In this context, conceptualisation assumes a new important role of supporting the knowledge-worker within collaborative organisations. Indeed, conceptualisation has been inspiring several research studies in several disciplines (from information science to AI, knowledge engineering or cognitive science). In software engineering, conceptualisation is a software development life cycle phase from great importance. It tries to design a solution from all the information gathered in the analysis phase. "The conceptualisation phase includes the objects presumed or hypothesized to exist in the word and its relationships. Its goal is to structure the domain knowledge in a conceptual model that describes the problem and its solution in terms of the domain vocabulary." as it is summarized by (Gomez-Perez, 1996). Within the scope of knowledge engineering research, more specifically organisational knowledge creation models, summarized in (Waltz, 2003) and described by (Oinas-Kukkonen and Oinas, 2001), and which aim at conducting organizational learning, conceptualisation is viewed as a key central role. It is considered a "collective reflection process articulating tacit knowledge to form explicit concepts and systemizing the concepts into a knowledge system (Oinas-Kukkonen and Oinas, 2001); or in other words, it aims at “structuring meaningful relationships of entities and events in time and space" (Waltz, 2003).

Cognitive semantics - for example - view the construction of meaning in terms of conceptualisation (Evans et al., 2007). Additionally, Kuhn (2004) states: "Probably the single most important idea from cognitive science to exploit for information modeling is that of conceptualization"

[1]FOLDOC - Free on-line dictionary of computing

(Kuhn, 2004). He continues, referring Langacker position about conceptualization in which he states as being: "the cognitive activity constituting our apprehension of the world". Still, the relation between cognitive semantics and conceptualization is better understood by considering the four principles that collectively characterize a cognitive semantics approach (Evans and Green, 2006):

1. The conceptual structure is embodied: The nature of conceptual organization arises from bodily experience, in other words, cognitive semanticists set out to explore the nature of human interaction with an awareness of the external world, and to build a theory of conceptual structure that is consonant with the ways in which we experience the world;

2. The semantic structure is equivalent to the conceptual structure: Semantic structure (the meanings conventionally associated with words and other linguistic units) is equated with concepts;

3. Meaning representation is encyclopedic: this means that words do not represent neatly packaged bundles of meaning (the dictionary view), but serve as ‘points of access’ to vast repositories of knowledge relating to a particular concept or conceptual domain;

4. Meaning construction is conceptualization: the language itself does not encode meaning. Instead, as we have seen, words (and other linguistic units) are only ‘prompts’ for the construction of meaning. According to this view, meaning is constructed at the conceptual level: meaning construction is equated with conceptualization, a dynamic process where linguistic units serve as prompts for an array of conceptual operations and the recruitment of background knowledge.

The figure (2.2) below depicts the model of meaning of cognitive semantics inspired by the previous principles.

To better understand the picture above read the labeling as follows:

- **LU** stands for Linguistic Unit;

- **C** stands for Concept and;

- **r** stands for rule;

Cognitive semantics studies language as a container and organiser of knowledge and comprises a model from three complementary domains:

1. **Domain of discourse** - the domain of discourse is created through discourse, which is considered to include text, or more accurately, verbal expressions of a particular discussion between two or more parties, which are composed by linguistic units. The semantic representation associated with domain of discourse (language) give us the semantic structure of meaning or, by other words, the linguistic knowledge. This knowledge is highly schematic and provides a schematic structure necessary for conceptual representations as mentioned earlier.

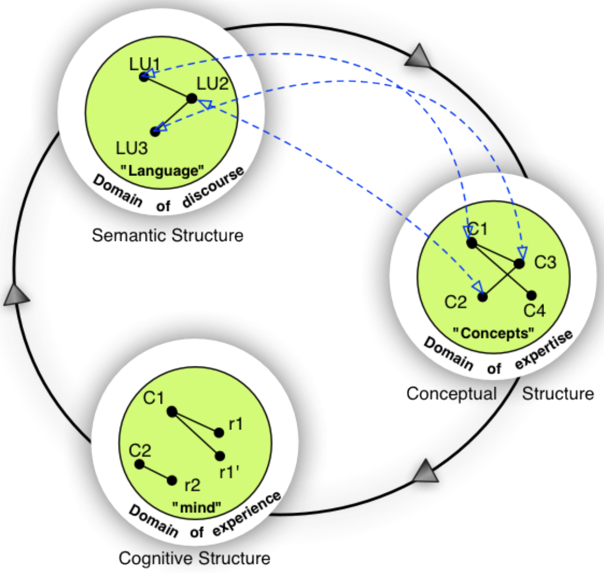


Figure 2.2: Cognitive semantics model of meaning

2. **Domain of experience** - It is related with the ever-changing mental models that both parties have of a particular discussion. The mental models are construed as modeling some object domain in the real word. They are directly dependent on or influenced by human experience and form the cognitive structure of meaning, which comprise the set of concepts and rules applied to explain and understand some phenomena

3. **Domain of expertise** - Here it is introduced the "know how" from very specific domains. It is about more rich conceptual structures. A rich conceptual structure is a representation of the structure of concepts, which belong to a specific subject field. It is related with rich aspects of perceptual and subjective experience.

Continuing on this approach to conceptualisation phenomenon grounded on cognition, (Barsalou, 2005) presents an interesting vision about conceptualisation and the way we think around concepts and represent them. For the author "conceptual representations are situated and contextualized dynamically to support courses of goal pursuit" (Barsalou, 2005). That means, concepts are not processed isolated "but are typically situated in background settings, events and introspections" (Barsalou, 2009). These conceptual representations result from situated conceptualizations, made explicit through multi-modal simulations. This means we tend to think about a concept not only in what concerns to its properties but also regarding its functions, and so forth. In summary, a "situated conceptualization is a pattern, namely, a complex configuration of multi-modal components that represent a familiar situation." (Barsalou, 2009). Yet, the most common notion of conceptualisation is from Artificial Intelligence (AI) area, given by Genesereth and Nilsson (1987) as follows:

> Conceptualisation contains the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them. A conceptualisation is an abstract, simplified view of the world that we wish to represent for some purpose. Every knowledge base, knowledge-based system, or knowledge-level agent is committed to some conceptualisation, explicitly or implicitly.

Still, the term conceptualisation appears frequently associated to ontologies. The notion above was in the base of Gruber's ontology definition (Gruber, 1993):

> "An ontology is an explicit specification of a conceptualisation." which was later redefined by Studer et al. (1998) as:
>
> > "An ontology is a formal and explicit specification of a shared conceptualisation."

Yet, Guarino (1998) partially disagree to the previous definitions of ontology mainly due to the unclarity of the term conceptualisation. According to his view:

> "An ontology is a logical theory accounting for the intended meaning of a formal vocabulary, i.e. its ontological commitment to a particular conceptualization of the world. The intended models of a logical language using such a vocabulary are constrained by its ontological commitment. An ontology indirectly reflects this commitment (and the underlying conceptualization) by approximating these intended models."

Whereas Guarino considers that an ontology refers to a restricted view of conceptualisation focused on intended meaning, which depends on the types of relations between objects, (Genesereth and Nilsson, 1987) notion of conceptualisation refers to a particular state of affairs. But, none of the definitions addresses the dynamic aspects of conceptualisation (time and context). In this still current debate about ontology and conceptualisation notions, raises the issue of ontological commitment. Inspired on Gruber (1995) and Guarino (1998) an ontological commitment is the intentional agreement on the vocabulary used to describe a particular view of a domain ensuring its consistency/correctness but without aiming to its completeness. This ontological commitment could be seen as the ultimate expression of a conceptualization.

Conceptualisation is a broad and cross-disciplinary concept and our interest on exploiting this concept on the ontology engineering domain is due to the fact that, ontologies have been used as an approach for capturing the knowledge represented by information sources (from different sources and formats) widespread across a CN or even the Internet itself. However representing knowledge through ontologies usually requires domain experts to commit to some particular formalism, which could derail or at least delay the overall process of achieving a shared representation of concepts and relationships between concepts. Unfortunately, is evident that "While different degrees of formalization have been well investigated and are now found in various ontology-based technologies, the notion of a shared conceptualization is neither well-explored, nor well-understood, nor well-supported by most ontology engineering tools" (Staab, 2008). From this, this work refers to the term conceptualisation in a more pragmatic sense and not to its formal notion. Thus departing from Huang et al. (2010) view, which refers to conceptualisation as "the relevant informal knowledge one can extract and generalize from experience, observation, or introspection", and according to the literature review, it is considered the following conceptualisation definition: conceptualisation is built upon the concepts gathered from a particular area of interest, either from observation, experience, introspection or other information sources, and the relations established among them. The goal is to express, in a conceptual model, the structure of a knowledge domain "in terms of a specific vocabulary".

Continuing in an informal tone, we can say that a conceptualisation is the result of a "conceptualisation process" that leads to the extraction and generalization of relevant information from one's experience (Huang et al., 2010). In fact, more than identifying the main constructs of a conceptualisation and the formal restrictions among them, current work is focused in conceptualisation as a process for the collaborative construction of meaning through the discussion around concepts.

2.1.1.2 The conceptualisation process

In the literature, conceptualisation has been seen more as a phase or step than a process itself and even less from a collaborative perspective. Considering a typical knowledge management process - comprising knowledge acquisition, knowledge transformation, knowledge maintenance and knowledge transfer sub-processes - conceptualisation is within the process of knowledge transformation (Waltz, 2003). SECI model, developed by Nonaka et al. (2000), describes an approach for creating and exchanging knowledge built upon a knowledge-conversion process, which contains four modes of conversion:

1. Tacit to knowledge - called socialization;
2. Tacit to explicit - called externalization;
3. Explicit to tacit - called internalization;
4. Explicit to explicit - called combination;

According to Waltz (2003), conceptualisation is the process that occurs for internal tacit knowledge, assisting its externalization by capturing human terms and explicit them as computable terms. Oinas-Kukkonen and Oinas (2001), inspired by the SECI model, goes a little further and advocates that conceptualization includes features both for externalization and combination.

Gómez et al. (2000), addresses conceptualisation as a process from a generic problem-solving point of view and framing it in the process of development of knowledge based systems (KBS). For the authors "human understanding of reality is determined by two interactive components":

a) The information gathered from the environment, either directly or indirectly - sensory component, and

b) The extraction from all the above information the relevant concepts for solving the problem at hand, their internal relations and the reasoning used to arrive at the right conclusions - conceptual component.

"So, conceptualisation is modeling by the problem solver. This modeling is represented by means of a conceptual model. This means that there are many ways of conceptualising, that is, many conceptual models for a problem." To the authors (Gómez et al., 2000), any conceptualization process must meet the following assumptions (rules):

1. **Rule of evidence**: Never accept anything as true which is not clearly and distinctly seen to be so. That is, take care not to act hastily or with prejudice, and admit only those judgements that appear to the mind so clearly and distinctly as not to aord the least doubt.

2. **Rule of analysis**: Divide each of the difficulties under examination into as many parts as possible.

3. **Rule of synthesis**: Order knowledge, beginning with the objects that are the simplest and easiest to know and so proceed, gradually, to knowledge of the more complex, also somehow ordering knowledge that is naturally devoid thereof.

4. **Rule of proof**: Always make lists and inspections that are exhaustive enough to assure that there are no omissions.

And is composed of a triplet: **concepts**, **relations** and **functions** respectively. Concepts are seen as the “mental building blocks used by human being to think. They are mental representation of things or experiences”. Relations by its turn, interlink concepts in an universe of discourse. **Functions** are “special cases of relations”. In other words, they **are relationships that only exist between two specific concepts**.

The conceptualisation process is then presented as follows:

Accordingly, "conceptualisation comes between knowledge-acquisition and formalisation [...]. Knowledge acquisition is the task of gathering the information required to build a KBS from any source [...]. This information is used during conceptualisation process to gain a clear understanding of the structure and relationships within a particular system or problem. Formalisation expresses the conceptualised knowledge [...] as structures that can be used by the computer." (Gómez et al., 2000)

The depicted process (figure 2.3) contains three phases, each one generating intermediate representations of knowledge, either through organisation charts, petri nets, decision trees, term lists or others:

1. **Analysis phase** - Analysis phase - is an activity of searching, understanding and (problem) decomposition. It comprises three stages, for a knowledge engineer to understand all parts of the problem:

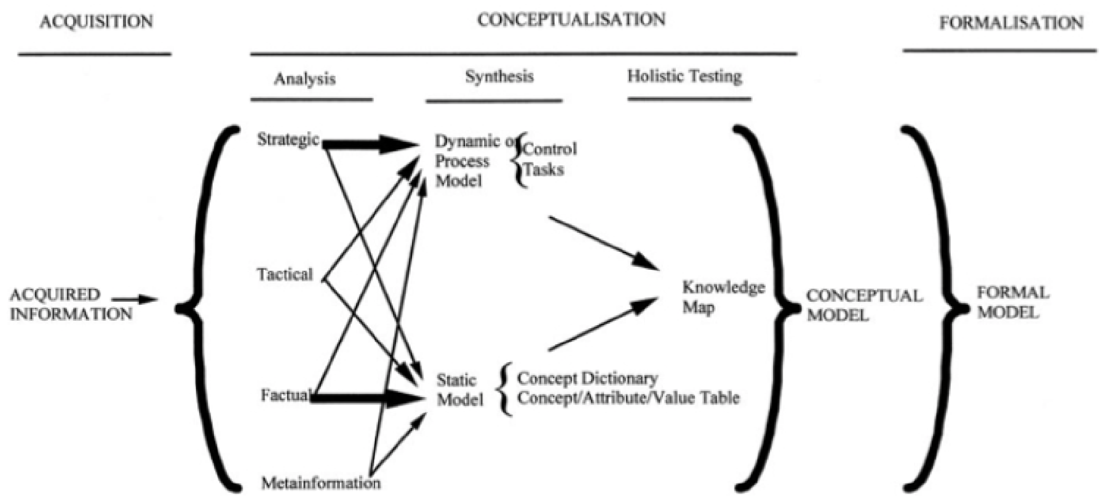


Figure 2.3: Analysis and synthesis for conceptualisation (Gomez et al., 2000)

- Strategic information identification - "Identify the steps into which the task can be divided and the order thereof"
- Tactical information identification - "Identify what happens in each step identified in stage 1"
- Factual information identification - "Identify concepts, properties, relations and functions in the domain of the task performed by the expert"

2. **Synthesis phase** - activity for organising and inter-relate the knowledge gathered in a static and dynamic model.
   - Static model - "Establish the domain structure and its components"
   - Dynamic model - "Define the functions to be performed to solve the problem"
3. **Holistic testing phase** - it intends to globally "examinate the conceptualised knowledge in order to ensure its validity and completeness before it is implemented in a software system". This is addressed by building a knowledge map "which integrates the knowledge represented in the static and dynamic models. The objective is to be able to jointly validate the knowledge expressed in both models".

Within ontology engineering field, in all different methodologies for ontology design, widely summarized and discussed in Fernández-López and Gómez-Pérez (2002); Roussey (2005), the life cycle of an ontology development process is composed of several iterative steps or stages:

1. Ontology specification
2. Knowledge acquisition
3. **Conceptualisation**

4. Formalisation
5. Evaluation
6. Documentation

The goal of the conceptualisation stage is to detect, define and organise concepts (Roussey, 2005) or more generically to choose, define and structure the conceptual elements of the domain model (Omrane et al., 2011). It is considered the core task of ontology building. Across the process of building a semantic artefacts, such as an ontology, **the focus of collaboration is at conceptualisation phase**, this is where the discussion/negotiation activities are concentrated towards a common shared structure of a portion of a knowledge domain. Moreover, conceptualisation does not requires fully automated tasks (as it may happens when formalizing or gathering domain information) but, **it asks for a continuous exercise of correlating the gathered information in earlier phases with the conceptual structure under construction, while interacting with the social network**.

In this work, the central notion of "conceptualisation process" (CP) is adopted following Pereira et al. (2013). In relation to an individual, a conceptualisation process of a given piece of reality is a collection of ordered cognitive activities that has as inputs information and knowledge internally or externally accessible to the individual, and as the output an internal or external conceptual representation. Furthermore, a "collaborative conceptualisation process" (CCP) is a conceptualisation process that involves more than one individual producing an agreed conceptual representation. In addition to an individual CP, the CCP involves social activities that include the negotiation of meaning and practical management activities for the collaborative process. In this work "knowledge representation process" is also used to refer, in practical terms, to a CP.

On top of this conceptualisation notion runs the ColBlend method (Pereira et al., 2013). ColBlend (see figure 2.4) was designed to support a collaborative conceptualisation process, based on conceptual blending theory (CBT) (Fauconnier and Turner, 1998). In practical terms ColBlend aims at supporting the co-construction of an agreed set of conceptual models, which could be translated into taxonomies, glossaries or ontologies. The method runs through a process involving explanation, discussion and negotiation.

In a synthesized way, the process comprises a set of virtual spaces: a) the input spaces - private to each party involved in the CP where the knowledge models proposals are built; b) the blend space - which contains the proposal resulting from the analysis of the input spaces and presented for discussion. Moreover it propose new concepts (originally not identified) from a global analysis of the current content of the spaces and; c) the generic space - which contains the common domain knowledge model composed by the all parts of the proposals that were accepted by all and "published" to this shared space.

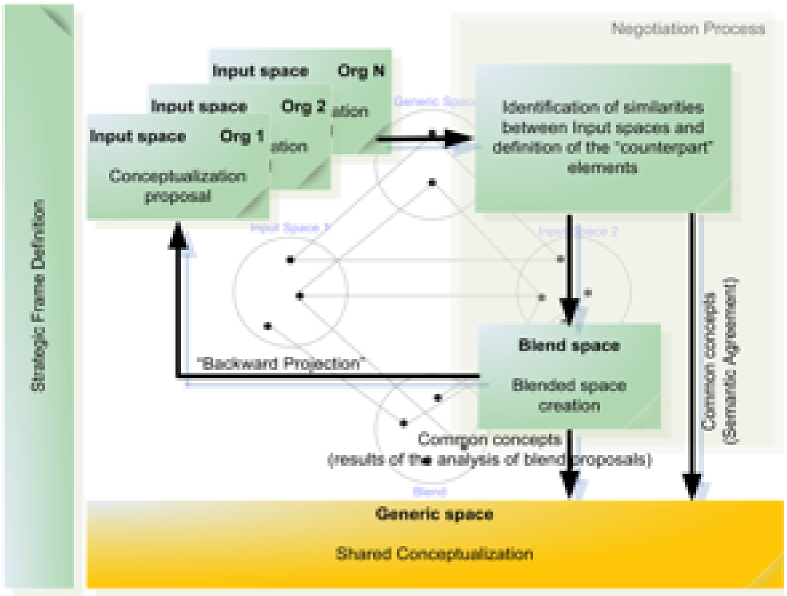


Figure 2.4: Colblend method (Pereira et al., 2013)

2.2 Representing a common conceptualisation: requirements and constraints

The collaborative conceptualisation processes are at the basis of the approach discussed in this thesis, whose focus is on empowering the domain experts in the production of useful and reusable conceptual representations, from semi-informal representation schemas to its rigorous definition (in terms of well-defined computer-readable artifacts). Thus, it is important to uncover the collaborative conceptualisation process and expose its main building blocks to better understand and define the needs for its effective utilisation.

The figure 2.5 depicts the composition of a collaborative conceptualisation process according to the notion provided earlier (see previous section).

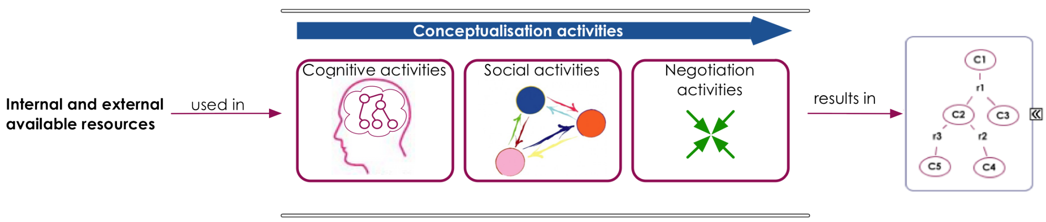


Figure 2.5: CCP composition

The following elements compose the abstract representation of a typical CCP:

1. The **inputs** are the resources used to implement the activities and, in this context, could be

systematized according to its source: internally available ("tacit" and implicit knowledge), external available (explicit knowledge) and socially generated (see next item in the list).

2. Beyond **individuals cognitive activities**, CCP addresses **social interactions** that emerge from the collaborative nature of the overall process. Social interactions have here a twofold role, they might contribute to move a resource from an internal source to an external source or, on the other hand, to bring up new resources crucial for negotiation activities (Nonaka et al., 2000).

3. The **negotiation ativities** could be supported by means of different and external perspectives (argumentation-based strategies or decision support methods, etc.)(Pereira et al., 2012). Yet, CCP takes into account a negotiation baseline comprising discussions activities around concepts, by exchanging contextual-information (i.e., information considered as relevant to an understanding of the concepts) enriching developed conceptual structures[2]. Additionally, it comprises a process of calculating the similarity degree among conceptual structures, conducted by the users, enabling the merging of two or more conceptual structures.

4. The **outcome**, in this particular case, are conceptual models made explicit by means of enriched visual representations of conceptual structures.

From the disclosure of the CCP components, there is the need for mechanisms to deal with the inputs for the process (information and knowledge internally or externally accessible) once they are crucial to conduct the conceptualisation tasks (concept elicitation to concept discussion). The possibility of the existence of unreliable sources of knowledge together with the slowness of the process and complexity on building transferable knowledge representations, constitute the major knowledge acquisition “bottleneck” (Wagner, 2008). The knowledge representation based theories, cannot deal with the issue alone, terminology could play a key role in the shortening of these constraints, mainly as regards the analysis of the corpus domain. Globally, terminology is focused on terms and their use. Those terms, framed by a context, acquire a specific meaning (Pearson, 1998). In others words, terminology provides a set of mechanisms to analyse terms in contexts, enabling the conceptualisation activities of finding concepts designations more accurately, which could decrease the time-cycle of concept elicitation and increase the foundation of the concept. From this perspective, since a collaborative conceptualisation is developed around concepts, domain experts engaged in the collaborative process and terminologists focus on the same object. From the creation and organization of a domain corpus, through term extraction methods for concept elicitation, until specific approaches to disclose semantic similarities during the discussion of achieved conceptual structures, terminology is a promising add-on of the process of conceptualisation.

Considering the previous elements and inspired by the empirical studies conducted as part of ColBlend (Pereira et al., 2013), a small set containing the major building blocks or top-level

[2]**Conceptual Structures** are models (or artifacts) representing a perception of reality by means of diagrammatic views forming a network of concepts interconnected by meaningful linking phrases.

requirements are described. The following building blocks (B1 to B3 in the list below) are considered fundamental for a collaborative elicitation of conceptual representations based on a corpus-based approach to be successful.

B1 - Terminological methods for corpus organisation : each organisation represents its conceptualisation proposal fed by a set of knowledge sources (such as URIs specifying documents, webpages) shared by all parties. The use of textual resources should be preceded by a task where the aim is to organise the domain corpus. Having such a repository could enable a more efficient extraction of term candidates for the construction of the initial conceptualisation proposals (Aussenac-Gilles et al., 2000; McEnery and Wilson, 1996).

B2 - Basic top-level conceptual structures : it is taken as an assumption that the main conceptualisation result is a less formal knowledge representation, which could be shaped into a shared conceptual model. One possible way of ensuring a common interpretation of the created conceptual models includes sharing a set of basic top-level conceptual structures and the meaning of their concepts and relations. Following a top- down approach on representing a domain knowledge, the process of creating conceptual structures could, among others, be based on patterns, which could be translated from text and from an ontological and synthesised perspective regarding its reuse potential. Gradually, these structures could be fine-tuned using term contexts and further negotiation activities.

B3 - Methods for analysing consistency in the conceptual structures : ensuring semantic consistency for conceptual structure interchange requires something more than just gathering conceptual structure patterns and defining a set of basic templates to be (re)used. If it is acceptable that templates could help create generic domain conceptual structures, inversely the specifics of a domain field enclose a set of particular details that typical top-level templates may not consider. Some assistance should be given to the users in order to help them determine the meaning of each structure. This could be performed on one hand by attaching specific metadata to conceptual structures and, one the other hand, by implementing real-time mechanisms to validate the ontological compliance of the model under construction. Examples include suggestion mechanisms, either based on context analysis or on cross-checking the various models of the same domain.

In a synthesizing overview, the collaborative conceptualisation is a process of continuous construction of meaning, providing a framework (a common structure) on what are the main activities to support the development cycle of semantic artefacts that represent the shared knowledge of a specific domain context. Considering the literature, conceptualisation could be presented or framed within a three-dimensional view as depicted below (see figure 2.6).

Each of the three different axles in the figure, shows the roles associated to knowledge representation, socio-semantics and cognitive perspective, respectively, in the conceptualisation activities. The following table (table 2) provides the justification for the roles identified in figure 2.6.

Table 2.1: The roles of the different conceptualisation dimensions

	Knowledge Representation	Socio Semantics	Cognitive dimension
Ontological commitment rule	Conceptualisation intends to achieve an agreement on the vocabulary used to describe a particular view of a domain ensuring its consistency/correctness but without aiming its completeness		
Interaction and content role		Conceptualisation is a process which involves social activities which leads to the development of semantically rich conceptual structures	
Foundational role			The conceptual structures emerge from bodily experience. Besides, its construction and interpretation depends on, or is influenced by lifetime experiences and acquired expertise

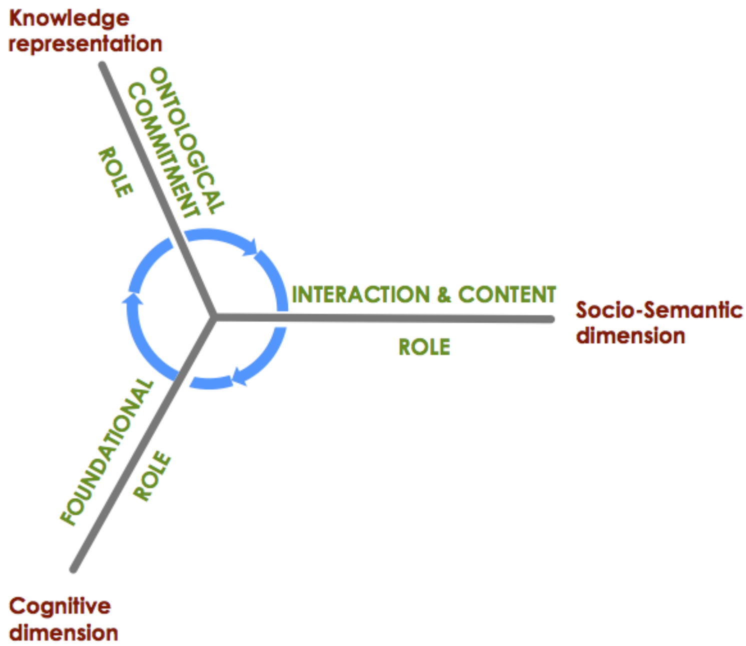


Figure 2.6: Conceptualisation dimensions

In order to find a pragmatic approach to support a CCP, that could provide a generic view on how to apply its sequence of activities, firstly, it is needed to overcome the challenge on how to assemble conceptual structures. The term conceptual structure (CS) is widely used in knowledge representation and conceptual modelling literature in general. According to Sowa (2000), conceptual structures express declarative knowledge by representing it as a connected bipartite oriented graph (conceptual graph). Mineau states that "*every network of concepts, whether an hierarchy, ontology, partonomy or semantic network can be called a structure of concepts. More specifically, CS is a representation of the structure of concepts, which belong to a subject field or domain. Conceptual structures are related with rich aspects of perceptual and subjective experience.*" (Mineau et al., 2000). The author considers that CS are models (or artefacts) representing a certain perception of reality.

Within the context of this thesis, a CS is a diagrammatic representation of a network of concepts interconnected through a linking phrase, here called conceptual relation. A CS represents a piece of knowledge (e.g. a simple fact) according to a particular view of a specific domain. Moreover, it is considered that a structure in the form of "$< concept >< conceptual\_relation >< concept >$", representing the minimum granularity of a CS, that is, a granularity of three. A final or complete conceptual representation, by its turn, encloses the maximum granularity of a CS, a granularity of multiples of three or greater than three.

In the end, CS constitute assets or artifacts that could be (re)used in a systematic way in a collaborative knowledge representation process. Hence, assembling conceptual representations is an iterative and incremental process of building and organising CS, where **relations play a key role**. The "art" of assembling CS, needs for an aided-approach to conceptual relations elicitation.

In a first stage there is the need to find the appropriate conceptual relation to interconnect two concepts. At a second stage, it is necessary to find the right place for a CS within the whole model, keeping the intended meaning of the artefact.

The problem of conceptual relations[3] elicitation as the main modeler agent of conceptual structures is debated in several domains (information science, AI, KR, Computer Science and terminology (Staab et al., 2000; Kharatmal and Nagarjuna, 2010; Auger and Barriere, 2010; Sousa et al., 2012; Elsayed, 2009). The is a broad consensus that the definition of relations between concepts are "*the major building blocks in common ontology definitions*", and that, "t*heir definition consumes much of the time needed for engineering an ontology*". Some authors, from information science domain (Elsayed, 2009) advocate that "*the information of conceptual relation is the most important part of building a conceptual representation*", and that, "*it is also the most demanding*". Additionally, the same author state that "*developing a framework of conceptual relations to support users in this task, not only reduces the variability of expressions, a necessary condition for sharing and exchange conceptual representations*" (Elsayed, 2009). The stance followed in this work is that the problem (conceptual relations elicitation) has to be tackled from the multi-disciplinary perspective. This is not an exclusive problem of knowledge representation in particular ontology engineering or terminology or information science. The construction of modelling structures in a consistent way is a challenge, specially when it intends to follow an informal approach to representing conceptual structures (e.g, concept maps). As stated by Kharatmal and Nagarjuna (2010), an informal representation of conceptual structures is harder than it may look like. And, in this context, the importance of the conceptual relations increases.

2.3 Approaches to conceptual relations definitions

The study of conceptual relations elicitation and the evaluation to which extend they contribute to the construction of well-defined conceptual representations, is one of the main concerns of this thesis, towards the definition of a comprehensive framework to assist domain experts performing their conceptualization activities.

Conceptual relations were studied according to three different perspectives or at three different levels (see figure 2.7):

- Conceptual,
- Logic, and
- Socio-semantics.

The ideia of studying conceptual relations in these three perspectives was to get some understanding on how to better assist domain experts on producing well-defined conceptual representations, starting to define semi-informal representation schemas and establishing a mapping between

[3] This work considers "conceptual relation" as a relation linking meanings of concepts, and "lexical or semantic relation" as a relation linking linguistic units and the meanings they denotes. Although concentrating on the identification and representation of the former, the latter is important to achieve that goal.

those structures and formal structures of knowledge. Thus, conceptual and logical foundations of conceptual relations followed an ontological approach, to understand the cognitive dimension of individual identification of conceptual relations and to support and maintain conceptual representations, respectively. Finally, conceptual relations were studied in specific applications in order to evaluate their utility.

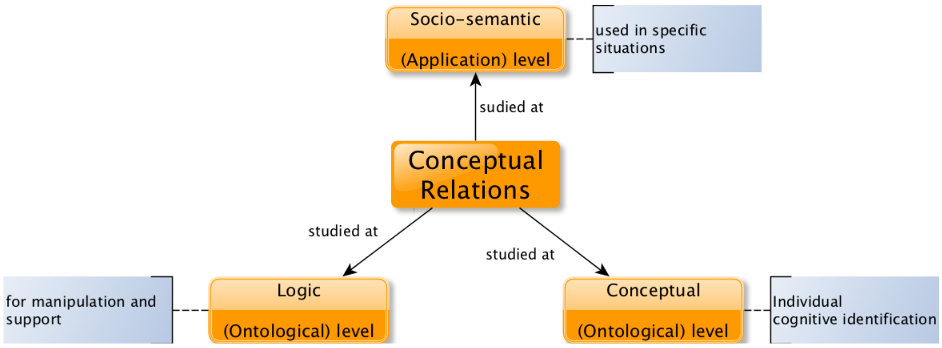


Figure 2.7: Conceptual relations study context

From the literature, there, there is an identifiable gap on the study and identification of non-taxonomic relations and relations on non-branching hierarchies. The research on this kind of semantic relations have been carried out according to several perspectives/approaches:

- knowledge representation (AI)
- knowledge organisation (information science)
- Semantic and lexical relations (linguistics)
- Ontological relations (ontology, formal ontology)
- Cognitive models (cognitive semantics)

2.3.1 Artificial Intelligence perspective

AI is mainly concerned with the aspects of formal knowledge representation in which the main goal is "t*o describe the terminology of a domain in terms of classes/concepts describing sets of individuals and properties/roles relating these.*" (Alvarez et al., 2007).

This approach derived from ontology in a search for methods for formally representing relations in a knowledge representation scheme for use in data modeling and knowledge-base influencing. Some of these schemas are Conceptual Graphs, Description Logic or even OWL. Through specific formalisms such as OWL "is possible to make statements about a set of concepts, such as to declaratively specify that two classes are disjoint, analogous declarative statements are not possible for relations. This also comprises the assignment of properties, while concepts are assigned with as many properties as needed, the same level of precision cannot be applied to semantic relations." (Alvarez et al., 2007).

In Sowa (2009) a brief overview on graph based knowledge representation is made. Semantic networks, which represents semantic relations among concepts, emerged along with the graph-based notation by Margaret Masterman. However semantic networks could not represent non-taxonomic knowledge. According to Sowa (2009) summary, "Silvio Ceccato presented correlational nets, which were based on 56 different relations, including subtype, instance, part-whole, case relations, kindship relations, and various kinds of attributes; and David Hays presented dependency graphs, which formalized the notation developed by the linguist Lucien Tesnière."All this graphical notation had its foundation in linguistics and computational linguistics. Sowa (2009) continues mentioning that none of this graph notations “could express full first-order logic". That is the reason why he developed conceptual graphs (CGs) as an intermediate language for mapping natural language questions and assertions to a relational database. The official standard for conceptual graph syntax and semantics is the ISO/IEC 24707 standard for Common Logic. According to ISO/IEC 24707:

> every conceptual relation r has a relation type t and a non-negative integer n called its valence.
>
> - The number of arcs that belong to r is equal to its valence n. A conceptual relation of valence n is said to be n-adic, and its arcs are numbered from 1 to n.
> - For every n-adic conceptual relation r, there is a sequence of n concept types t1,...,tn, called the signature of r. A 0-adic conceptual relation has no arcs, and its signature is empty.
> - All conceptual relations of the same relation type t have the same valence n and the same signature s.
> - The term monadic is synonymous with 1-adic, dyadic with 2-adic, and triadic with 3-adic.

In the Online Conceptual Graphs course at Aalborg University - Department of Communication ($http://www.huminf.aau.dk/cg$), we can find an overview of the relation types and its meaning (Petersen et. al, 2005):

On (on)

In (in)

Agnt (agent)

Thme (theme)

Ptnt (patient)

Rcpt (recipient)

Dest (destination)

Semantic networks, by they turn, were developed earlier, in 1968 ny Quilian with the purpose of representing dictionaries for AI and machine translation (Baader, 1999). It's an old formalism for knowledge representation where the concepts and its relations within a specific domain are expressed in a declarative graphical notation. There are three basic elements in semantic networks: i) concepts; ii) relations and; iii) instances.

In a semantic network, each node represents a concept, object or situation connected by labeled arcs or edges which represent a semantic relation between concepts. In such representation, meaning is implied by the way a concept is connected to other concepts (Heflin, 2001). There is no standard set of relations for semantic networks, but the most important or the most common are:

- **INSTANCE** relation (i.e. generic-specific relation)
- **IS-A** relation (i.e. subset/sub-kind/sub-type relation)
- **PARTHOOD** relation (i.e. part-of/has part relation)
- **SAME AS** relation

Therefore, semantic network may use special labeled arcs giving a partial order on concepts. Such arcs can be from different types:

- Arcs that assign properties to concepts or objects.
- Is-a arcs that introduce hierarchical relationships. Is-a arcs indicate that a concept is a subclass of another.
- Instance-of arcs indicates that a concept is an example of another.

Considering knowledge representation within AI, frames are another representation formalism to have into account. Frames representation scheme became popular in the 70s when Minsky introduced a completely different representation of knowledge for that time. In general, 2 frames are structures representing classes of objects in terms of properties that their instances must satisfy. Such properties are defined by the frame slots, which constitute the items of a frame definition" (Calvanese et al., 1999).

Frames are something close to the database records but with more capacity to express data. It is a complex structure composed by:

- An identification, which is basically a name.
- Slots, which correspond to the object attributes. Each slot has its identification; its default value and its current value.

It can be said that frames are "object-oriented" representation formalisms and each frame has at least another frame, which is hierarchical superior. Hence, frames allow the implementation of the inheritance mechanism, and thus, frames can establish a set of relations between each other according to following list:

- **"is-a" relation**, which is a relation of hierarchical dependency.
- **Inherit relation**, which is a relation that allows inheritance of frame slots that are not in the same hierarchical line.
- **Do not inherit**, where there are no inheritance of frame slots in the same hierarchical line.
- **A slot**, which can be also a frame

Frames had a huge impact as KR formalisms and quite a few languages and tools implementing frames emerged subsequently. Other ones, like Ontology Web Language (OWL), emerged from the new approach to Frames based ontology languages. Regarding frame-based implementation tools, Protégé, which is the most used software tool for knowledge acquisition and representations, uses frames in its core specification.

2.3.2 Information science perspective

In the scope of IS, there is no clear view about the theoretical foundations about semantics. The available research found inspiration in other domains such as linguistic (Hjorland, 2007) or cognitive science for example. In what regards to semantic relations, specifically, there is the concern to clarify the concepts of semantic relation and lexical relation (Khoo and Na, 2006; Hjorland, 2007) in which semantic relations are those relations between concepts - called conceptual relations -, and lexical relations are the relations between words. In Khoo and Na (2006) the authors also stressed about other terms, such as "lexical-semantic relations", used to refer to "relations between lexical concepts, that is, concepts denoted by words". Furthermore, two more categories could be found distinguishing between paradigmatic and syntagmatic relations (Khoo and Na, 2006; Peters and Weller, 2008; Stock, 2010). Syntagmatic relations occurs between concepts in specific documents; it refers to relations between words that co-occur (often in close syntactic positions) in the same sentence or text, and the meaning comes from syntactic and grammatical rules. Paradigmatic relations are relations between pairs of words or phrases, that can occur in the same position (grammatical class, that is, noun, adverb, adjective, verb, etc.). (Peters and Weller, 2008).

The work performed by Khoo and Na (2006) approaches semantic relations according to a linguistic and psychological perspectives. The authors advocate that there is the need for natural language processing to identify semantic relations, in order to accomplish activities related to ontology construction, information representation, information extraction or information retrieval. A set of semantic relations, considered as unitary primitive relations, were selected and summarised by the authors.

"*The distinction between meronymy and hyponymy relations is clear for concrete concepts but fuzzy for abstract concepts. Hyponymy relations can be said to exist within concepts, while meronymy relations are between concepts.* " (Khoo and Na, 2006). Moreover, " *hyponyms inherit features from the hypernyms but parts do not inherit features from the whole*" (Khoo and Na, 2006).

Table 2.2: Semantic relations catalog from (Khoo and Na, 2006)

Semantic Relation	Short Description
I - Paradigmatic relations	
1. Hyponym - hypernym	Imposes a hierarchical structure. Called "is-a" relations, "kind-of" relations or even superordinate-subordinate, genus- species and class-subclass relations. The relation implies class inclusion.
2. Synonym	Implies that two concepts are similar in their meaning within a specific context. Could be expresses as "same as".
3. Antonymy	Imposes a relation between opposite meanings within a specific context.
4. Meronymy - holonym	It refers to a "Part-Whole" relation, expressed as "part- of". It encloses 6 types of relations.
4.1 Component - object	
4.2 member - collection	
4.3 portion - mass	
4.4 matter - object	
4.5 task - activity	
5. Toponymy	Refers to broader-narrow relations between verbs. Relation between verbs (that occupy the same hierarchical position and super ordinated to a general event) and more specific verbs. This relation emerged by noting that hyponym relation could also occur between verbs beside nouns. Hence, hyponym among verbs is a toponymy.
I - Syntagmatic relations	
1. Cause-Effect	Very difficult to define (Khoo and Na, 2006). In order to be established a cause-effect relation, some necessary conditions should be met, according to Hume, cited by (Khoo and Na, 2006): a) **contiguity** in time and space - the relation between A and B must not occur by means of third concept; b) **priority** in time - implies a temporal antecedent condition (e.g. A preceeds B); c) **constant conjunction** between cause and effect - A and B must be related;

Beyond the aforementioned synthesis of semantic relations, discusses about the most common relations used in specific knowledge structures such as thesauri and ontologies.

Table 2.3: Common relation used in knowledge structures (Khoo and Na, 2006)

KOS	Type of Relation
Thesauri	a) Equivalence relation expressed as use and use for; b) Hierarchical relation - broader term and narrower term; c) Associative - related term
Ontologies	"*In practice, an ontology is expressed as a taxonomy*": a) IS-A relation; b) Part-Whole relation; c) Attribute-value relation; d) other kinds of relations as well as additional rules or constraints called axioms.

Peters and Weller (2008) concludes that the most used types of relations in domain-specific knowledge representation and organization models are: i) **relations of equivalence**; ii) **hierarchical relations** - comprising meronymy (mereology, part-of relation, part-whole relation, partonomy) and and hyponymy (kind-of-relation, taxonomic relation, taxonomy) and; iii) **Associative relations**. The following picture represents a classification of some knowledge organization systems according to the common types of relations.

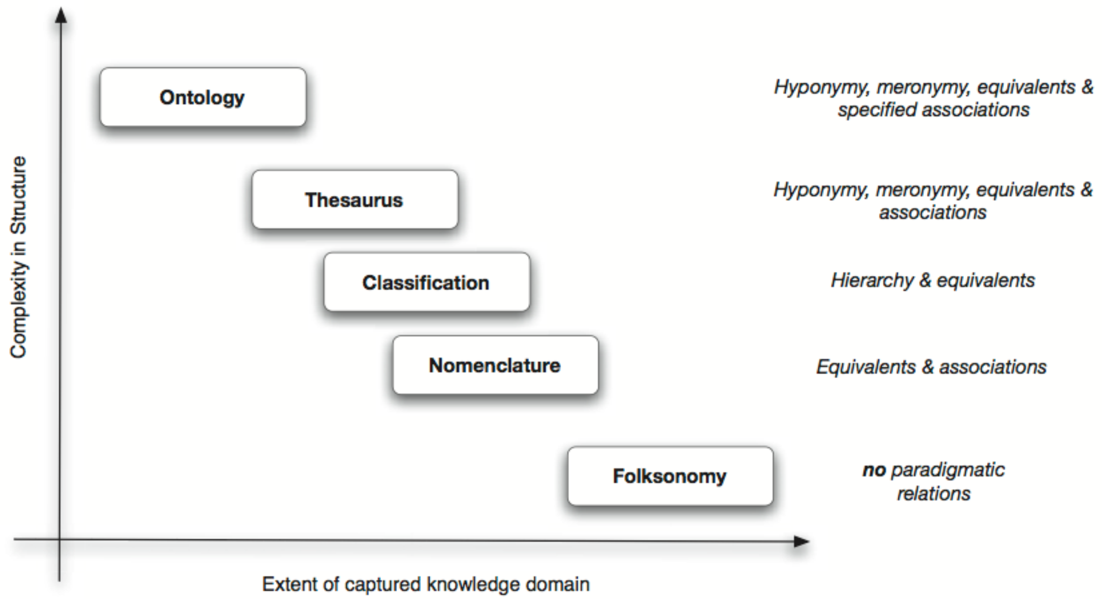


Figure 2.8: Classification of popular knowledge representation systems according to complexity in relational constructions (Peters and Weller, 2008)

A similar "exercise" was done by Stock (2010), systematised in the following picture, in which the author relates the KOS (folksonomy, nomenclature, classification, thesaurus and ontology) to the semantic relations used by each one.

	Folksonomy	Nomenclature	Classification	Thesaurus	Ontology
Term	Tag	Keyword	Notation	Descriptor	Concept
Equivalence	–	Yes	Yes	Yes	Yes
– Synonymy	–	Yes	Yes	Yes	Yes
– Gen–Identity	–	Yes	–	–	Yes
– Antonymy	–	–	–	–	Yes
Hierarchy	–	–	Yes	Yes	Yes
– Hyponymy	–	–	–	Yes	Yes
– simple Hyponymy	–	–	–	–	Yes
– Taxonomy	–	–	–	–	Yes
– Meronymy (unspecific)	–	–	–	Yes	–
– specific Meronymies	–	–	–	–	Yes
– Instance	–	–	–	As req.	Yes
Specific Relations	–	–	–	Yes	Yes
– "see also"	–	As req.	As req.	Yes	Yes
– further Relations	–	–	–	–	Yes
Syntagmatic relation	Yes	Yes	Yes	Yes	No

Figure 2.9: KOS and therapies they use (Stock, 2010)

Further, the authors (Peters and Weller, 2008) presented a set of semantic relations and their specification (as depicted in Figure 2.10), arguing that the filed of specifying semantic relations in the development of Knowledge representation artefacts of information science domain towards a general theory of relations, is still an open discussion.

The work of Elsayed and also Shams (Elsayed, 2008; Shams and Elsayed, 2008; Elsayed, 2009) has been towards the identification of a framework of conceptual relations to support conceptual modelling, mainly based on informal concept mapping techniques. From Elsayed (2009), a set of classified relations were disclosed into 7 categories that fall within 4 main themes: taxonomy/hierarchy, organisation, space & time and action-based relations in a total of 31 relations (see figure 2.11.

According to the work accomplished by Shams et al. (2010), a framework for semantic relations was elaborated to be used for corpus knowledge mapping, that is, for modeling activities from text.

The framework comprises a taxonomy of relations, starting from three main relation categories (predicate relations, instantiation and extension), which are subdivided into two more tiers of semantic relations. At the end, it is possible to find a set of specific relations for each entry of the last tier, as is shown in figure 2.12.

For this, the author "*had to resort to language independent representations used in various linguistic and knowledge representation computational tasks*" (Elsayed, 2009). Still in the domain of conceptual modelling, specifically in the information science (IS) domain, Storey (2005) proposed the use of ontologies to "*capture the semantics of relationships verb phrases*". Note that, for the author, a relationship assumes the form of $Entity - verb_{p}hrase - Entity$. Eight categories were proposed: i) common verb phrases (obtained from wordnet); ii) data abstraction (is-a, part-of and member-of); iii) wordnet verb files; iv) business process; v) temporal; vi) event and vii) associated_with as the default one. Beyond that, an empirical study was presented and a relationship ontology system described in which terms candidate to name relations, could be classified

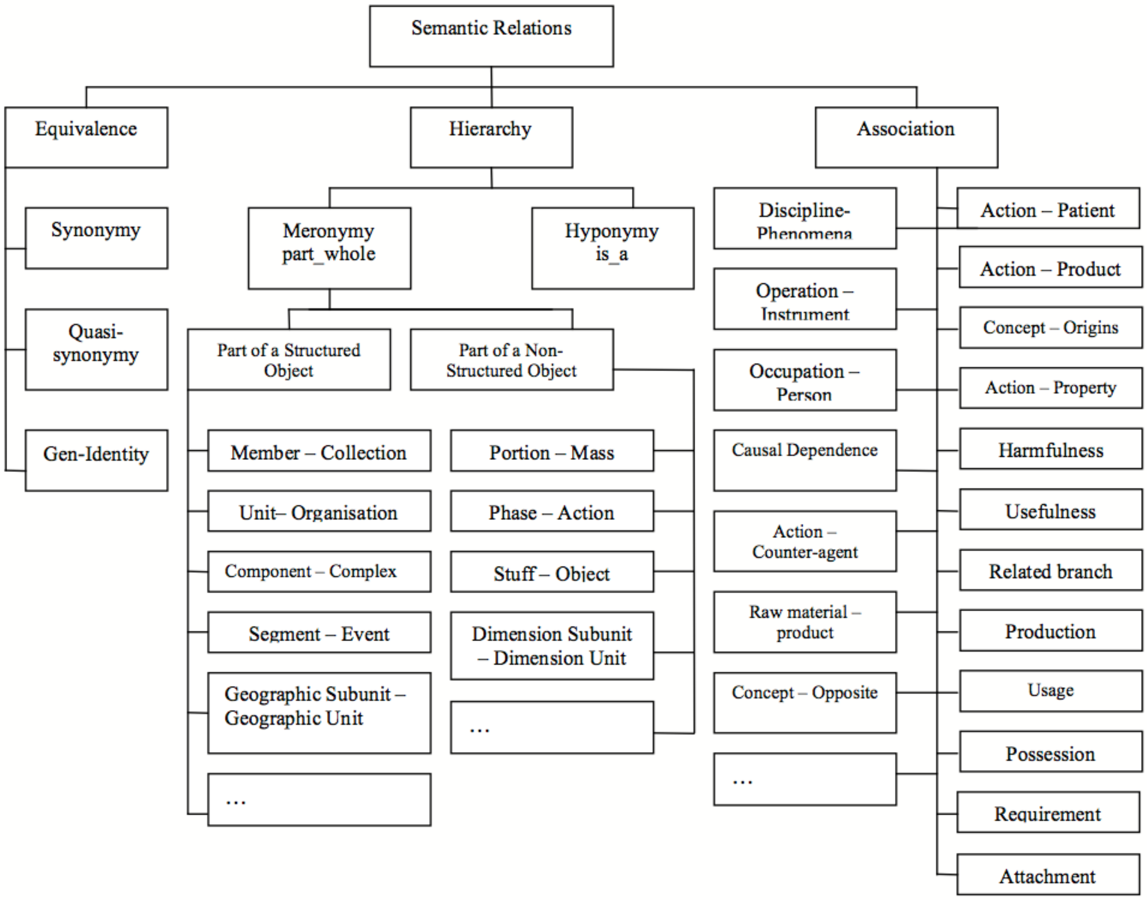


Figure 2.10: Semantic relations and specification (Petters and Weller, 2008)

Main themes	Categories	relations
Taxonomic	Hierarchy	Isa A type of A kind of Instance of
Organization	Generic	Has Such as Also known as Attribute of
	Structural	Part of Member of Consist of Made of Associated with
Space & time	Spatial	Inside Outside Adjacent Through
	Temporal	Follow Occur at Occur with Precede
Action	Actor	Cause Act Perform Receive Experience
	Activity / event	By means of Condition for Lead to Manifestation of Result of

Figure 2.11: Classified Conceptual Relations (Elsayed,2009)

into a repository organised by domain. It is important to mention that this work was fundamentally an exercise of extracting the relations that best fit the database design needs. For that, the author looked for inspiration in several domains: conceptual modeling (for is-a and member-of relations), cognitive science (based on mereology), linguistics (based on wordnet) and business. More recently, (Stock, 2010), come up with a semantic relation classification, in order to provide Knowledge Organisation Systems of sophisticated and effective mechanisms of searching, browsing, querying, etc,. He followed the drive that it could be done by means of accurated knowledge representations, based on concept theory. According to the author (Stock, 2010) there are five approaches for knowledge representation regarding knowledge organisation systems: nomenclatures, classification systems, thesauri, ontologies and folksonomies). Since there are no concept system without relations between concepts it was presented a semantic relation classification to a specific knowledge representation approach (see figure 2.13).

As we could see from the classification depicted in the figure 2.13, beyond association relation type, all others are more or less commonly agreed despite of the approach or perspective. The specific one (Association - various others) still non-specified.

2.3.3 Semantic and lexical relations (linguistics perspective)

The main distinction that we make between conceptual relations and lexical- semantic relations is associated to the scope and purpose of their usage. Lexical- semantic relations are related to

Relation Category	Tier 1 Semantic Relations	Tier 2 Semantic Relations	Predicate Relations	Inverse Predicate Relations
Predicate Relations	Hierarchy		Have type	Type of
	Physically Related	Parts	Have component	Component of
		Constituent Material	Make, Produce	Made of, Produced by
	Spatial Relations	Location of Objects	Take place between, Connected to, Flows through, Have direction	Direction of
		Location of Activities	Transfer, Find, Divide, Commence from	Transferred by, Found by, Divided by, End to
	Causally/ Functionally Related	Effect/ Partial Cause	Affect, Cause, Vary in, Resist, Force, Limit, Opposite to, Related to	Affected by, Caused by, Resisted by, Forced by, Limited by
		Production/ Generation	Produce	Produced by
		Destruction	Collide, Melt	Collided by, Melted by
		Manifestation	Represent	Represented by
		Conversion	Convert, Convertible to	Converted by, Convertible from
	Instrumental Function/ Usage	Functions	Carry, Measure, Supply, Share, Depend on, Protect, Absorb	Carried by, Measured by, Supplied by, Shared by, Depended by, Protected by, Absorbed by
		Use	Use, Do not use	Used by, Not used by
	Human Role		Deal with	Dealt by
	Conceptually Related	Topic	Govern	Governed by
		Representation	Represent, Characterize	Represented by, Characterized by
		Property	Have state, Have unit, Have source, Have Magnitude, Have Terminal	State of, Unit of, Source of, Magnitude of, Terminal of
	Similarity	Synonymy	Is, Referred to	Is
		Hyponymy	Have type	Type of
	Quantitative Relations	Numerical Relations	Proportional, Inverse proportional to, Gain, Lose, Do not gain, Do not lose	
Instantiation			Have instance	Instance of
Extension			Have Extension	Extension of

Figure 2.12: Framework for semantic relations in the corpus (Shams et al., 2010)

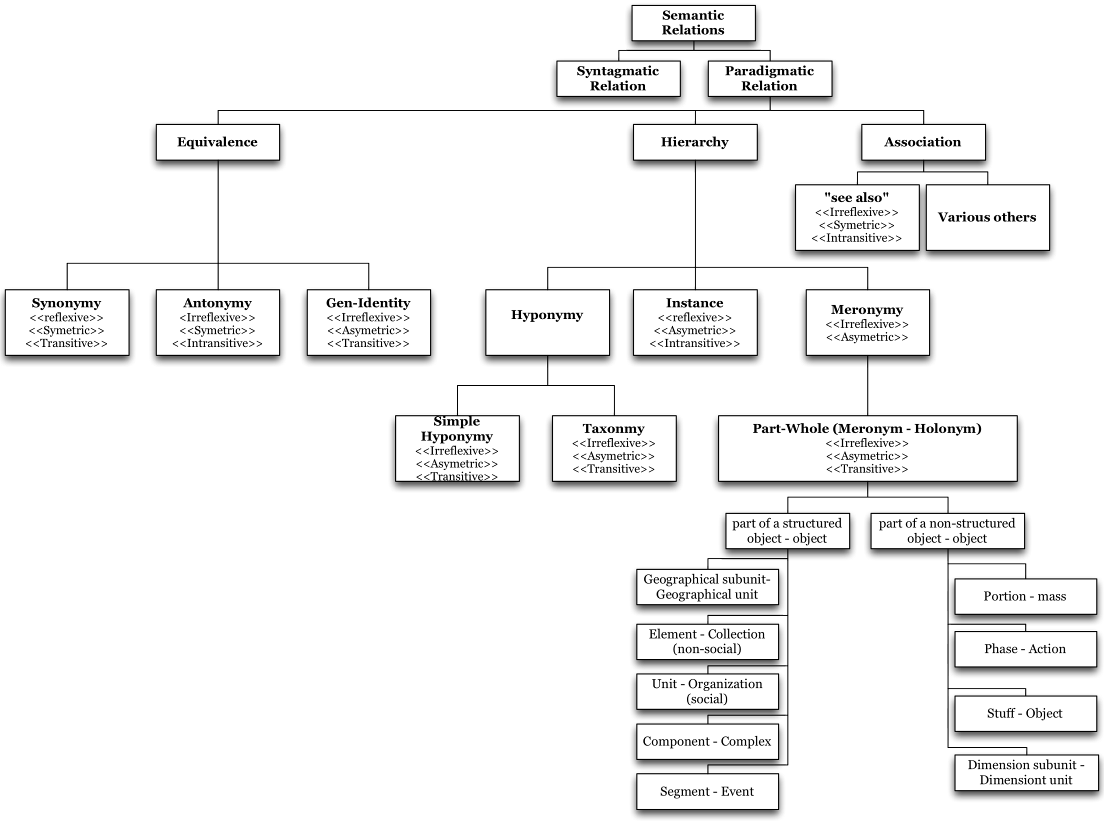


Figure 2.13: Semantic relations classification for KOS (Stock, 2010)

the process of linking words in order to create text, whereas conceptual relations are used for non-textual knowledge representations through the construction of conceptual structures.

Lexical semantic relations have been widely studied and we found them well defined and accepted among scientific community. This relations could be classified into classical and non-classical relations (Morris, 2004; Chiu et al., 2007).

Classical relations, according to (Morris, 2004; Chiu et al., 2007) are those from wordnet such as:

- hypernymy (some times called hyperonym);
- hyponymy, shares a type-of relation with hypernymy and it is its sub term;
- meronymy (also called partonomy), corresponds to a part-whole relation;
- antonymy, and
- synonymy.

However, the majority of lexical semantic relations found in real-world text are in fact non-classical (Morris, 2004; Chiu et al., 2007).

Manfred Krifka (1998) made another classification on lexical relations types:

- **the basic ones**: synonymy, hyponymy, complementaries and Antonyms;
- **taxonomies**: taxonomic hierarchies, folk taxonomies, natural kinds, mereonomies (partonomies)

Conceptual relations have been reviewed and discussed within the field of Terminology most notably by (Sager, 1990; Madsen et al., 2001; Nuopponen, 2005) , following the classification proposed by Wüster and Cabré (1998). All these works follow a classification approach on conceptual relations. In fact, we are witnessing an interesting variety of classifications in which the most common division is between generic relations, part-whole relations and a set of others (depending on the classification). This division may be more or less detailed. Nevertheless, there is consensus in the fact that the "list" of the relations that will underpin the construction of conceptual structures for knowledge representation is not very extensive.

Nuopponen (2005), built an extensive concept relation classification, gathered through applying terminological methods. The result was an overview on the most often relations but also a more rare set of relation types. The author considers that the most common relations used in traditional terminology (generic, partitive and associative comprising temporal and causal relations) are not enough for studies performed within advanced terminology management systems or semantic web applications, concept modeling, etc.

The exhaustive classification made by the author comprises a set of relation types to allow the description of activities, actions and processes. This classification intends to answer the following basic questions:

- Who? (agent);

- What? (patient);
- With what? (instrument);
- How? (method);
- Why? (cause);
- Where? (place);
- From where? (place of origin);
- To where? (destination);
- Through what? (intermediary route);

The achieved classification starts with two main categories of relations, namely:

1. **logical concept relations** as direct relations between concepts. Also designated as abstract relations, categorial relation, genus-species relation and generic relations;
2. ontological concept relations as relations arising indirectly between concepts. Logical concept relations are divided according to two different criteria:
   - the concept positions in the concept system and;
   - concepts intension (concept formal definition) and extension (concept range of applicability)[4].

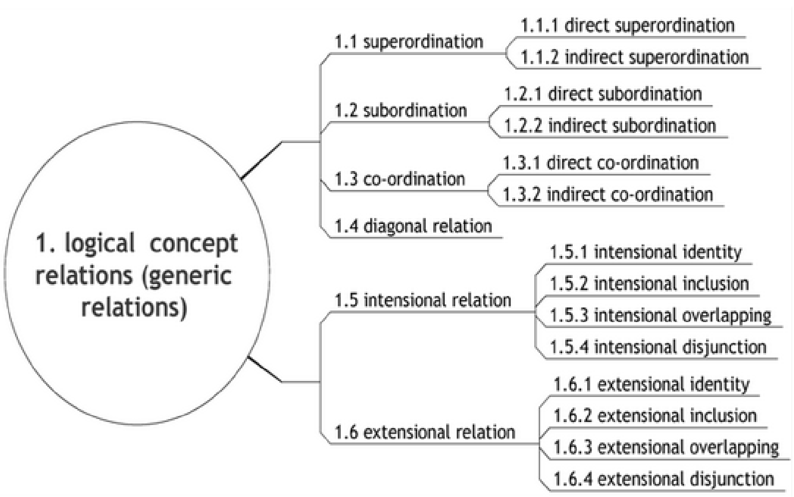


Figure 2.14: Logical concept relations (Nuopponen, 2005)

Ontological concept relations by their turn were divided according to:

[4]Concept intension means all characteristics that make up a concept. Concept extension means all the objects to which a concept refers. In ISO/FDIS 1087-1:2000 (Terminology work: vocabulary)

1. Spatial and temporal contiguity;

2. Influence relations (which encloses a causal component

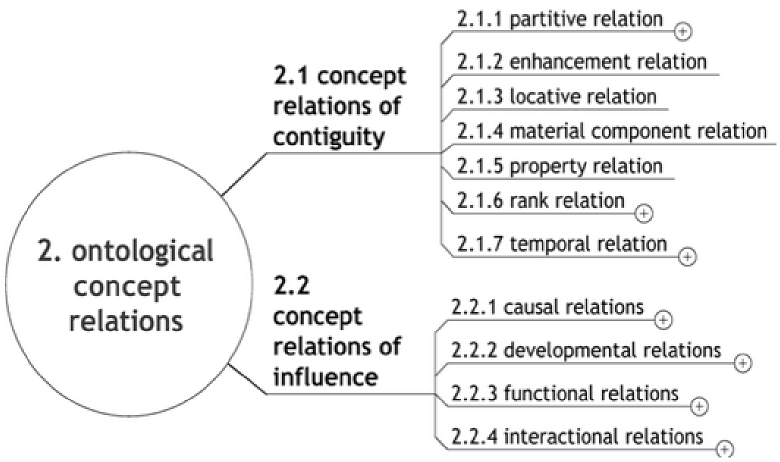


Figure 2.15: Ontological concept relations (Nuopponen, 2005)

According to the picture bellow, concept relations of contiguity cover relations that are based on some contact in space or time between concrete or abstract phenomena. By other words, that means the relation between A and B must not occur by means of a third concept C.

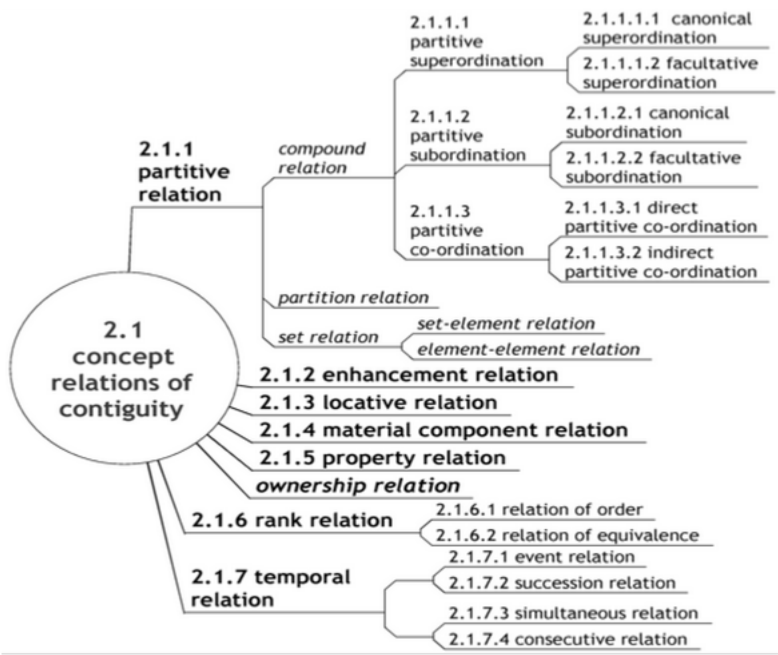


Figure 2.16: Concept relations of contiguity (Nuopponen, 2005)

Nuopponen (2005) continues with her classification proposal according to the next figures.

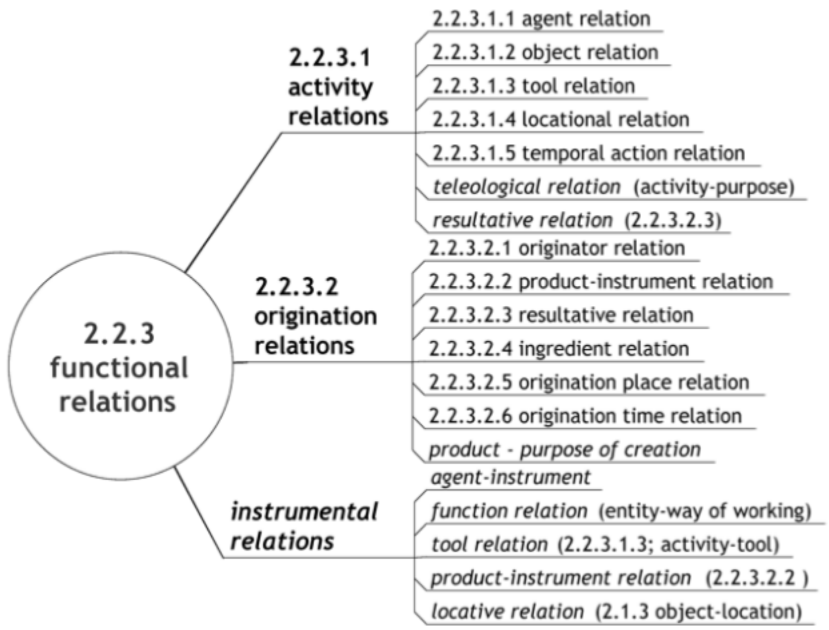


Figure 2.17: Influence relations (Nuopponen, 2005)

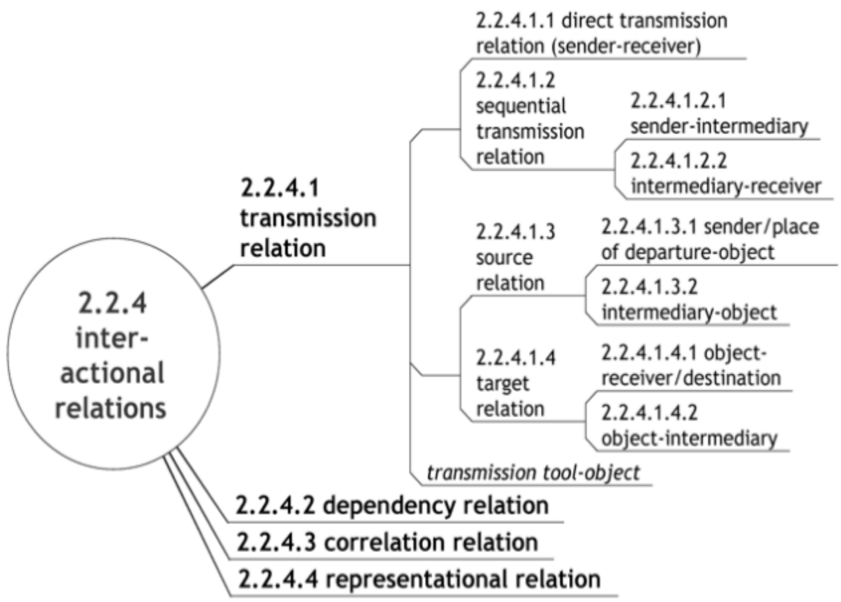


Figure 2.18: Functional relations (Nuopponen, 2005)

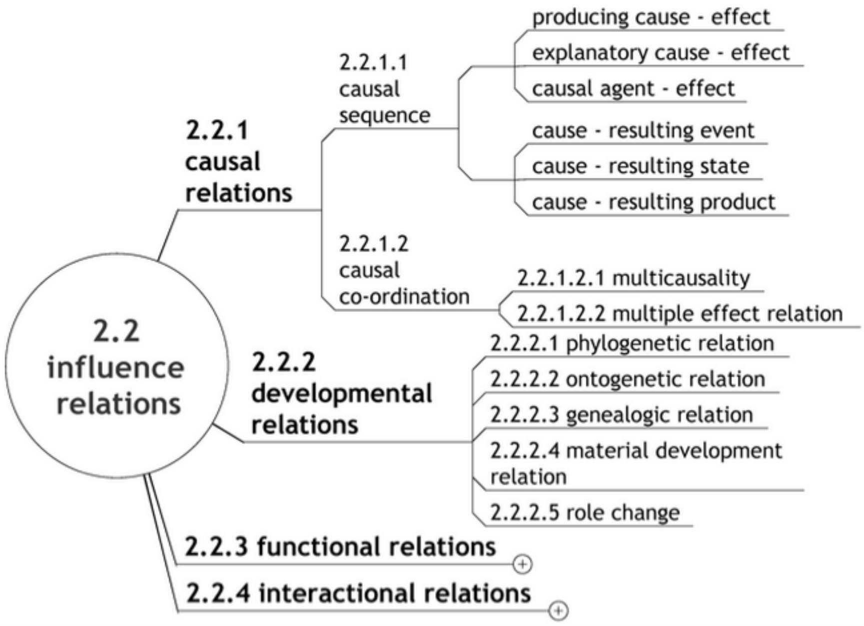


Figure 2.19: Inter actional relations (Nuopponen, 2005)

Other classifications are synthesized by Madsen et al. (2001), within two different fields, namely lexical semantics and terminology. Here we present two classifications used in EuroWordNet (was an EU-project with the aim of building a multilingual lexical database with wordnets for the following European languages) and SIMPLE (Semantic Information for Multifunctional Plurilingual Lexica – another EU project)

The UMLS (Unified Medical Language System) classification is also referred to as a 'classification of semantic relations', although it was designed to model concepts of the medical domain.

UMLS - version of 2010 - has two types of relations: the *isa* relations and the *associated_with* relations.

2.3.4 Cognitive models (cognitive semantics perspective)

The main theories of cognitive semantics are (Evans and Green, 2006): categorisation and idealised cognitive models (ICMs) (Lakoff, 1990), conceptual metaphor theory (Johnson and Lakoff, 1980), mental spaces theory and conceptual blending theory (Fauconnier and Turner, 1998).

From these, categorisation is one of the oldest and simplest with regard to structured organisation of knowledge, which was started by Aristotle. Wordnet lexical database defines categorisation as the "*basic cognitive process of arranging into classes or categories*", as Rosch and Lloyd (1978) state that "categorisation provides maximum information with the least cognitive effort", which "*is achieved if categories map the perceived world structure as closely as possible*". Following this, we focused essentially in the categorisation and idealised cognitive models (ICMs) in particular. Still, categorisation itself, dives into several theories of which the following stand out: i) classical theory; ii) conceptual clustering (refines classical approach based on set theory) and;

Table 2.4: Semantic relations in EuroWordNet (Madsen et al., 2001)

Semantic Relation in Euro WordNet	**Example**
SYNONYMY	ANTONYMY
near_synonym (not in same synset)	tool <> instrument
antonym	good <> bad
HYPONYMY	
has_hyponym	vehicle > car
has_hyperonym	car > vehicle
PART-WHOLE RELATIONS	
has_meronymy	(for underspecified cases such as: has as parts)
has_holonymy	(for underspecified cases such as : is a part of)
has_mero_part	hand > finger
has_mero_member	fleet > ship
has_mero_made_of	book > paper
has_mero_portion	bread > slice
has_mero_location	desert > oasis
has_holo_part	finger > hand
has_holo_member	ship > fleet
has_ holo_made_of	paper > book
CAUSE RELATIONS	
is_caused_by (for underspecified cases)	
causes (for underspecified cases)	
results_in to kill > to die	
for_purpose_of to search > to find	
enables_to vision > to see	
SUBEVENT RELATIONS	
is_subevent_of	to snore > to sleep
has_subevent	to sleep > to snore
INVOLVED/ROLE RELATIONS	
involved_agent	to bark > dog
role_agent	dog > to bark
involved_patient	to teach > learner
role_patient	learner > to teach
involved_instrument	to paint > paint-brush
role_instrument	paint-brush > to paint
involved_location	to swim > water
role_location	water > swim
involved_source_direction	to disembark > ship
role_source_direction	ship > disembark
involved_target_direction	rincasarse > casa
role_target_direction	casa > rincasarse
involved_result	to freeze > ice
role_result	ice > to freeze
involved_manner	shout > loudly
role_manner	loudly > shout

Table 2.5: Semantic relations in SIMPLE (Madsen et al., 2001)

Semantic Relations In SIMPLE	**Examples**
FORMAL RELATIONS	
is_a	(yacht, boat)
CONSTITUTIVE RELATIONS	
is_a_member_of	(senator, senate)
has_as_member	(flock, bird)
is_a_part_of	(head, body)
has_as_part	(airplane, wings)
instrument	(paint, brush)
relates	(kinship, brother)
resulting state	(die, dead)
is_a_follower_of	(marxist, marxism)
made_of	(bread, flour)
is_in	(oasis, dessert)
has_as_colour	(lemon, yellow)
constitutive activity	(bird, fly)
produces	(bird, egg)
produced_by	(honey, bee)
property of	(intelligence, intelligent)
concerns	(hepatitis, lever)
contains	(wineglass, wine)
quantifies	(bottle, liquid)
measured_by	(temperature, degree)
related_to	(second, two)
successor_of	(two, one)
has_as_effect	(storm, thunder)
typical_of	(distemper, dog)
causes	(measles, fever)
TELIC RELATIONS	
indirect_telic	(eye, see)
purpose	(send, receive)
object_of_the_activity	(book, read)
is_the_activity_of	(doctor, heal)
is_the_ability_of	(painter, pain)
is_the_habit_of	(smoker, smoke)
used_for	(crane, lift)
used_by	(lancet, surgeon)
used_against	(chemoterapi, cancer)
used_as	(wood, material)
AGENTIVE RELATIONS	
result_of	(loss, loose)
agentive_prog	(pedestrian, walk)
agentive_experience	(fear, feel)
caused_by	(infection, bacterion)
source	(law, society)
created_by	(book, write)
derived_from	(petrol, oil)

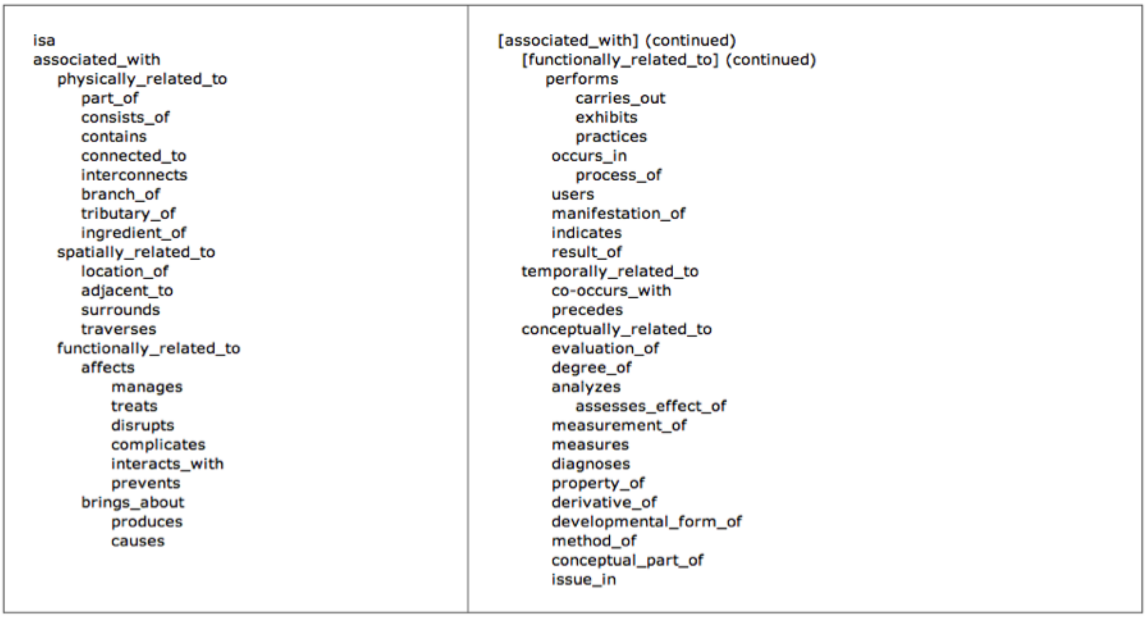


Figure 2.20: Current relations in UML semantic network

iii) prototype theory (which explores the family resemblance). According to the classical theory, categories are entities characterised by a set of properties shared by its members. Following this way to define categories it is impossible to find its typical elements. This gap is filled by prototype theory in which categories are formed by means of a typical model. Prototypes already includes the notion of mental images and provides some insight into the way we conceive certain ideas/objects but, no artefacts are provided to create and understand categories. Riemer (2010) synthesises the main drawbacks of prototype theory, namely: i) attributes can often only be identified after the category has been identified; ii) attributes are highly context-dependent and iii) there are many different alternative descriptions of the attributes of a given category. Afterwards, Idealised Cognitive Models (ICMs) emerged inspired by prototype theory aiming to explain the prototype effects upon categorisation. This was intended by means of image- schema diagrams which "*allow a rich description of meaning that seems to make contact with perceptual and cultural aspects of language – aspects which are easily left out in other types of analysis*" (Riemer, 2010).

ICMs, proposed by Lakoff (1990), structure mental spaces (conceptual ‘packets’ of knowledge constructed during ongoing meaning construction). As Lakoff observes, a mental space is a medium for conceptualisation and thought and ICMs provide the background knowledge that structures those mental spaces.

Categorisation could be implemented through ICMs which provide the general principles on how to organise knowledge. In order to deal with categories we may use cognitive models of five kinds: 1) propositional models, which specify elements, their properties and relations among them; 2) Image-schematic models, which specify schematic images; 3) Metonymy models, which map an element of a model to another; 4) Metaphoric models, which map from a model in one domain to a model in other domain and; 5) Symbolic models. Each ICM model comprises one or more schema, which provide a specific structure through which we understand the world. From these models, image-schemas (from image schematic models) are the most basic and concrete of all, intimately linked to both bodily and linguistic experience on knowledge structures construction.

As asserted by Riemer (2010), image schemas are usually represented diagrammatically and are particularly useful as representations of the meanings of prepositions. Hereupon, ICMs are especially suitable to withstand graphical knowledge representation, providing the basic constructs to build and understand conceptual structures.

Despite of ICMs, throughout image schemas, provide sufficient valuable arguments to support the construction of conceptual models, there could be ambiguity of diagrammatic representations as it could happen when drawing informal concept map to model a domain. We could fall into the situation in which we have the same representation for two different meanings, even using ICMs. The view outlined here, is that ICMs won't be the ultimate solution for conceptual modeling but they could provide common structures of visual representation (non-textual) of knowledge from which we could extract relations types among structures of meaning, and define a catalog of conceptual relations regarding its in conceptual modeling of several different domains.

According to "Specialisation of form hypothesis", which is described by Lakoff (1990) it is possible to find specific conceptual structures associated to ICMs: "*Strictly speaking, the Spatialization of Form hypothesis requires a metaphorical mapping from physical space into a 'conceptual space'. Under this mapping, spatial structure is mapped into conceptual structure. More specifically, images schemas (which structure space) are mapped into the corresponding abstract configurations (which structure concepts). The Spatialization of Form hypothesis thus maintains that conceptual structure is understood in terms of image schemas plus a metaphorical mapping.*"

Table 2.6 summarises the exercise of of making the correspondence between cognitive models and its associated structure and enclosed relation type, according with the Specialisation of form hypothesis.

Table 2.6: Cognitive models associated structure and relation type

Cognitive model	Associated structure	Type of conceptual relation
Container	Is_A hierarchy	Taxonomy
Container	Nested hierarchy	Class inclusion
Part-whole	Has_A hierarchy	Partonomy
Front-back	Radial	
Center-periphery	Radial	Participation
Source-path-goal	Linear hierarchy/Non-branching hierarchy	Interface
Link	Non-hierarchical	ER

2.3.5 Ontological perspective to conceptual relations elicitation

Although the specification of conceptual relations have been approached according to different theoretical perspectives, of conceptual relations still as a relevant and interesting research topic, though hard to approach. The interest on relations specification is increasing within the research community when conceptualizing a specific domain, either for creating knowledge base systems or upper-level ontologies.

A closer look into the literature revealed that ontology engineering together with the development of knowledge bases form the main research topics in what regards to the representation of knowledge, wherein terminology plays a base role. Among the reported research, the elicitation of conceptual relations is addressed with variable emphasis and in different forms. There are several researches addressing the construction of ontologies grounded in terminology (Gillam et al., 2005; Yu-Liang, 2007; Buitelaar et al., 2009) . In Gillam et al. (2005), terminology plays the role of term system provider which act as input for the construction of the ontology. The authors propose an automatic process to identify a tree of lexical related terms, which constitute a candidate conceptual structure. Yu-Liang (2007) motivated by the lack of reference models in the process of building ontologies, presented a three step process, grounded in extraction techniques and textual corpus analysis, comprising: i) recognize terminology in text (using statistical analysis and association rules created using TexAnalyst software, plus semantic network analysis, in order to overcome the problem of ignoring the terms with a low frequency) ; b) name tags in terminology (in order to face the synonyms or variance issue. Repertory Grid Technique was used); c) derive hierarchies (using Formal Conceptual Analysis). His stance is that "*linguistic perspectives should be considered while building ontologies*". Further he underlines the need to develop a 'lightweight ontology' which "*is a schema like taxonomy which comprises a conceptual system used to model knowledge. Consequently, ontology editors must first construct a conceptual system, after which editors should identify hierarchical structures among concepts*".

Buitelaar et al. (2009) argue - once again - that ontologies should be grounded in linguistics. The goal was to enrich current formalisms such as RDFS/OWL to include linguistic information such as "part-of-speech metadata of the lexical items", morphological information and variations, expressed as RDFS/OWL properties.

One of the main areas where terminology interacts with ontology engineering is that of ontology learning. As mentioned by (Buitelaar et al., 2009) "Term extraction is a prerequisite for all aspects of ontology learning from text". However we consider that the use of terminology within knowledge representation contexts is wider than the use given by ontology learning field, that is, mainly corpus tagging for information extraction. Learning ontological relations is the most recent target in the scope of ontology learning. This is a fact that the identification of relations between concepts has a significant importance in the creation of artefacts to represent a specific domain. Other authors place conceptual relations as the core issue either on developing ontologies (Alvarez et al., 2007; Faber et al., 2009) or in representing a conceptual system in general (Elsayed, 2009; Auger and Barriere, 2010; Storey, 2005).

2.3.5.1 Foundational ontological analysis

As mentioned by Gruber (1993), an ontology is a "*formal, explicit specification of a shared conceptualisation*". Additionally, he states that “an ontology is a description of the concepts and relationships that can formally exist for an agent or a community of agents”. Citing (Arvidsson, F. and Flycht-Eriksson), an ontology "*renders shared vocabulary and taxonomy which models a domain with the definition of objects and/or concepts and their properties and relations*". More recently, Klein and Smith (2010) provided a definition of ontology, towards an effort for a consensual terminology for ontologies, in the context of the development of domain specific (technical) terminologies. The authors considered an ontology as "*a representational artifact, comprising a taxonomy as proper part, whose representational units are intended to designate some combination of types, classes, and certain relations between them*". It is quite evident, within scientific community, that relations are a fundamental component on building ontologies. This section aim at find out answers for the following questions:

- How relations are treated when building an ontology?
- Is there a set of fundamental ontological relations which could be used in the construction of any ontology despite of its domain?

Hence, it was followed a study on the main foundation/upper-level ontologies. The basic idea was that, such as ontologies describe the very general concepts that are the same across all knowledge domains, it would be possible to get some insights about a set of domain-independent relations to be used. "*Ontologies are often equated with taxonomic hierarchies of classes, class definitions, and the subsumption relation*" (Gruber, 1993), however the goal was to identify other than only these.

According to Guizzardi (2005) a reference ontology (foundational ontology) could be used to help the representation of a conceptualisation in an explicit and formal way. The concept of formal ontology was initially defined by Edmunf Husserl. A formal ontology deals with formal ontological structures, namely: the theory of parts and wholes, types, instantiation, identity, dependency and unity.

The issues related to knowledge representation are addressed by scientific community, within informatics engineering domain for decades (Brachman, 1983; Sowa, 2000). However its foundations are much older, based on philosophy, and come from one of the oldest (if not the oldest) ontology, developed by Aristotle: Aristotle’s categories. The main goal of an ontology is to represent or to support the representation of a certain reality by providing a set of basic categories. According to Hjorland (2007) "*the most commonly used semantic relations have resemblance with lists of categories,*" either grammatical or with for example, Aristotle’s 10 categories:

- substance (fundamental entities)
- quantity
- qualification (quality)

- a relative (relation)
- where (place)
- when (time)
- being-in-a-position (position)
- having (state)
- doing (action)
- being-affected (affection

After Aristotle several other authors presented their view about a system of ontological categories (Thomasson, 2012). One of them was Lowe (2005). He argues that are exactly four categories namely **object**, **modes**, kinds and **attributes** (as is the next list). These are related in such a way in what Lowe calls ’the ontological square’.

- Entities[5]
  - Particulars[6]
    - **Objects**
      - Substances[7]
      - Non-Substances[8]
    - **Modes**[9]
  - Universals[10]
    - **Kinds**
    - **Attributes**

The Arstotle’s ontological square summarises and provides a categorical schema, recurrently used in foundational ontologies for representing a reality and its entities. This four-categorial scheme is obtained by crossing two formal distinctions which underpin conceptual modelling languages and top-level ontologies alike: that between universals (or types or kinds) and particulars

[5] An entity is the top-level or root category of the things that exist.

[6] Particulars refers to a specific object in the real world. Particular means being an individual. Particulars are also called instances or tokens or simply objects.

[7] Substances are also called constituents or endurants corresponding to the entities that exist independent from time.

[8] Also known as occurrent or perdurant or accident it refers to the entities that partially exist on a certain time-frame moment (e.g. a process or an event)

[9] Modes (or moments) are instances of attributes, that is, the combination between an attribute and its bearers. Also called individual accidents

[10] Universals are the kinds of things that exist in real world, that is, recurrent entities sharing some characteristics that could be instantiated or exemplified by more than one particular thing. Universals could be seen as abstract entities and we may think about universals as concepts (Earl, D., 2005) representing a certain view of a particular reality. Universals are also called types.

(instances or objects or tokens) on the one hand, and that between features[11] (characters), and their substrates14 (or bearers) on the other hand. "*Thus the Ontological Square consists of particular substrates, called substances, and universal substrates, called kinds, as well as particular characters, called modes or moments, and universal characters, called attributes.*" (Schneider, 2009).

	Substrates	**Characters**
Types	Kinds (e.g. *Man*)	Attributes (e.g. *Wisdom*)
Tokens	Substances (e.g. *Socrates*)	Modes (e.g. *Socrates' wisdom*)

Figure 2.21: Ontological square sort of things. (Schneider, 2008)

Following this, there are some basic relations that could be obtained among entities in the ontological square, namely (Lowe, 2005) :

- Modes (Individual accidents) **inhere in** objects (individual substances);
- Attributes (non-substance universals) **characterize** Kinds (substance universals)
- Objects (individual substances) **instantiate** kinds (substance universals);
- Modes (individual accidents) **instantiate** attributes (accident universals);
- Objects (individual substances) **exemplify** attributes (accident universals);

In his book 'The Four-Category Ontology' Lowe (2005), grounded on the ontological square, goes further and identify some formal ontological relations[12], in the context of an account of ontological categorization. The author considers that at least the following different relations need to be invoked: identity, instantiation, characterization, exemplification, constitution, composition and dependence.

Kless (2011) on his presentation at the 10th European Networked Knowledge Organisation Systems (NKOS) Workshop, gives an interesting summary on foundational ontology relationships, depicted below (see figure 2.22.

Schwarz and Smith (2008), continue with a characterisation of the relations among entities in those categories at the level of universals, which, according to the authors are the ones with more interest for ontologies. Either due to temporal dimension issues or due to the way that, sometimes, the relations between generic terms despise the way that in reality the instances (particulars) relate each other, Ulf Schwarz and Barry Smith found the need of a more rigorous and unambiguous definition of a basic set of ontological relations applied to generic terms. The most basic ontological relations (considered primitive and domain-independent), were typified by the authors as following:

[11] The features of something (universals or particulars) are called characters or simply properties/attributes.

[12] Formal ontological relations are a set of basic relations between entities belonging to the same or different ontological categories.

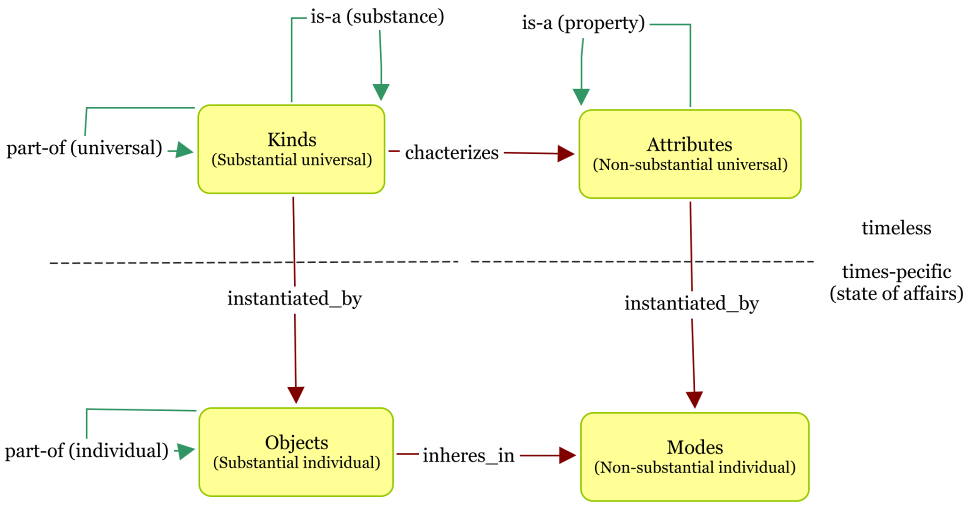


Figure 2.22: Fundamental Ontology Relations. (kless, 2011)

- subsumption (*subtype*);
- parthood (*part-whole*);
- participation and;
- class-inclusion.

For each one of these relation types, a way of expressing/naming it, together with its definition was provided by the authors:

- subsumption (subtype) relation could be expressed by “*instance_of*”;
- parthood (part-whole) relation could be expressed by “*part_of*”;
- participation relation could be expressed by “*has_participant*”, and;
- class-inclusion could be expressed by “*located_in*”

The authors also distinguish between **primitive relation**s and **logical relations**, being the last ones, the transformation of the first, according to its inverse specification:

- subsumption (subtype) has “*has_subclass*” as logical relation and;
- parthood (part-whole) has ”*has_part*” as logical relation;

These short ontological notions are commonly used in foundational ontologies, hereupon, it were considered the main upper-level ontologies, namely: CyC[13], BFO[14], GFO[15], UFO (Guiz-

[13]http://www.opencyc.org/
[14]http://www.ifomis.org/bfo
[15]http://www.onto-med.de/ontologies/gfo.html

zardi, 2005), SUMO[16], COSMO[17], DOLCE[18], PROTON[19] and Sowa's ontology, in order to gather a common basic set of ontological relations.

In Sowa's ontology there is only one primitive relation - the relation 'Has'. But, when entering in a more detailed analysis of these ontologies it was found a considerable difference between their conceptual structures (regarding the size and content). COSMO Ontology, for instance, has over 700 relations and 6400 Classes and its conceptual structure is translated into hierarchical relations. BFO is much smaller but contains only taxonomic relations. Yet, GFO and even UFO, provide a more interesting conceptual structure that goes beyond a taxonomy. Other issue on this study was the fact that some of these ontologies overlap each other. COSMO uses elements from CyC, SUMO, BFO and DOLCE. BFO, for example, overlaps DOLCE and SUMO. By its turn, the second version of UFO combines elements from DOLCE. A more detailed catalog of the relations which each ontology uses, could be found in the annex accompanying to this report.

To fully approach "upper-level" ontologies, GOL must be considered. GOL, stands for General Ontology Language. It is a modeling language containing the set of ontologically basic categories aiming at provide a system of formalized and axiomatized top-level ontologies which can be used as a framework for building more specific ontologies. GOL consists of a syntax, and of an axiomatic core which captures the meaning of the introduced ontological categories. GOL shares some of its development history with BFO but currently it is a complete independent project (Degen and Herre, 2001). Once GOL is a language for building ontologies grounded on a set of ontologically basic categories, then, its syntax could give some clues on the set of basic formal ontological relations. The figure below illustrates both the symbols and relations available in GOL language (figure 2.23).

$\in$	(membership)	$::$	(instantiation)	$:\succ$	(inherence)
$<$	(part-of)	$\leq$	(reflexive part-of)	$:\mu(x,y,z)$	(rel. part-of)
$:\sqsubset$	(framing)	$:\perp(x,y)$	(foundation)	$:\triangleright$	(is contained in)
$:occ(x,y)$	(x occupies y)	$:ass(x,y)$	(y ass. to y)	$ext(x,y)$	(is extension)

Figure 2.23: Symbols for binary and ternary basic relations (Degen et al., 2001)

A closer look at the figure above, three main categories could be identified, one per line. It could be said that *membership*, *instantiation* and *inherence* share the same root. All of them establish relations of dependence. The second line, following a top-down reading, contains relations connecting parts and wholes, whether the parts exist or not apart from the wholes. The next group (*framing*, *foundation* and *is contained in*) share a containment dependence root. At the end, a location or spatial dependence is the common part of *occupies*, *ass. to* and *is extension* relations. Identical conclusions were achieved by Gangemi et al. (2001). The authors have focused on top-level ontologies, and identified a set of primitive formal relations towards a sys-

[16]http://www.ontologyportal.org/
[17]http://micra.com/COSMO/
[18]http://www.loa.istc.cnr.it/DOLCE.html
[19]http://proton.semanticweb.org/

tematic methodology for selecting general ontological categories to be used for multiple practical purposes. Accordingly, the following formal relations were identified:

1. Instantiation and membership;
2. Parthood;
3. Connection;
4. Location and extension and;
5. Dependence;

Summarising, and considering the review of the literature the following ontological categories of formal relations were selected: constitution and containment dependence, existential dependence, generic dependence, historical dependence (Thomasson, 1999). In this work it will not be considered the Existential Dependence since it has to do with relations between entities and its examples and the intent, at this level (conceptualization), is to avoid mixing classes (concepts) with its instances. But, in fact, the individuals which belongs to a specific category should be known in order to a new category/concept be added accurately. Constitution and Containment dependence was detailed as a Part-Whole conceptual relation as it is more common across literature. Following the same purpose, generic dependence was detailed into the Generic-Specific category. Historical dependence is related with temporal location relations. These kind of relations are treated differently (in terms of each taxonomy of categories used) in the available upper-lever ontologies. Historical dependence could have a space or time boundary considering physical or non-physical objects respectively, hence it was decided to detail it into two more specific conceptual relations, namely: Temporal Conceptual Relation and Spatial Conceptual Relation. Inspired mainly by GFO, it was decided to include Participation relation. Participation could be considered as an extension of historical dependence relation, however, in the context of collaborative networks of organizations, participation relation has an important role on offering an orthogonal view of an event or process. It can also offer a brief overview on the social interaction network around an event or process. Finally it was also considered the Cause-Effect Conceptual Relation. Casualty could easily be associated to space and time relations to describe events and consequently considered as not adding value for the current purpose. However, Cause-Effect Conceptual Relation is fundamental to add some dynamicity to conceptual representations on describing phenomenons and agents of change within some process or event or object state. Finally, it was achieved the following taxonomic:

- Constitution and Containment Dependence
  - Part-Whole Conceptual Relation
  - Containment Conceptual Relation
- Generic Dependence

  - Generic-Specific Conceptual Relation

- Time and Space Dependence

  - Spatial Conceptual Relation
  - Temporal Conceptual Relation

- Cause-Effect Conceptual Relation

- Participation Conceptual Relation

- Usage Conceptual Relation

Chapter 3

Research design

"Though this be madness, yet there is method in it!"

Shakespeare in *Hamlet*

Research design comprises a plan to conduct a study to tackle a specific scientific problem, detailing how to acquire the knowledge to develop innovative artifacts "*through which the analysis, design, implementation, management, and use of information systems can be effectively and efficiently accomplished*" (Hevner et al., 2004).

Research design is common to all scientific domains enclosing several approaches, either following natural or artificial studies. "*Whereas natural science tries to understand reality, design science attempts to create things that serve human purposes. It is technology-oriented. Its products are assessed against criteria of value or utility.*"(March and Smith, 1995).

"*Design science is fundamentally a problem-solving paradigm. It seeks to create innovations that define the ideas, practices, technical capabilities, and products through which the analysis, design, implementation, and use of information systems can be effectively and efficiently accomplished.*" (Hevner and Chatterjee, 2010). The research challenge raised in this thesis is artificial as opposed to natural phenomena and calls for novel IT artefacts (methods and/or processes and tools) to assist a set of identifiable human needs, . Accordingly, this research thesis embraces a design science paradigm to address the formulated research problem, whose characteristics are artificial and technological based. Thus, this section explains how the research was conducted and how its outcomes analysed and validated following a design science research approach.

3.1 Design science research overview

Within design science scientific domain, Hevner et al. (2004) argue that there are two kinds of approaches to deal with IS research: behavior science and design science. According to the authors, behavior science "*seeks to develop and verify theories that explain or predict human and* *organisational behavior*", whereas design science "*seeks to extend the boundaries of human understanding of human and organisational capabilities by creating new and innovative artifacts.*". The main difference is that "*While behavioural IS research aims at 'truth' i.e., at the exploration and validation of generic cause–effect relations, IS design science research aims at 'utility" i.e., at the construction and evaluation of generic means–ends relations.*" (Winter, 2008).

This thesis does not fit into the explanatory principles of research, seeking to discover new theories or explain a particular phenomenon, nor it is an engineering solution for a specific design problem. The awareness of current research (from literature review) push this research project beyond the theories, on pursuit of innovative artifacts for a specific type of unsolved problem. In fact, there are already some theories explaining collaboration strategies for information and knowledge management within networked organisations, however, the methods and the tools to support the underlying social processes leading to a shared conceptualization, still scarce.

Focusing on design science research, Hevner (2007) discusses an approach based on a complementary 3-cycle model: (1) relevance cycle; (2) design cycle and; (3) rigor cycle.

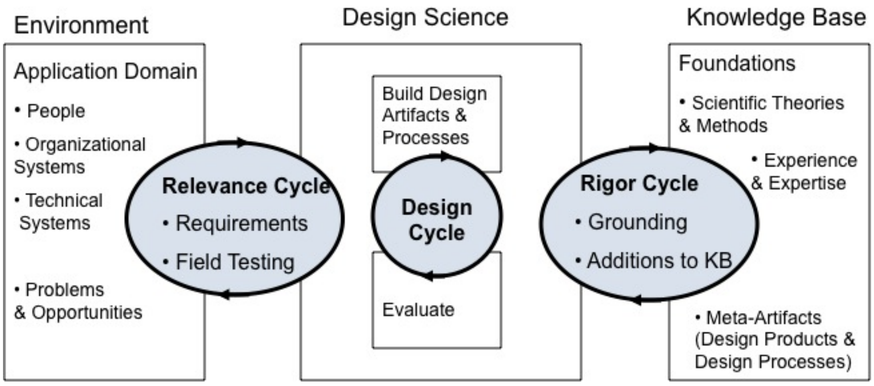


Figure 3.1: Design Science Research cycles (Hevner et al. 2007)

As figure 3.1 portrays, design science applies to a socio-technical perspetive from where the requirements for the research come and to where the design artifacts should be returned. This is ensured by this 3-cycle approach:

Relevance cycle DSR acquire knowledge from the environment, in order to develop an innovative artifact, thus "*good design science research begins by identifying and representing opportunities and problems in actual application environments*". The relevance cycle initiates the design science having as input the requirements for the research, that means, problem to be addressed (Hevner and Chatterjee, 2010).

Rigor cycle the rigor cycle is where the outcomes are grounded according to scientific theories and methods and the contributions are discussed, enriching the scientific knowledge base.

Design cycle "*Iterates between the construction of the artifact, its evaluation, and subsequent feedback to refine the design further.*" (Hevner and Chatterjee, 2010).

3.2 Applying Design Science Research: overview

Figure 3.2 depicts how DSR principles and methods were instantiated in the course of this research.

The research process begins by identifying a scientific problem, continuing to its characterization (i.e., challenges and opportunities). Based on this, the research problem and research questions were formulated. Afterwards, the justification and definition of the scope research strategy is provided, both to answer the posed research questions and to validade the formulated problems and objectives. This was achieved by the literature review together with experience coming from previous research. After the closure of the relevance cycle, the focus was given to a specific part of the literature, in order to identify the class of artifacts that will meet the aims and goals of this research project. The information is synthesized and the design of the artifact starts. At this particular stage it is very important to start defining the evaluation strategy.

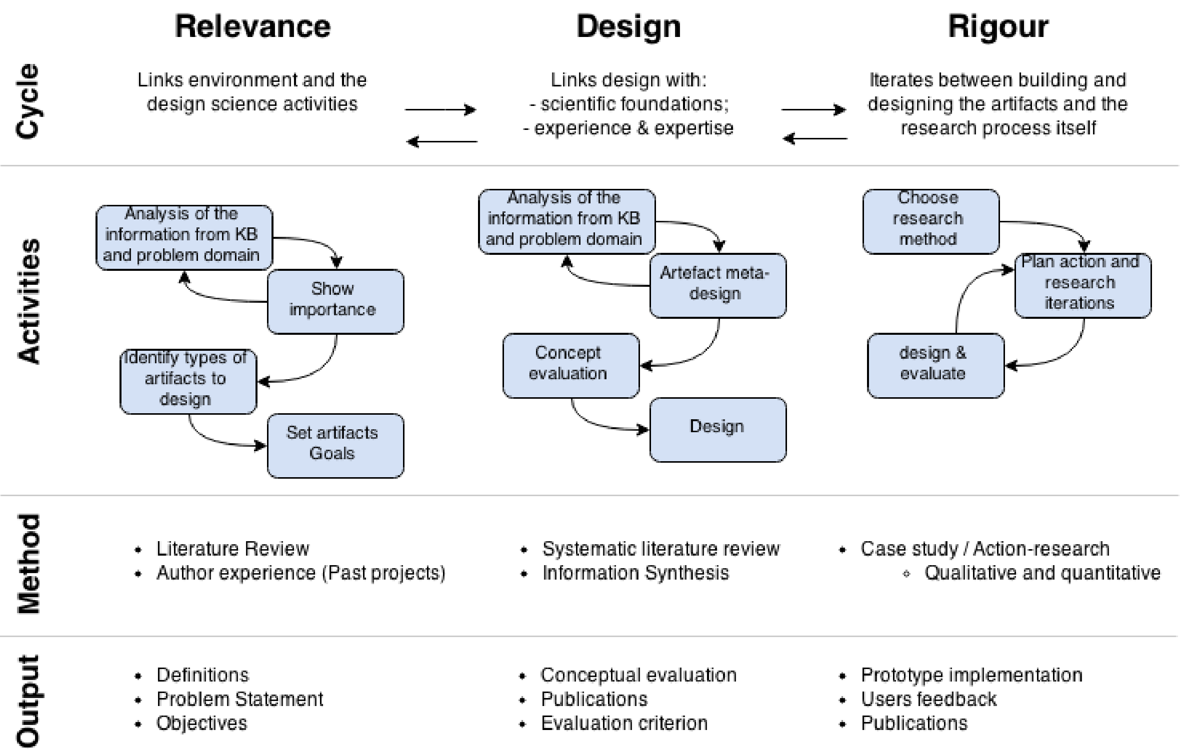


Figure 3.2: Followed design science research design

The evaluation strategy in design science research is focused in two main components:

1. the evaluation criteria
2. the process

Therefore, while the design cycle is concerned about the evaluation criteria (i.e., the "what"), the rigor cycle is focused on the evaluation process and its methods (i.e., the "how").

In what regards to the first component, a set of criteria, inspired on semiotics were selected, namely:

- Usability and completeness, related to the use (pragmatics level).
- Easiness to understand and interpretability, at the semantic level.
- Consistency, in relation to the representation structure.

These criteria are based in the Helfert et al. (2012) approach based on semiotics where they discuss a set of criteria for each semiotic level (pragmatic, semantic and syntax) (see the table of the figure 3.3).

Semiotic Level	Example Criteria
Pragmatic	Relevance, usability, completeness, timeliness, actuality, efficiency,
Semantic	Precise definitions and terminology, easy to understand, interpretability, accuracy (free-of error), consistent content
Syntax	Consistent and adequate syntax, syntactical correctness, consistent representation, accessibility

Figure 3.3: Design evaluation framework (Helfert et al. 2012)

These criteria correspond to quality attributes required for all the conceptual representations that result from typical conceptualisation process. This research aims to ensure these quality attributes, by developing a set of artifacts that will be available to domain experts during their conceptualisation activities. Thus, each artifact built under this thesis, might contribute differently but in a complementary way for conceptual representations' quality. The selected criteria, the artifacts contributing to them and the way they contribute, are explained in more detail in the next section (section 3.2.1).

Regarding the second component (i.e., the process), Hevner et al. (2004) discuss around methods and the corresponding evaluation processes, mapping them as the table 3.1 shows.

Table 3.1: DSR: Evaluation processes and methods (Hevner et al., 2004)

Method	Evaluation process
Observational	Case or study filed
Analytical	Static analysis, architecture analysis, optimization, Dynamic
Experimental	Controlled experimental
Testing	Functional (black box), structural (white box)
Descriptive	Informed argumented scenarios

Therefore, in the perspective of the processes and the underlying methods, current research project follows, mainly, observational methods focusing on qualitative evaluation approaches. Observational methods are considered the most suitable having into account the fundamental research assumptions, whose basis is grounded in socio-semantic principles. The research problem statement itself is focused on "what" and "how" research questions, rather than "how many". In other words, the main interest is on the semantic processes and tools that provide the adequate support to the social construction of conceptual representations, ensuring: a) its effective use in real IS contexts and; b) its reusability, either for reconceptualisation purposes or information management purposes.

3.2.1 Relevance and rigor

The main DSR challenge is to maintain the consistency of the relation between the relevance of the research problem and the application of rigorous methods in both construction and evaluation of the design artifact. Meanwhile, the aims and goals should not be neglected, but rather used to mediate the research flow towards answering the posed research questions.

For Winter (2008), "*IS DSR aims at 'utility',i.e., at the construction and evaluation of generic means–ends relations*" to achieve certain goals in order to satisfy some relevant research needs. Establishing these relations could be a way of attaching the three research cycles to the same path. Figure 3.4 shows, generically, how to map in this thesis *the means* and *the ends* according to the DSR framework.

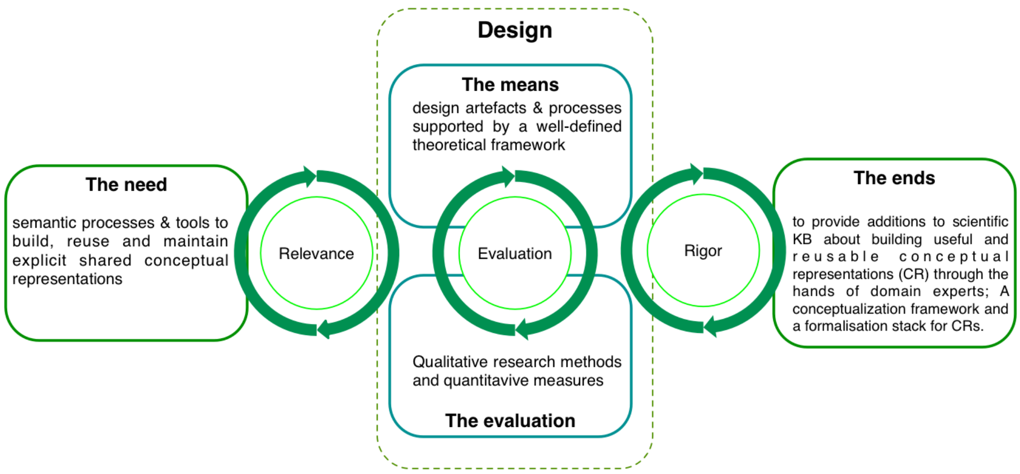


Figure 3.4: Ensuring rigor through means-ends relations: overview

3.2.1.1 The need

Figure 3.4 describes, briefly, the central need debated in this thesis, identifying explicit shared conceptual representations as the main desired output of a collaborative conceptualisation process. The need to develop means to produce such an artefact implies to know its fundamental characteristics. As shown in figure 3.5, conceptual representations refer to a specific and concrete domain model, containing the concepts and the relations between concepts, and providing an understanding about a concrete situation. These representations should be agnostic as to the representation formalisms, and agreed (shared) within a community. This requires that the concepts and the relations that form and combine the conceptual structures, towards the final conceptual representation, to be explicit defined, i.e., the concepts do not exist in isolation, but interconnected by means of a set of meaningful relations and attached to a particular context.

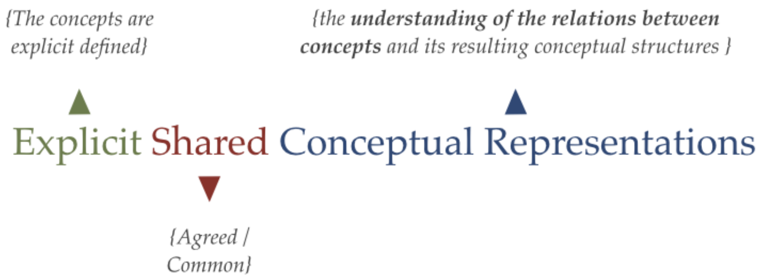


Figure 3.5: Explicit shared conceptual representations

3.2.1.2 The means

In general terms, conceptualisation is the abstract means to create, interprete and communicate signs and/or symbols through conceptual representations. Thus, conceptual representations themselves, might be seen as a specific case of semiotics, since it corresponds to a system of shared symbols by which a group of social actors communicate within the same context. In this particular case the sign is a term that designates (or simbolizes) a specific concept of the real world. Thus, considering that: a) a conceptual representation is both a domain model and a semiotic system, and; b) the quality of a conceptual representation indicates to which extent it is useful and reusable; semiotics principles could be seen as the umbrella that houses the theoretical framework (the concrete means) that leads to the construction of the research artefact(s). In fact, and considering the work done by Krogstie et al. (2006), semiotics might contribute to understand quality in conceptual representations, considering already the conceptual representations quality criteria (the what).

Krogstie et al. (2006), in his semiotic quality framework, defines several quality aspects considering the relationships between a model, a body of knowledge, a domain, a modeling language, and the modeling activities. The framework was initially developed grounded on Peirce's semiotics view and extended later according to the Stamper's *semiotic Ladder* including three more layers: the physical world, empirical and social layer (Stamper, 1996), wherein each layer identifies a specific semiotic quality to evaluate conceptual models.

Hereupon, the generic description about the means, i.e., the theoretical framework employed in this thesis to design and evaluate the research artefact(s) can be posed as follows:

The construction of useful and reusable conceptual representations, through collaborative conceptualisation processes, requires socio-semantics approaches based on semiotics principles and mediated by methods of terminological analysis and knowledge representation. Moreover, the cycle of development and evaluation of the artefact will follow qualitative evaluation methods combined with qualitative measures, builtin a comprehensive action-research approach.

The aforementioned theoretical overview might be better understood through the following representation (figure 3.6), which depicts the generic "*means and ends relations*" aiming at utility and reusability.

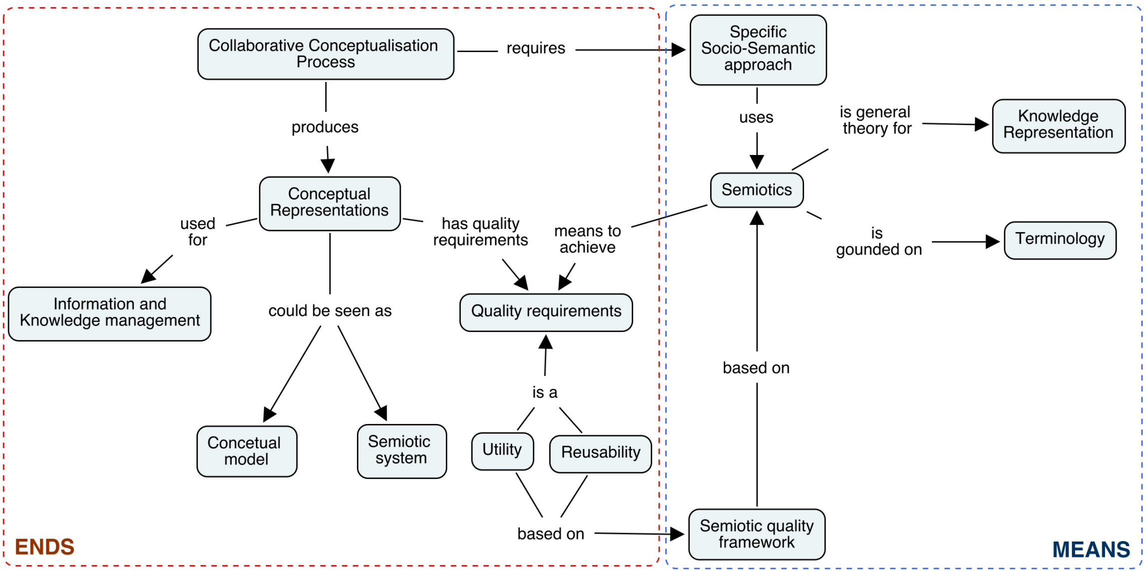


Figure 3.6: Theoretical framework for design cycle: means-ends relations

Semiotics is the common basis wherein current multidisciplinary theoretical approach makes sense and is understood, connecting the *means* to the *ends*. The table 3.2 synthesizes how semiotics contributes to achieve quality conceptual representations through concrete artefacts designed in a twofold perspective, combining knowledge representation and terminology in general.

3.2.1.3 The validation

On tying means-end relations, and in order to validate and refine our results, an hybrid approach will be followed combining qualitative and quantitative approaches. On one hand it was conducted qualitative research studies, to be carried out in different applications scenarios: (1) a transnational research project, and; (2) with a group of researchers, whose high level of expertise is well known in the domain of production planning and scheduling. On the other hand, it is intended to evaluate quantitatively the experiences and case results according to specific quality requirements. Additionally, a third experiment was considered in order to test and validate a specific part of the designed artifacts related to the use of specific terminological methods to aid conceptualisation activities.

Table 3.2: Contribution of semiotics to conceptual representations's quality

Semiotic quality	**Description**	**Artefact**	**Theoretical lens** (underlying theory)
Primary qualities			
syntatic quality	degree of correspondence between the model and the language	Conceptual Relations Reference Model	Knowledge representation (ontology engineering) and terminology
Semantic quality	degree of correspondence between model and domain, i.e., what is expressed in the model?	Competency Questions	Knowledge representation (conceptual graphs) and terminology (Query types and patterns)
Pragmatics quality	degree of correspondence between the domain model and the domain experts interpretations	KR templates	Knowledge Representation
Secundary qualities			
Physical quality	reusability (the ultimate goals); how a model is represented, stored and retrieved, i.e., its life-cycle	CG[1] Engine	Knowledge Representation and Semiotics
Empirical quality	readability of the domain model	CG Engine, transforming semi-informal KR to conceptual graphs	Knowledge Representation
Social quality	degree of agreement, i.e., is there a common agreement?		This is a transversal quality or the ultimate goal

The evaluation of the research is detailed in the Section 6.1. Basically the evaluation is divided in two approaches (reflecting two complementary evaluation stages): the verification and validation. The validation approach is focused on the produced artefacts (conceptual representations) whereas verification is concerned with the process. For the validation it was adopted a qualitative approach. The verification was qualitative based on action-research (AR) methods.

AR consists in a holistic approach of problem solving where knowledge is learned by working in a context of action and where people try to work together to address key problems in their organisations. Typically an AR based project involves more or less systematic cycles of action and reflection: in action phases co-researchers test practices and gather evidence; in reflection stages they make sense together and plan further actions.

The qualitative approach consists of gathering a set of metrics from the analysis of the conceptualisation results carried out during the case studies. Those metrics will be used to determine to which extend a conceptual representation satisfies the aforementioned quality requirements (i.e., utility and reusability).

3.3 Conclusion

This thesis is concerned with developing and evaluating new instruments and processes to ensure the involvement of the domain specialists in conceptualisation activities in producing useful and reusable conceptual representations. With this emphasis on producing artefacts that are useful to domain experts, it is argued that the most suitable research design is one adopting Design Science. Behavioral or explanatory research offers the best fit for a social science basis of this kind of research. The model of Design Science outlined by Hevner et al. (2004) is appropriate for this thesis purposes and so it was adopt their terminology and guidelines.

More concretely, the relevance phase was conducted by a review of relevant literature to justify the relevance of the problem. At the design phase, besides the literature, the experience and the scientific feedback conducted the design of the artifacts in a multidisciplinary approach. The justify/evaluate phase proceeds with an hybrid approach combining qualitative methods based on action-research study, followed by a quantitative evaluation examining the conceptualisation results produced in the case studies.

Part II

Design and Implementation

Chapter 4

Conceptual design of the research artefacts

"Design is a plan for arranging elements in such a way as best to accomplish a particular purpose."

Charles Eames

It was evidenced earlier in this thesis the importance of the conceptual relations in the conceptualisation process since they influence the organisation and, consequently, the interpretation of the conceptual structures, decreasing the probability of reusability and utility of the produced conceptual representations. For this reason, designing an artefact to assist the domain experts eliciting conceptual representations is the main research topic of this thesis. Following this, in this chapter it is discussed how to support conceptual relations elicitation, discussing the Conceptual Relation Reference Model (CRRM). Furthermore, a framework providing a structured and multidimensional view over the conceptualisation process, combining terminological and knowledge representation processes is described. Then, the Conceptual Graphs (CGs) formalism is reviewed, emphasizing how it is used to process and manipulate conceptual representations as well as to assist the definition of competency questions. Meanwhile it is reported how the role of terminology enhances the conceptualisation process results. The designed artefacts described in this section were accommodated in a technological platform, whose functional description was included to better understand the artefacts themselves and how they interact to provide an effective support to the conceptualisation.

4.1 ConceptME functional overview

ConceptME[1] is a technological platform where groups of users can find tools and resources to collaboratively develop conceptual representations (e.g., concept maps). The platform was designed to allow multi-domain modeling, conducted by activities characterized by social interaction, holding a twofold perspective commitment, aligning terminology and knowledge representations within technological collaborative spaces.

The conceptME started to be developed under the CogniNet[2] project in order to implement the ColBlend method (Pereira et al., 2013), designed to support the conceptualisation process. The artefacts designed in the scope of this thesis contributed to enhance the conceptME support on domain conceptualisation, both at knowledge representation and terminological level. Figure 4.1 depicts the functional architecture of conceptME, designed to manage the conceptualisation life cycle (CLC).

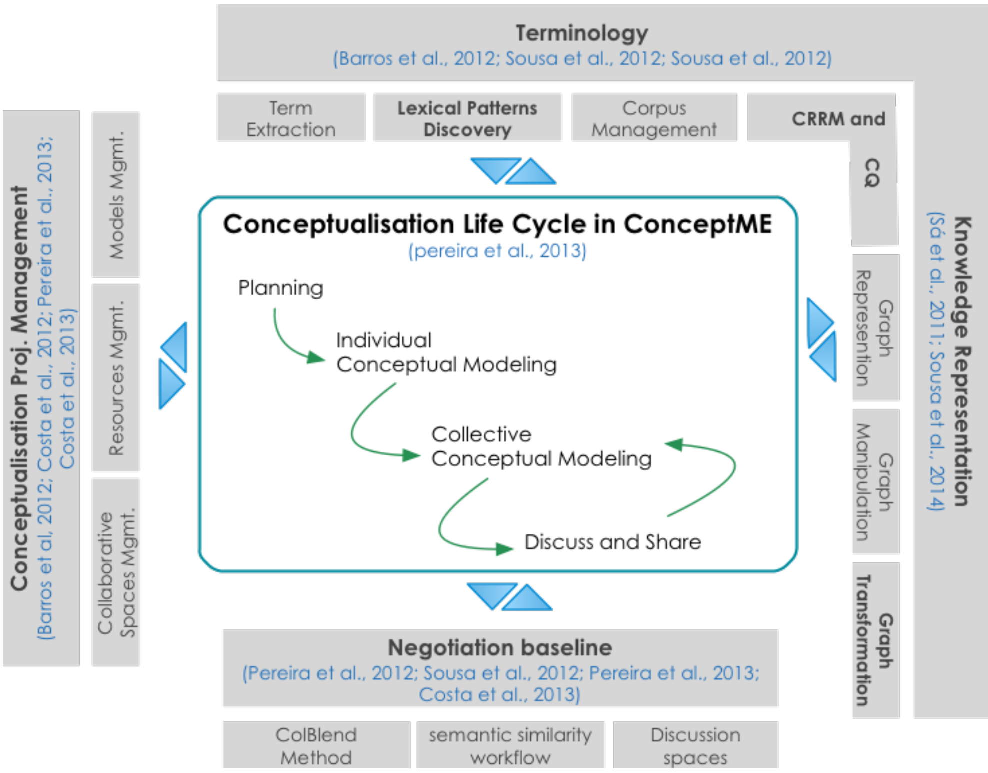


Figure 4.1: ConceptME functional overview

Inspired by ColBlend method, CLC comprises the following activities:

- Planning activities, which encloses structuring a conceptualisation project and defining its goals through competency questions.

[1] *http://www.conceptme.pt*

[2] The cogniNet was a project funded by the *Fundação para a Ciência e Tecnologia* under the contract PTDC/EIA-EIA/103779/2008

- Individual conceptual modeling, where the parties (groups) involved in the conceptualisation project, individually, build their proposal. A proposal is a conceptual representation, which might be published in a shared collaborative space and made available to all the other parties. On publishing an individual conceptual representation, the domain expert should inform about which of the existing competency questions are addressed in the shared proposal.
- Collective modeling. At this stage, all parties are involved. The main difference between collective modeling and individual modeling is on the negotiation features available. At the shared space there are features such as: specific discussion spaces (e.g., semantic forums) and a semantic similarity workflow, in addition to the terminological specific features.
- Discuss and Share. Discuss and Share and Collective modeling occurs in an incremental and iterative manner. The intention (or proposal) of any group to add or delete a conceptual structure, or to merge or detach conceptual representations, implies a discussion/negotiation cycle.

ConceptME follows a modular architecture based on services (as described in Section 5.2) and the modules are categorized as showed in Figure 4.1. In general the conceptME platform contains four main packages, described bellow:

Terminological package Terminological package implements the methods for term extraction, lexical pattern discovery and corpus management, allowing to carry out the following activities:

- Resource collection - allowing the domain expert to add and pre-classify new resources (e.g., documents, web page, . . .) to the platform.
- Concept organisation - when adding new conceptual structures, terminological methods are used, either to assist the identification of term candidates in corpus, or to retrieve the context of a specific term within the current conceptual representation, (e.g., to get clues on possible new concepts and relations). Term contexts can also support the discussion phase, justifying the use of a specific term, by showing evidence about that term's occurrence in the corpus.

Terminology is also in the basis of CRRM specification, together with knowledge representation discipline.

Knowledge Representation package Knowledge representation (KR) package implements the methods for visual graph representations (e.g., concept mapping), graph manipulation (i.e., functions to build and edit conceptual structures) and graph transformations. In addition it has implementations for CRRM usage and CQ specification) The KR methods endow conceptME with:

- a graphical-driven approach to the construction of conceptual representations;

- a user-friendly interface to interact with the produced content;
- a better navigation over the content, which might the aided through the conceptual representations themselves;
- mechanisms to ensure (at the end of the conceptualisation process) the formalisation of the conceptual representations.

Project management package Project management package is based on a structure driven by virtual spaces (or collaborative spaces):

a) the input spaces, which are private collaborative spaces where each group, involved in the conceptualisation process, models their conceptual representations proposals. Typically, there are as many input spaces as the number of involved groups;

b) the collective spaces, where each group shares its proposal. Typically there is one common space by project.

Furthermore, there are a space where it is possible to acquire and organise specific domain-related resources. Resources are collected either internally or from external sources and uploaded into a common repository, where they are classified using specific classification schemas (Barros et al., 2012).

Negotiation package Negotiation package implements a baseline to help domain experts reaching consensus around a particular conceptual representation. ColBlend method provides the process through which it is possible to ease reaching consensus.

4.2 On supporting conceptual relations elicitation

Representing a common conceptualisation requires an iterative and incremental process of building and organising conceptual structures, ensuring that they can be used and reused. For this, the conceptualisation process was decomposed in order to unveil its core needs. From the description of those needs (cf. Section 2.2) and according to the literature (cf. Section 2.3), conceptual relations are commonly understood as the cornerstone for building up the IT artefats that will provide support for the construction of *explicit shared conceptual representations*, e.g., *lightweight* ontologies. Hereupon, a Conceptual Relations Reference Model (CRRM) was developed providing a common baseline for conceptual representations construction. CRRM is an ontology used to build and assess conceptual representaions.

Figure 4.2 shows the conceptual view of the CRRM, according the conceptual relations study in Section 2.3. Figure 4.3 shows its implementation in Protégé, but the full ontology could be found in appendix A.

CRRM artefact includes the following information:

1. a taxonomy of conceptual relation types (or classes);

2. a taxonomy of classes of terms (*e.g., Part, Whole, Generic, Specific, Cause, Effect, Local,* ...);

3. a taxonomy of conceptual relation templates - one for each conceptual relation type. Each instance of a template encloses a conceptual structure in the basic form of "*concept - relation - concept*";

Beyond that, CRRM was built according to a threefold purpose:

1. to contribute, directly, to the conceptual relations elicitation, providing a common baseline for the creation of basic conceptual structures by means of the pre-defined templates;

2. to help on the definition of the conceptualisation goals, i.e., specifying the requirements or questions to which the final conceptual representation should provide an answer;

3. to provide a set of metrics for conceptual representations (i.e., the conceptualisation result) evaluation;

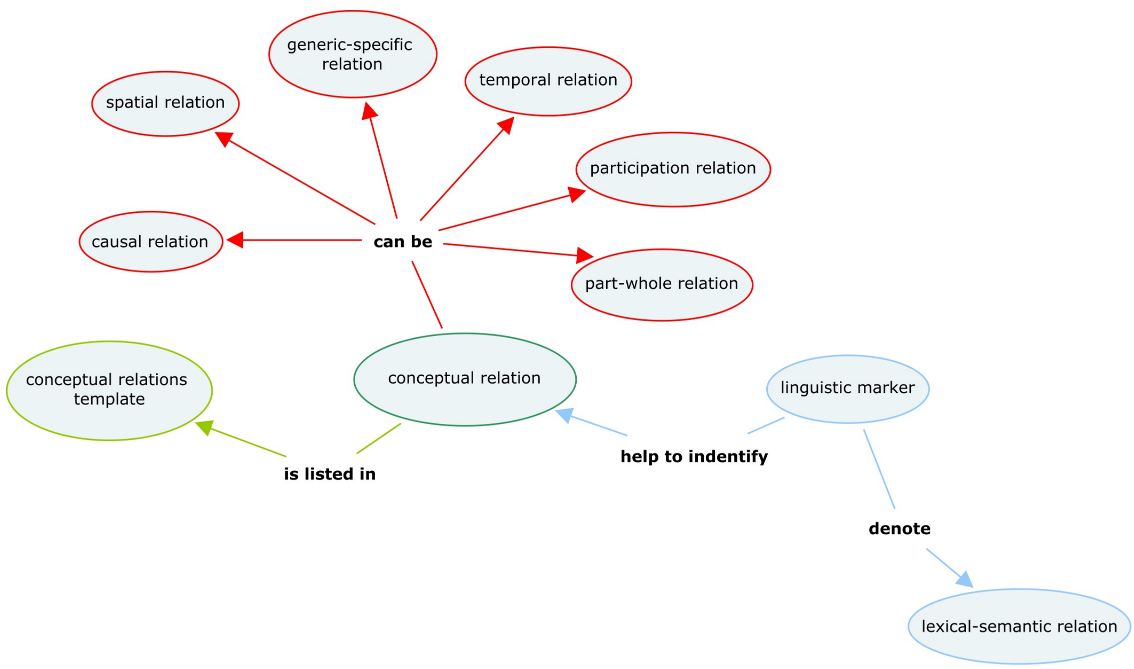


Figure 4.2: CRRM conceptual view

In CRRM, each conceptual relation is defined by: i) an intent; ii) a set o competency questions, and; iii) a linking phrase (derived from a linguistic marker) that designates and represents an instance of a conceptual relation. The intent is the goal or "usage scenario" of a certain type of relation, whereas the competency questions purpose is to define the scope of a conceptual relation. In this case it is possible to define more than one question.

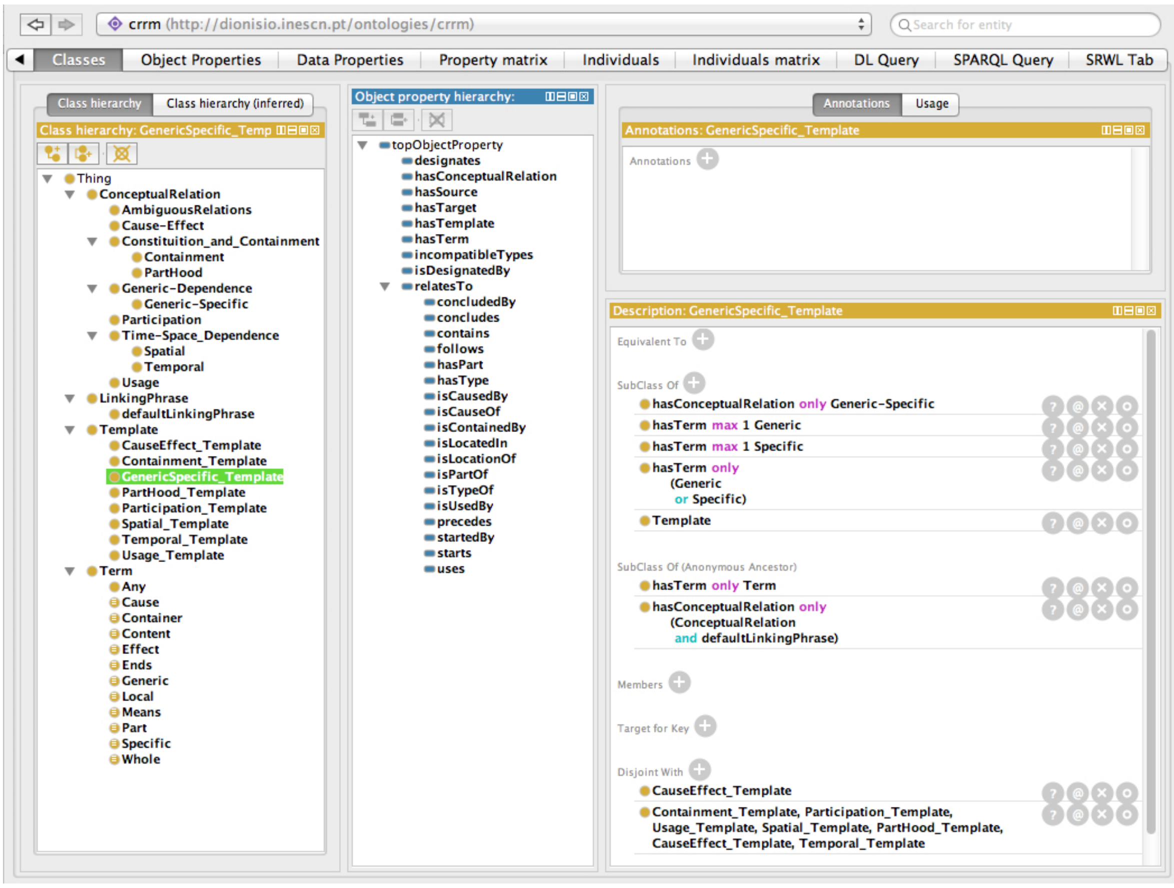


Figure 4.3: CRRM implementation in Protégé

CRRM assists users along the conceptualisation process through templates. A template consists of a specific type of a conceptual relation and two distinct terms, allowing users to instantiate new conceptual structures.

In practice CRRM is intended to be a baseline model which could be extended either by adding new linking phrases to designate conceptual relations or detailing the model adding more specific types of conceptual relations.

Additionally, and besides the class taxonomy (and each class restrictions), CRRM also includes a set of *Object Properties*[3] and SWRL[4] rules (see Table 4.3). *Object Properties* are used to describe Conceptual Representations for its evaluation rather than its construction. Still in a perspective of evaluating a conceptual representation, there is a sub-set of *Object Properties* that allow, through inferred rules (using SWRL), to discover new knowledge used to determine if conceptual relations carry ambiguity or inconsistency. In this scenario it is assumed that the evaluated conceptual representations were created based on CRRM.

Summarising, the procedure is as follows:

1. Conceptual representations are built by the domain experts and the conceptual structures are loaded into CRRM ontology (Table 4.1) and described using a set of *Object Properties*.

2. Through the reasoner engine, the SWRL rules are executed and extra knowledge is gathered (Table 4.2).

3. Simultaneously the SWRL rules try to identify relationships that are ambiguous or inconsistent. According to the example presented in Table 4.1, the "*is a part of*" linking phrase contains some ambiguity since it is both a descendant and a part of "*Engine*". So, the "*is a part of*" linking phrase is classified as AmbiguousRelations, according to the following rule:

   - $LinkingPhrase(?x)$, $Term(?y), Term(?z)$, $(not(Generic-Specific))(?x)$, $hasSource(?x,?y)$, $hasTarget(?x,?z)$, $isTypeOf(?y,?z) \rightarrow AmbiguousRelations(?x)$

4. Additionally, SWRL also helps to identify the incompatible terms. Two terms are incompatible if they have ambiguous incoming and outgoing conceptual relations.

At the end, the total number of terms, relations, linking phrases classified as Ambiguous Relations and the number of incompatible types of terms are the metrics used to determine the reusability degree of the conceptual representation (c.f. Chapter 6).

CRRM example: Let us consider a simple conceptual representation as depicted in the first column of the Table 4.1. Loading the conceptual representation into CRRM ontology implies defining a set of assertions using the existing *ObjectProperties*, as shown in the second column

[3] Object properties are relations between instances of two classes
[4] Semantic Web Rule Language

of Table 4.1. Afterwards, a reasoner (pellet[5]) is triggered and additional knowledge is gathered (Table 4.2).

Table 4.1: Load a conceptual representation in CRRM

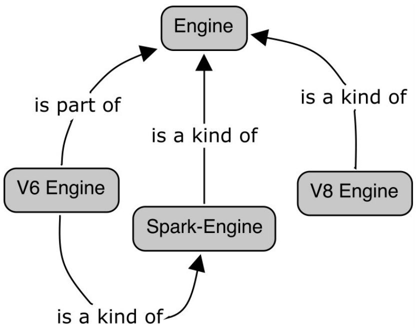
Conceptual Representation	**Asserted facts in CRRM**
	*hasSource*("is a kind of", V8 Engine) *hasTarget*("is a kind of", Engine) *hasSource*("is a kind of", Spark Engine) *hasTarget*("is a kind of", Engine) *hasSource*("is a kind of", V6 Engine) *hasTarget*("is a kind of", Spark Engine) *hasSource*("is a part of", V6 Engine) *hasTarget*("is a part of", Engine)

Table 4.2: Inferred a conceptual representation with CRRM

Concepts	**Inferred facts**
	precedes V6_Engine
	precedes Spark-Engine
	precedes V8_Engine
	hasType V6_Engine
Engine	*hasType* Spark-Engine
	hasType V8_Engine
	relatesTo V6_Engine
	relatesTo Spark-Engine
	relatesTo V8_Engine
	follows Engine
	follows Spark-Engine
	isTypeOf Engine
V6 Engine	*isTypeOf* Spark-Engine
	relatesTo Engine
	relatesTo Spark-Engine
	follows Engine
V8 Engine	*isTypeOf* Engine
	relatesTo Engine

[5] http://clarkparsia.com/pellet/

Table 4.3: CRRM SRWL rules

	CRRM SWRL rules
1.	$LinkingPhrase(?x),\ Term(?y),\ Term(?z),\ Usage(?x),\ hasSource(?x,?y) \rightarrow isUsedBy(?y,?z)$
2.	$inkingPhrase(?x), Term(?a), Term(?b), Usage(?x), hasSource(?x,?a), hasTarget(?x,?b) \rightarrow isUsedBy(?a,?b)$
3.	$Term(?a), Term(?b), startedBy(?a,?b) \rightarrow isUsedBy(?b,?a)$
4.	$Term(?a), Term(?b), concludedBy(?a,?b) \rightarrow isUsedBy(?b,?a)$
5.	$Constituition\_and\_Containment(?x), LinkingPhrase(?x), PartHood(?x), Term(?y), Term(?z), hasSource(?x,?y), hasTarget(?x,?z) \rightarrow isPartOf(?y,?z)$
6.	$LinkingPhrase(?x), Spatial(?x), Term(?y), Term(?z), hasSource(?x,?y), hasTarget(?x,?z) \rightarrow isLocationOf(?z,?y)$
7.	$CauseEffect(?x), LinkingPhrase(?x), Term(?y), Term(?z), hasSource(?x,?y), hasTarget(?x,?z) \rightarrow isCauseOf(?y,?z)$
8.	$GenericSpecific(?x), LinkingPhrase(?x), Term(?y), Term(?z), hasSource(?x,?y), hasTarget(?x,?z) \rightarrow isTypeOf(?y,?z)$
9.	$Containment(?x), LinkingPhrase(?x), Term(?a), Term(?b), hasSource(?x,?a), hasTarget(?x,?b) \rightarrow isContainedBy(?a,?b)$
10.	$CauseEffect(?y), LinkingPhrase(?x), LinkingPhrase(?y), Term(?a), Term(?b), (\_and\_Containment\ or\ GenericDependence\ or\ Participation\ or\ Spatial\ orusage)(?x), hasSource(?x,?a), hasSource(?y,?a), hasTarget(?x,?b), hasTarget(?y,?b) \rightarrow AmbiguousRelations(?y)$
11.	$CauseEffect(?y), LinkingPhrase(?x), LinkingPhrase(?y), Term(?a), Term(?b), Term(?c), (CauseEffect\ orGenericSpecific\ or\ PartHood\ or\ Participation\ or\ Temporal)(?x), hasSource(?x,?a), hasSource(?y,?b), hasTarget(?x,?b), hasTarget(?y,?c), relatesTo(?c,?a) \rightarrow AmbiguousRelations(?y)$
12.	$CauseEffect(?x), LinkingPhrase(?x), Term(?a), Term(?b), Term(?c), Term(?d), Term(?e), follows(?e,?a), hasSource(?x,?a), hasTarget(?x,?b), isLocatedIn(?a,?c), isLocatedIn(?b,?d), precedes(?e,?b) \rightarrow AmbiguousRelations(?x)$
13.	$CauseEffect(?y), LinkingPhrase(?x), LinkingPhrase(?y), Term(?a), Term(?b), Term(?c), (not(CauseEffect))(?x), hasSource(?x,?b), hasSource(?y,?a), hasTarget(?x,?c), hasTarget(?y,?b), isCauseOf(?a,?b) \rightarrow AmbiguousRelations(?y)$
14.	$LinkingPhrase(?x), Term(?y), Term(?z), (not(Generic-Specific))(?x), hasSource(?x,?y), hasTarget(?x,?z), isTypeOf(?y,?z) \rightarrow AmbiguousRelations(?x)$
15.	$Containment(?x), LinkingPhrase(?x), Term(?y), (hasTarget\ min\ 2\ Term)(?x) \rightarrow AmbiguousRelations(?x)$
16.	$isContainedBy(?a,?b), isContainedBy(?b,?a) \rightarrow incompatibleTypes$
17.	$Term(?x), Term(?y), isPartOf(?x,?y), isPartOf(?y,?x) \rightarrow incompatibleTypes(?y,?x)$
18.	$Term(?x), Term(?y), isTypeOf(?x,?y), isTypeOf(?y,?x) \rightarrow incompatibleTypes(?y,?x)$
19.	$Term(?x), Term(?y), isUsedBy(?x,?y), isUsedBy(?y,?x) \rightarrow incompatibleTypes(?y,?x)$
20.	$Term(?a), Term(?b), isTypeOf(?a,?b) \rightarrow precedes(?b,?a)$
21.	$Term(?a), Term(?b), isCauseOf(?a,?b) \rightarrow precedes(?a,?b)$
22.	$Term(?a), Term(?b), isPartOf(?a,?b) \rightarrow precedes(?a,?b)$

In CRRM ontology a relation is considered ambiguous if it violates at least one of the following conditions (see line 10 to 15 of the Table 4.3):

- if there is a CauseEffect relation among two terms (a cause and an effect respectively), then, no other relation between those terms may occur, except a temporal relation.
- if there is a CauseEffect relation linking two terms (a cause and an effect respectively), then the Effect cannot be linked to other concepts connected to the Cause through the following conceptual relations: PartHood, GenericSpecific, CauseEffect, Participation or Temporal.
- if there is a CauseEffect relation linking two terms (a cause and an effect respectively), then the cause always precede the effect.
- If there is term that is a specialization of another term, then the specific term cannot be linking by a CauseEffect relation to the generic term.
- if there is a Containment relation that does not meet the "one-to-one" cardinality between the part and the whole. It means that in a Containment relation the part can only be linked to a single whole, which calls for a nested relationship.

Furthermore, two term are considered incompatible when in the CRRM ontology (see line 16 to 19 of the Table 4.3) :

- it is found that a term is both a part and a whole.
- it is found that a term is both te container and the content.
- it is found that a term is both a generic and a specific.
- it is found that a term is both a means and a end.

However, in order to determine the ambiguity of a relation or the incompatibility of terms, a set of facts must be inferred. In Table 4.3 there are rules that define the conditions under which, from the stated facts, it is possible inferred:

- two terms related through an *isUsedBy Object Property* (see line 1 to 4 of the Table 4.3);
- two terms related through an i*sPartOf Object Property* (see line 5 of the Table 4.3);
- two terms related through an i*sLocationOf Object Property* (see line 6 of the Table 4.3);
- two terms related through an *isTypeOf Object Property* (see line 8 of the Table 4.3);
- two terms related through an *isCauseOf Object Property* (see line 7 of the Table 4.3)
- two terms related through an *isContainedBy Object Property* (see line 9 of the Table 4.3);
- two terms are related through an *precedes Object Property* (see line 20 to 22 of the Table 4.3)

Nevertheless, if generic conceptual relations can ease the conceptualisation process to identify the nature of the relation among the top-level concepts of a domain, it is not so easy to find the appropriate naming for such relation. For that, a set of linking phrases were collected according to upper-level ontology analysis and made available to the user. Yet, the user might not agree with the provided standard options to name a conceptual relation, in which case he/she could use the “text” (i.e., using terminological methods over an existing corpus) to get clues about new possible labels for a conceptual relation. Following this, let us consider the existence of a domain corpus, specifically from the urban rehabilitation domain, gathered in the scope a the H-Know [6] project. Additionally, let us consider two terms: **Diagnosis** and **Observation**, obtained by "inspecting" the domain corpus, more specifically the following retrieved text snippet:

"*The diagnosis consists essentially in the process for identification or determination of the nature and the cause of the anomalies, through observation and investigation, using several tests, historical research and the expert opinion.*" In H-Know Project corpus.

So far the user has been assisted by terminological extracting services to identify two candidate terms, but then the challenge is on building the conceptual structure itself, that is, linking the collected terms properly together. Here, the templates can guide the users to complete the process of concept organization. Browsing through the templates available, the user is informed about the context of use (intent) of each template. Part-Whole, for instance, has the following intent: “*Used to represent relations between concepts in which a concept has another concept as its constituent forming a whole, which could be dependent or independent from its parts.*” Accordingly, it is acceptable to consider that Part-Whole template is suitable to link diagnosis and observation. Having identified the appropriate template, the next step is to check the feasibility of the proposed link between the terms, through the verification of a set of competency questions. Thus, considering the terms under analysis (Diagnosis and Observation), the correspondent competency questions should look like what follows:

1. (Is) *Observation* a component/constituent or is attached to *Diagnosis*?
2. (Are) *Observation* and *Diagnosis* nested?
3. (Are) *Observation* and *Diagnosis* physically engaged?

By confirming the questions the user is able to select the appropriate linking phrase between the terms. In this particular case, the selected questions could point to different sets of linking phrases. For example, if the user agrees with questions 2 and 3, the resulting conceptual structure could be: "*Diagnoses **includes** Observation*". On the other hand if the user agrees with question 1 and 3, the resulting conceptual structure could be: “*Observation **isPartOf** Diagnosis*”.

CRRM, as described, might be useful from the perspective of a knowledge engineer, but it is quite complex considering a regular domain expert using it during his conceptualisation activities.

[6]H-KNOW was an European research project in the area of building rehabilitation, restoration and maintenance, particularly in the cultural heritage domain. The project objective was to develop an ICT solution, to support SME’s collaborative networks in integrating collaboration, knowledge and learning in the RR&M field ($http://h-know.eu$)

Therefore, CRRM was included in a more comprehensive framework towards its implementation on an IT platform. The next section presents and describes the conceptualisation framework.

4.3 Conceptualisation framework (CF)

The CF depicted in Figure 4.4 provides the conceptual overview of the model tailored to support a build-to-reuse approach of conceptual representations.

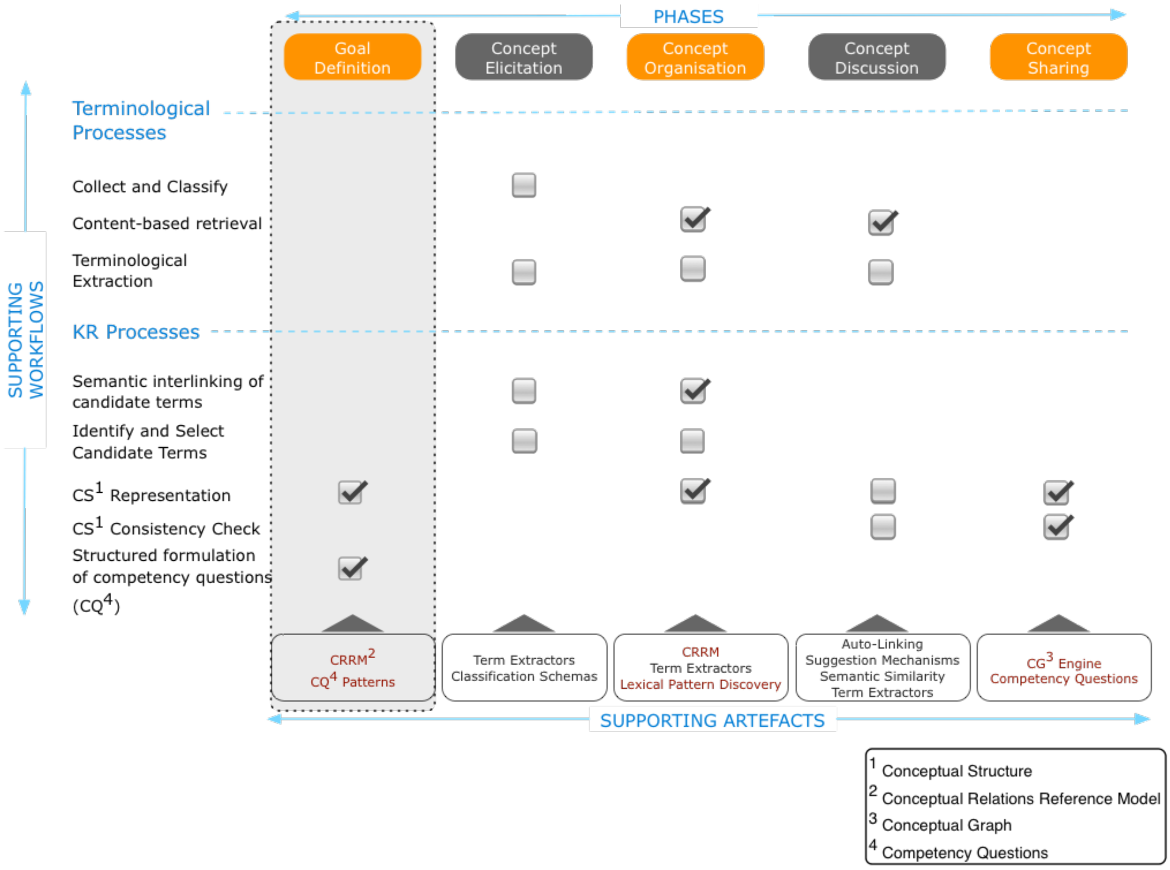


Figure 4.4: Conceptualisation framework (CF)

According to this framework a conceptualisation process is divided into five phases – goal definition (preliminary phase), concept elicitation, concept organisation, concept discussion and concept sharing, whose execution depends on a set of supporting workflows (processes). Typically, the phases run in several iterations until some conditions are verified, as Table 4.4 briefly systematizes.

In the CF two sets of processes are considered: terminological processes and knowledge representation (KR) processes. Generically, the terminological processes encompass methods for identifying/selecting lexical resources and their classification, as well as terminological extraction techniques. Additionally, they are also used to assist the negotiation activities during concept discussion. KR processes encompass activities to assist the production of well-defined concep-

Table 4.4: Transition criteria among the phases during the conceptualisation phases

Phases				
Goal definition	**Concept elicitation**	**Concept organisation**	**Concept discussion**	**Concept sharing**
Are all the Competency Questions defined?	There are the necessary and sufficient domain terms?	All Conceptualisation proposals were shared?	There is consensus around the final conceptual representation?	Is the conceptual representations formalized? Is the degree of reusability above a predefined threshold?

tual structures. Terminological processes support concept elicitation, overcoming the difficulties related to concept identification (such as naming, meanings, contexts of use). In order to accomplish domain structuring, users could resort to available templates; however each domain has its own specificities, which asks for specific relations. As the detail level of the domain description increases, it calls for real-time term contexts where some words or compound words are highlighted. Those highlighted terms could be used to detail existing representations, either to designate new concepts or new conceptual relations.

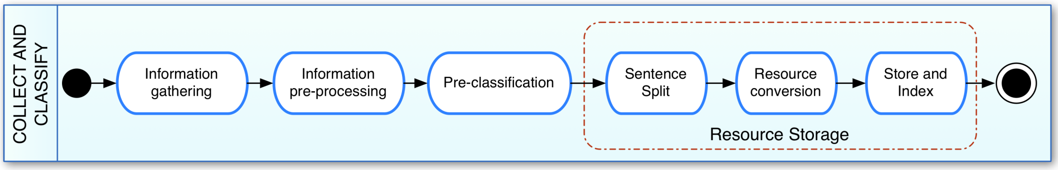
The research focus of this thesis was both in the design of the CF and its implementation in a prototype tool. However, the efforts were concentrated in just some of the enumerated workflows and phases as indicated by the checkboxes in figure 4.4. Moreover, not all the supporting artefacts were designed and implemented. From the list depicted in Figure 4.4, the following were considered:

- CRRM;
- Competency Questions (CQ) and CQ patterns;
- Lexical Pattern Discovery (LPD);
- Conceptual Graphs (CG) Engine;

The workflows may be triggered throughout the conceptualisation process making use of specific artefacts to assist/support the execution of their inclosing activities. The following tables (from table 4.5 to 4.11) detail each of the conceptualisation supporting workflows.

The CF presents a dual perspective combining terminology and knowledge representation. The approach discussed in this thesis assumes that some kind of transformation of semi-informal conceptual representations (built by the domain experts) into more rigorously defined and computer-readable conceptual representations. The definition of the core formalism for knowledge representation is crucial to ensure the CF's feasibility. During the the next section, conceptual graphs (CGs)

Table 4.5: Collect and classify workflow

Collect and Classify workflow
Purpose: To gather relevant informational items in a shared repository. **Assumptions**: There is a common list with the needed metadata attributes to describe the identified informational resources (e.g. a subset of the DublinCore vocabulary). **Limitations**: It is advisable to have a list of metadata descriptors. **Inputs**: Classification schemas. **Outputs**: Domain corpus **Main artefacts used**: a) classification schemas; b) NLP services; c) Indexing and search engine.
Basic flow


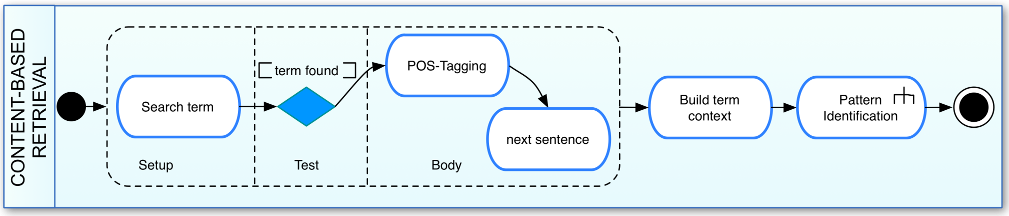
are explained and presented as the formalism for conceptual representations formal definitions and processing.

Table 4.6: Content-based Retrieval workflow

Content-based Retrieval workflow

Purpose: Retrieve all sentences from all available resources, which contain at least one concept from the current model. Each lexical pattern identified within a sentence is highlighted.
Assumptions: There is a domain corpus.
Limitations: A domain corpus is needed.
Inputs: Index domain corpus.
Outputs: The annotated context for each concept is obtained.
Main artefacts used: a) Term extractors; b) NLP services; c) Lexical pattern discovery rules; d) Indexing and search engine.

Basic flow



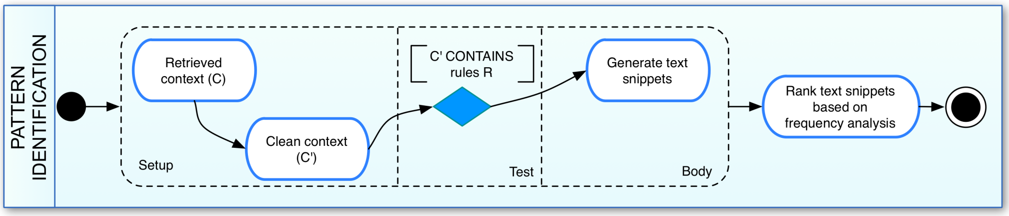


Table 4.7: Terminological extraction workflow

Terminological Extraction workflow

Purpose: To retrieve candidate terms.
Assumptions: There is a domain corpus.
Limitations: Corpus main language.
Inputs: Domain resources (e.g. uploaded documents or other URIs).
Outputs: List of candidate terms.
Main artefacts used: a) Term extractors; b) Indexing and search engine.

Basic flow

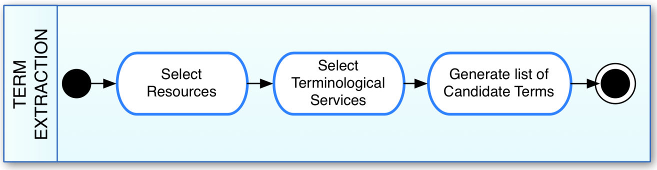


Table 4.8: Semantic Interlinking of domain terms workflow

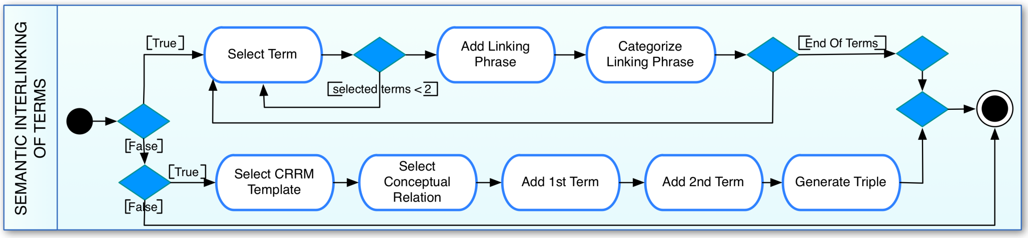
Semantic Interlinking of domain terms workflow
Purpose: Build and interlink conceptual structures. *The minimum granularity of a conceptual structure is a triple in the form of "Concept-Relation-Concept"* **Assumptions**: NA **Limitations**: No CRRM available. **Inputs**: Candidate terms and conceptual relations to link concepts. **Outputs**: Conceptual Structure(s). **Main artefacts used**: a) CRRM; b) Term Extractors.
Basic flow


Table 4.9: Identify and Select candidate terms workflow

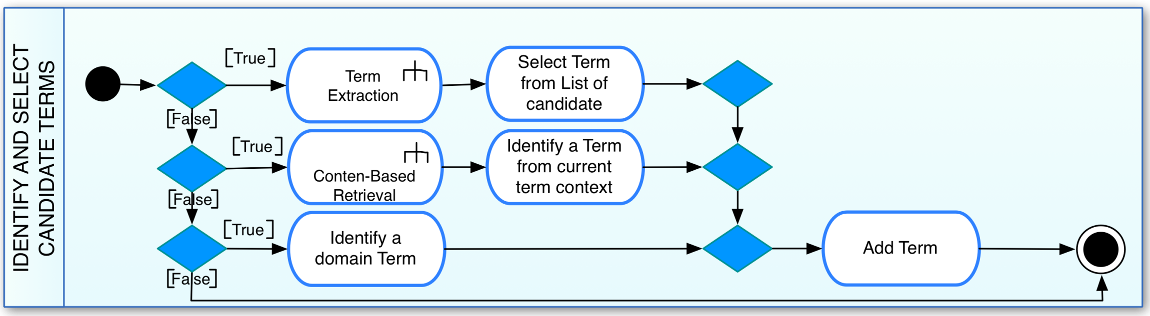
Identify and Select candidate terms workflow
Purpose: Selection of the fundamental domain candidate terms. **Assumptions**: There is a domain corpus. **Limitations**: NA **Inputs**: List of candidate terms; List of domain resources. **Outputs**: List of domain concepts. **Main artefacts used**: a) Lexical Pattern Discovery; b) Term Extractors.
Basic flow


Table 4.10: Conceptual Structures Representation workflow

Conceptual Structures Representation workflow
Purpose: To ensure that the developed conceptual structures are computational-ready and well-formed (that is, compliant to a well defined representation schema). **Assumptions**: NA **Limitations**: NA **Inputs**: Informal Conceptual Structures. **Outputs**: Semi-formal and formal Conceptual Structures. **Main artefacts used**: a) CQ patterns; b) CG Engine.
Basic flow

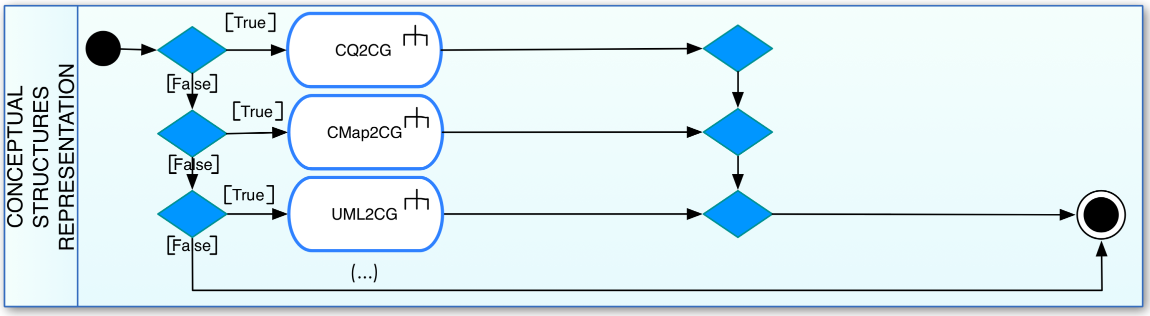
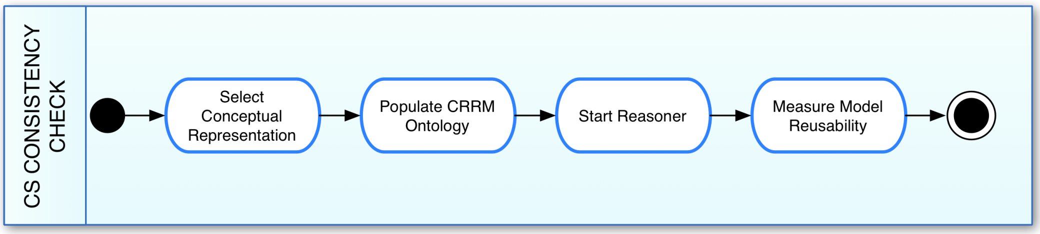


Table 4.11: Conceptual Structures Consistency Check workflow

Conceptual Structures Consistency Check workflow
Purpose: Verify the semantic consistency of the available conceptual structures. **Assumptions**: The conceptual relations were properly categorized. **Limitations**: Available CRRM rules. **Inputs**: Known conceptual structures. **Outputs**: List of ambiguous conceptual relations and incompatible terms.
Basic flow



4.4 Conceptual graph theory

Conceptual Graphs (CGs) model emerged within the artificial intelligence (AI) scientific domain by John Sowa, but its foundations come from several areas such as: natural language processing, semantic networks, databases and logics, especially the existential graphs of Pierce, which form a diagrammatical system of logics (Chein and Mugnier, 2008). CGs have been used as a paradigm of reference for knowledge representation, greatly due to its design principles[7], which "*emphasize the requirements for a cognitive representation: a smooth mapping to and from natural languages; an 'iconic' structure for representing patterns of percepts in visual and tactile imagery; and cognitively realistic operations for perception, reasoning, and language understanding. The regularity and simplicity of the graph structures also support efficient algorithms for searching, pattern matching, and reasoning*" (Sowa, 2009).

Following this, a set of characteristics are found in CGs which make them well suited to broad-scope knowledge representations [8] purposes.

- **Conceptual graphs could be seen as a semiotic system** - CGs uses a specific notation to create symbolic models for knowledge representation, whose elements are considered as signs. Through the use of "symbolic diagrams", a CG communicates the knowledge of a particular worldview.

- **Conceptual graph forms models with logical consistenc**y - CGs are a system of logic with a graph-based formalism, allowing the creation of models in the logical sense of structures for which some set of axioms are true (Sowa, 1979).

- **Conceptual Graphs claim to have great expressive power** - "*The design goal for conceptual graphs is a balance between the simplicity of Peirce's existential graphs and the flexibility, adaptability, and expressive power of natural languages.*" (Sowa, 2003). Its aim is "*to express meaning in a form that is logically precise, humanly readable, and computationally tractable*" (Sowa, 2003). CG graph-based formalism, can be expressed in FOL [9] and by means of a graph notation, it is possible to capture sentences into conceptual graphs (Mineau et al., 2000).

- **Pragmatic functionality** - According to De Moor et al. (2002) CGs are well suited to model evolving knowledge structures of different levels of detail, but retaining the idea from Peirce's existential graphs of which "*complex ideas can be expressed with repeated application of a small number of primitives*" (Sowa, 2003).

4.4.1 Conceptual graph model

The conceptual graph model is composed by an ontology and the graph itself (Laudy et al., 2007). The ontology defines the *graph vocabulary* in what regards to the different types of concepts and

[7]Expressivity; Simplicit; Readability
[8]Domain independent. Not committed to a particular domain
[9]First Order Logic

conceptual relations, forming a hierarchy of types and type labels. Concept types and conceptual relations types (or just relation types) are two partially ordered sets. The partial order is interpreted as categorical generalisation relation: T1 $\succeq$ T2 meaning that T2 is a specialisation of T1 (or T1 is a generalisation of T2, T2 is a subtype of T1, T1 subsumes T2, T2 is of type T1). For the construction of conceptual graphs (the graph itself) it is assumed that concept types and conceptual relation types are given.

A CG is a graph with two kinds of nodes (or vertices) - concepts and conceptual relations - and edges (or arcs) linking the nodes. From the graph-theory perspective, a CG is a bipartite directed graph with labelled nodes, which means that edges connect two nodes of two disjoint sets (concepts and conceptual relations), ordering them to denote edge orientation. The nodes should be labelled (e.g., an identifier), thus, all edges either go from a concept to a conceptual relation or from a conceptual relation to a concept. Graphically, concepts and conceptual relations are represented by two different shapes: boxes and ovals respectively. Inside boxes goes the information defining a concept, which is composed by the following entities:

- **Concept types**, denoted by a concept type label. Example: [INDIVIDUAL], [SAND], [STONE], etc.
- **Referents**, denoting individuals, values or sets. Example: ’Gravel’, ’1978’, ’Alfredo’. And could be represented as follows: [INDIVIDUAL: Alfredo], [STONE: gravel].

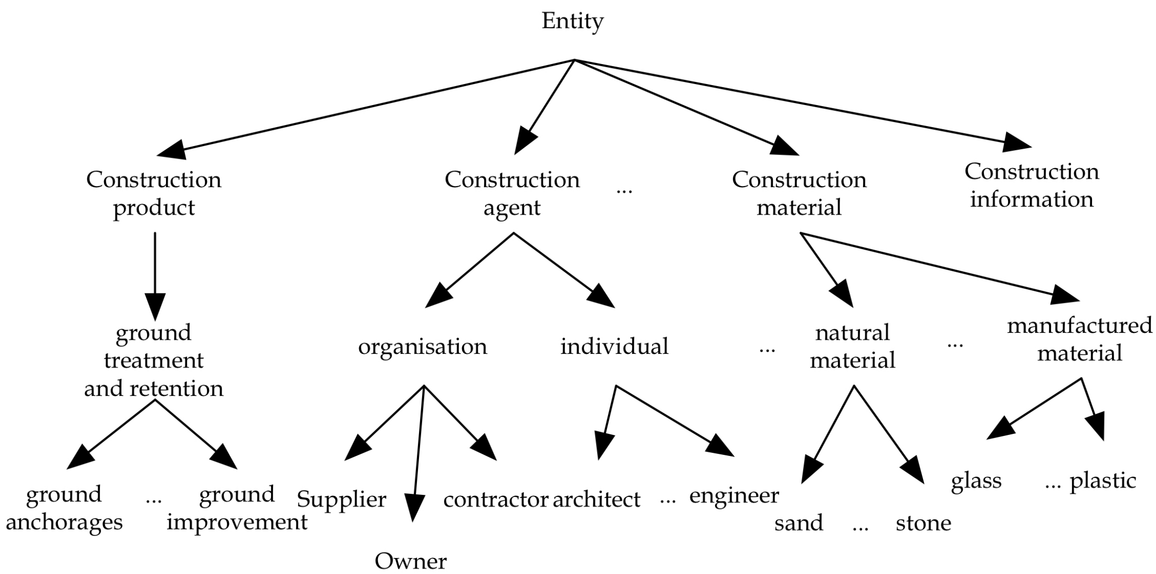


Figure 4.5: Concept type hierarchy. An excerpt produced in H-KNOW[10]project

[10]H-KNOW was an European research project in the area of building rehabilitation, restoration and maintenance, particularly in the cultural heritage domain. The project objective is to develop an ICT solution, to support SME’s collaborative networks in integrating collaboration, knowledge and learning in the RR&M field ($http://h-know.eu$)

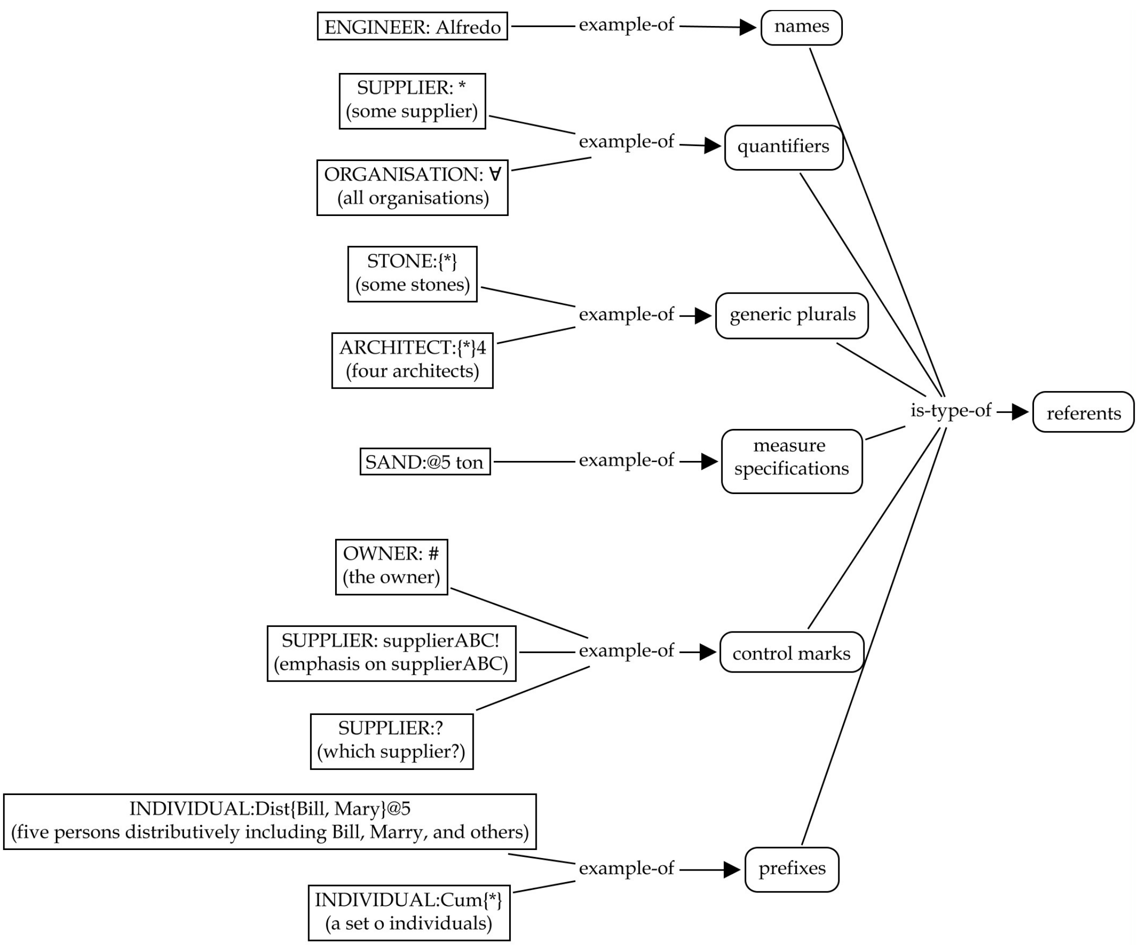


Figure 4.6: Type of referent with examples, denoting the set of individual markers

Figures 4.5 and 4.6 represent a concept type hierarchy and the different types of 'referents', respectively.

Regarding conceptual relations, they are represented, graphically, as "ovals" and have the following associated entities:

- **Relation types**, denoted by a relation type label. Example: [PartWhole], [Cause-Effect], etc. (Figure 4.6)

- **Valence**, denoting de number of arcs belonging to each relation type.

- The **Signature**, which defines the orientation of the relation and the type of concepts allowed or involved in that same relation. Let us consider the following representation of the *belongsTo* relation signature: <ENGINEER, ORGANISATION>, which denotes that the first arc should have a concept as its origin, whose type is Engineer or a subtype of Engineer, and the second arc should have another concept as its target, whose type is organisation or a subtype of organisation.

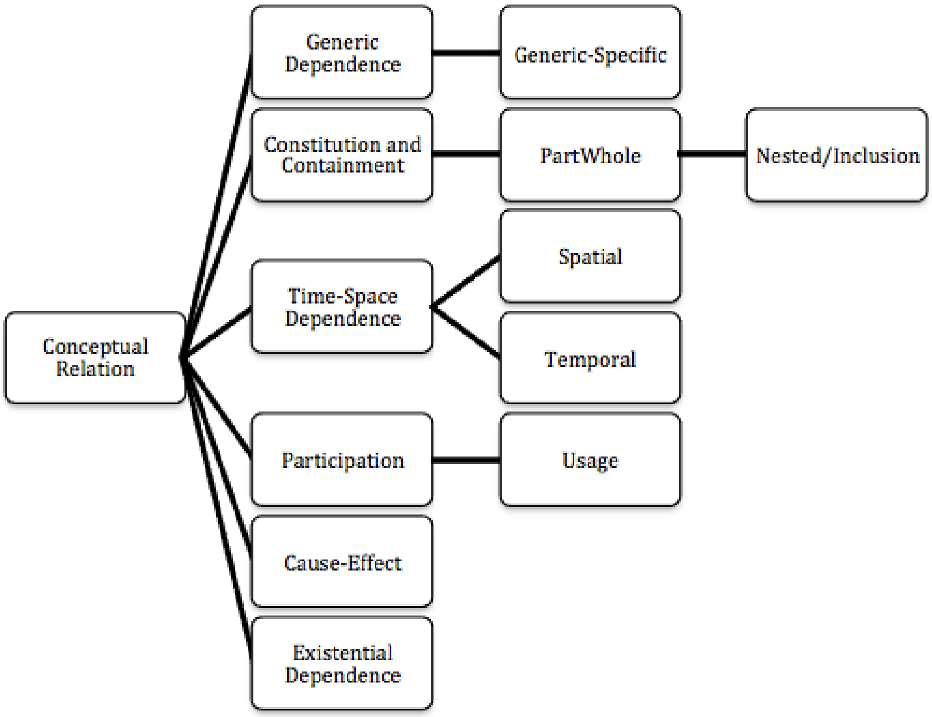


Figure 4.7: Relation type hierarchy (based on conceptual relations catalog)

A CG is assumed complete when its structure is compliant with the example depicted in Figure 4.7. However, CGs may take a more descriptive representation called linearized form, as portrayed by Figure 4.8.

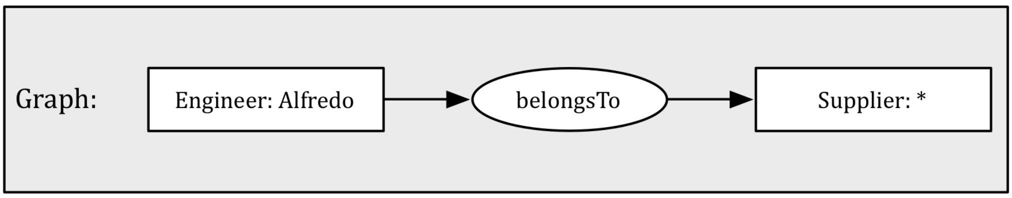


Figure 4.8: Typical conceptual graph (graphical representation)

According to the generic description of the CG model, it was shown a set of information that is assumed to exist, in order to be feasible the representation of knowledge based on CG formalism. That set of information is here designated by "CG knowledge Base", and its structure is systematised in the concept map in Figure 4.10.

CG formalism may be considered with more or fewer elements, providing a family of languages based on CGs, each one enclosing slight differences. For the purpose of the current work, we will consider only the basic CG elements, which form what is called Basic Conceptual Graphs

[Engineer: Alfredo] -> (belongsTo) -> [Supplier: *]

Figure 4.9: Typical conceptual graph (linear representation)

(BGs) or Simple Conceptual Graphs (SGs) (Chein and Mugnier, 2008).

The aim of our approach is to design artefacts to allow the representation, in the conceptual graph model, of *lightweight* formal ontologies, nevertheless we might not have access to any hierarchy of types of concepts, but we do provide a conceptual relation classification (Sousa et al., 2012). Hence, the interest is focused on gathering and organising knowledge around the domain concepts, which, ultimately, may result in a taxonomic hierarchy of concepts (Chein and laure Mugnier, 1992) (Baget, 2003).

4.4.2 Conceptual graph formal definition

Beyond the principles stated by Sowa (1979) the CG model is also "*a declarative model encoding knowledge in a mathematical theory, namely labelled graph theory, which has efficient computable forms, with a fundamental graph operation on the encodings to do reasoning, projection, which is a labelled graph morphism.*" (Chein and Mugnier, 1995).

Let us consider Figure 4.11, portraying a labelled bipartite directed graph. The displayed structure could be seen as a CG representation from the graph-theory perspective, where the relations are denoted by a diamond and labelled as r_{ij}, and concepts are denoted by a circle and labelled as C_i. Edges are connecting concept nodes and relation nodes, where e_1 and e_2 are indicating the connection from $r_{4,1}$ to C_1 and from $r_{4,2}$ to C_2 respectively. Note that edges are only allowed to connect two different types of nodes.

As mentioned previously, CGs require a set of information that will support their creation and manipulation. The most important part of this information is denominated *vocabulary*, which, as shown in Figure 4.10, might have more or less elements. Considering its basic elements, a vocabulary might be defined as follows:

Definition 4.1 (Vocabulary)**.** *The basic vocabulary of a CG is a triple (T_C, T_R, I), where:*

- *Unlike the CG's classical theory, in which T_C is set of concept types partially ordered by a specialisation relation (cf. Figure 4.5), here T_C = "concept". That means T_C is composed by a single concept type designated by the term: concept;*
- *T_R = set of relation types. It is partially ordered by the specialisation relation (cf. Figure 4.5). Here, T_R corresponds to the CRRM;*
- *I = set of individual markers, which are disjoint of T_R and T_C. Moreover, $M = I \cup *$, which denotes the set of individual markers (where * is a generic marker) and it is ordered as follows: "* is greater than any element in I and elements in I are pairwise disjoint". In*

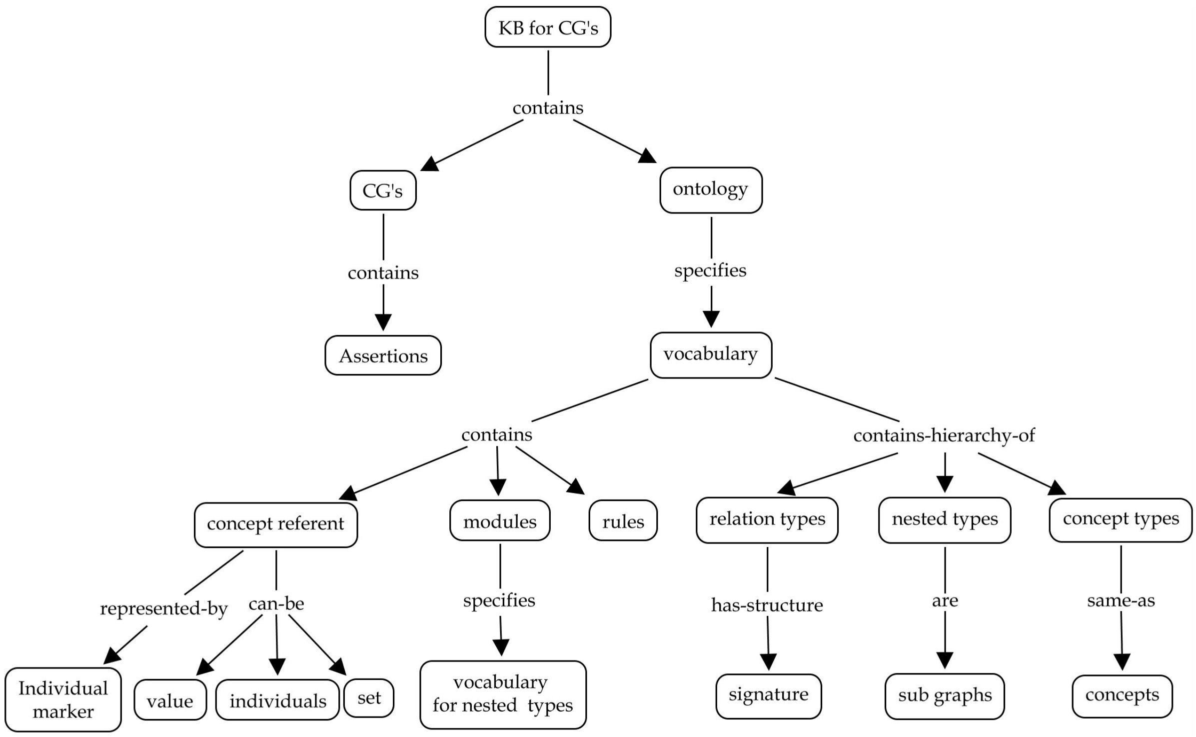


Figure 4.10: Conceptual Graphs knowledge base

our approach, $I = t_1, t_2, \ldots t_n$ is considered as a set of conceptual markers, whose elements correspond to domain terms that may be used to designate specific concepts.

Beyond these basic elements, a third element could be included in the CG knowledge base, the relation symbol signatures, where symbol is here considered as a label designating the conceptual relation.

Definition 4.2 (Relation Signatures)**.** *Relation signatures, are formally defined by a mapping ∂, which associates types of relations to types of concepts, such as:*

- $\forall r_1, r_2 \in T_{R^j}, r_1, r_1 \leq r2 \rightarrow \partial(r_1) \leq \partial(r_2)$ *meaning that, when a symbol (label) of a relation r_2 is specialised in a relation label r_1, its arguments could be specialised but nor generalised.*

For a better comprehension let us considerer the following conceptual relation labels: *partOf*, *belongsTo*, *usedIn* and *resultsIn*. The respective signatures for these relations would be: *partOf* (Individual, Organisation), *belongsTo*(Engineer, Organisation), usedIn(Construction Resource, Construction Process), *resultsIn*(Construction Process, Construction Result). From the relation type hierarchy (Figure 4.7), the term *belongsTo* is a relation's label specialisation of *partOf*. Accordingly, it is allowed to specialise the arguments that compose the conceptual relation signature, but generalisation cannot be performed. Following this, it is possible to define a conceptual graph as follows.

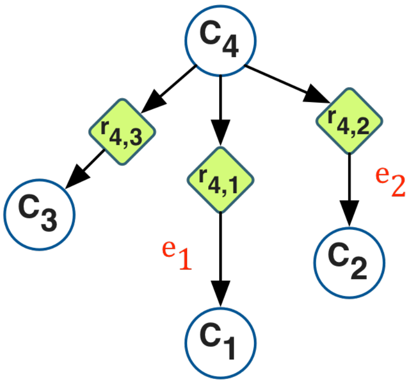


Figure 4.11: Labelled bipartite directed graph

Definition 4.3. *Assuming the vocabulary defined earlier, a conceptual graph is defined as a 4-tuple CG=(C, R, E, λ), where:*

- *C = finite non-empty set of concept nodes $c_1, \ldots, c_n$;*
- *R = finite non-empty set of conceptual relations $r_1, \ldots, r_n$;*
- *E = set of edges which connect two different type of nodes;*
- *λ, is a labelling function such as:*
  - *a concept node c is labelled as: (type(c), marker (c)), where type(c) $\in T_c$ and marker(c) $\in I \cup *$*
  - *a relation r is labelled as $\lambda(r) \in T_R$, which is the same as type(r)*
  - *an edge labelled i between a relation r and a concept c is denoted (r,i,c). According to Figure 4.11, the edge indicating the connection from $r_{4,1}$ to C_1, is designated $e_1 = (r_{4,1},'2',C_1)$. Accordingly, the set of the edges adjacent to each conceptual relation $r \in R$ is numbered from 1 to degree(r), and $C_i(r)$ denotes the i^{th} concept node $c \in C$ adjacent to r.*

4.4.3 CGs' operations definitions

Introduced the formal definition and main assumptions associated to the creation of CGs, we move forward in order to understand how it is possible to ensure that the conceptualisation process is driven by a "build to reuse" approach, that is, to build conceptual representations towards its retrieval and reutilisation in a simplified way. To cope with this stance, we considered the six canonical rules proposed by John Sowa, which allow any operation on CGs. Canonical rules act as a graph grammar, enforcing selection constrains in CGs. The fundamental principle is simple and each rule could be applied according to one of the following four possibilities:

1. Creating a new graph v, following the specialisation of an existing graph u, preserving the logical equivalence between v and u.

2. Creating a new graph v, following the generalisation of an existing graph u, preserving the logical equivalence between v and u.

3. Creating a new graph v, through an exact copy of an existing graph u, preserving the logical equivalence of v and u.

4. Creating a new graph v, by changing the form of an existing graph u, preserving the logical equivalence of v and u.

Canonical rules come in three blocks of two rules: copy and simplify; restrict and unrestrict; join and detach; allowing specialising, generalising or simply copying an existing CG. In each pair, any operation performed by one rule can be reversed by the other rule. Basic CG operations are found synthesised in Figure 4.12.

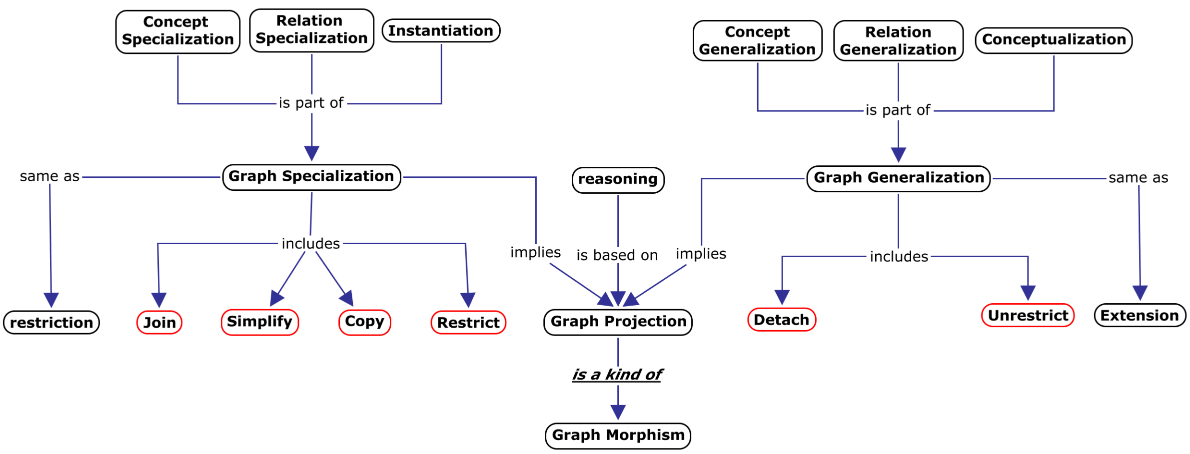


Figure 4.12: CGs' operations summary

From these rules it is possible to derivate new graphs among which is possible to define generalisation/specialisation relations. According to Sowa (1984), "*a conceptual graph has no meaning in isolation. Only through the semantic network and its concepts and relations linked to context, language, emotion and perception.*". This means that it only makes sense to create domain assertions, upon the existence of an ontology with the generic knowledge of a domain (cf. definition 4.3). From our approach, it might happen that such artefact does not exist. Moreover, and as mentioned earlier, it may happen that the output of the conceptualisation process is actually a concept type hierarchy. Even so, we keep the view that CGs are still interesting media-independent formalisms to represent expert's conceptual representations in a formal way. For that the way to perform some of the elementary CG operations was redesigned and adapted for our specific purposes and context. The main change is to focus on conceptual relations rather than on concept hierarchy, together with concept and relation designations and categories, edge orientation and concept neighbours. Following that, let us considered the specialisation rules: simplify, restrict, join and detach.

- **Simplify**: Simplify is a rule that applies to conceptual relations. It consists in removing duplicate relations in a CG. Given two relations r and s of a graph G, r and s are considered duplicate if they share the same type and neighbors. Removing s or r implies removing its edges.

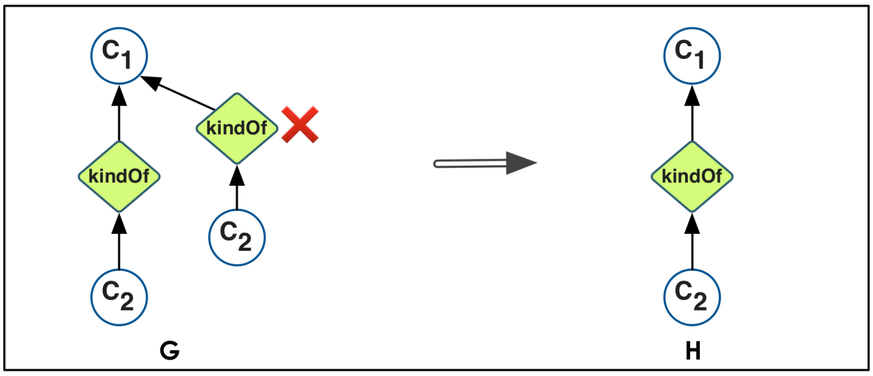


Figure 4.13: Simplify rule

- **Restriction**: Typically, restriction is a rule that applies both to concepts and conceptual relations. It consists in decreasing the label of a concept or conceptual relation. For any concept c in a graph G, $type(c)$ may be replaced by a subtype (Figure 4.16). If c is generic, its referent may be replaced by an individual marker (Figure 4.15). Additionally, for any relation r in a graph G, $type(r)$ may be replaced by a subtype (Figure 4.16).

Restriction rule is grounded on subsumption principles, whose criteria are given by an existent concept lattice (concept type hierarchy). For concept restriction and in the absence of a concept lattice, new criteria were defined based on conceptual relations. For any relation r of arity one, indicating subsumption between concepts c and d, belonging to graph G, it is possible to obtain a graph H by removing c and r, and linking to d all edges of conceptual relations that had been connected to c (Figure 4.17).

- **Join**: Join is a rule that applies to concepts. Whenever two concept nodes from two different CGs are identified as identical, the concepts are joined in a single concept, but keeping all edges. If a concept c in a graph G is identical to a concept d in H, then it is possible to obtain a graph W by deleting d and linking to c all edges of conceptual relations that had been connected to d.

In our approach, two concepts are said to be identical or similar according to their designation and structure, that means, two concept are similar if:

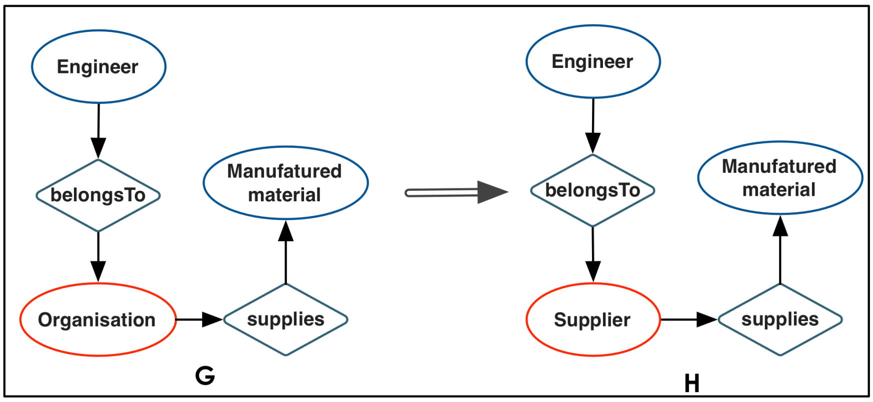


Figure 4.14: Typical concept restriction

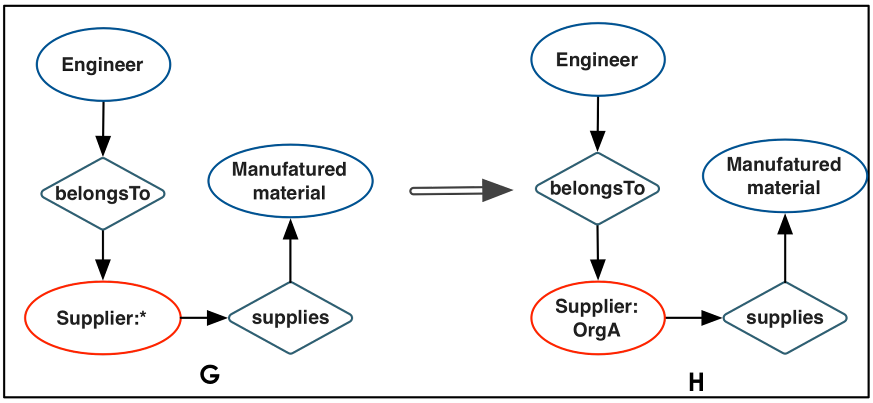


Figure 4.15: Concept restriction with referent

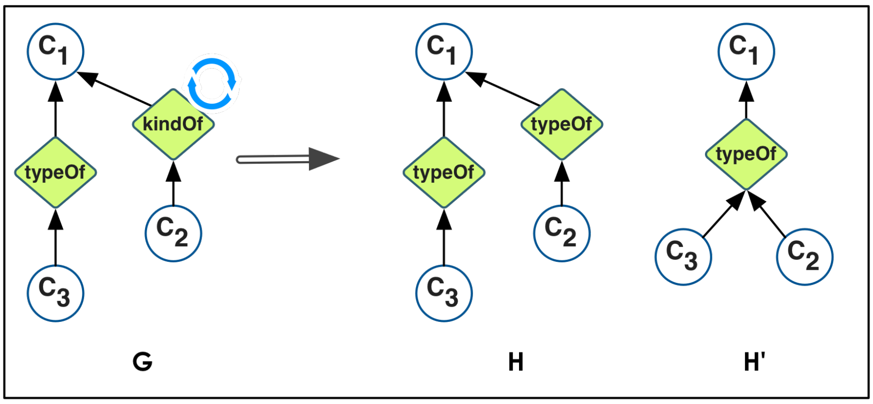


Figure 4.16: Relation restriction (Note that H and H' are equivalent

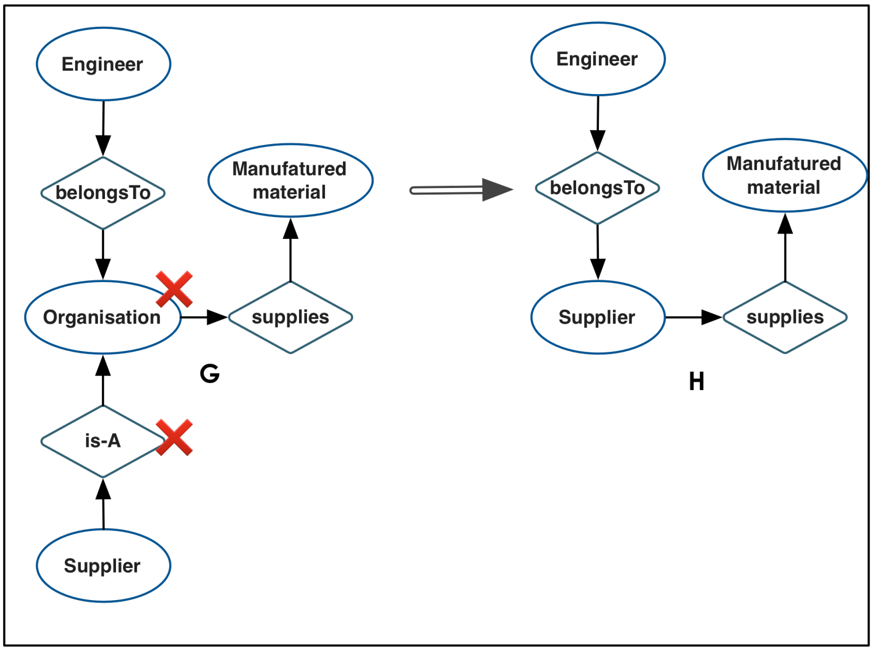


Figure 4.17: Concept restriction (subsumption based on conceptual relations)

- the concepts are syntactically equivalent;
- the concepts share the same number of concept neighbours, connected by the same number of edges of conceptual relations;
- the concepts belong to the same category (inferred by the CRRM ontology).

For each of the aforementioned criteria, it is possible to measure the similarity between two concepts, given by a value in the interval [0,1], where 0 means not similar and 1 highly similar.

Two concepts are syntactically similar if they share the same designation. Syntactical similarity is calculated after normalising the concept designations (tokenize, stemming, etc, ...). On the other hand, the calculation of the structural similarity measure is inspired in the Dice similarity (Montes-y Gómez et al., 2000). Dice measure is calculated considering twice the number of common neighbours divided by the sum of the degree of the vertices [11]. The result is a pairwise similarity value, as shown by the following equation:

$$sim(c,d) = \frac{2 * nN(G_c, H_d)}{deg(G_c) + deg(H_d)} \tag{4.1}$$

According to our approach, it is considered that the number of common neighbours depends on the relation similarity, that is, the count of neighbours will only consider those that are connected from the current vertex through the same kind of relation. The type of relation could be obtained by CRRM.

Let us consider the example depicted in Figure 4.18, where:

[11]The degree of a vertice corresponds to the number of incoming and outgoing edges

- G' corresponds to the sub graph of G containing the direct neighbors of vertex a.
- H' corresponds to the sub graph of H containing the direct neighbors of a'

Accordingly, the number of common neighbors between vertex G_a and $H_{a'}$, given by the function $nN(G_a, H_{a'})$, is equal to two. From the three possible common neighbors, there are only two vertices in G connected to G_a and two vertices in H connected to $H_{a'}$ that share the same relation type. After determining the value of $nN(G_a, H_{a'})$, it is calculated the vertex degree of G_a and $H_{a'}$, then it is possible to obtain the similarity between vertex a of G and vertex a' of $H : Sim(G_a, H_{a'}) = 0,66(\ldots)$.

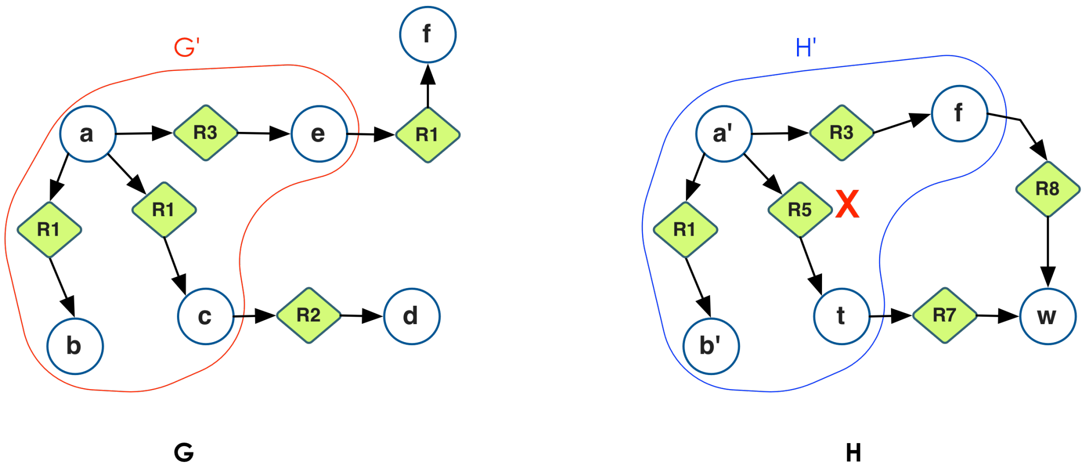


Figure 4.18: Concept join (example)

- **Detach**: detach is a rule that applies to concepts. It consists on deriving H from G making a copy of d from concept node $c \in G$ and detaching one or more incoming arcs of conceptual relations that had been attached to c, and attaching them to d.

In the sequence of the foregoing descriptions, not all CG operations apply to all kind of nodes (concepts and conceptual relations), the challenge is not to find which operation applies to which node, but to find the appropriate sequence of possible operations to retrieve a new conceptual structure that satisfies the expert's needs. In order to discover a generic workflow to perform the CG operations, these were classified according to their scope and granularity in relation to a graph. Whereas scope could be internal or external, vertex granularity could refer to an element-level or to a structure-level. The following table summarizes the adopted CG operations classification schema.

Having the classification schema as a baseline (Table 4.12), it was defined a process (see Figure 4.19) through which it is possible to extract new CGs from existing ones, applying some of the defined CG's operations. The process adopts a bottom-up approach; firstly by applying those operations classified as internal with an element-level granularity, to operations classified as external, which may occur both at element and structure-level.

Table 4.12: CGs' operations classifications

Scope	Internal	Operations performed within the graph
	external	Operations considering more than one graph
Granularity	Element-level	Operations performed in isolated concepts
	Structure-level	Operation considering more than one concept

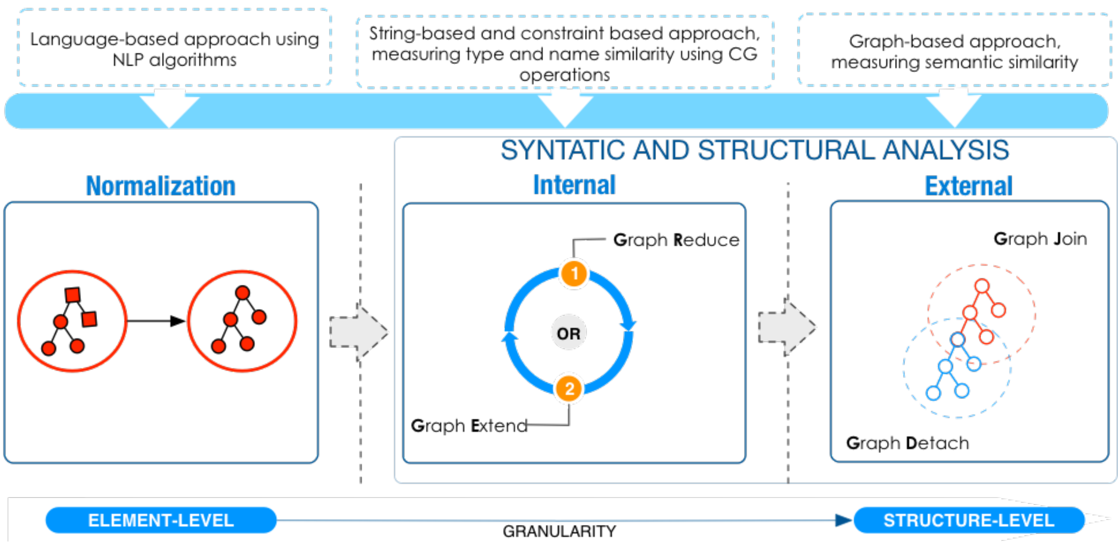


Figure 4.19: Derivation process of new graphs, from existing ones and supported by CG operations

The depicted process (Figure 4.19) was designed to act as a support workflow for the collaborative conceptualisation process, aiming at, on one hand, reusing existing conceptual structures when creating new conceptual representations and, on the other hand, helping to reach consensus during the discussion of conceptual representations. So, the current process supports both the activities of individualized creation of new conceptual representations, and activities inherent to the "shared spaced" of the conceptualisation process, such as the merge operations. Accordingly, the depicted process (Figure 4.19) encloses the following main activities:

- Normalization and;
- Internal and external syntactic and structural analysis;

The Normalization activity follows a language-based approach using NLP algorithms such as *stemming* and *tokenizing* to normalize concepts and conceptual relations designations. It occurs at element-level within a graph. After normalization, users may choose to specialize or generalize a CG by performing graph reduce or graph extend operations respectively. These operations occur both at element-level and at structural-level within a graph, wherein graph reduce includes *simplify* and *restrict* CG operations and graph extend encloses *unrestrict* CG operations. Finally, users may proceed to external natured operations, applicable either to element-level or structure-level. Graph *join* and graph *detach* are the operations associated to this stage of the process.

On despite of relying on several mechanisms to conduct the process, experts keep all the main decisions even on how to begin. After normalization it is possible to, manually, identify the conceptual structures to start working with, or let it be done automatically. In this case, the choice on which node to start analysing the model (concept map) will follow on the leaf with less incoming edge or on the node with less incoming and outgoing edges. After that, the model is transversed and conceptual structures are annotated according to the conceptual relations type. All relations denoting a generic-specific connection between two or more concepts are highlighted, according to the following conditions:

- $\forall r \in R \exists r' : r' = f(r) \rightarrow r' \leq r \wedge r.r' \in T_R$, where:
  - r is a relation from a set of conceptual relations R with and image (correspondence) on CRRM;
  - r' is a specialisation of r;
  - $f()$ is a specialisation function for r;

The experts are invited to choose from "*graph reduce*" or "*graph extend*", applying one of these operations to each of the annotated conceptual structures. After the internal analysis, from which may result a new conceptual model, it is possible to move further combining the achieved results with other models. At this stage it is possible to find similarities between elements from two different models forming a common one by running "*graph join*" operation (see Figure 4.18). Similarly, a model may be split into one or more components or sub-models, if the following condition is met:

- Let c be a concept node (vertex) of a connected conceptual graph $G = (C, R, E, \lambda)$ identified by the domain expert, then:
  - $\exists c \in C : cut\_vertex(c) \rightarrow G \backslash c = G'$ is disconnected.

That means, if removing c and detaching its boundary arcs of conceptual relations from G, then G without c is disconnected. So, c is a *cut_vertex* of G.

4.5 Establishing the conceptualisation objectives

The process of establishing the conceptualisation objectives by the stakeholders is fundamental for the conceptualisation success. This process is unstructured by nature, being a continuum from the initial discussion of the purpose to the first agreed conceptual representation. The inherent (inter) subjectivity of the process and its outcomes, together with an excessive informality are perhaps the reasons for being overlooked in the literature. This thesis proposes an approach integrating competency questions (CQ) and conceptual graphs to the support of domain experts and knowledge specialists in defining the purpose and fundamental conceptual commitments of the conceptual representations to be developed.

Conceptual representations, due to its informal basis, allow poor computational processing and only have non-inferable constructs, which means there is not a complete axiomatic definition of domain concepts and relations (e.g. an ontology). To fully axiomatize a conceptual representation there is the need to formalized it using, for example, OWL[12] and using DL[13], which is not affordable for a domain experts (DE). They can only provide the relevant conceptual knowledge. Moreover, the need for a formal conceptual representations i.e., an ontology, in the sense of being fully axiomatized, is not the case in many situations (e.g. for structuring and maintaining knowledge bases information). So, the fundamental problem debated in this thesis (i.e., how to assists DE to produce useful and reusable conceptual representations?), encloses two challenges from the outset: 1) the conceptual representations must conform to the intended needs and; 2) the conceptual representations must be rigorously defined, which means they must be computational-ready and well-formed (compliant with a well defined representation scheme).

Typically, the exact statement of the particular needs to be satisfied by the conceptual representations might be addressed by the concept of competency questions (CQ). In fact, several research works have been investing on CQ-based approaches not only for ontology validation but also for building them from scratch (Bezerra et al., 2013; Zemmouchi-Ghomari and Ghomari, 2013; Fernandes et al., 2011; Malheiros and Freitas, 2013; Ren et al., 2014). Despite valuable, they tend to focus on knowledge engineers or ontologists rather than DE (Bezerra et al., 2013; Zemmouchi-Ghomari and Ghomari, 2013), or it requires previous knowledge of specific semi-formal knowledge representations (Fernandes et al., 2011). Others (Malheiros and Freitas, 2013),

[12]Ontology Web Language
[13]Description Logic

beyond focusing on the ontology axiomatization, are using CQs but in an ad-hoc manner (using natural language). Some works share our vision that CQs "*can have a clear structure and relatively simple syntactic pattern*" (Ren et al., 2014), however their approach is too focused on creating OWL ontologies and are deeply committed to this particular application, violating one of the Grubber's conceptualization principles (Gruber, 2008). Our CQ based approach can be distinguished from others in 4 main aspects:

1. CQs are used not only for conceptual representations specification and validation, but also as an aid to its initial building;

2. the formulation of the CQ itself, is keyword-based rather than natural language-based. The experts are invited to, wittingly, think about the more appropriate keywords to formulate the questions (typically associated to candidate concepts and relations), rather writing CQs according to his own writing standards (or habits) and interpretations. A CQ catalog was implemented where a mapping between CQ's structure and the underlying semi-formal conceptual representations was defined based on the conceptual relations that each CQ type comprises. This calls for DE to double check the intended meaning for a given CQ;

3. CQs are "internally" represented according to the conceptual graphs (CG) structure (Sowa, 1992), and;

4. the achieved CQ's conceptual representations, which define the conceptual representations, are further extended towards its completion by means of terminological methods.

4.5.1 Building conceptual representations based on competency questions

For a conceptual representation to be useful and reusable it is necessary to find a tradeoff between its commitment with the real world and the experts' vision, in order to avoid either its under specification or over specification (Gruber, 2008). In other words, building useful conceptual representations means that they should be specified at the knowledge level (Uschold, 1998) and represent what was initially intended to. Furthermore, for conceptualisations to be reusable they should not be too complex or overcommitted to a particular application (Shotton et al., 2010) and its structure should be rigorously defined. On pursuing straightforward mechanisms to stimulate DE involvement in the conceptualisation process to build conceptual representations, an approach was designed, aiming at producing (re)usable conceptual representations for knowledge organization purposes (e.g. to capture, organize, categorize and maintain knowledge base information).

Generically, the process runs as follows:

1. CQs are formulated by means of the set of templates available in the CQ catalog and;

2. a transformation operation occurs, in order to convert CQs into semi-formal conceptual representations (e.g. concept maps);

3. DE validate the CQ's graphical representations, which are then represented (internally) under the CG formalism. After the formulation of the CQs and its acceptance by the DE, the resulting conceptual structures can be extended with more concepts and relations by means of terminological services, namely:
   - extracting relevant terms from a previously organised corpus and;
   - retrieving contextual information according to existing concepts and queries (using the corpus). This aims at discovering new concepts and/or conceptual relations through specific lexical patterns.

CQs, should be easy to specify and clear to understand. Thus, in order to commit the experts to the task of specifying the goals for the conceptual representations, fostering the development of concept-based queries, it was followed a template-based approach to assist the formulation of each objective, based on typical query types and underlying patterns (Cao and Mai, 2010). The identified patterns can be associated to the form of the following query types: i) what; ii) why; iii) how and; iv) where. For each query type, one or more formulation patterns can be derived, corresponding to a typical abstract syntactic structure through which a CQ statement is obtained. This set of patterns is made available through the CQ catalog in the form of templates used by DE to provide the fundamental keywords instead of a complete formulation of the CQ. This allows building simpler and unambiguous CQ sentences. From the CQ analysis is possible to establish a connection between query types and conceptual relation types, which each CQ purpose encloses (Nuopponen, 2005).

Table 4.13: Query type vs. query purpose vs. conceptual relation (CR) type

Query Type	Purpose	CR Type	Examples of CR
What/Which	Used to represent or claim about the "types and sub-types" or "parts and wholes" of an entity that has a certain interaction with another specific entity.	Generic/specific relation or containment relation	*type of, part of, included, in, ...*
How	Used to represent or claim about what might be used to accomplish an action or function.	Usage relations	*used by, ...*
Why	Used to represent or claim about what might be the cause (reason or motivation) through which some entity (or agent) accomplishes/realizes some action/function.	Causal relations	*result of, caused by, ...*
Where	Used to represent or claim about what might be the local where a specific interaction between two entities occurs	Spatial relations	*located at, next to, in, ...*

Furthermore, for each query type it was identified its typical structure - here called pattern - and the corresponding formal representation by means of conceptual graphs.

4.5.1.1 What/Which query type

The “what/which” query type, has the following pattern:

- **What/Which** $<$ **entity** $><$ **objectProperty**|”**part of**” $> * <$ **entity** $> *$, where:

  a) all data presented between “< >” is provided by domain expert when formulating the CQ;

  b) *objectProperty* – refers to a possible relation phrase, linking the object to another object. Typically corresponds to a verbal form. Users may choose from a predefined set of linking phases;

  c) entity - refers to an entity of the domain of the discourse. Typically corresponds to a noun;

  d) the "*" symbol means mandatory input and;

  e) the "|" symbol means a logical or. That is, either the user provides an arbitrary *objectProperty*, or it chooses specifically the "part of" label;

The typical CG representation of the what/which pattern is depicted in Figure 4.20, and it can be read as follows: which kind of entities interact by means of *objectProperty* with a specific domain entity?

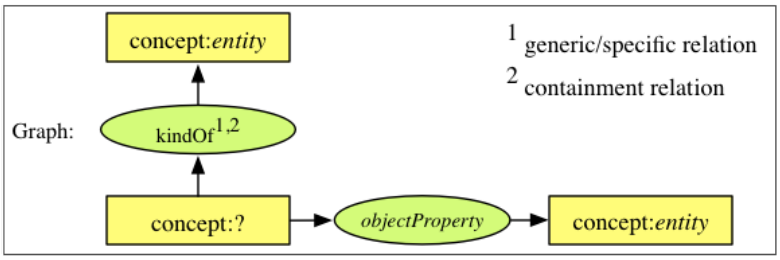


Figure 4.20: Conceptual graph view of the CQ What/Which pattern

In a CG linear form (textual) and according to the information provided by the DE, the “what/which” pattern may assume other two conceptual structures:

- $[concept :?] \rightarrow (Kind\ of) \rightarrow [concept : entity] \rightarrow (part\ of) \rightarrow [concept : entity]$, and read as follows: which kind of parts (entity) a whole (another entity) holds?

- $[concept :?] \rightarrow (Kind\ of) \rightarrow [concept : entity]$, and can be read as follows: what sub-types an entity holds?

Let us consider the CQ query example (Q.1.) and the correspondent CG representations of the Figure 4.21, which considers the existence of a domain concept type hierarchy; or as depicted in Figure 4.22, where there is not such hierarchy (which is the scenario discussed in this thesis).

Q.1. Which supplier provides sand?

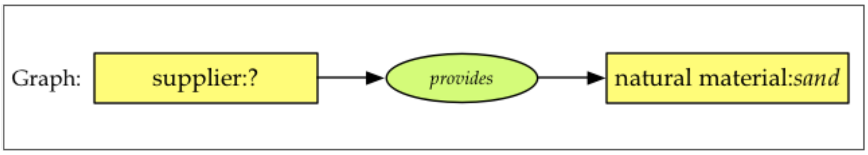


Figure 4.21: CG representation of a CQ What/Which pattern considering the existence of a concept type hierarchy

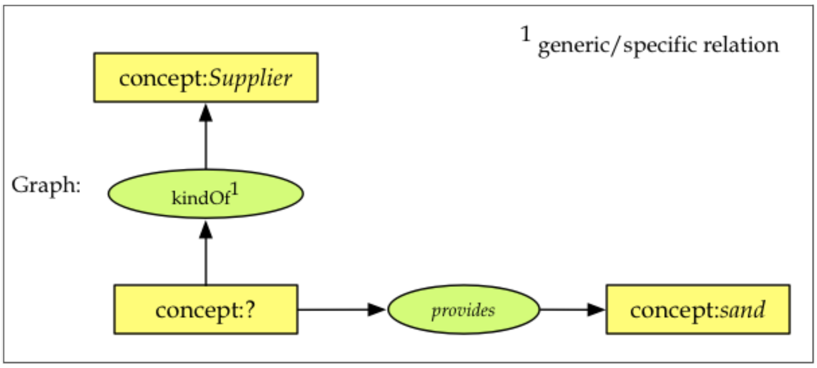


Figure 4.22: CG representation of a CQ What/Which pattern considering the absence of a concept type hierarchy

The first scenario (Figure 4.21) depicts the most common case found in the literature for building CGs. In the second scenario there is no support of a concept type hierarchy. In this case, every entity is of type *concept* and the linking phrase between two entities is a conceptual relation. The CG of the figure 4.22 could be read as follows: *Which is the specific supplier that supplies sand?* .

Further examples on this document about the CQ patterns will only consider that there is no concept type hierarchy available.

For the second pattern, it was selected an example (Q.2) where the experts provide the keywords corresponding to the *objectProperty* and only to an object, leading to a simpler representation, as showed in Figure 4.23.

Q.2. What is the contractor responsible for?

Table 4.14 synthesizes all the CQ types and the corresponding patterns and the CG representations.

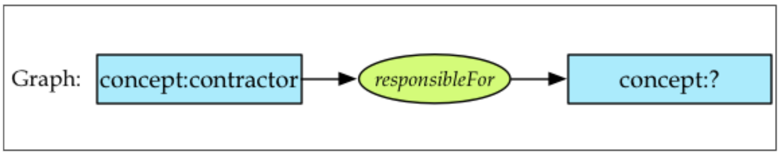


Figure 4.23: CG representation of a second CQ What/Which pattern

Table 4.14: Synthesis of CQ patterns and conceptual structure

Query type	Pattern and CG form	(Conceptual) structure
What	Pattern	$\textbf{What}\mid\textbf{Which} < \textbf{entity} >< \textbf{objectProperty}\mid\textbf{"part of"} > * < \textbf{entity} > *$
	CG form	$[concept :?] \rightarrow (Kind\ of) \rightarrow [concept : entity] \rightarrow (part\ of) \rightarrow [concept : entity]$
		$[concept :?] \rightarrow (Kind\ of) \rightarrow [concept : entity]$
How	Pattern	$\textbf{How} < \textbf{entity} >< \textbf{objectProperty} > * < \textbf{entity} > *$
	CG form	$[concept :?] \rightarrow (usage\ relation) \rightarrow concept : *] \rightarrow (objectProperty) \rightarrow [concept : entity]$
		$[concept :?] \rightarrow (usage\ relation) \rightarrow [concept : entity] \rightarrow (objectProperty) \rightarrow [concept : entity]$
Why	Pattern	$\textbf{Why} < \textbf{entity} >< \textbf{objectProperty} > * < \textbf{entity} > *$
	CG form	$[concept : *] \rightarrow (objectProperty) \rightarrow [concept : entity] \rightarrow (causal\ relation) \rightarrow [concept :?]$
		$[concept : entity] \rightarrow (objectProperty) \rightarrow [concept : entity] \rightarrow (causal\ relation) \rightarrow [concept :?]$
Where	Pattern	$\textbf{Where} < \textbf{entity} >< \textbf{objectProperty} > * < \textbf{entity} > *$
	CG form	$[concept :?] \rightarrow (spatial\ relation) \rightarrow [concept : *] \rightarrow (objectProperty) \rightarrow [concept : entity]$
		$[concept :?] \rightarrow (spatial\ relation) \rightarrow [concept : entity] \rightarrow (objectProperty) \rightarrow [concept : entity]$

4.5.1.2 How query type

The “how” query type, has the following pattern:

- **How <"many" | "much"> <entity> <objectProperty | dataProperty >* <entity>***, where:
  - *dataProperty* refers to an entity attribute or characteristic. Typically corresponds to an adjective.

However the following configurations (or forms) were disregarded:

1. "How much" . . . ;
2. "How" followed by an *abjective* (e.g., How tall?);
3. "How many" . . . ;

The first, for instance, is related to an amount of something, regarding its value or importance, whereas the second is related to a specific characteristic of an object/entity/concept. Both try to represent specificities that are not considered at the conceptual level of knowledge representation, i.e., the identification and representation of concepts and their relationships. As for the "how many", its intention is focused in counting the number of specific concepts of the most generic concept (named as object). The interest of this variant is focused on the after conceptualisation activities for typical search purposes. Its interest as a competency question to define the goals of the conceptual model is limited.

For better understanding let us consider the following CQ query example (Q.3.), which might be represented as illustrated in Figure 4.24:

Q.3. How are Construction materials Shipped ? Or, How to ship Construction Materials?

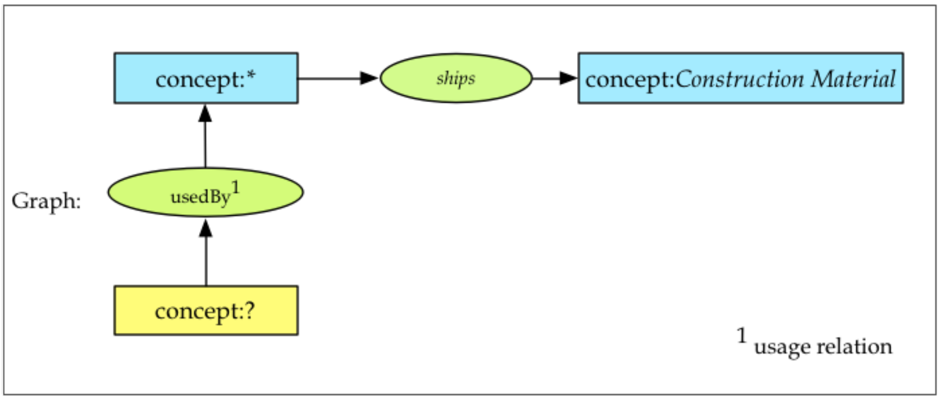


Figure 4.24: CG representation of CQ How pattern

The CG query depicted in Figure 4.24 may be read as follows: *Which is the concept (method) that is used by some other generic concept that ships construction materials?* Note that, conceptually, a usage relation is able to represent what may be used, but not how it is used specifically.

That would imply to axiomatize the conceptual representation to constrain their intended specific interpretation. For a better understanding, current conceptual vision over the usage relation is similar to the usage relation in UML.

The exception for these patterns occurs when the *objectProperty* token refers itself to a usage relation. In those cases the pattern correspond to the "What/Which pattern" in its simpler form, as the following textual representation indicates:

$[\textbf{concept} : \textbf{object}] \rightarrow (\textbf{objectProperty}) \rightarrow [\textbf{concept} :?]$

Example: What/Which uses construction materials?

$[concept : construction\, materials] \rightarrow (usedBy) \rightarrow [concept :?]$

4.5.1.3 Why query type

The "Why" query type, has the following pattern:

- **Why <entity >< objectProperty >* < entity >***, denoting two different configurations or forms (cf. Table 4.14).

Implicitly, the "why" question refers to a cause, that is, it seeks to identify the concept representing the motivation (reason) through which some other generic concept accomplish/realize some action/function. This encloses a cause-effect relation between the concept representing the cause and the concept that executes some action/function (*objectProperty*) over a specific concept (object). So, the interpretation for this pattern is as follows:

- There is an object, corresponding to the *effect* or *result*;
- There is a specific object ([concept:?]), corresponding to the *cause*;
- There is an agent of change, corresponding to some generic concept/object, ([concept:*]);
- There is an *objectProperty*, corresponding to the function or action;

For a better understanding let us consider the following CQ query example (Q.4.), which might be represented as illustrated in Figure 4.25:

Q.4. Why construction industry (can) produce (some) constructions results?

Thus the pattern could be read as follows: There is an agent (construction industry) (capable of) performing some action (*to produce*) over an object, which has a specific concept as root cause/motivation/justificaton.

Similar to the previous pattern, it can arise the case in which the *objectProperty* token refers itself to a causal relation, and, in that case the pattern is, again, reduced the "What/Which" pattern. The corresponding textual representation goes as follows:

$[\textbf{Concept} : \textbf{object}] \rightarrow (\textbf{objectProperty}) \rightarrow [\textbf{Concept} : \textbf{object}]$

Example: What is the cause of construction result production?

$[concept : constructionresult] \rightarrow (causedBy) \rightarrow [concept :?]$

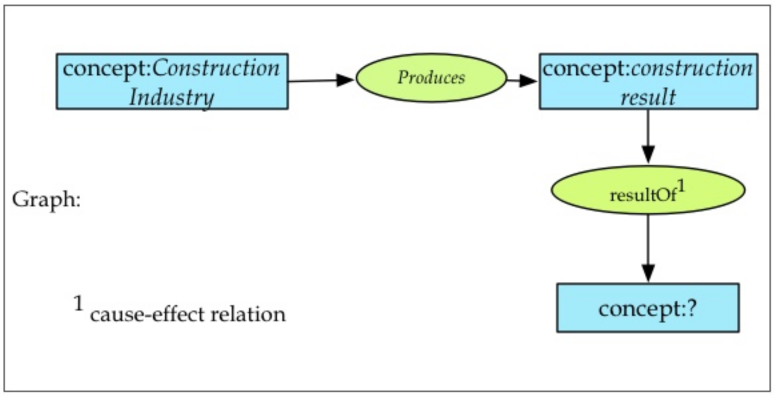


Figure 4.25: CG representation of Why CQ

4.5.1.4 Where query type

The "Where" query type, has the following pattern:

- **Where < entity >< objectProperty >*< entity >***, denoting two different configurations or forms (cf. table 4.14).

Implicitly, the where question refers to a space, that is, it seeks to identify the concept representing the physical location where two objects interact according to the defined *objectProperty*. This encloses a spatial relation as shown in Table 4.14.

Let us consider the following CQ query example (Q.5.), which might be represented as illustrated in Figure 4.26 and the expected retrieved after a CQ-driven conceptualisation process in Figure 4.27

Q.5. Where are Construction Materials stored ?

It may happen that *object Property* encloses itself a spatial relation and, in this case, the pattern reverts to a "What/Which" pattern. The corresponding textual representation goes as follows:
$[\textbf{Concept} : \textbf{object}] \rightarrow (\textbf{objectProperty}) \rightarrow [\textbf{Concept} : \textbf{object}]$
Example:what/which is the location of construction materials?
$[Concept : construction\ materials] \rightarrow (locatedAt) \rightarrow [Concept :?]$

4.5.1.5 CQ graph structure formalization

As mentioned earlier a CQ can be represented as a CG and thus, described in a formal way.

Definition 4.4 (CQ Vocabulary)**.** *CQ vocabulary is composed as:*

- *a single concept type designated by the term : $concept$;*
- *a set of conceptual relations (T_R) corresponding to CRRM;*

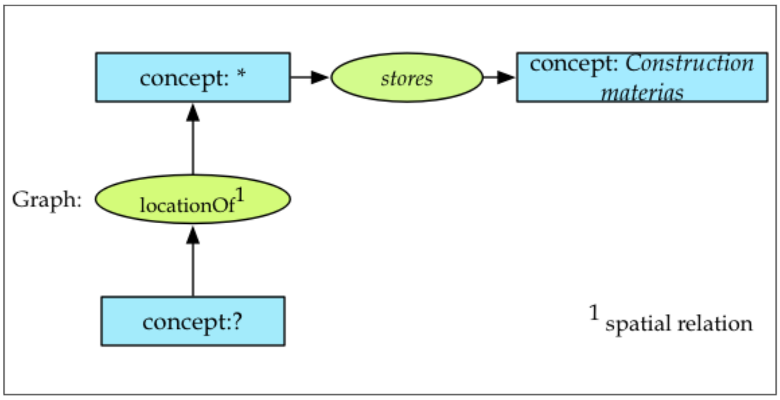


Figure 4.26: CG representation of CQ Where pattern 1

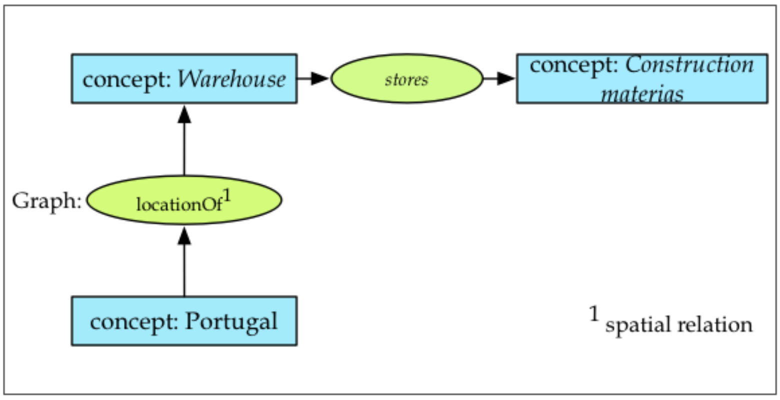


Figure 4.27: CG representation of a retrieved CQ Where query

- *a set of unordered query types ,* $T_Q = \text{"what"}, \text{"how"}, \text{"why"}, \text{"where"}$;
- *a set of conceptual markers* $I = t_1, t_2, ..., t_n$, *whose elements correspond to specific domain terms that can be used to designate specific concepts.*

Definition 4.5 (Assignment Relation). *Assignment relation is a surjective relation given by function* θ *, which associates types of conceptual relations*(r) *to types of CQ queries* (q)*, such as*
$\theta : T_R \longrightarrow T_Q$*, where:*$\forall r \in T_R, \exists q \in T_Q, \theta(q) = r$

Definition 4.6 (Competency Question). *A competency question (CQ) is a CG corresponding to a 4-tuple* (C, R, E, λ)*, where:*

- *C, is triple of concept nodes* c_1, c_2, c_3;
- *R, is a tuple of relations* r_1, r_2;
- *E, is a set of edges* e_1, e_2, e_3, e_4 *connecting two disjoint sets C and R;*
- λ*, is a labeling function, such as:*
  - ***A concept node*** *c is labeled as:* $(type(c), marker(c))$*, where* $type(c)$ *is always "concept" and* $marker(c)$ *corresponds to: i) the specific* $concept_1$ *we are focused on, in case of concept* c_1*, and; ii) a domain term from I, in case of* c_2 *and* c_3*, where* c_3 *is explicit indicated by experts and* c_2 *may be any term from domain terminology represented by a generic concept.*
  - ***A relation*** *r is labeled as* $\lambda(r) \in T_S \subseteq T_R$ *(which is the same as* $type(r)$*), according to a source query q, whose type is given through the label function* $\lambda(q) \in T_Q$ *(which is the same as* $type(q)$*), where:*

$$\forall q, \exists r_1, r_2 : \theta(\lambda(r_1)) = \begin{cases} GENERIC\_SPECIFIC\_TYPE & if \lambda(q) = WHAT \\ CONTAINMENT\_TYPE & if \lambda(q) = WHAT \\ USAGE\_TYPE & if \lambda(q) = HOW \\ CAUSE\_EFFECT\_TYPE & if \lambda(q) = WHY \\ SPATIAL\_TYPE & if \lambda(q) = WHERE \end{cases} \bigwedge \lambda(r_2) \in T_R$$

  - *an edge labelled i between a relation r and a concept c is denoted* (r, i, c)*. The conceptual structure is given by the edge labeling function and the underlying edge orientation, which depends, by its turn, on the CQ type and the underlying CG form. So, for the CQ query of type "WHY", it goes as follows:*
    - $e_1 = (r_2, '1', c_2)$*, indicating the connection from* c_2 *to* r_2;
    - $e_2 = (r_2, '2', c_3)$*, indicating the connection from* r_2 *to* c_3;
    - $e_3 = (r_1, '1', c_3)$*, indicating the connection from* c_3 *to* r_1;

  * $e_4 = (r_1,' 2', c_3)$, *indicating the connection from* r_1 *to* c_1*;*

- *whereas for the other CQ query, it goes as follows:*
  * $e1 = (r_1,' 1', c_1)$, *indicating the connection from* c_1 *to* r_1*;*
  * $e2 = (r_1,' 2', c_2)$, *indicating the connection from* r_1 *to* c_2*;*
  * $e3 = (r_2,' 1', c_2)$, *indicating the connection from* c_2 *to* r_2*;*
  * $e4 = (r_2,' 2', c_3)$, *indicating the connection from* r_2 *to* c_3*;*

4.6 Corpus-based elicitation of conceptual representations

Over the last decade, along with the maturity level of NLP [14] tools and a multidisciplinary view on knowledge representation, corpus-based approaches for conceptual modelling have gained proeminence in the literature, whether by new methodologies/frameworks (Aussenac-Gilles et al., 2000) or information extraction algorithms for knowledge discovery (Sarawagi, 2008). Despite their quality, existing extraction algorithms usually require a large customisation effort as mentioned in (Baroni et al., 2010). Within this thesis it was designed a terminological approach to provide to domain experts artefacts that were almost exclusive to computational linguistic experts.

The idea is based on the assumption that some linguistic patterns could be found in texts, denoting the existence of a domain concept or a relation. A *verb* or a *verb* plus a *pronoun*, for instance, could indicate a possible designation for a conceptual relation, while a *noun* could indicate a a term candidate. Moreover, the pattern $< noun >< verb >< noun >$, within a sentence, could indicate a brand new conceptual structure. During the artefact design a special effort was made to ensure: a) term context and metadata retrieval from unstructured data, and b) derived facts taken from achieved conceptual representations. At this level, contexts are equivalent to the sentences extracted from the several resources made available for the CCP project, in relation to a specific term.

As mentioned earlier, the CCP requires an organised domain corpus as it could be used to validate the use of specific terms within conceptual representations. This pre-required activity includes tasks such as collecting, describing, storing and classifying provided resources. The main objective is to describe a specific resource regarding its bibliographic properties and later to classify it according to terminological characteristics. Upon creating the corpus, the resources should be indexed. Afterwards, and at any time during the conceptualisation process, term contexts could be immediately retrieved in order to obtain clues on new possible concepts or relations linked to the current concept and to the corresponding conceptual representation. Moreover, term contexts could additionally support discussion around a specific concept, justifying its use by showing evidence of term occurrence in a corpus or inferring on concept semantic metadata (e.g., by using an RDF [15] triple store), or even highlighting patterns ($< noun >< verb >< noun >$)

[14] Natural Language Processing
[15] Resource Description Framework

within the text where at least one of the terms in the pattern is already in use. This suggests that the overall conceptual structure is incomplete.

To identify patterns in the retrieved term contexts, a simple method is proposed. It depends on pre-processing tasks, such as sentence split, resource indexing and *part-of-speech tagging*. This requires a set of services to support the pre-processing tasks associated to index the resources of the domain corpus (see collect and classify workflows on Table 4.5) and to interact with the indexing engine in order to return the contexts associated to a specific term (see Content-Based Retrieval workflow on Table 4.6).

The **Collect and Classify workflow** (Table 4.5) encloses activities to support users in collecting relevant resources related to the intended conceptualisation goal. For each textual resource attached, the sentences are identified and the document is converted to a standard format (XML) and indexed. The **Content-Based Retrieval workflow** (Table 4.6) encloses activities to support the search, identification and retrieval of the context related to each term/concept in the current model.

The **Content-Based Retrieval workflow** works as follows: for each term found, the sentence containing the term is tagged. At the end, the context is retrieved and specific words/compound words are highlighted (coloured) according to the tagged value (e.g., *Noun, Verb, Adverb. ...*). This retrieving task could occur at several stages of the conceptualisation process according to the following scenarios: a) any time a concept is identified and collected, its context could be retrieved and viewed; b) when linking two concepts, a possible linking phrase (conceptual relation) is suggested, if the terms co-occur; c) at any time during the construction of the conceptual representation the user may call for neighbor terms of a certain concept, thus making it possible to collect possible related concepts; d) at any time during concept discussion on finding new concept variances.

The method for identifying and extracting patterns in the retrieved contexts calls for the aforementioned workflows and it is proposed as illustrated in table 4.6.

For a context C retrieved for a term t, all numeric values D, punctuations P and other special characters and words from the stop-words list L will be eliminated from the context. Moreover, a *stemming* task will be performed. The resulting $cleanedcontext$ C' will be segmented into text snippets S according to the following rules R:

1. S must contain exactly one occurrence of t and another term whose tag denotes a *noun* n

2. between t and n there should be at least a *verb* or *verb-phrase* v

3. the maximum length of S is w word

4. S could contain t at the beginning, middle or at the end of the text snippet. S will then be ranked according to the frequency of each S: $f(S_i)$

5. $f(S_i)$ is calculated by counting how many times the term t co-occurs with other term either present in the model or in C in each S. (see Figure 4.28).

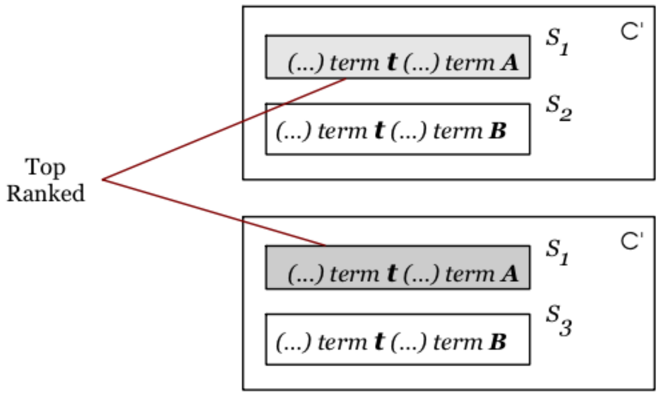


Figure 4.28: Ranking patterns

4.6.1 Illustrative Experiment

An experiment was conducted illustrating a possible scenario where the approach presented here could support the collaborative construction of conceptual representations. The objective was to: i) qualitatively assess the relevance of the description and understanding of the facts behind the approach; and ii) to legitimise the approach as a means to improve the creation of an agreed conceptualisation with a group of domain experts. Therefore, the illustration example started to be outlined as part of the H-Know ($http://www.h-know.eu/$) project. H-know was an European project (FP7-NMP-2007-SME-1) in the area of old building restoration and maintenance, particularly in the cultural heritage domain. It was a large project involving 15 partners from 5 countries and 7 work packages. Our research group has participated as an RTD partner, responsible for the work package 2, whose goal was to specify the methodology, ontology framework and the services for the H-Know solutions. This revealed to be a fitting backdrop to apply our approach, taking advantage of an existing domain corpus. The corpus was comprised by scientific, technical, juridical and didactic texts according to a predefined typology that took into consideration the domain's communicative and professional specificities. It had 532 000 tokens.

The scenario that would underpin the example was established, selecting two specific groups (G1 and G2) from two different partners (P1 and P2). There was a third group acting as moderators/observers (the author). Due to the very short time-frame available, and restrictions of the several partners' agendas, only two groups were invited. The approach was explained to both groups, but while G1 had complete access to the context-based features developed so far, the access to those features was denied to G2. Additionally, three documents selected from the urban rehabilitation corpus were indexed using the Solr server [16]. While the groups were performing their modelling tasks, their actions were being monitored. Along the conceptualisation process (CP), users added concepts and relationships between concepts, either on their own using extraction services or available templates. After releasing a term (here referred to as concept) on the

[16]http://lucene.apache.org/solr/

IT tool canvas, several tasks may follow: a) providing a definition to the concept; b) completing the structure adding another term (concept) and a linking phrase between two terms (concepts). Context-analysis could be helpful on both tasks.

Still in the scope of urban rehabilitation, at a certain stage of CP, the conceptual structure started being defined around the concept of *moisture control*. The task was initiated by adding the *moisture control* term. After that, the main challenge was on finding how and to which term *moisture control* should be linked to. For G1 the solution was achieved after analysing the contexts by identifying terms (nouns) that co-occur with the added term (*moisture control*) and possible linking phases connecting them. The linking phases are typically *verbs* or expressions that match the following lexical patterns:

1. a verb preceded or followed by a preposition or subordinating conjunction;
2. a verb preceded or followed by a coordinating conjunction;
3. a verb preceded or followed by a "TO";
4. verb preceded or followed by a determiner;
5. a verb preceded or followed by another verb.

The information retrieved by the contexts related to moisture control provided the following information:

> *However, good* ***moisture control*** *design depends on {pattern} a variety of parameters {noun} such as climate conditions {noun} and construction type {noun} which changes {verb} from region {noun} to region noun.*

The tags indicating if a word is a *noun*, a *verb* or other *lexical pattern* such a candidate linking phrase, were added to the example for information purposes only. According to the context retrieved, it is possible to see that "*moisture control depends on climate conditions and construction type*", since *climate conditions* and *construction type* co-occur with *moisture control* mediated by the lexical pattern "*depends on*".

Additionally, the following context was also retrieved:

> *Effective moisture control has to {pattern} respond to {pattern} the exterior as well as the interior moisture loads {noun} acting on {pattern} building constructions {noun}.*"

From this context it can be understood that "*moisture control responds to moisture loads*, where: *moisture control* and *moisture load* are two candidate terms linked by the linking phrase: *responds to*.

G2 started a discussion on the possible terms and linking phrases when browsing the available documents, while G1 was indeed faster in finding the appropriate term with a high level of agreement as the achieved conceptual structure turned out to be justified by the information that came from the contexts.

After analysing the context, G1 added the following propositions:

a) *Moisture control depends on climate conditions*;

b) *Moisture control depends on construction type*;

c) *Moisture control responds to moisture loads*.

The same approach was used for the "*moisture load*" term, which led to the following assertion: "*Construction moisture* is a *moisture load*." The process continues for "*construction moisture*" until the conceptual structure is achieved.

This test scenario was based on the simple observation of the process of creating conceptual representations, with or without a specific variable. In this case, the variable was the presence or lack of specific features related to the retrieval of term contexts and pattern identification. The results from the described example showed that the presence of such a corpus-based approach is an interesting and promising add-on to assist groups of users in creating conceptual representations.

4.7 Formalisation stack: Formalising semi-formal conceptual representations

To ensure that a group of experts is able to represent a sharable view of a domain, there must be available, both informal and formal knowledge representation formalisms. On one hand DEs are reluctant in using technical formalisms outside their domains, but on the other hand the conceptual representations should be rigorously defined in order to be machine-readable. So, a kind of formalisation stack allowing to go from an informal conceptual level to a more formal representation of knowledge is needed.

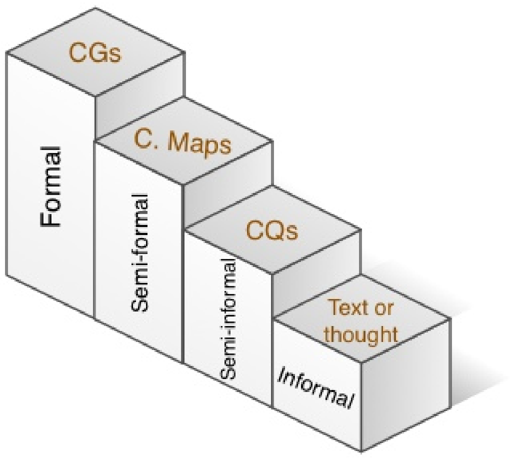


Figure 4.29: Formalisation Stack

According to Figure 4.29, the first step towards explicit knowledge might be through the specification of CQs (see Section 4.5). This structured-based approach to CQs specification allows to move from a highly expressive but ambiguous knowledge representation (the natural language), to a semi-formal knowledge representation.

After structuring the domain knowledge (using CQs), it is possible to convert the resulting conceptual assertions into a generic semi-formal representation such as concept maps. The last step consists in transforming the semi-formal conceptual representations into a formal knowledge representation such as CGs.

4.7.1 Transforming semi-formal in formal knowledge representations

There are similarities between semi-formal and formal conceptual representations, namely:

- Both are represented as graphs (i.e., have nodes and arcs and edges);
- Both have concepts;
- Both have relations;
- Both can be translated into FOL (First Order Logic).

A concept map could be seen as a direct labeled graph (Figure 4.30), whose basic elements are the *concepts*, and the *relations* (labeled links connecting two or more concepts).

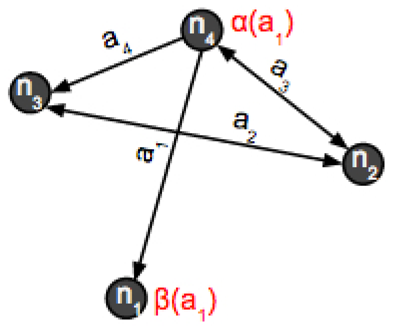


Figure 4.30: Concept map as labelled direct graphs

Considering the Figure 4.30, the concepts and relations of a concept map can be represented in FOL as follows:

- A concept c is an *unary* predicate named concept and with one variable as argument.
  Example: $concept(n_i) = n_i$ is a concept;
- A relation r is a *2-ary* predicate and t_1, t_2 are terms, where the order of the arguments indicates the direction of the relation.
  Example: $a_1(n_4, n_1) = a_1$, is directed from n_4 to n_1.
- The predicate name is the labeled *arc* and the arguments are $\alpha(a_1)$ and $\beta(a_1)$.

Definition 4.7. *A labeled direct graph is defined as a 5-tuple* $CM = (N, A, \alpha, \beta, \lambda)$*, where*

- N *= finite non-empty set of nodes representing concepts* $n_1, \ldots, n_n$
- A *= finite non-empty set of arcs representing relations* $a_1, \ldots, a_n$
- α *amd* β *are mappings from arc* $a \in A$*, with:*
  - $\alpha(a) \in N$ *as the source of a and*
  - $\beta(a) \in N$ *as the target of a*
- *an arc* a_i *is labelled by* $\lambda(a_i)$
- $\forall a_i[\lambda(a_i) \rightarrow \tau(n)]$*, where* $\tau(n)$ *is a tuple (ordered list) of nodes (concepts)*

Definition 4.8. *Additionally let us consider and define the concept of edge as triple* $E = (\alpha(a), a, \beta(a))$ *containing the source node, the arc and the target node. The set of edges of a concept map are denoted by* $SE_{CM}\{(\alpha(a), a, \beta(a)\}$

Example: According to Figure 4.31, the triple (n_4, a_1, n_1) is here considered an edge.

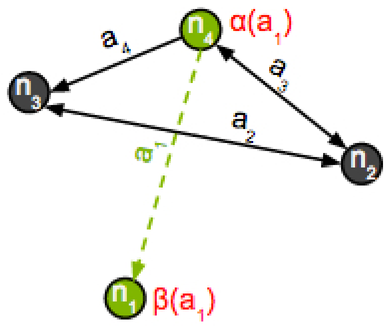


Figure 4.31: Labeled direct graph: edge definition

Concept maps and conceptual graphs are semiotically similar since both can be viewed as a set of nodes and arcs representing a conceptual view of a particular domain. In that sense it its possible to define a direct mapping between the two graphs.

Definition 4.9. *The complete transformation between a concept map and a conceptual graph is given by the function* $f : G_{CM} \rightarrow G_{CG}$ *in which:*

- G_{CM} *: A conceptual model represented as a concept map graph*
- G_{CG} *: A conceptual graph represented as a graph*

The first step towards transforming G_{CM} to G_{CG} is to separate node and edge transformation, where:

- A node transformation specifies which elements of G_{CM} are represented as a node of G_{CG}

- An edge transformation defines which elements relate the nodes

The transformation function $f : G_{CM} \rightarrow G_{CG}$ encloses two partial transformations functions and an operation, namely:

- function $fn : G_{CM}(N_1) \rightarrow G_{CG}(N_2)$, corresponding to the transformation between a concept of concept map and a conceptual graph concept node, such as:
  - $\forall n \in N_1, \lambda(n_1) = \lambda(fn(n_1))$
- function $fr : G_{CM}(A) \rightarrow G_{CG}(R)$, corresponding to the transformation between a concept map arc to a conceptual graph relation node, such as:
  - $\forall a \in A, \lambda(a) = \lambda(fr(a))$
- Edge addition operation $EA[G_{CG}, SE_{CM}]$, such as:
  - $\forall e_1 \in SE_{CM}, \exists e_2, e_3(e_2 = (fr(a),' 1', fn(\alpha(a))) \wedge e_2 = (fr(a),' 2', fn(\beta(a))))$, considering that
  - set of edges of a concept map is denoted by: $SE_{CM} = \{(\alpha(a), a, \beta(a)\}$
  - set of edges of a concept graph, incident to a relation $r : SE_{CG} = \{(r, i, c)\}$, where $i = 1$ and c is a source concept

4.8 Short Conclusion

In conclusion, it seems legitimate to approach the conceptualisation process according to a holistic perspective in order to better support the socio-technical phenomenon behind it. This has been the purpose of this chapter through the design of the CF and its supporting specific artefacts, allowing its implementation in an IT tool. As planned, the artifacts described herein have their foundations shared between the terminology and knowledge representation. This "symbiosis" allowed to bring domain experts to participate in a process traditionally exclusive to knowledge engineers and / or terminologists. The design of the artefacts was done iteratively and under the observation of the domain experts themselves. The preliminary results showed to be promising in order to carry out research towards a more rigorous evaluation.

In this chapter it was discussed:

- how to enable domain experts to elicit conceptual representations requirements in a structured way, through the use of Competency Questions;
- how to organise and represent conceptual structures easing sharing, through CRRM;
- how to transform semi-formal conceptual representations into computer-readable ones, using a graph-based approach to convert conceptual representations in formal conceptual graphs.

- how domain resources (e.g., documents) help to discover new domain knowledge (e.g., terms and linking phrases), following a corpus-based approach.

Chapter 5

Build-to-reuse conceptual representations in ConceptME

"The number one benefit of information technology is that it empowers people to do what they want to do. It lets people be creative. It lets people be productive. It lets people learn things they didn't think they could learn before, and so in a sense it is all about potential."

Steve Ballmer

The challenge related to the identification of the concepts and conceptual relations, as well as their intelligible graphical representation, requires a multidisciplinar approach, which is configured in a technological environment around the conceptualisation process. A theoretical binomial based on knowledge representation and terminology, was incorporated in the conceptualisation framework (CF), which describes the model to support the conceptualsation process. The CF, by its turn, was designed to address both the domain expert's and the computational processing perspectives on producing useful and reusable conceptual representations. Hereupon, in this chapter, it is discussed the technological approach to implement the CF and its supporting artefacts.

5.1 Technological environment

Conceptual modeling encloses the main activities inherent to the conceptualisation process. The way those activities are executed depends on the technological availability of the artefacts described in the previous section. The figure 5.1, defines the necessary environment to implement the vision portrayed by the CF.

During the different conceptual modeling stages, the domain experts must have available the appropriated means to:

- structure a set of questions (CQs), to which the final conceptual representation should provide an answer, easing the initial conceptualisation steps;

- collect and organise relevant information of the domain and identifying term candidates;
- graphical organise the concepts and relations through a set of templates based on CRRM towards the definition of meaningful conceptual structures.

The result of conceptual modeling activities is a semi-formal model, whose "shareability" and reusability depends on its formalisation. The CG Engine allows the transformation of a semi-formal conceptual model (e.g., concept map) into formal conceptual graphs.

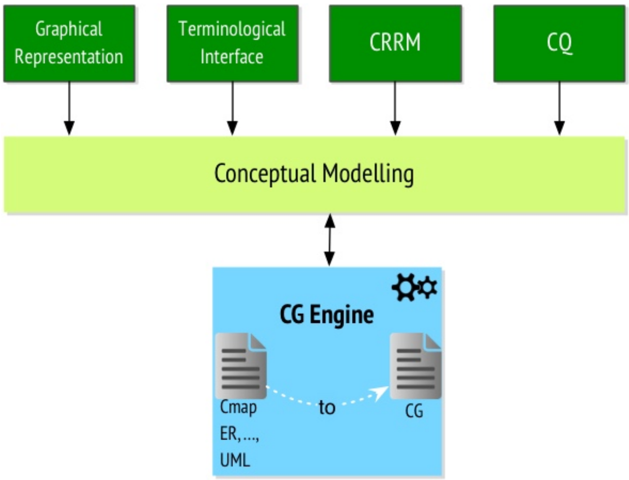


Figure 5.1: Tehnical Environment

The environment described in Figure 5.1 was assembled in the conceptME platform., whose architecture is described here.

5.2 conceptME architecture

As described earlier in this thesis, the conceptME started to be developed under the CogniNet project in order to support the conceptualisation process. The platform is based on semantic technologies and started to offer a set of basic functionalities to manage (create, read, update and delete) conceptual models and conduct discussions around them. Meanwhile, further developments were required in order to cope with the pragmatic properties of conceptual representations (i.e., short-term validity, contextual and situational dependency), seeking to ensure the double commitment between informal and formal conceptual representations, implementing the formalisation stack. Therefore, specific methods were designed to be accommodated into service libraries within conceptME, supporting the synthesis of the domain experts knowledge towards the development of fully interoperable conceptual representations.

At its core, conceptME is a wiki-based platform joining semantic technologies with content and metadata management. Within conceptME, the conceptual models are mapped into wiki pages, creating a semantic repository of wiki articles. Figure 5.2 shows the correspondence between the typical organisation of the conceptual structures built in conceptME and the organization of that information in MediaWiki. Accordingly, both concepts and relations correspond to wiki pages belonging to a specific category. In order to extend MediaWiki base model to accommodate conceptual structures, a set o categories were created: Concept and Relation categories.

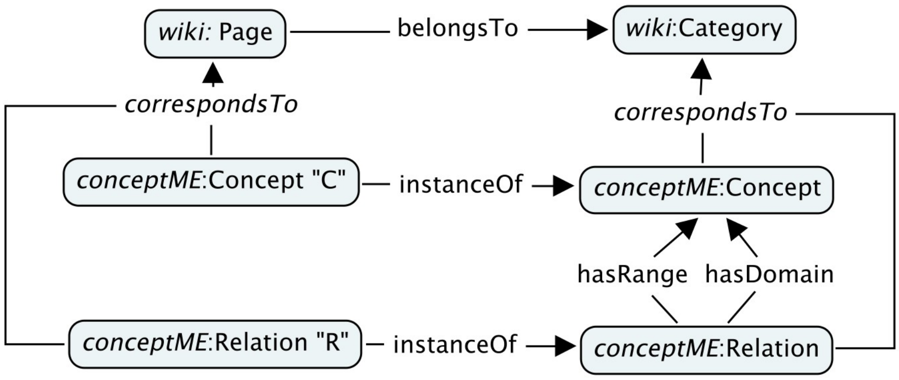


Figure 5.2: ConceptME to Mediawiki metamodel

Regarding the interface with the user, there was the concern to enrich the user experience adding a more friendly layer to ease the interaction with the wiki and its semantic features. This was achieved combining SemanticForms[1] and a set o templates using Smarty framework[2]. Hence, the technological stack was assembled as portrayed in Figure 5.3, i.e., on top of a standard MediaWiki[3] installation, there is the Semantic MediaWiki extension[4], adding semantic capabilities to the standard MediaWiki platform (e.g., tagging, content organisation around semantic properties, export wiki pages to RDF format, facet browsing, advanced search mechanisms, ...). Additionally, SemanticForms together with specific developed templates enhanced the user interface.

Additionally, a set of specific extensions (named as *ConceptMEXtensions*), complete the *conceptMEngine* as described in Figure 5.3). Four extensions were designed:

- *conceptMExtract*;
- *conceptMEXport*;
- *conceptMEVis*;
- *conceptMENegotiation*.

These extensions implement the baseline workflows to support the CF and the underlying conceptualisation process in a collaborative way. However, specific operations associated to each module are available through a *service library*, following a service-oriented architecture paradigm. The provision of services is made by the WSO2 Server[5]. Finally an Apache Solr[6] server is be-

[1] http://www.mediawiki.org/wiki/Extension:Semantic_Forms
[2] http://www.smarty.net
[3] http://www.mediawiki.org/wiki/MediaWiki
[4] https://semantic-mediawiki.org
[5] http://wso2.com/products/enterprise-service-bus/
[6] http://lucene.apache.org/solr/

ing used for indexing purposes, allowing rapid attainment of the indexed information about the documents. The obtained information is used during term context retrieval (as described in 4.6).

The components that are implemented in the service bus environment (WSO2) and decoupled with the conceptME basis tool, are:

- **Terminological service**, including both term extraction an NLP[7] services (e.g., stemming, POS tagging, Sentence Split, tokenize, . . .)
- **CRRM services**, which assist conceptual relations elicitation, feeding the templates used to add new conceptual structures.
- **CQ services**, used to define the goals of a conceptualisation process project and to build goal-oriented conceptual structures.
- **Transformation services**, converting: i) semi-informal conceptual structures in semi-formal representations and; ii) semi-formal conceptual representations in formal conceptual graphs.
- **Graph operations services**, used to derive new graphs from existing ones.

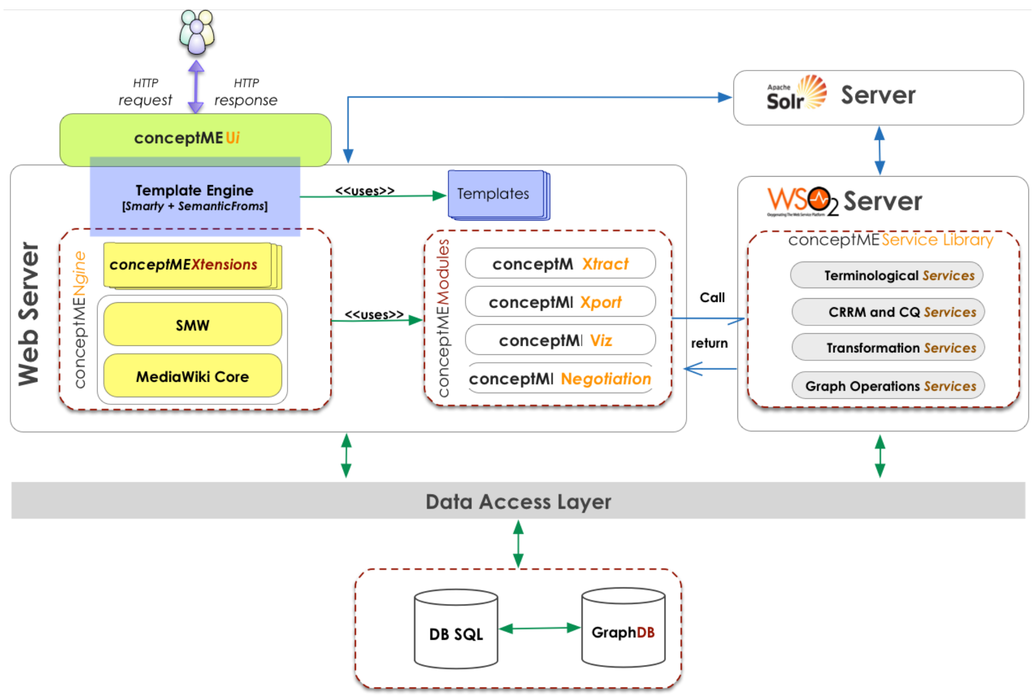


Figure 5.3: ConceptME architecture

ConceptME*Xtract* ConceptMEXtract module implements the workflow that allows the execution of the terminological methods needed for term extraction from one or more resources. The terminological extraction service implements the GaleXtract[8] as it delivers results from documents

[7]Natural language Processing
[8]http://gramatica.usc.es/ gamallo/gale-extra/index2.1.htm

in the Portuguese, English and French language (Barros et al., 2012). However, given its limitation in not allowing the extraction of mono-terms (terms with a single word), the extraction service was extended with implementations of AlchemyApi[9] and FiveFilters[10], providing more extraction opltions to the user.

ConceptME*Xport* Semantic MediaWiki extension has already specific features for RDF import/export, but it exports all the content of a wiki page. According to the conceptME metamodel 5.2, a wiki page refers only to a particular concept or relation. The export module allows to export all wiki pages related to a particular conceptual representation to RDFS[11] format, considering only the concepts and conceptual relations and excluding all specific wiki content.

ConceptME*Vis* In ConceptME, conceptual modeling activities follows a graphical-driven approach in order to provide more interactivity when dealing with models, facilitating their navigation and reading. The model that supports the development of conceptual models, despite being simple and providing the wiki pages with semantic integration capability, does not have a hypertext model to support browsing and manipulating the several models, concepts and relations. The absence of such model requires the browsing among several pages to depend on the graphical models represented in conceptME. The graphical modeling capabilities of conceptME allow that the (hyper)textual descriptions (concept and relation definition, model discussions, etc.) developed are semantically integrated. Moreover, it allow the users to add concepts (represented as circles) and edges (represented as arrows) connecting concepts, directly to the canvas[12]. The graphical modeling environment was designed to provide, in a single space, all the tools and resources necessary for the creation and manipulation of conceptual models, allowing users to focus on modeling activities. In fact, users can enable the full screen mode of the modeling space hiding all wiki like features and keeping just what is needed to build a conceptual model proposal (see Figure 5.4).

CoceptME*Negotiation* This module provides a negotiation baseline enclosing a set of basic features such as:

1. suggestion mechanisms;

2. discussion forums with automatic linking (i.e., when somebody writes a term in a discussion post that corresponds to the designation of an existing concept , then an hyperlink to that concept is automatically incorporated in the forum post);

3. auto-complete;

[9]http://www.alchemyapi.com/
[10]http://fivefilters.org/term-extraction/
[11]RDF Schema ($http://www.w3.org/TR/rdf-schema/$
[12]Canvas is an HTML element designed to enclose an area for dynamic graphics rendering

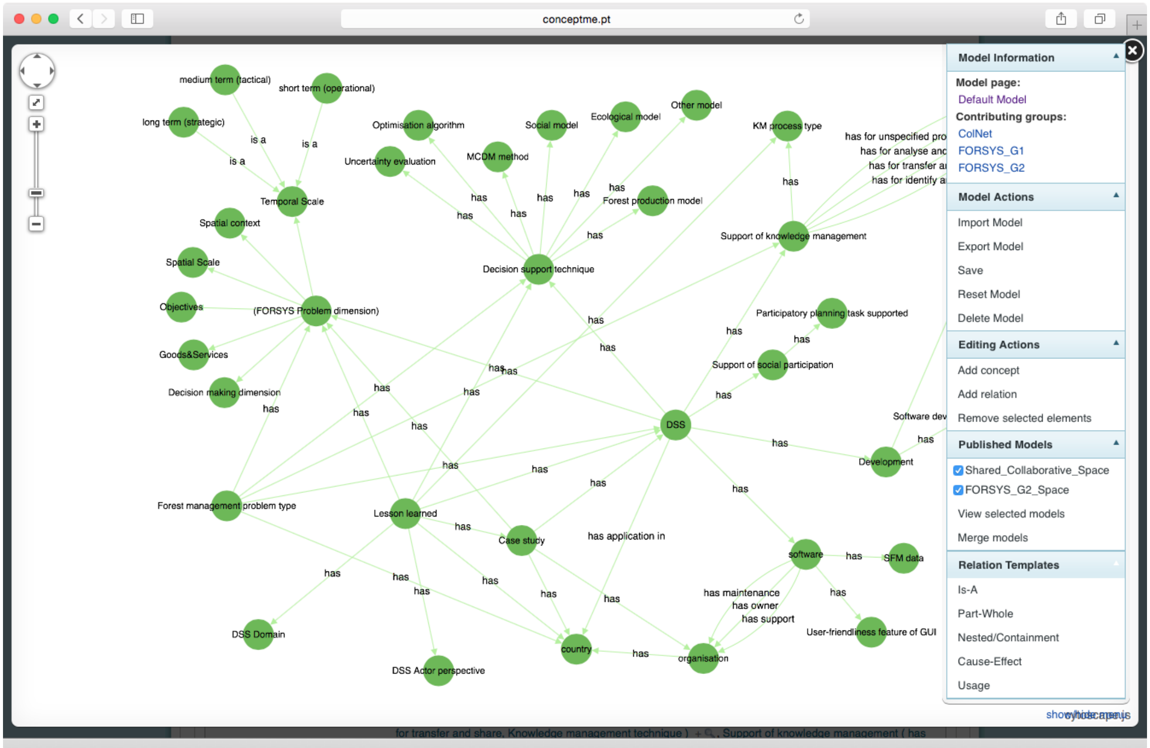


Figure 5.4: ConceptME modeling layout

4. possibility to define alias and term definitions. This module provides the interface and environment conditions, allowing to connect other advanced negotiation mechanisms, such as, the argumentation-based conceptual negotiation approach described in (Pereira et al., 2012).

Furthermore, the conceptME allows a complete managing of the collaborative conceptual modeling projects. The conceptual structure depicted in Figure 5.5 allows to support the major collaborative conceptual modeling activities envisaged by the conceptualization process.

Since conceptME includes in its core an SMW instance, each instance of a project, collaborative space, model, concept or relation, corresponds to a wiki page belonging to a specific category. The pages are semantically integrated, either by the taxonomy shown in Figure 5.5, or through a predefined list of metadata (properties) associated with each informational item created in conceptME. Thus, and as an example, it is possible to know to which model a certain concept or relationship "belongs" and to which project belongs the respective model. Even though this model is considered suitable for content browsing, it demands efficient visualization and manipulation mechanisms to streamline the creation, organisation and discussion of the models, without the need of the expert to lose track of collaborative workspace or "interrupt" his system thinking process of to navigate in the content.

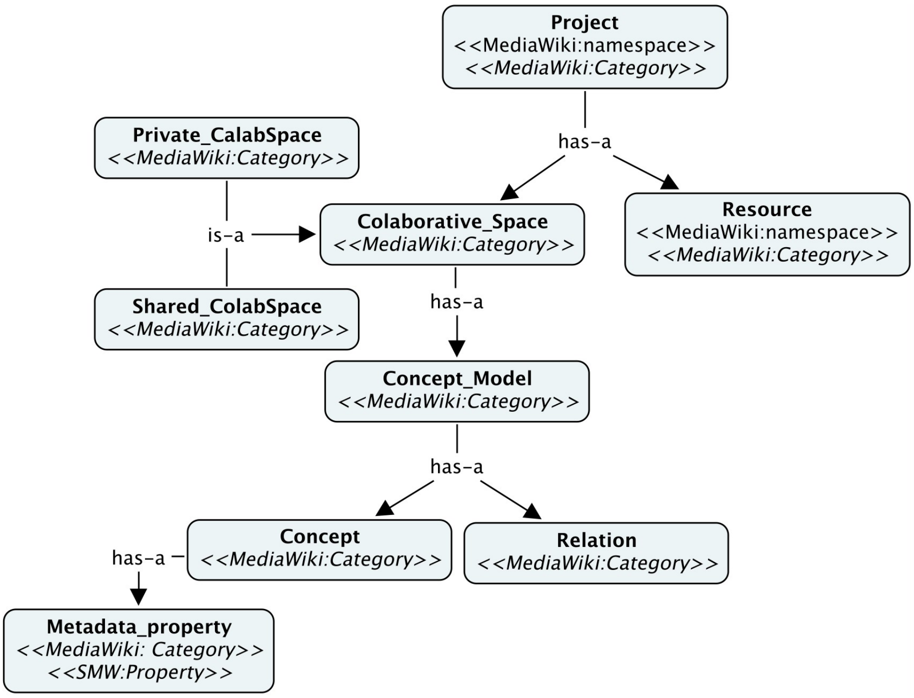


Figure 5.5: conceptualisation project metamodel: the conceptME approach

5.2.1 Modeling conceptual graphs in conceptME

Revisiting the main research goals of this thesis, the reusability of conceptual representations is at the core of the CF requirements. Thus, from a technological point-of-view, the persistence of the developed models is of utmost importance. The design of the data layer for conceptME implies the existence of two kind of data repositories: 1) a relational database for supporting the wiki base model, and; 2) a graph database to support conceptual graphs processing (see Figure 5.3). "*Graph databases are optimized for the efficient processing of dense, interrelated datasets. In these databases, the atomic entity is the graph as a whole. The typical data model is the property graph 5.6. By supporting the interrelation of data, graph databases allow for fast traversals along the edges between vertices.* ". (Rodriguez and Neubauer, 2010).

A property graph, also known as general graphs, are "*directed, labeled, attributed, multi-graph. The edges are directed, vertices/edges are labeled, vertices/edges have associated key/-value pair metadata (properties), and there can be multiple edges between any two vertices. (...) The property graph is common because allow to express other types of graphs by simply abandoning or adding particular bits and pieces.*" (Rodriguez and Neubauer, 2010). With the implementation of a graph database it is possible to process conceptual representations more efficiently without having to handle the wiki's core and wiki's database. Additionally, the platform benefits from a more interoperability "cleanness" e.g., the models can be easily exported to other formats such as OWL. However, there is not conceptual graph databases, and the native graph databases do not have such kind of support. For that reason it was used the Gremlin[13] language, which is a graph transversal language, implementing a property graph model. Through gremlin it is possible to develop a domain specific language allowing to create conceptual graphs. Figure 5.6 represents the DSL model to carry out conceptual graphs processing interacting with the graph database.

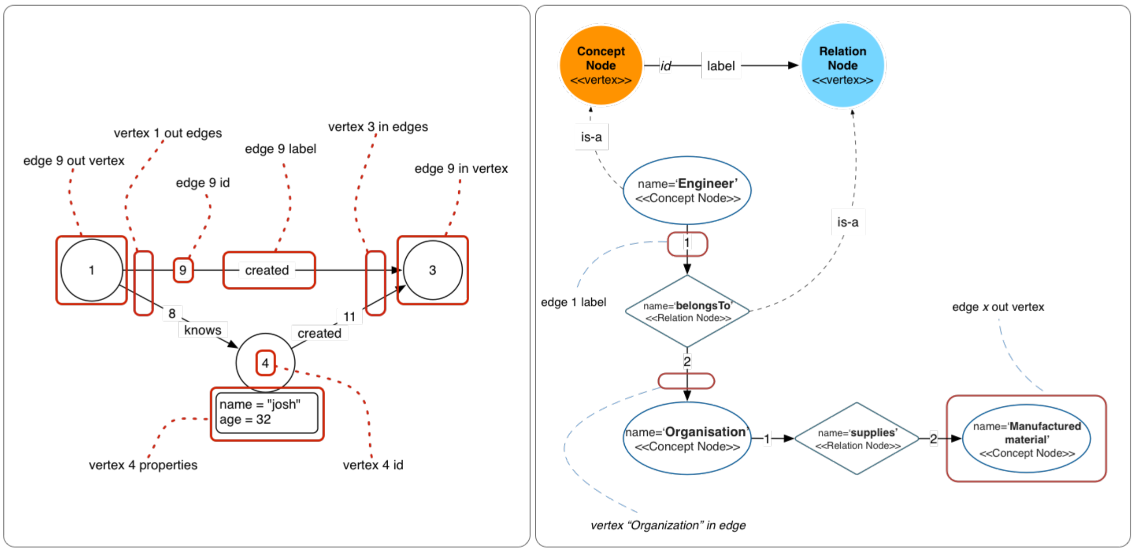


Figure 5.6: Property graph model (on the left) and CG DSL metamodel

Modeling CGs using Gremlin DSL implies to "emulate" a bipartite graph using the property model features, so that all relationships and concepts become nodes and use edges to interlink the two nodes. The edge label would indicate the *arity* of the conceptual relation. This approach, compared to the use of the edge labels as conceptual relations, offers more reliability in what regards to scalability, information retrieval performance and to model other types of graphs, e.g., UML [14].

According to the model depicted in Figure 5.6, anytime a CG is created, two vertex are defined: concept vertex and relation vertex. During the edition of a CG, the creation of a new concept implies to link that node to *concept vertex* node through an edge labeled as "is-a". Similarly, when a new conceptual relation is added, it should be connected to *relation vertex* node by an edge whose name is "is-a".

5.2.2 Specifying Competency Questions

Competency questions (CQs) are created during the configuration of a conceptualisation project. The CQs define a set of claims that the resulting conceptual representations should address. For that it was designed a four-step *wizard* domain experts (DE) are invited to create their claims. Firstly domain experts specifies the claim type according to the CQ types defined in section 4.5.

[13] https://github.com/tinkerpop/gremlin/wiki
[14] Unified Modeling Language

Next the DE choose which is the suitable pattern for the claim. To support the choice, DEs have a set of simplified tips, whith the following information:

- an example of a claim;
- a short description of the pattern;
- the pattern structure and its main building blocks.

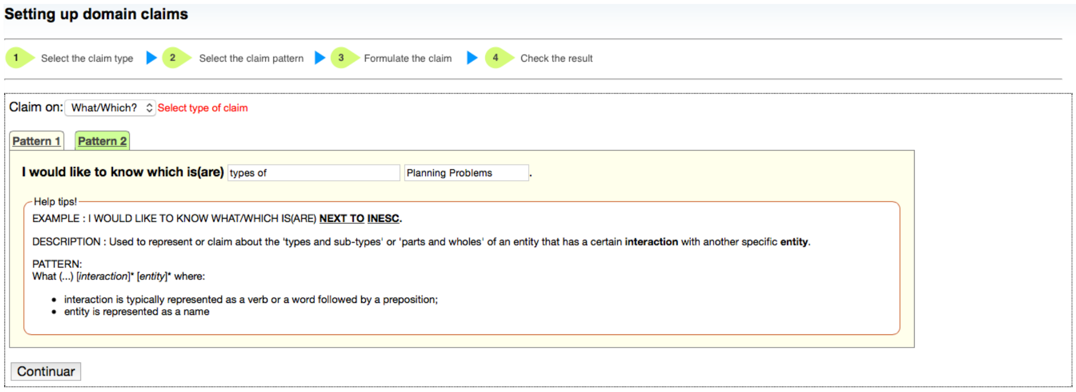


Figure 5.7: CQ prototype: query (claim) specification

In a third action step, DEs formulate the claim providing the required mandatory data. Afterwards the formulated claim (i.e., a semi-formal conceptual structure) is transformed in a semi-formal representation (see Figure 5.8). At this point the DE might validate or reject the proposed representation. In case of acceptance the claim is associated to the conceptualisation process.

When all the parties involved in the conceptualisation process share their proposals, the DE add information about which of the existing claims are addressed in the proposal. This information is associated to the model to be published (or shared), and it might be used to enhance information retrieval tasks, to define priorities during the calculation of semantic similarity between conceptual representations (e.g., start to find similarities in the models addressing the same claims) or even to use the unaddressed claims to query the domain corpus or other external repositories in oder to complete the common conceptual representation.

The conceptME platform brings together in one place a wide range of features, designed for the domain experts as the main modelers of their own knowledge and the technical knowledge that shapes their domain. Although tools and methodologies for the collaborative development of semantic artefacts (e.g., ontologies) have been emerging in the last years , there is no evidente of such a comprehensive tool. Next is a list of the most similar tools with brief description and evaluation:

- OntoShare (Davies et al., 2004) is based on On-To-Knowledge methodology and allows users to collaborate in a virtual community, contributing for the growth of a shared ontology. However it does not offer graphical modeling support nor discussion features. Moreover, it is not focused on domain experts but in knowledge engineers.

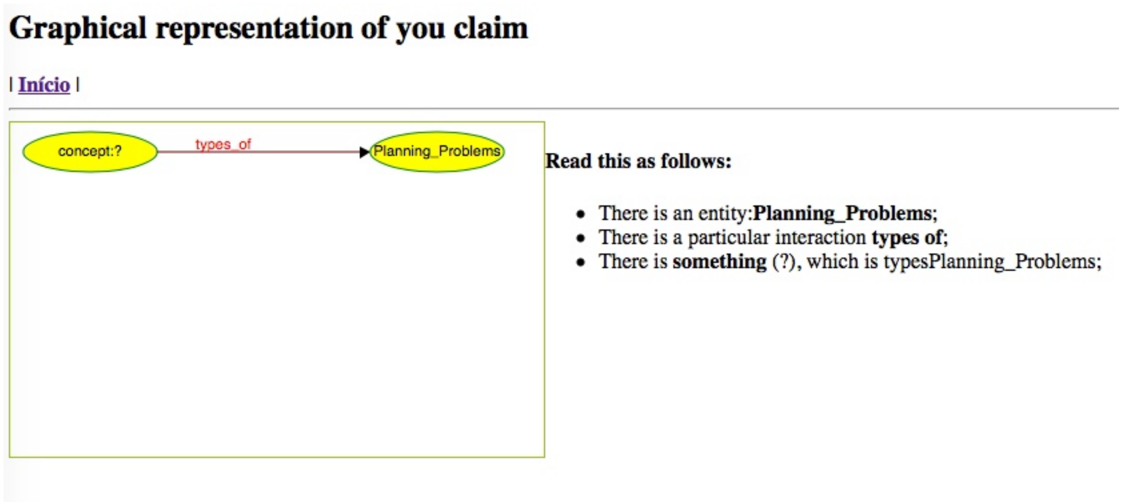


Figure 5.8: CQ prototype: semi-formal representation of a query (claim)

- NeOn Toolkit[15] provides an extensive set of plug-ins (some very complex) covering a variety of ontology engineering activities, including annotation, documentation, and interaction among other ontology editing features. It is so huge that it is necessary to be a NeOn expert to use the tool.

- OntoWiki[16] is semantic wiki more oriented for ontology management and creation of semantic knowledge bases. Definitely is a tool for ontologist and not for regular users (i.e., domain experts) who may not have any knowledge in formal notations such as OWL[17].

- MokiWiki (Ghidini et al., 2009) is another Semantic Wiki platform focusing in enterprise modeling (specifically ontologies and business processes). It is based on Semantic Media Wiki, like conceptME, and supports the collaboration in the development of enterprise models from an informal to a formal state. In fact Mokiwiki is an interesting tool, but it has a less versatile visual modeling environment, it lacks terminological extraction features and the reusability mechanisms are absent.

- CmapTools COE[18] is presented under the slogan "*ontologies for the rest of us...*". It is a graphical modeling environment based on concept maps (as conceptME). It follows a graphical template-based approach for the creation of formal ontologies. However some fundamental knowledge about OWL is needed to understand the meaning of each template. Moreover it has no terminological features and the collaborative environment encloses some limitations due to the platform architecture. Though useful, the focus questions functionality offered by Cmaptools COE is not as evolved as conceptME competency questions. Indeed, competency questions is one of the most valuable assets of ConceptME

[15] http://neon-toolkit.org/wiki/Main_Page
[16] http://ontowiki.eu/Welcome
[17] Ontology Web Language
[18] http://www.ihmc.us/groups/coe/

Grounded on a multidisciplinary framework, conceptME offers tools and resources allowing its usage in different domains and for different purposes (e.g., Building classification schemas for information management; Rich enterprise each systems; Software engineering; New product development; . . .).

The main characteristics that distinguish ConceptME from other approaches mentioned in section are: the use of integrated graphical representations for visualization of conceptual models, the paradigm of continued support and collaborative development of shared conceptual models, conception and association of semantic information and the automated support for the formalization, integration and interoperability of conceptual models. Another important differentiating factor lies in the generic nature of the platform, in which conceptual models can be modeled independently of its domain and on the other hand, the possibility to customize the platform for specific conceptualization processes (Sá et al., 2011).

Part III

Evaluation and Case Studies

Chapter 6

Research evaluation

"Not everything that can be counted counts,
and not everything that counts can be counted"

Albert Einstein

6.1 Research evaluation approach

According to the principles of design science research (the research methodology followed in this research) the evaluation of the produced artefacts (whether methods and tools) is focused on the examination of its usefulness in a given context situation, for which a specific evaluation method is adopted. Generically, it seeks to evaluate the validity and reliability of the artefacts. The validity addresses the usefulness of the artefacts, whereas the reliability addresses the reproducibility The main motivation driving the current research work is grounded on the challenge of developing an holistic approach to promote the involvement of domain experts in the production of computer-ready, useful and reusable conceptual representations of a particular domain. Therefore, the evaluation focus is to investigate to which extend the provided semantic processes contribute to obtain high quality representational artefacts, i.e., conceptual representations endowed of the characteristics mentioned above (computer-ready, useful and reusable). This means the evaluation has a double purpose:

1. to validate de conceptual representations, and;
2. to verify the conceptualisation framework.

The validation approach is focused on the produced artefacts (conceptual representations) whereas verification is concerned with the process. For the validation it was adopted a hybrid approach combining quantitative and qualitative approaches. The verification was qualitative based on action-research methods.

6.2 Validation approach of the conceptual representations

Utility and reusability criteria (described within the next sections) are the major building blocks to measure conceptual representations artefacts. Both utility and reusability requires that conceptual representations can be described under some computer readable formalism such as XML-based languages. We could say that formalisation is a pre-condition to measure reusability and also to determine utility.

6.2.1 Utility Criterion

The utility criterion comprises: (i) qualitative indicators, which arise from an empirical evaluation about the usage of the resulting conceptual representations, and (ii) quantitative indicators, related to the answers of the formulated competency questions.

Assuming the existence of a set of competency questions, the ability to quantify the number of questions to which a conceptual representation can provide an answer, is an indicator that allows evaluating the utility of the result of the domain conceptualisation. Therefore, let's consider the following example of a semi-formal representation of a competency question (figure 6.1), formulated bearing in mind the domain of the H-KNOW European project.

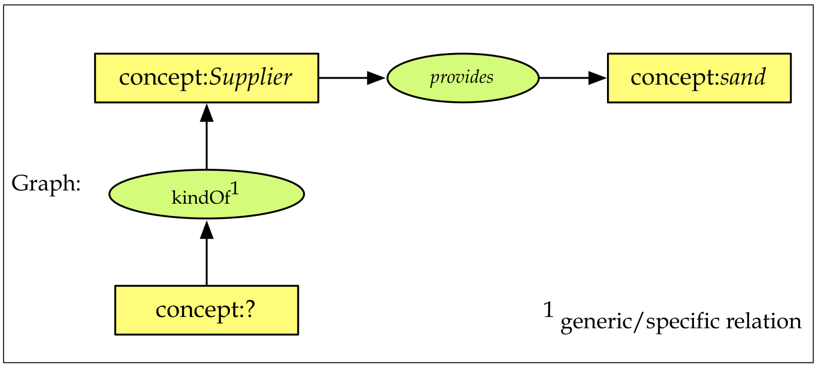


Figure 6.1: Competency Question example

The above representation (figure 6.1) might be formally represented in a query language such as SPARQL[1].

```
CONSTRUCT { ?x crrm:isTypeOf crrm:Supplier }
WHERE {
  ?x crrm:isTypeOf crrm:Supplier.
  ?y rdfs:label "provides"@en.
  ?y crrm:hasSource crrm:Supplier.
  ?y crrm:hasTarget crrm:Sand.
}
```

Listing 6.1: SPARQL representation of a competency question

[1] SPARQL Protocol and RDF Query Language (http://www.w3.org/TR/rdf-sparql-protocol/)

Considering the existence of competency questions' patterns, this transformation is easy to accomplish, and the mapping process capable of being automated. Thus, once the model is complete, SPARQL queries (corresponding to the formulated competency questions) may be triggered. Therefore, the utility degree will depend on the number of queries returning a valid result.

Assuming that the developed conceptual representation was previously populated into CRMM ontology in Protégé Ontology Editor, the result data of the SPARQL query are RDF triples, where “Subject_A” and "Subject_B" represent the concepts, whose definitions are according to the SPARQL restrictions, i.e., both “Supplier_A” and “Supplier_B” are types of suppliers, which are able to provide Sand.

Subject	**Perdicate**	**Object**
Supplier_A	isTypeOf	Supplier
Supplier_B	isTypeOf	Supplier
(...)	isTypeOf	Supplier

Table 6.1: Example of the SPARQL result

Accordingly, the qualitative indicator for utility is given by the ratio between the number of queries (obtained from the CQ), which returned a **valid** result (nQ_a), and the total number of queries (nQ). So, the utility degree is given by the following formula: $deg_{utility} = \frac{nQ_a}{nQ}$.

The utility criterion is better illustrated in Section 7.2.2.1, during the case study discussion.

Note that a **valid** query result is a a non-null or non-empty result whose retrieved content is as the domain expert expected. This aspect encloses que qualitative perspective of the utility criterion.

6.2.2 Reusability criterion

Reusability criterion measures the extent to which a common conceptual representation might be reused. A Conceptual Representation is prone to reusability if it is well-formed (computational ready) and its terms and conceptual relations are defined unambiguously, that is, there is a common understanding about the relations among terms. Considering a formalised conceptual representation, the reusability degree is expressed as a cumulative measure combining two components, wherein one provides a value (degree) for (i) Conceptual Relations ambiguity (deg_{CRa}) and one for (ii) Concept Type incompatibility (deg_{CTi}). Both (deg_{CRa}) and (deg_{CTi}) are determined with the support of the CRRM ontology, which implements a catalog of conceptual relations. At a glance, the process runs as follows:

1. The CRRM ontology is populated with the data from an existing conceptual representation (or Lightweight ontology) developed by the domain experts;

2. A set of (SWRL) rules is executed and a small set of indicators are gathered;

3. The indicators allow calculating $\deg_{\mathrm{CRa}}$ and $\deg_{\mathrm{CTi}}$ for a particular conceptual representation.

6.2.2.1 Conceptual relations ambiguity ($\deg_{\mathrm{CRa}}$)

For Conceptual Relations ambiguity ($\deg_{\mathrm{CRa}}$) let us consider a counting function μ, where:

- $\mu(\mathrm{R_t})$ = Number of existing relations in the LO (Lightweight Ontology) developed by the domain experts;
- $\mu(\mathrm{R_a})$ = Number of ambiguous relations inferred (according to the SWRL rules);
- $\mu(\mathrm{R}_\theta)$ = Number of uncategorized relations;

The values of $\mu(\mathrm{R_a})$ and $\mu(\mathrm{R}_\theta)$ are then normalized, by scaling between 0 and 1, obtaining:

- $\mathrm{N}\mu(\mathrm{R_a})$, the normalized value of $\mu(\mathrm{R_a})$, calculated as follows: $\mu(R_a)/[\mu(R_t)-\mu(R_\theta)]$
- $\mathrm{N}\mu(\mathrm{R}_\theta)$, the normalized value of $\mu(\mathrm{R}_\theta)$, calculated as follows: $\mu(R_\theta)/\mu(R_t)$

After normalizing these values, the conditions are met for calculating ($\deg_{\mathrm{CRa}}$), which is obtained through the expression:

$$deg_{\mathrm{CRa}} = \begin{cases} 1 & if\,\mu(R_\theta)=\mu(R_t) \\ N\mu(R_a)\times\omega_a + N\mu(R_\theta)\times\omega_\theta & if\,\mu(R_\theta)<\mu(R_t), where: \end{cases}$$

- ω_a and ω_θ, are weights reflecting the importance of R_a and R_θ respectively;

The Knowledge engineer might adjust the weight values, other wise it is used the default weight scale tables, however the proper identification of ambiguous relations, requires its classification through the CRRM ontology. Thereby, if $\mu(R_\theta)$ has a direct implication on $\mu(R_a)$, then the value of ω_θ should always be higher than ω_a. Figure 6.2, shows an indicative scale of how ω_θ can evolve according to the variation of R_θ percentage, having in mind that the higher the percentage of R_θ, the greater its weight (ω_θ) should be.

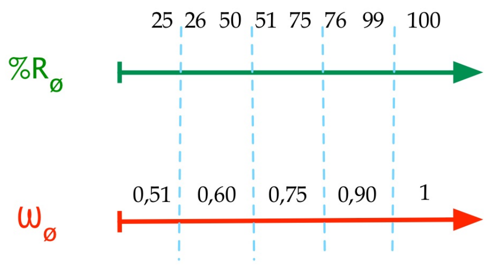


Figure 6.2: Weight scale of uncategorized relations

In short, the main restrictions for ω_θ and ω_a are:

- $0,51 \leq \omega_\theta \leq 1 \wedge 0 \leq \omega_a \leq 0,49 \wedge \omega_\theta + \omega_a = 1$

6.2.2.2 Conceptual types incompatibility ($\mathbf{deg_{CTi}}$)

For Concept Type incompatibility (deg_{CTi}), let us consider also a counting function μ, where:

- $\mu(T)$ = Number of existing terms in the LO (Lightweight Ontology) developed by the domain experts;
- $\mu(T_\theta)$ = Number of uncategorized terms (by inference);
- $\mu(T_i)$ = Number of incompatible related terms;

The values of $\mu(T_i)$ are then normalized, by scaling between 0 and 1, obtaining:

- $N\mu(T_i)$, the normalized value of $\mu(T_i)$, calculated as follows: $\mu(T_i)/[\mu(T) - \mu(T_\theta)]$

After normalization, the value for deg_{CTi} is equal to $N\mu(T_i)$, only if $\mu(T_\theta < \mu(T)$. Otherwise, deg_{CTi} is impossible to calculate.

$deg_{CTi} = N\mu(T_i), if \mu(T_\theta) < \mu(T)$

Finally, and as aforementioned, the reusability degree results on the combination of deg_{CRa} and deg_{CTi}, by multiplying each other, in order to keep the proportionality of both. However, the two components might not have the same importance. Actually, deg_{CTi} is less relevant, because the classification of each term depends on the classification of the relation that binds to it. Moreover, even if there are no incompatible concepts, it is possible to calculate the reusability degree. The opposite however: despite the proportionally of the cumulative measure (degR), it should not be zero, even if deg_{CTi} is zero. The final expression for calculating the reusability degree is:

$$degR = \begin{cases} 1 - deg_{CRa} & if \mu(T_\theta) = \mu(T) \\ 1 - (deg_{CRa} \times (\omega_a + \omega_t \times deg_{CTi})) & if \mu(T_\theta) < \mu(T), where \end{cases}$$

- ω_a and ω_t, are weights reflecting the importance of deg_{CRa} and deg_{CTi} respectively;

Typically, domain experts might adjust the weight values for ω_a and ω_t as well, but there are some boundaries. deg_{CTi}, is a secondary importance factor, its calculation depends on deg_{CRa}. Actually, the reusability degree is always given by deg_{CRa} in the irst place. Afterwards, deg_{CRa} is obtained in order to fine-tune the reusability degree. Based on this, the value for ω_t obey to the following assumptions:

- $0 \leq \omega_t \leq 0,5 \wedge \omega_a + \omega_t = 1$
- $\omega_t = 0, if \mu(T_\theta) = \mu(T)$

Considering the assumptions listed before, figure Figure 6.3 presents an indicative scale of how ω_t can evolve according to the variation of of T_ω percentage.

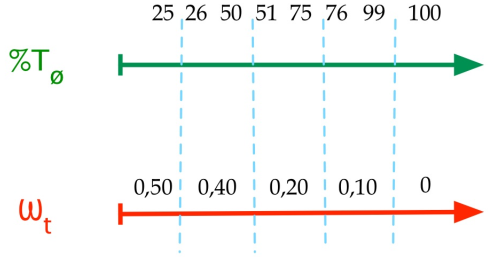


Figure 6.3: Weight scale of uncategorized term types

6.2.3 Illustrative example for measuring reusability

In order to explain and demonstrate the reusability evaluation criteria, a simplification of the SEON[2] general concepts ontology was used and represented as shown in figure 6.4. A correspondence was made between the SEON general concepts ontology and CRRM, where all relations matched to some CRRM category were evidenced in the figure bellow with the CRRM prefix.

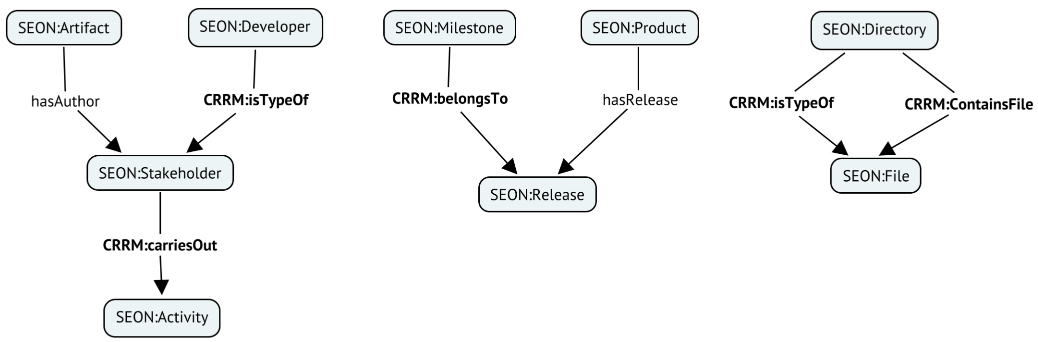


Figure 6.4: SEON ontology example

The above ontology was populated into CRRM (see figure 6.5). The pellet reasoner was started up and the metrics gathered as summarized in table 6.2.

Table 6.2: SEON ontology reusability metrics

$\mu(R_t)$	$\mu(R_\theta)$	$\mu(R_a)$	$\mu(T)$	$\mu(T_\theta)$	$\mu(T_i)$
7	2	1	9	2	0

From the above metrics and considering the indicative weight scale from Figure6.2, it was found that the value form Conceptual Relations ambiguity (deg_{CRa}) is 0,236. As for Concept Type incompatibility (deg_{CTi}), the value is 0. From deg_{CRa} and deg_{CTi} and considering the indicative

[2]SEON stands for Software Evolution ONtologies and representes an attempt to formally describe knowledge from domain of software evolution analysis and mining software repositories.

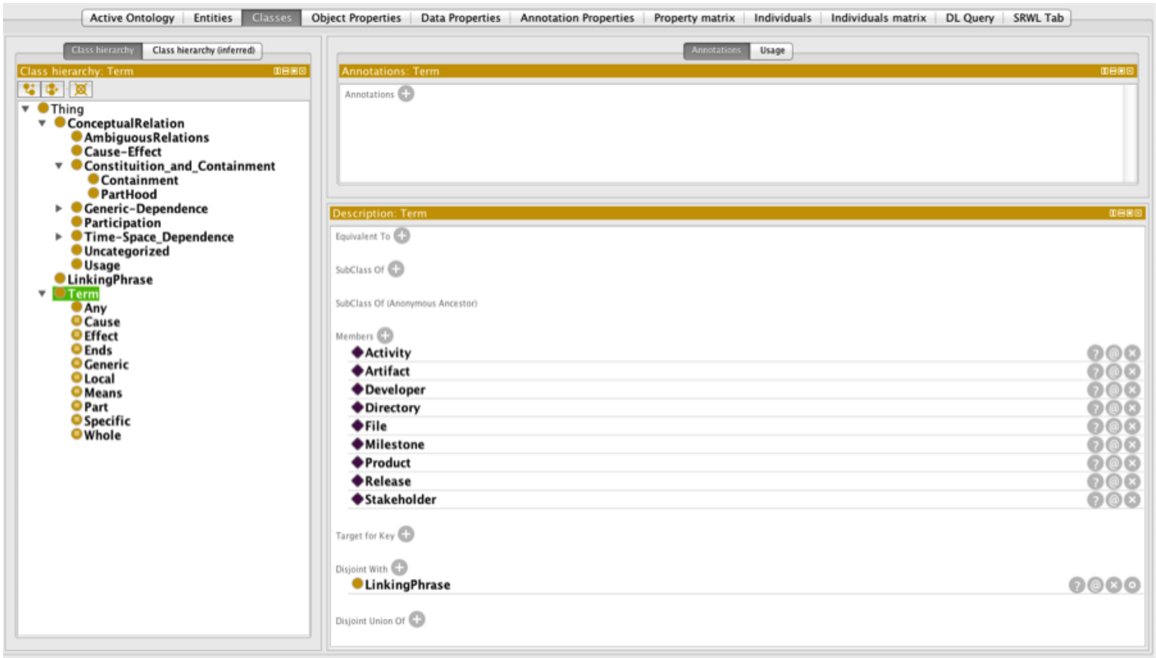


Figure 6.5: SEON ontology populated in CRRM

weight scale depicted in figure 6.3, reusability degree is obtained as follows: $deg_{CRa} = 0,236 \times (0,51 + 0,49 \times 0)) = 0,88$

A degree of $0,88$ means that 88% of the conceptual representation content is fully reusable. In this context reusability means that the conceptual representation content can be formalised and retrieved by other user, maintaining its intended meaning, since the employed relations belong to the CRRM catalog. The results indicate that 12% of the conceptual representation content is highly susceptible to misinterpretations when reused by others.

6.3 Approach to the verification of the conceptual representations

The verification approach relies on monitoring the conceptualisation process, investigating the suitability of the conceptualisation framework on supporting domain experts to establish domain-valid (i.e., useful within a particular community) conceptual commitments, through semi-formal knowledge representations. In this case, we are concerned in understanding the experience of a community during the development of the conceptual representations. According to the designed research approach, the focus is on studying how the means (i.e., the conceptualization framework) contributes to the ends (i.e., useful conceptual representations). For this, a qualitative evaluation was followed, based on the principles of Action Research (AR).

AR consists in a holistic approach to problem-solving where knowledge is learned working in a context of action and where people try to work together to address key problems in their organisations. Typically an AR based project involves more or less systematic cycles of action and reflection: in action phases co-researchers test practices and gather evidence; in reflection stages they make sense together and plan further actions (Reason and Bradbury, 2008). Often, research approaches in this category are also referred as participatory AR (Baskerville, 1999; Flick, 2002; Baskerville and Myers, 2004). What separates this type of research from general professional practices is the emphasis on the scientific study, which is to say the researcher studies the problem systematically and ensures the intervention is informed by theoretical considerations. Therefore, much of the researcher time is spent on refining the methodological tools to suit the exigencies of the situation, and on collecting, analysing, and presenting data on an ongoing, cyclical basis (Reason and Bradbury, 2008).

AR offers a systematic and self conscious research design, data collection, interpretation and communication, allowing to mitigate some of the problems associated with qualitative approaches, including: i) research bias restrictions; ii) reproducibility scarcity; iii) lack of generalisability. Moreover, the considered research cases were conducted with groups of people whose needs were aligned to the conceptualisation phenomena under study, rather than inviting groups of people to participate in a controlled experiment.

Chapter 7

Case Studies

"To acquire knowledge, one must study;
but to acquire wisdom, one must observe."

Marilyn Vos Savant

7.1 Case study I

7.1.1 Description

This case refers to a three years project on forest management decision support systems - FORSYS ($http://fp$0804.$emu.ee/wiki/index.php/Main_page$), and more precisely to its so-called knowledge management system. The goal of the project was to provide an aggregate view about the European experience with developing and applying decision support systems for forest management as a solid foundation for technological innovation and collaboration between research partners. The project was structured in 4 working groups, with 20 to 50 participants each from a wide range of countries. The case studied in this thesis concerns WG1, holding 25 participants from 15 countries, whose major goal was to "develop a procedural framework, information standards and guidelines for the development, testing and evaluation, as well as the application of decision support systems for forest management problems in multifunctional forestry". This involved "evaluating innovative decision support systems for their potential to improve the efficiency and effectiveness of forest management and promoting the use of these resources for enhanced decision-making". The project developed an information management system using a wiki platform and, in their discourse, assumed it as the project's knowledge management system. This system was developed to support the aforementioned goals and to support knowledge sharing within a community of practice (the project members and future users) dedicated to decision support systems for forest management. The system would provide a repository on the subject of decision support in forest management, continuously evolving over time and serving as a reference for future projects on decision support systems.

The first version of the wiki system provided an informational support (creation, organisation, storage) to the project operation and an informational structure to codify and organise the knowledge created within the project's community. This would cover the project duration as well as after-project activities. The idea was to codify, organise and provide an efficient access to the knowledge created around the concept of Decision Support Systems for forest management. It would include information about Decision Support Systems - the central concept - and information about related concepts such as: possible system structures, types of problems addressed, possible techniques and decision models. Furthermore, other important concepts for this community such as the lessons learned, social participation, and knowledge management would be included.

Regardless the proven usefulness of the wiki platform, the FORSYS community representatives (developers and final users) identified limitations to the effective achievement of the goals referred above. Two main problems affecting the efficacy of information retrieval were identified. The lack of a common conceptual model of the domain, together with a poor strategy to develop and maintain the content structure of the wiki system led to poor results - long wiki pages, lack of advanced search feature, no semantic entailment among content and ad-hoc hypertext model - as well as the inability to ensure the continuous growth and evolution of the content.

A more detailed conceptual model of the domain, shared by all the stakeholders, was needed. This model aimed to increase the user's capability for successfully undertaking project support tasks such as organising, searching and retrieving information. An action-research project to improve the platform's knowledge organisation structure, involving as much as possible the FORSYS community, was setup.

7.1.2 Method for the FORSYS KMS redesign

7.1.2.1 An action research approach

The redesign of the project's wiki system was taken as an opportunity to investigate the practices of knowledge organisation and representation of a specific scientific and technical community. Therefore, the dual objective of this research project was: (i) to improve the knowledge organisation of the FORSYS platform (problem solving) and (ii) to know more about the practices of knowledge organisation and representation in technical communities (creation of scientific knowledge). For this, the principles of Action Research were applied. Questionnaires and interviews were used to collect preliminary data while the main data collection technique was participatory observation (Reason and Bradbury, 2008). Content analysis was made by text interpretation (from interviews transcriptions, observation notes), without codification. The findings were collectively interpreted regarding the suitability of the results according to the users expectations. The validity of the results depends on (1) the degree to which the developed artefacts provide a common understanding about the context in which the research was conducted; and (2) to which extent it provides the essential rational not only about how semantic categories are related, but how the achieved models provide the gateways for FORSYS semantic wiki structure.

7.1.2.2 Overview and Characterization for the FORSYS knowledge management system redesign

An iterative and incremental model was adopted for the research method. It resulted in a process comprising 4 phases, each having its own specificities - number and type of roles, specific goals, allocated time frame - requiring some adjustments to the action-research approach (see figure 7.1):

1. phase one, consisting in the analysis and feasibility study, from which an initial conceptualisation was obtained;

2. phase two, entailing the collaborative development of the model for the knowledge organisation systems, based on the new conceptualisation; it uses the developed initial conceptualisation as input and it was carried out following a participatory action-research;

3. phase three, including the implementation of a new model for knowledge organisation systems, and;

4. phase four, enclosing the design and implementation of a semi-automated linking mechanism between conceptME and FORSYS's knowledge management system.

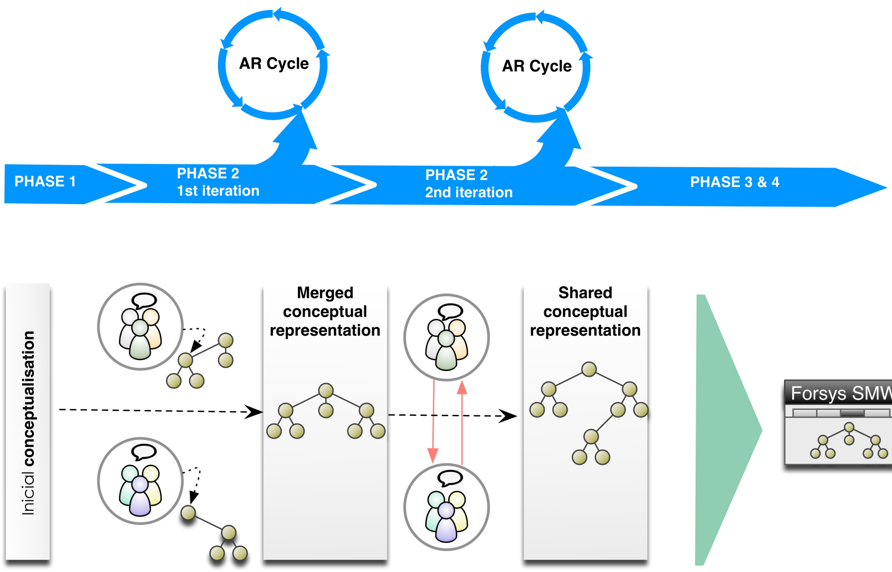


Figure 7.1: Research method overview

Five project members were involved in the redesign of the project wiki, becoming the main data source for the research part. Data was collected during the two phases by means of interviews, observation and focus groups, as will be detailed bellow.

7.1.2.3 The research process

In practical terms the process started with an initial iteration wherein a group composed by two FORSYS representatives and two knowledge management specialists paved the way for the rest of the process and produced a first conceptual representation of the Decision Support System domain area. A proposal on how to address the improvement of the conceptual models for the knowledge organisation system was outlined. The process continued to the second phase with two iterations, which occurred in different action-research cycles. The first iteration of the second phase involved two independent FORSYS expert groups debating the two other main domain areas of the FORSYS project: Lessons Learned and Case Study. It resulted in two models that were the inputs for creating a merged, agreed model in the second iteration. At the end of this second phase, we were able to know more about the implicit conceptual relations structures used by the domain experts.

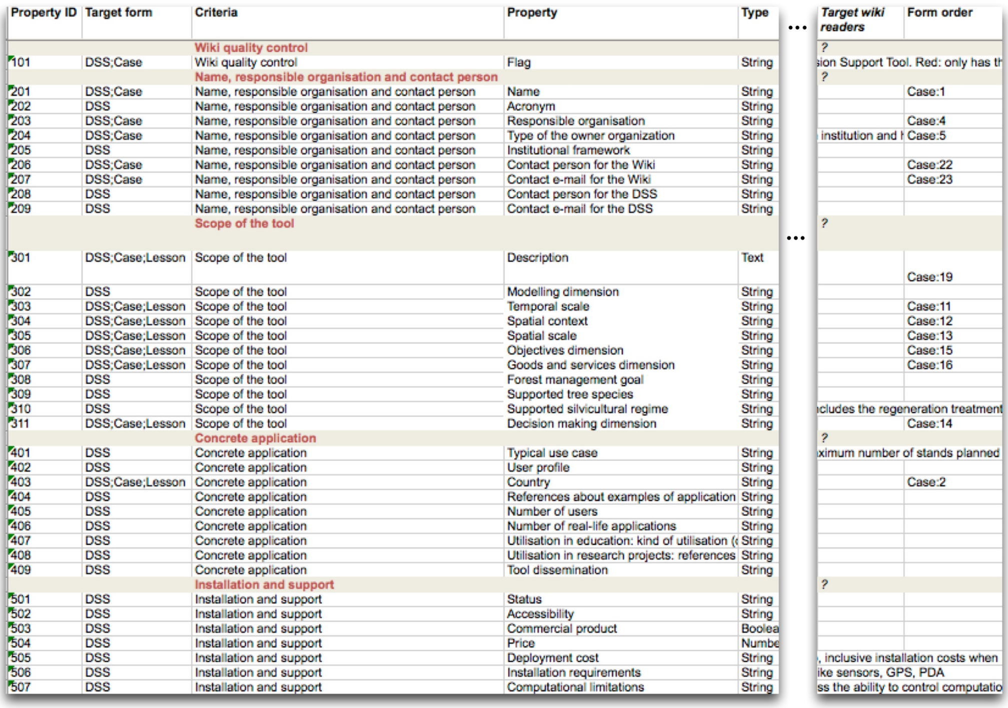
The second iteration included the negotiation and agreement regarding the conceptual models developed by the several groups in the first iteration.

The main result was a complete conceptual model of the FORSYS domain, agreed by the stakeholders and improved for implementing the knowledge organisation system of the wiki platform. At the end of this iteration (phase 2 - 2^{nd} iteration), a proposal to conduct the elicitation process of conceptual relations in conceptual modelling was completed. The goals and expected results of each phase are systematized in table 7.5 in section 7.1.4 with the conslusions.

Phase 1 - The development of the initial conceptualisation. Phase 1 comprised a conceptual analysis of the FORSYS categories and property model (which were available in a spreadsheet document) and the "as-is" wiki implementation. Moreover, an experiment was conducted in order to develop a "proof of concept" for the new FORSYS knowledge organisation. This experiment was performed by a group of domain experts from five different partner institutions of the FORSYS project. During the experiment the candidate took the role of observer, but providing some technical support when needed. This group of experts, performed the conceptualisation activities in the conceptME tool, giving focus to the definition of the wiki structure, accounting for all the FORSYS working domain concepts (but focused on decision support system), reviewing the wiki content and contributing to the identification of relevant queries. The result was a first version of a shared model (the initial conceptualisation).

Once the approach has been validated, a new group of domain experts came into play to collaboratively develop a new wiki metamodel (the model for the new knowledge organisation system).

Phase 2 - Reconceptualisation. Phase 2 entailed the development of a common view on the FORSYS conceptual model, with particular focus on the following concepts: decision support system, lessons learnt and case study. This phase underwent some changes regarding the approach followed in the previous phase, namely:



Property ID	Target form	Criteria	Property	Type	…	*Target wiki readers*	Form order
		Wiki quality control				?	
101	DSS;Case	Wiki quality control	Flag	String		ion Support Tool. Red: only has th	
		Name, responsible organisation and contact person				?	
201	DSS;Case	Name, responsible organisation and contact person	Name	String			Case:1
202	DSS	Name, responsible organisation and contact person	Acronym	String			
203	DSS;Case	Name, responsible organisation and contact person	Responsible organisation	String			Case:4
204	DSS;Case	Name, responsible organisation and contact person	Type of the owner organization	String		institution and I	Case:5
205	DSS	Name, responsible organisation and contact person	Institutional framework	String			
206	DSS;Case	Name, responsible organisation and contact person	Contact person for the Wiki	String			Case:22
207	DSS;Case	Name, responsible organisation and contact person	Contact e-mail for the Wiki	String			Case:23
208	DSS	Name, responsible organisation and contact person	Contact person for the DSS	String			
209	DSS	Name, responsible organisation and contact person	Contact e-mail for the DSS	String			
		Scope of the tool			…	?	
301	DSS;Case;Lesson	Scope of the tool	Description	Text			Case:19
302	DSS	Scope of the tool	Modelling dimension	String			
303	DSS;Case;Lesson	Scope of the tool	Temporal scale	String			Case:11
304	DSS;Case;Lesson	Scope of the tool	Spatial context	String			Case:12
305	DSS;Case;Lesson	Scope of the tool	Spatial scale	String			Case:13
306	DSS;Case;Lesson	Scope of the tool	Objectives dimension	String			Case:15
307	DSS;Case;Lesson	Scope of the tool	Goods and services dimension	String			Case:16
308	DSS	Scope of the tool	Forest management goal	String			
309	DSS	Scope of the tool	Supported tree species	String			
310	DSS	Scope of the tool	Supported silvicultural regime	String		icludes the regeneration treatment	
311	DSS;Case;Lesson	Scope of the tool	Decision making dimension	String			Case:14
		Concrete application				?	
401	DSS	Concrete application	Typical use case	String		iximum number of stands planned	
402	DSS	Concrete application	User profile	String			
403	DSS;Case;Lesson	Concrete application	Country	String			Case:2
404	DSS	Concrete application	References about examples of application	String			
405	DSS	Concrete application	Number of users	String			
406	DSS	Concrete application	Number of real-life applications	String			
407	DSS	Concrete application	Utilisation in education: kind of utilisation (	String			
408	DSS	Concrete application	Utilisation in research projects: references	String			
409	DSS	Concrete application	Tool dissemination	String			
		Installation and support				?	
501	DSS	Installation and support	Status	String			
502	DSS	Installation and support	Accessibility	String			
503	DSS	Installation and support	Commercial product	Boolea			
504	DSS	Installation and support	Price	Numbe			
505	DSS	Installation and support	Deployment cost	String		, inclusive installation costs when	
506	DSS	Installation and support	Installation requirements	String		ike sensors, GPS, PDA	
507	DSS	Installation and support	Computational limitations	String		ss the ability to control computatio	

Figure 7.2: FORSYS property list

1. the definition of new working groups;

2. the change of the research view (myself) role from simply "observer" to "participant" (following now a participatory action-research);

3. a major effort was made on framing the problem; and

4. some knowledge organisation guidance was provided beyond technical support, comprising:

    - distribution of a set of common best-practices for creating good concept maps;

    - a catalogue of conceptual relations, their common meaning and typical use cases (to assist users on the specification of the appropriate relations among concepts).

Figure 7.3 depicts the adopted cycle model of action research. According to the action-research cycle portrayed in figure 7.3, preliminary activities comprising information gathering and the problem definition were performed before the action-research cycle began. Besides the study of the domain (using the project documentation), a questionnaire was designed and it was aiming at profiling the domain experts in their basic skills and goals regarding knowledge management and representation practices. From the analysis of the answers we became aware that, unsurprisingly, the participants were comfortable with the forest management DSS concept, but not so much with the current common shared view. Additionally, we verified that all participants had information and knowledge management concerns in their daily activities and they revealed their preferences for graphical knowledge representations, despite of being aware of the difficulties of building such representations.

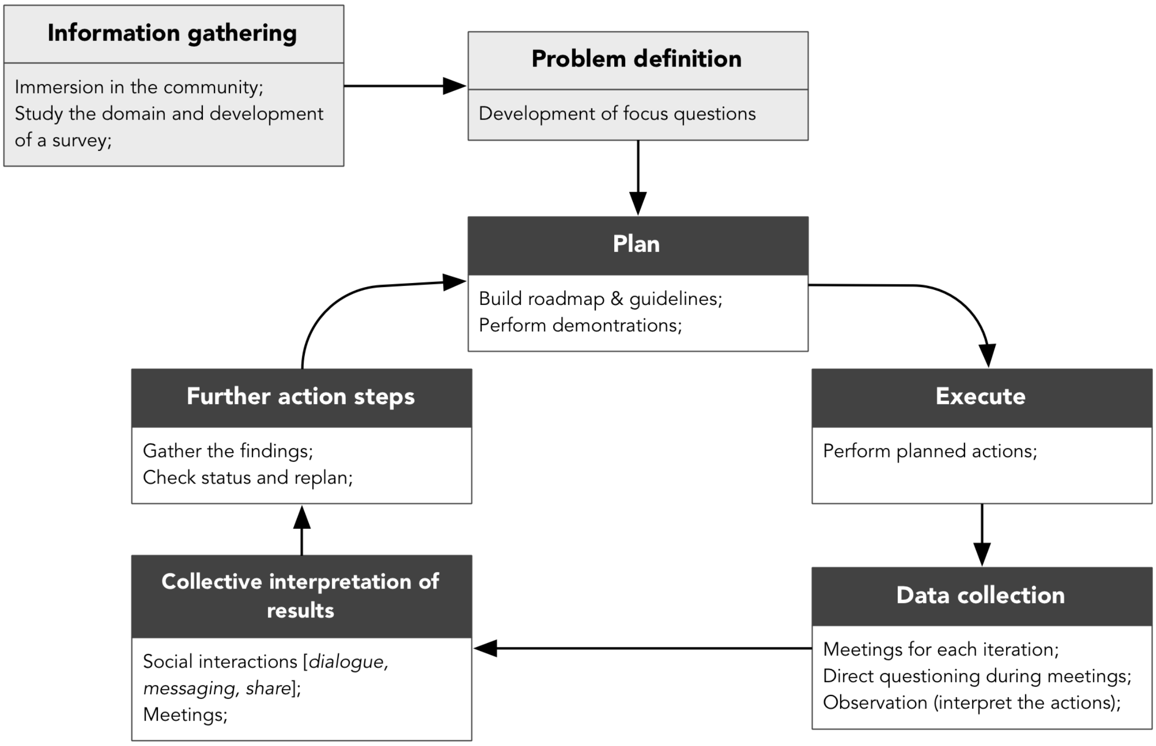


Figure 7.3: Adopted participatory action-research cycle

The boundaries of the domain being conceptualised were specified by formulating a set of focus questions (as shown above), to which the models should provide answers. The goal was, on one hand, to frame de conceptualisation project and, on the other hand, to verify whether or not the models answered all the posed questions, that is, if the models were or not "complete".

FQ1 what kind of lessons learnt exist? What caused/produced them?

FQ2 what kind of problem a lesson helps to solve?

FQ3 how should the lessons learnt be applied?

FQ4 what phase does it support? Which role(s) should participate?

FQ5 which methods/models could be used?

After the conclusion of this two initial activities the plan was outlined and a kick-off meeting was scheduled. The objectives of the meeting were to: a) validate the roadmap; b) distribute the guidelines for this study; c) share the goals; d) provide specific information about conceptME platform (running according to the conceptualization framework) and perform a short demonstration; and e) the group formation. Afterwards, the process continued with two groups of domain experts (GE1 and GE2) with elements from 4 different partner institutions with different backgrounds, but all of them specialized in the forestry management domain. The area of expertise of the group members included: i) decision support systems for forest and natural resource management; ii) forest process optimisation and; iii) geographical information systems. A third group of non-domain experts (GF1 - the researchers) participated with the role of process facilitators and as observers, collecting and registering all taken actions. GF1 domain of expertise is knowledge engineering.

The planned actions were developed based on conceptualization framework (Sousa et al., 2012) and inspired on the principles and assumptions of the ColBlend method (Pereira et al., 2013) . Basically, this method proposes three steps: (i) the development of the individual proposals, (ii) publishing of the proposals and (iii) the discussion/negotiation/merging of the proposals. Each step of the ColBlend method was performed in different iterations after which, a meeting took place and the results were discussed based on the data collected, both from interviews and direct observations, as well as data gathered from the conceptME platform. At this meeting the next steps were also discussed with all participants and the "initial roadmap" reviewed according to the progress of the conceptualisation process and a next iteration took place.

Phase 3 and 4 - New content organisation model and conceptME for FORSYS KMS semi-automated linking. The last two phases (3 and 4) sought to find how conceptME and FORSYS platforms could be used in a complementary way. A mapping between the conceptual models developed in conceptME and the Semantic MediaWiki instance was defined. The goal was to determine to what extent conceptME could serve as a modelling front-end to specify the meta-model of the FORSYS wiki platform, replacing the spreadsheet currently specifying the FORSYS properties model. Figure 7.4 depicts the metamodel, which matches the structure of concepts developed in conceptME with the content structure (metamodel) of semantic mediawiki.

The conceptME platform runs itself on top of an instance of semantic mediawiki. This means that every content (such as a specific instance of concept, relation or property) added to the platform is a wiki page belonging to a specific category, namely a concept, a relation or a property. The mapping between the developed models in conceptME and the semantic mediawiki follow a simple set of basic rules:

r1 a concept within a conceptME model (represented as Concept "A" in the figure 7.4 above), corresponds to a specific category in semantic mediawiki;

r2 a relation connecting two concepts within a conceptME model (represented as Relation "R" in the figure 7.4 above), corresponds to a specific property in semantic mediawiki;

r2.1 a property (in semantic mediawiki) resulting from a relation between two concepts (from conceptME) must be of the Page type and its value should be a wiki page;

r3 a concept within a conceptME property model (represented as Property "P" in the figure 7.4 above), corresponds to a specific property in semantic mediawiki;

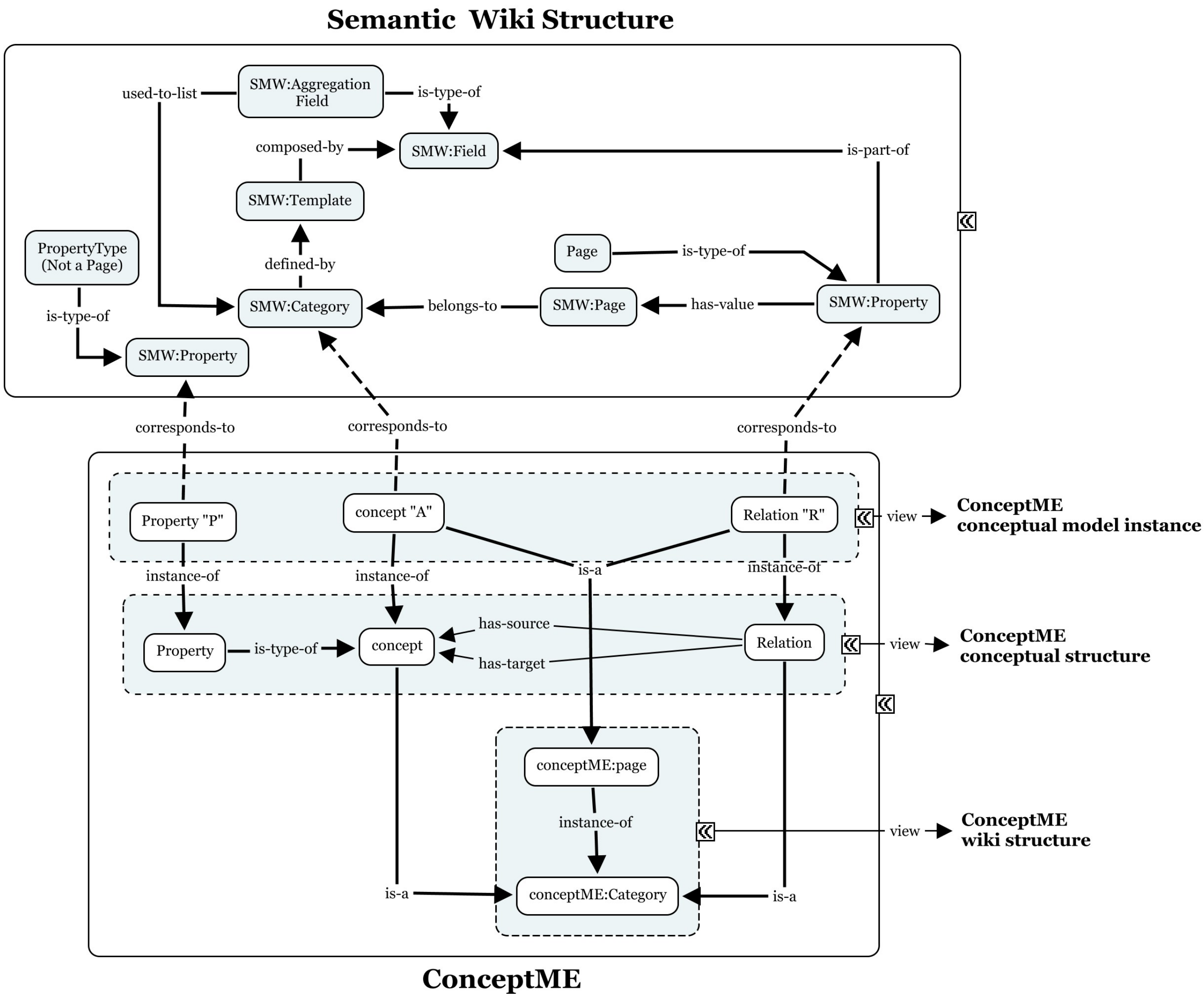


Figure 7.4: conceptME to semantic mediawiki metamodel

r3.1 a property (in semantic mediawiki) resulting from a concept in a property model (from conceptME) can be of any type except Page;

r4 a template to define new categories in semantic mediawiki should be defined to cope with this rules and its member fields types should correspond to the properties translated from conceptME model.

Additionally, and in order to maintain conceptME and some particular instance of semantic mediawiki tied together in a kind of topic map approach, it was necessary to consider the following additional rules in the transformation procedure described above:
i) for each concept in the final conceptual representation in conceptME, a default query is built on the SMW instance in the form of **{{#Ask: [[Category: concept]]}}**. For this, it is assumed that a specific conceptME semantic wiki property exists, and that it stores the Unified Resource Identifier (URI) of the recently built semantic mediawiki query page.

7.1.3 Results and discussion

Given the action-research approach followed here, due to the intertwining between results and their analysis, it makes more sense to join the description of the results and the discussion into a single section, rather than having them separate, as usual. Thus, the results are analysed according to the process followed and the models developed. Given the action-research approach, the discussion is based on the data collected by the researchers through participatory observation and joint reflection with the domain experts (acting as co-researchers).

The process. As described above, the process designed to improve the knowledge organisation structure of the FORSYS platform (the problem solving part of the action-research) followed the conceptualization framework, which is supported by the conceptME modelling environment. Two groups of specialists addressed the same focus questions and independently developed a concept map for them. Then, the groups debated the similarities and differences of the respective outcomes and, supported by the model merging facilities of conceptME, negotiated a common model (concept map).

The process and the underlying framework (the conceptualization framework) contributed to achieve conceptual representations that were actually useful for domain experts in real applications. The resulting conceptual representations were effectively used for structuring information for a community of practice (CoP) information system and to define a set of basic queries to interact with the system. The CoP web application is available at *http://www.forestdss.org/CoP/*

From our observation, and from the reflection made with the experts, it was evident the existence of an initial difficulty in understanding entirely the concept maps produced by the other group due to the lack of textual definitions for the concepts. This was overcome by the debate and discussion between the two groups, albeit at the expense of more time to reach an agreement (Stock, 2010). In fact, an extra effort to sustain and argue the conceptual structures was observed.

The definition of focus questions was very useful as a pragmatic way to define the domain boundaries as well as to assist in the validation of the completeness of the developed concept maps. The early definition of different roles (ensuring the involvement of domain experts beyond knowledge engineers) and the composition of the teams developing the models revealed to be important both for the process and the quality of the results. The expert role was fulfilled with people with different backgrounds, differences that contributed positively for the debate of the different points of view when merging the concept maps. The facilitator role also revealed a major importance. As in every social process involving collaboration, facilitation and leadership are fundamental for the efficiency and effectiveness of the results. From the observations made during the project, we conclude that it was beneficial that the facilitation encompassed both the knowledge representation and the FORSYS domains.

The models. The following aspects of the resulting conceptual models were analysed:

1. the set of elicited concepts,
2. the differences between those and the set of concepts (categories and properties) extracted from the existing metamodel,
3. the differences between the elicited concept sets from the two groups,
4. the set of elicited relationships,
5. the meaning of the basic conceptual structures (a concept and directly related concepts).

We will give particular attention to 4) the set of elicited relationships. The elicitation of conceptual relations is recognized as one of the most difficult problems in a conceptualisation process (Auger and Barriere, 2010; Elsayed, 2009). While the elicitation of concepts is close to the basic cognitive process of categorisation (Rosch and Lloyd, 1978), the same is not true for the relationships. In this case, additional ontological knowledge is needed.

In this project particular case, the conceptual relations elicitation was of great importance because of the need to transform part of the relatively big sets of concept properties into conceptual structures. As explained before, this would lead to a better organised, less monolithic content organisation in the wiki platform. A great part of the conceptual structures represented in the models were not precise enough due to the use of the "has" relationship. In fact, this relationship had already been amply used in the starting conceptual models (developed during the first phase). Although at the beginning of the second phase (the reconceptualisation phase) the participants have been invited to use a catalogue of predefined types of relations provided by conceptME, and despite of reporting its usefulness, in most cases the groups tended to accept the conceptual structures proposed during the first phase as final and stable. That means, the conceptual relations catalogue was mostly used to build new conceptual structures during the reconceptualisation phase.

Observing in detail those structures (i.e., the structures containing the "has" conceptual relation label), the intention of their creators seem to have been to represent:

1. a chain of entities and their member fields;
2. a compound structure, in which an entity is composed or made up of other entities;
3. a containment structure, in which an entity belongs to another entity.

These results were expected as Shams et al. (2010) concludes that the "has" relation is a too generic conceptual relation and most of the times it is used to represent a physical relation between entities. In the same line, Nuopponen (2005) concludes that "has" is generally used to mean some kind of contiguity engagement.

The confirmation of such assertion came form the domain experts themselves by completing a questionnaire. For the questionnaire preparation it was collected (form the conceptualisation result) a set of 42 conceptual structures [1], each of which containing a conceptual relation labelled as "has". For each conceptual structure, domain experts should provide its intended meaning by selecting one of the following options:

a) membership relation

b) compound relation

c) containment relation

d) "generic-specific" relation

e) other

Together with each of the options depicted above, a short explanation was provided. Furthermore, the questionnaire also included some conceptual structures that employed conceptual relations from CRRM ontology. Again, domain experts were invited to confirm the intended meaning of those conceptual structures.

The questionnaire was sent to the five groups that participated in the reconceptualisation process (the 2^{nd}). Four of the five groups replied. The answers were collected and then analysed. It was verified that, despite the domain experts having agreed upon the produced conceptual representation, in fact we discovered that there was no consensus around the intended meaning of the collected conceptual structures (c.f. appendix C). The exception were those conceptual structures composed with CRRM conceptual relations.

As an example, let us consider the following conceptual structures in the final conceptual representation of the the 2*nd* phase:

CS-1: Decision support technique **has** MCDM method.

[1] "Concept-Relation-Concept" structures

CS-2: Decision support technique **has** Optimisation algorithm.

CS-10: DSS **has** Decision Support Technique.

CS-21: Forest management problems type **has** FORSYS problem dimension.

CS-24: Lesson Learned **has** Case Study.

CS-46: Knowledge source **isSupportedBy** Developer.

CS-47: Knowledge Map **is-a** Knowledge management technique.

CS-48: Case Study **isTypeOf** Source.

The answers provided by domain experts (considering the aforementioned conceptual structures) indicated that:

a) CS-1 had 3 different interpretations.

b) CS-2 had 3 different interpretations.

c) CS-10 had 2 different interpretations.

d) CS-21 had 2 different interpretations.

e) CS-24 had 2 different interpretations.

f) CS-46 was consensual.

g) CS-47 was consensual.

h) CS-48 was consensual.

Accordingly, for the first five conceptual structures (CS), there was no agreement on the underlying intended meaning, unlike the last three ones wherein it was verified a complete engagement to its interpretation.

If, on one hand, domain experts were comfortable enough with the conceptualization result (a necessary condition for its practical applicability), on the other hand, the ambiguity of some employed conceptual relations raised reusability issues to the achieved conceptual representations, eventually, to some other community of experts. So, after the last iteration of the the second phase of the conceptualisation process, wherein domains experts expressed their agreement as to the content, structure and completeness of the produced conceptual representation, the reusability of such artifact was investigated. The conceptualisation output comprised a representation of a conceptual model containing 52 terms and 79 conceptual relations, from which 23 terms and 45 relations were uncategorized. This can be explained due to the fact that during the first phase of the process there was not available any sort of support to the elicitation of conceptual relations.

Only at the second phase, the conceptualisation framework was introduced to the users (domain experts). The framework included the conceptual relations catalog based on CRRM.

In order to evaluate the conceptual representation reusability, we took the facts expressed through the conceptualisation process and uploaded them into CRRM ontology to gather further metrics and determine the reusability degree of the conceptual representation.

The metrics collected are represented bellow (table 7.1).

Table 7.1: FORSYS - Reusability metrics of the final conceptual representation

Number of Relations $\mu(R_t)$	Uncategorized Relations $\mu(R_\theta)$	Ambiguous Relations $\mu(R_a)$	Total Terms $\mu(T)$	Uncategorized Terms $\mu(T_\theta)$	Incompatible Terms $\mu(T_i)$
79	45	0	52	23	0

From the metrics above, and according to the indicative weight scale from figure 6.2, it is was calculated the Conceptual Relations ambiguity degree as follows:

- $degA = N\mu(R_\theta) \times \omega_a + N\mu(R_a) \times \omega_b = 0,427$, where ω_a = 0,75 and ω_b = 0,25;

By its turn, Concept Type incompatibility is zero according to the expression:

- $deg_{CTi} = N\mu(T_\theta) = 0,0$

From deg_{CRa} and deg_{CTi} and considering the indicative weight scale depicted in figure 6.3, reusability degree is obtained as follows:

- $deg_{Reusability} = deg_{CRa} \times (\omega_a + \omega_t \times deg_{CTi} = 0,744)$, where ω_t = 0,49 and ω_a = 0,51;

If we have into account the output of the first phase of the conceptualization process, and proceed with the same analysis, we get the results as shown in table 7.2.

Table 7.2: FORSYS - Reusability analysis of the initial conceptual representation

	CR metrics	deg_{CRa}	deg_{CTi}	$deg_{Reusability}$
Initial version	$\mu(R_t) = 59$ $\mu(R_\theta) = 56$ $\mu(R_a) = 0$ $\mu(T) = 36$ $\mu(T_\theta) = 32$ $\mu(T_i) = 0$	$deg_{CRa} = 0,854$	$deg_{CTi} = 0$	$deg_{Reusability} = 0,231$

The same exercise was performed considering each of the answers provided by the different groups of domain experts through a questionnaire, which aimed to clarify the meaning of the conceptual structures containing relations whose linking phrase (label) was "has". The results are synthesized in the table 7.3 bellow .

Table 7.3: FORSYS - Reusability analysis according to the domain experts answers

	CR metrics	deg_{CRa}	deg_{CTi}	$deg_{Reusability}$
Group I	$\mu(R_t) = 79$ $\mu(R_\theta) = 14$ $\mu(R_a) = 2$ $\mu(T) = 52$ $\mu(T_\theta) = 7$ $\mu(T_i) = 0$	$deg_{CRa} = 0,105$	$deg_{CTi} = 0$	$deg_{Reusability} = 0,946$
Group II	$\mu(R_t) = 79$ $\mu(R_\theta) = 7$ $\mu(R_a) = 0$ $\mu(T) = 52$ $\mu(T_\theta) = 13$ $\mu(T_i) = 0$	$deg_{CRa} = 0,045$	$deg_{CTi} = 0$	$deg_{Reusability} = 0,977$
Group III	$\mu(R_t) = 79$ $\mu(R_\theta) = 7$ $\mu(R_a) = 0$ $\mu(T) = 52$ $\mu(T_\theta) = 13$ $\mu(T_i) = 0$	$deg_{CRa} = 0,045$	$deg_{CTi} = 0$	$deg_{Reusability} = 0,977$
Group IV	$\mu(R_t) = 79$ $\mu(R_\theta) = 7$ $\mu(R_a) = 0$ $\mu(T) = 52$ $\mu(T_\theta) = 13$ $\mu(T_i) = 0$	$deg_{CRa} = 0,045$	$deg_{CTi} = 0$	$deg_{Reusability} = 0,977$

The interpretation of the results indicates that the more the catalogue of conceptual relations was being used in the conceptualisation process, the more the reusability degree increases. Indeed, the biggest increment on the reusability degree was verified when the relations from CRRM started to be used in the process.

According to the analysis of the shared model and considering its low reusability degree, a new proposal was developed and it derived from the merged one, while also trying to focus on the mentioned gap. The intention was, on one hand, to clarify (in practice) the use of the "has" conceptual relation and, on the other hand, to re-build the conceptual structures considering a set of primitive relation types (Sousa et al., 2012). This remake had a reorganisation purpose, maintaining the domain concepts and following the initial focus questions. Hereupon, for this action, it was considered the conceptual structures linked by the "has" linking phrase, and their types were now discussed together with all groups. Table 7.4 presents some examples of the exercise.

Table 7.4: Examples of the transformation of "has" conceptual relations

Conceptual structure (CS)	Posed question	New CS according to the answer
*"Lesson learned **has** DSS"*	**has** Lesson learned a DSS **as** Scope?	*Lesson learned **has-scope** DSS*
*"Lesson learned **has** Source" and "Case study **is-type-of** Source*	**has** Lesson learned a Case Study **as** a Source?	*Lesson learned **has-source** Case Study*

The new proposal was submitted to the consideration of a second group of domain experts, including an explanation of its *rationale*. At the beginning, the experts generally disagreed with the proposed model. The reasons pointed for that were basically that the model seemed more complex than necessary. This disagreement can be explained in part due to the fact that they were already committed to a more logical view of the 'decision support system' concept structure, regarding the construction of the FORSYS wiki. Moreover, it can be easily observed that maintaining the concepts but changing the relations can result in a substantially different perspective over the subject. The relationships are, indeed, the dynamic part of the structures of knowledge.

Finally, the developed conceptual models were subjected to an assessment against the competency questions. The conclusion was that the models succeeded in answering all the competency questions except one: "*How should lessons learned be applied?*", because it was considered, by the domain experts, out of the scope of the project. Figure 7.5 depicts an extract of the final conceptual model (2nd iteration).

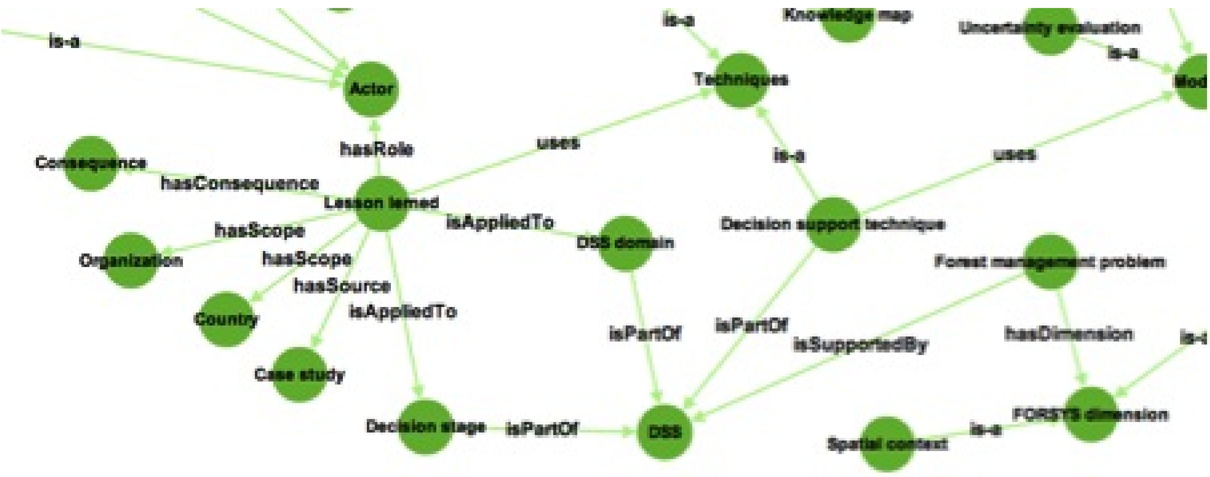


Figure 7.5: Extract of the final conceptual model after the second iteration

Table 7.5 synthesizes the whole research process in terms of goals and results, both for the problem solving and the research dimensions.

7.1.4 Conclusions

In particular, the described case study demonstrated the utility of the conceptualisation framework in a real case-scenario testing several of its valencies, in particular the conceptual relations elicitation, term extraction and conceptual representations evaluation with CRRM.

In general the work performed in the scope of the FORSYS project, contributes to the scarce literature on domain conceptualisation processes in the context of communities of experts. First, it supported a community of forest planning experts in the creation of a shared conceptualisation of the decision support systems for the forest management domain by debating and learning from the joint construction of conceptual models. At the end of this process, the participants acknowledged, both individually and in-group, the achievement of a higher level of conceptual understanding of the domain. Second, the empirical findings achieved during the action-research cycles will shed more light on the characterisation of socio-semantic processes involving knowledge representation. This knowledge was also valuable to improve the conceptME platform in a way that supports more effectively the collaborative conceptualisation processes. In particular, the conclusions about the identification and use of conceptual relations by the experts were of utmost importance.

Table 7.5: Action and research goals and expected results

AR approach	action goals	action results	research goals	research results
Phase 1	(i) to define the requirements for the metamodel improvement; (ii) to elaborate a "proof of concept" for the decision support system conceptual model.	(i) a strategy for the improvement of the metamodel; (ii) a decision support system conceptual model; (iii) the decision support system metamodel part and FORSYS wiki template structure.	(i) to characterise a process of reconceptualisation by the FORSYS domain experts in knowledge representation oriented to information organisation and retrieval.	(i) recommendations on how to address the improvement of conceptual models aimed at specifying knowledge organisation systems.
Phase 2, 1^{st} iteration	(i) to debate and learn, individually, about the lessons learned and case study conceptualization.	(i) conceptual representations of the lessons learned and case study areas of FORSYS (one for each group); (ii) a better and shared understanding of the FORSYS domain (partial).	(i) to characterise the use of conceptual relations by the FORSYS domain experts; (ii) to assess the value of using focus questions to assist the collaborative development of the models.	(i) knowledge of the implicit conceptual relations structures used by the domain experts;
Phase 2, 2^{nd} iteration	(i) to debate, learn and agree about the conceptualisation of the main areas of FORSYS: decision support system, lessons learned and case study; to complete the FORSYS conceptual specification of the knowledge organisation system for the wiki platform.	(i) a complete conceptual model of the FORSYS domain, agreed by the stakeholders for implementing the knowledge organisation system of the wiki platform.	(i) to further characterise the use of conceptual relations by the FORSYS domain experts; (ii) to improve the conceptual negotiation process centered around the discussion of conceptual structures	(i) improvement of the recommendation process for conceptual relations elicitation in conceptual modelling
Phase 3 and 4	(i) to specify the semi-automated linking between conceptME and the FORSYS wiki.	(i) validated specification of the semi-automated linking between conceptME and the FORSYS wiki.		

7.2 Case study II

7.2.1 Description

A group of researchers, whose high level of expertise is well known in the domain of production planning and scheduling, was conducting a study towards the development of a decision-making tool for the chemical-pharmaceutical industry. The development of a tool to support planning is considered very challenging because of the complexity of the decision making process. In particular, the tool must integrate data sourced from distinct departments such as Sales, Research and Development, Procurement, and Production Planning, posing problems of information organization and retrieval. Information (or knowledge) organization models need thus to be understood and used in the same way by all the stakeholders of the decision-support tool. Due to the complexity of the decision and information models involved, decision-makers (DEs from our perspective) ended up spending a significant amount of time trying to understand and explain each other (several companies involved) the concepts and idiosyncrasies of the decisions and methods. This required some sort of representation of the concepts associated with the problem solving methods i.e., a conceptual model that could be used for the DEs to understand what methods are more suitable for a given planning problem. In research terms, we saw the opportunity to face this particular modeling challenge together with DEs by applying the approach described in this paper. The general goal was twofold: first, to identify the relations between concepts, structuring the concepts and developing a common comprehension of the existing knowledge; second, to harmonize concepts targeting the generalization of the data structures to be used by the tool. DEs of this domain have been involved and validated the approach.

To accomplish that, a participatory action-research (Baskerville and Myers, 2004) method was followed.

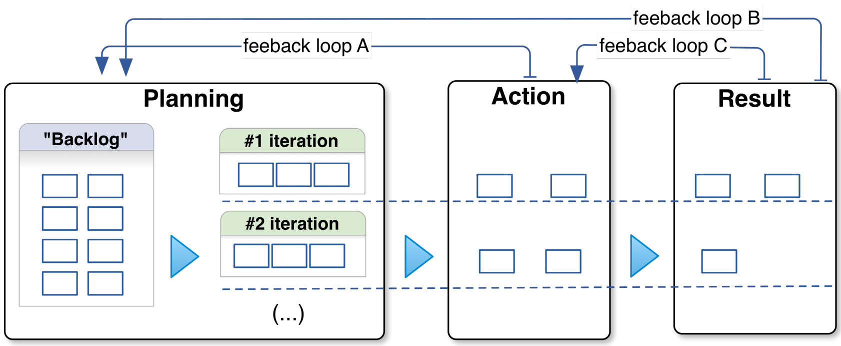


Figure 7.6: Adopted action-research scheme

Figure 7.6 outlines how this study was structured and carried out. We call it an (agile) action research approach, comprising a planning phase, as well as the action and result phase. During the planning phase, it was defined how and when the group (DE and the researchers) would interact. The objectives were collected and introduced into the planning structure. Thus, the "backlog" contains the guidelines for the development of this case study, wherein each "backlog" item corresponds to a specific goal. Later the "backlog" items are reorganized and distributed by several iterations (or sprints). Each interaction contains one or more tasks in order to fulfill one or more goals. Either the "backlog" or the iterations can be readjusted according to the performance of the team during the action phase or the analysis of the results. This updates are represented by feedback loop A and B respectively. Feedback loop C occurs within each iteration. In this case, the researcher, as an observer, is able to perceive if the specialist feels some resistance or difficulty on performing his actions. Thus, some additional information to unblock the process is immediately provided.

Three main goals were set for this real world experiment:

1. to validate the usefulness of the competency questions query patterns;
2. to validate the transformation rules that convert semi-informal assertions into semi-formal conceptual representations, and;
3. to ascertain whether the designed method approach is capable of guiding experts onto the specification and production of conceptual representations from scratch.

In order to evaluate these goals, four iterations were planned. In the first iteration the DEs tried to formulate "what/which" CQs through a simple user interface. During the second iteration the DEs were invited to formulate more complex questions such as "how" and "why" CQs. In both iterations, for each formulated CQ, a conceptual representation (in the form of concept map) with a short textual description was provided. It was the DEs' function to accept or reject the CQ after its analysis. This encloses a double check validation of the formulated queries. In a third iteration, the DEs were confronted with the network of concepts created by the interconnection of the formulated CQs. At this stage, the DEs validate or reject the proposed CQs' interlinking. The final iteration comprised the creation of a domain corpus with a short sample of documents (mainly scientific articles). Afterwards, terminological methods were used to: i) extract domain relevant terms and; ii) extract the contextual information for each concept within the network, in order to identify and retrieve possible concepts and conceptual relations to complete the ontology.

7.2.2 Research results

During the real world experiment case study, a dozen of CQs were produced (see Table 7.6. After the CQ formulation, only one was rejected when the DEs were invited to check CQ's visual conceptual representations. This is an indicator that the transformation rules are correct and aligned with the structure of the different CQ types. Figure 7.7 depicts a subset of CQ conceptual representations using the concept map notation. Note that the "which question" (e.g. "which means?") designating some of the concepts in the next images, refers to a placeholder to be substituted by the DEs.

Having formulated all the CQs, the common terms were identified and merged forming a network of concepts.

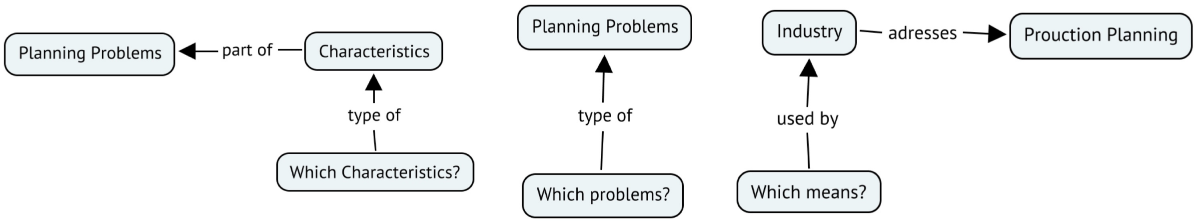


Figure 7.7: Sub-set of CQ conceptual representations (concept map notation)

Example: Figure 7.7 shows a subset of CQ conceptual representations. The 1^{st} and the 2^{nd} representation share the same concept designation, i.e., "Planning Problems". So, syntactic similar terms are merged forming a network of concepts.

The DEs made no rejections at this stage. Finally, DEs proceeded to complete the conceptual model by retrieving the appropriate terms to designate the concepts initially represented by placeholders (e.g. Which Characteristic?). Terminological methods supported DEs to find occurrences, within domain corpus, that might indicate new concepts and/or conceptual relations to be included in the model (c.f. evidenced bellow). The result is evidenced in the figure 7.8 bellow, where the dashed boxes represent the concepts, whose designations were retrieved form corpus, indicating types of characteristics.

Term context excerpt Within the domain corpus the following sentence was retrieved according the terms in the conceptual representation:

> *This work deals with the optimal short-term scheduling of general multipurpose batch plants, considering multiple operational* ***characteristics*** *such as sequence-dependent changeovers, temporary storage in the processing units, lots blending, and material flows traceability.*

In the retrieved context it is possible to get some clues about the types of operation **characteristcs**. Accordingly, the conceptual representations could be extended to included the terms: *sequence-dependent changeovers*; *temporary storage*; *lots blending*, and; *material flows traceability*; as types of characteristics.

7.2.2.1 Utility

The usability of the developed conceptual representations is based on the utility criterion defined before (see section 6.2.1). Accordingly, the utility criterion contains two indicators: the first is related to the conceptual representation usage in a real world application, whereas the second is concerned to the evaluation of the conceptual representations against the formulated competency questions. The utility of the developed conceptual representation was validated qualitatively, based in domain experts inputs, and quantitatively by checking if all requirements (competency questions) were satisfied. For that, we took the facts expressed in the conceptual representation and

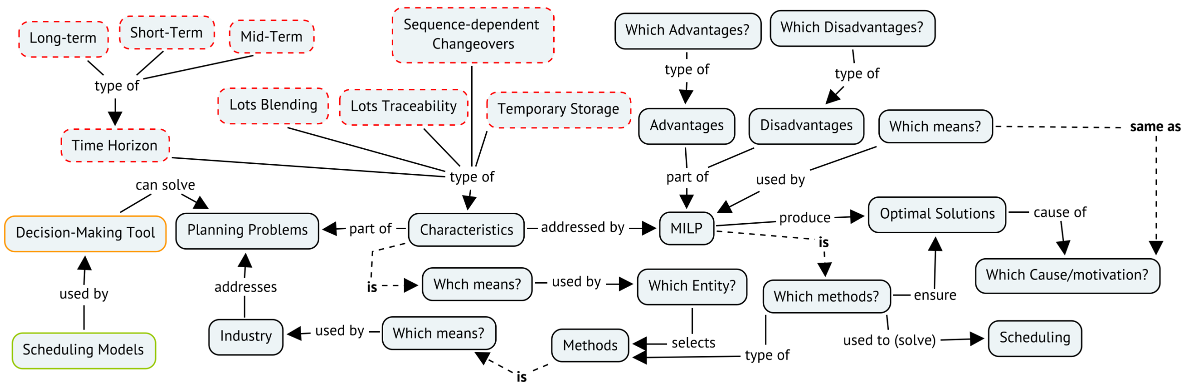


Figure 7.8: Final conceptual representation

uploaded them into the CRRM ontology. Afterwards, we executed all the competency questions as SPARQL queries (as depicted in figure 7.9) and confirmed the output. It was verified that all competency questions returned a result that was confirmed by domain experts as appropriated.

Table 7.6: Summary of the requirements (competency questions) validation

Competency question	It returned an output?	Was the output appropriated?
Which are the types of Planing Problems?	yes	yes
Which kind of Characteristics are part of Planning Problems?	yes	yes
What kind of Methods ensure Optimal Solutions?	yes	yes
What kind of planning problems can be solved by MILP Methods?	yes	yes
How to solve Planning Problems?	yes	yes
How does Industry perform Production Planning?	yes	yes
How do MILP Methods produce Optimal Solutions?	yes	yes
What kind of Problem Characteristics are addressed by MILP Methods?	yes, as exemplified in figure 7.9	yes
How to select Methods?	yes	yes
How does Manufacturing Industry apply Methods?	yes	yes
What kind of Disavantages are part of MILP Methods?	yes	yes
Where does Construction Industry use MILP Methods?	yes	yes

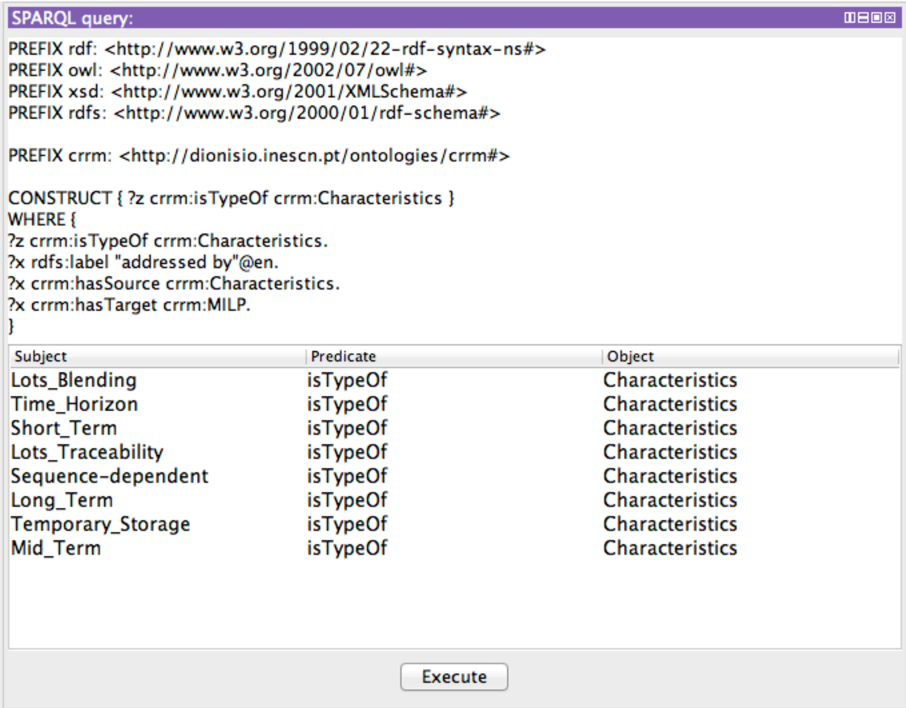


Figure 7.9: Validating conceptual representation through competency questions

7.2.2.2 Reusability

To evaluate reusability there was the need to collect a set of metrics associated to the developed conceptual representation. Following the procedure described in the section 6.2.2, the facts expressed in the conceptualization result were uploaded into the CRRM ontology (in protégé). A set of metrics were then gathered (see 7.7), and the reusability measure calculated.

Table 7.7: Reusability metrics

$\mu(R_t)$	$\mu(R_\theta)$	$\mu(R_a)$	$\mu(T)$	$\mu(T_\theta)$	$\mu(T_i)$
26	4	0	28	2	0

Considering the indicative scales depicted in figure 6.2 and figure 6.3 the reusability degree is **0.96**.

The results shows that the conceptual representation has a high reusability degree. In particular, 96% of the conceptual representations content can be formalised without further processing tasks. Moreover, it means that the conceptual representation content could be retrieved for being reused in another conceptualisation project, maintaining its intended meaning (since its conceptual relations are defined in CRRM ontology). On the other hand 4% of the conceptual representation content is highly susceptible to misinterpretations when reused.

Comparing the reusability of the conceptual representations developed in this case study and the ones developed within the previous case study (see Section 7.1.1, it is clear that following a CQ-based approach together with CRRM, increases the probability of creating conceptual representations with a high reusability degree.

7.3 Conclusion

In particular, the described case study demonstrated the utility of the conceptualisation framework in a real case-scenario testing several of its valencies, in particular the competency questions approach, term extraction and conceptual representations evaluation with CRRM,

In general the work performed in the scope of this case study, showed evidence that our approach contributes for a systematic manner of guiding experts on establishing conceptual commitments in the development of ontologies. In this action-research experiment we could observe that the semi-informal specification of CQs can be effective in overturning the initial barrier to build an ontology. This CQ-approach guides specialists through the knowledge-level of a conceptualization, towards the development of "well-defined" conceptual representations. At the end almost all CQ queries were answered and domain experts knew exactly what was needed to complete the conceptual representations through the use of the terminological methods available in conceptME

Chapter 8

Conclusions and Future Work

8.1 Summary

The quality of the domain conceptual models is a key challenge in designing and maintaining information systems in general. The involvement of domain experts in conceptualisation procedures is essential, in particular to promote knowledge management and facilitate knowledge sharing. This requires the development of semantic tools (e.g., thesaurus, taxonomies, ontologies, ...), whose tasks are usually left on the hands of the knowledge engineers. The involvement of the domain experts in the production of those kind of artefacts, poses new challenges. For that, the developed conceptualisation framework (CF) offers a set of constructs to help building semantically enriched conceptual representations of a particular domain and to assess them. Further, CF framework employs a 3-layered approach for conceptual representations definition (i.e., the formalisation stack), allowing domain experts to formally represent a domain model without the need to commit to a particular formal knowledge representation formalism.

The research process followed a Design Science approach, employing qualitative research methods underpinning by action-research, and complemented by a quantitative evaluation focused on the produced conceptual representations. Here the CF was considered an abstract artifact with utility and reusability as the goal. The artifacts were accommodated in conceptME platform and domain experts were invited to use them in real-world contexts. Finally, the CF was successfully evaluated against utility and reusability criteria from the field of Design Science.

8.2 Research Findings and contributions

Conceptualisation processes has been inspiring several studies in several disciplines. The notion of conceptualisation shared in this thesis is grounded on cognitive semantics, and despite general, it has been applied in an ontological perspective, underpinning collaborative knowledge representation (i.e., for the development of semi-formal ontologies' representations).

In this field, few efforts were directed towards an holistic approach to conceptualisation (combining terminology and knowledge representation), to which the major efforts should be put in the informal and/or semi-formal development of ontologies. Many of the approaches found in literature are focused on its formal specification, disregarding the early phases of the process, given rise to conceptual misalignments, which could render useless and non-reusable representations of ontologies. Other problems arise due to misunderstanding the information context.

Recently (Pereira et al., 2013), there has been a paradigm shift, focused on collaboration and in the early stages of ontology development (the conceptualization phase), wherein domain experts assume a leading role on producing semi-formal ontologies - here called conceptual representations. However, conceptual representations allow poor computational processing and it only contains non-inferable constructs, which means there is not a complete axiomatic definition of domain concepts and relations. To fully axiomatize a conceptual representation there is the need to formalize it using OWL and Description Logic (DL), which is not affordable for a domain expert. They can only provide the relevant conceptual knowledge. Moreover, the need for a formal ontology, in the sense of being fully axiomatized, is not the case in many situations (e.g. for structuring and maintaining knowledge bases information). So, to produce useful and reusable conceptual representations a tradeoff between informal and formal conceptual representations, in the sense that (at the end) they are computational-ready and well-formed (i.e., compliant with a well defined representation scheme) was approached in this thesis.

One of the most important key finding was that conceptual relations are recognised as an important enabler of the conceptualisation activities within a collaborative environment. In fact, the reusability of the conceptualisation results depends on the employed conceptual relations and the way they are interpreted by the domain experts. However, as argued in this thesis, conceptual relations relevance is proportional to the difficulty of its proper elicitation. From the domain expert point-of-view, the elicitation of conceptual relations is not feasible without the adequate means to facilitate the identification, selection and application of conceptual relations in a awareness-driven process (i.e., the domain experts are aware of the context of use of a conceptual relations).

Another important finding is that the process of establishing the conceptualisation objectives (shared by all stakeholders involved in a conceptualisation project) is fundamental for the success of the conceptual representations. This process is unstructured by nature, being a continuum from the initial discussion of the purpose to the first agreed conceptual representation. The inherent (inter) subjectivity of the process and their outcomes together with an excessive informality are perhaps the reasons for being overlooked in the literature. This thesis proposed an approach to integrate competency questions (CQ) and conceptual graphs to the support of domain experts and knowledge specialists in defining the purpose and fundamental conceptual commitments of the semi-formal ontology to be developed.

In summary, the contributes of this thesis are offered by means of an abstract artefact , i.e., the conceptualisation framework, which provides the necessary support for domain experts to build reusable conceptual representations. Moreover, it was implemented in a collaborative technological platform (i.e., conceptME) providing support to:

1. conceptual relation elicitation, through the Conceptual Relations Reference Model (CRRM), which implements an ontology, allowing:

- a template-based construction of domain conceptual structures;
- the assessment of the development conceptual representations, if they were built using the CRRM;

2. the formulation of structured competency questions, specifying "claims" that conceptual representations should address. In fact, conceptual representations could be built based on the designed CQ approach.
3. term extractions feature, in order to find lexical pattern in corpus that could indicate new term candidates;
4. formalise conceptual representations in conceptual graphs, empowering its computation capabilities, facilitating knowledge sharing and reusability.

Finally, the conceptualisation framework – comprising all the supporting artefacts – was evaluated against the reusability and utility criteria, as proposed in Section 6.1. The framework (as an abstract artefact) was found to be a purposeful, innovative and a generic solution to the problem of semi-formal knowledge representation reusability.

8.3 Further research

This work paved the way for new research challenges to be tackled in a near future. We envisage to promote the continuation of current research considering several aspects:

New case studies Promoting new qualitative studies to obtain further feedback from domain experts in several distinct domains. The epistemological basis of domain experts vary considerable according to the domain and it would be interesting to have empirical evidences from a wider spectrum of domains on using conceptME and the conceptualisation framework to build conceptual representations for different purposes.

Extending CRRM: mapping conceptual relations to ontological relations CRRM allow the identification and selection of top-level conceptual relations. It might be necessary to add more specific relations to CRRM due to a couple of reasons:

1. the specificities and the level of detail needed to describe a particular technical domain may require more specific conceptual relations, otherwise the concepts might remain underspecified.
2. it might be necessary to understand the restrictions of the associations between two or more concepts in order to fully axiomatize the conceptual representation.
   Let us consider two terms (A and B). At the moment, the domain experts can easily add (through CRRM) a "Part-Whole" conceptual relation between the two terms, but is not possible, in a pragmatic way, to detail the "Part-Whole" relation in order to define if term A is, for instance, a "proper part" of term B.

Competency question could be used to cope with this challenge from de domain experts perspective.

Knowledge representation From a semiotic point-of-view and considering the graph model developed in this thesis, it would be interesting to extend the formalisaton stack to give users the opportunity to use other representation formalism besides concept maps (e.g., UML class diagrams, BPM, ER, . . .). In fact the use of UML classes is already under study by the author.

Incorporate linked data principles in conceptME In a more technological perspective, we are convinced that the conceptME architecture could be extended according to the principles of linked data and transform the conceptME platform in a global multi-domain repository of conceptual representations.

In a final remark this thesis is closed with a vision of the future of conceptME in Figure 8.1.

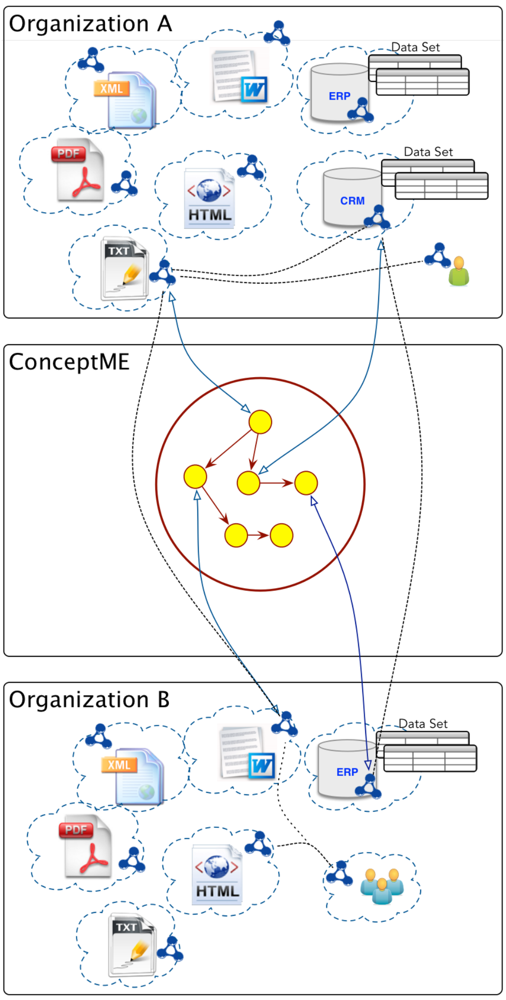


Figure 8.1: The future of conceptME: overview

References

A. R. Hevner, S. T. March, J. Park, and S. Ram, “Design science in information systems research,” *MIS quarterly*, vol. 28, no. 1, pp. 75–105, 2004.

C. Pereira, C. Sousa, and A. L. Soares, “Supporting conceptualisation processes in collaborative networks: a case study on an r&d project,” *International Journal of Computer Integrated Manufacturing*, vol. 26, no. 11, pp. 1066–1086, Nov. 2013. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/0951192X.2012.684714

U. Frank, “Conceptual modelling as the core of the information systems discipline - perspectives and epistemological challenges,” *AMCIS 1999 Proceedings*, Dec. 1999. [Online]. Available: http://aisel.aisnet.org/amcis1999/240

J. Mylopoulos, “Conceptual modelling and telos,” 1992.

Y. Wand and R. Weber, “Research commentary: Information systems and conceptual modeling?a research agenda,” *Information Systems Research*, vol. 13, no. 4, pp. 363–376, Dec. 2002. [Online]. Available: http://isr.journal.informs.org/cgi/doi/10.1287/isre.13.4.363.69

N. Roussopoulos and D. Karagiannis, “Conceptual modeling: Past, present and the continuum of the future,” in *Conceptual Modeling: Foundations and Applications*, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, A. T. Borgida, V. K. Chaudhri, P. Giorgini, and E. S. Yu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, vol. 5600, pp. 139–152. [Online]. Available: http://www.springerlink.com/index/10.1007/978-3-642-02463-4_9

N. Guarino and G. Guizzardi, “In the defense of ontological foundations for conceptual modeling,” *Scandinavian Journal of Information Systems*, vol. 18, no. 1, p. 1, 2006.

J. A. Gulla, “Using models in enterprise systems projects,” in *Conceptual Modelling in Information Systems Engineering*, J. Krogstie, A. L. Opdahl, and S. Brinkkemper, Eds. Springer Berlin Heidelberg, Jan. 2007, pp. 107–122. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-540-72677-7_7

M. Kharatmal and G. Nagarjuna, “Introducing rigor in concept maps,” in *Conceptual Structures: From Information to Intelligence*, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, M. Croitoru, S. Ferré, and D. Lukose, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, vol. 6208, pp. 199–202. [Online]. Available: http://www.springerlink.com/index/10.1007/978-3-642-14197-3_22

A. Auger and C. Barriere, “Probing semantic relations,” *Probing Semantic Relations: Exploration and Identification in Specialized Texts*, vol. 23, p. 1, 2010.

A. Elsayed, “Relations for graphical conceptual representations,” in *2009 Ninth IEEE International Conference on Advanced Learning Technologies*, Riga, Latvia, Jul. 2009, pp. 724–725. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5194353

S. Seufert and A. Seufert, “Towards the continuously learning organization through knowledge networking-case swiss re group,” in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, 2000.* IEEE, Jan. 2000.

G. D. Bhatt, “Organizing knowledge in the knowledge development cycle,” *Journal of Knowledge Management*, vol. 4, no. 1, pp. 15–26, Mar. 2000. [Online]. Available: http://www.emeraldinsight.com/journals.htm?articleid=883690

H. J. Levesque and R. J. Brachman, “Expressiveness and tractability in knowledge representation and reasoning1,” *Computational Intelligence*, vol. 3, no. 1, pp. 78–93, 1987. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1111/j.1467-8640.1987.tb00176.x/abstract

H. Takeda, P. Veerkamp, and H. Yoshikawa, “Modeling design process,” *AI magazine*, vol. 11, no. 4, p. 37, 1990.

K. Siau and Y. Wang, “Cognitive evaluation of information modeling methods,” *Information and Software Technology*, vol. 49, no. 5, pp. 455–474, May 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950584906000905

G. Hodge, *Systems of knowledge organization for digital libraries*. digital library federation, council on library and information resources, 2000. [Online]. Available: http://www.sims.monash.edu/subjects/ims2603/resources/Assignment2Papers/SKOforDigLib.pdf

S. Martin, A. Aurum, R. Jeffery, and B. Paech, “Requirements engineering process models in practice,” 2002, pp. 141–155.

P. Martin, “Knowledge representation, sharing, and retrieval on the web,” in *Web Intelligence*, N. Zhong, J. Liu, and Y. Yao, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 243–276. [Online]. Available: http://link.springer.com/10.1007/978-3-662-05320-1_12

A. Gomez-Perez, *Ontological engineering : with examples from the areas of knowledge management, e-commerce and the Semantic Web*. London ;;New York: Springer, 2004.

W. Reinhardt, B. Schmidt, P. Sloep, and H. Drachsler, “Knowledge worker roles and actions—results of two empirical studies,” *Knowledge and Process Management*, vol. 18, no. 3, pp. 150–174, 2011.

E. Waltz, *Knowledge management in the intelligence enterprise*. Boston [Mass.]: Artech House, 2003.

H. Oinas-Kukkonen and H. Oinas, “The 7c model for organizational knowledge creation and management,” *Infotech Oulu OASIS & Dept. Information Processing Science, University of Oulu, Finland*, 2001. [Online]. Available: http://www.cream.oulu.fi/english/documents/Harri_Oinas-Kukkonen_artikkeli.pdf

V. Evans, B. K. Bergen, and J. Zinken, "The cognitive linguistics enterprise: An overview," *The cognitive linguistics reader*, pp. 2–36, 2007.

W. Kuhn, "Why information science needs cognitive semantics," in *Workshop on the Potential of Cognitive Semantics for Ontologies (FOIS 2004), Torino, Italy*, 2004.

V. Evans and M. Green, *Cognitive linguistics: An introduction.* Lawrence Erlbaum Associates Publishers, 2006.

L. W. Barsalou, "Situated conceptualization," *Handbook of categorization in cognitive science*, vol. 619, p. 650, 2005.

——, "Simulation, situated conceptualization, and prediction," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 364, no. 1521, pp. 1281–1289, 2009.

M. R. Genesereth and N. J. Nilsson, "Logical foundations of artificial," *Intelligence. Morgan Kaufmann*, 1987.

T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing," in *International Journal of Human-Computer Studies*. Kluwer Academic Publishers, 1993, pp. 907–928.

R. Studer, V. R. Benjamins, and D. Fensel, "Knowledge engineering: principles and methods," *Data & knowledge engineering*, vol. 25, no. 1, pp. 161–197, 1998.

N. Guarino, *Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy*. IOS press, 1998, vol. 46.

T. Gruber, "Toward principles for the design of ontologies used for knowledge sharing," 1995. [Online]. Available: http://xml.engineeringvillage2.org/controller/servlet/Controller?CID=expertSearchDetailedFormat&EISESSION=1_caf6c1121c469fcc32233ses4&SYSTEM_USE_SESSION_PARAM=true&SEARCHID=14d581b121a75fcb0d6518prod4data1&DOCINDEX=1&PAGEINDEX=1&RESULTSCOUNT=1&database=2&format=expertSearchDetailedFormat

S. Staab, "On understanding the sharing of conceptualizations," http://userpages.uni-koblenz.de/ staab/Research/Talks/20080314SharedConceptualizations.pdf, March 2008.

C.-R. Huang, N. Calzolari, A. Gangemi, A. Lenci, A. Oltramari, and L. Prévot, *Ontology and the Lexicon*, 2010.

I. Nonaka, R. Toyama, and N. Konno, "Seci,< i> ba</i> and leadership: a unified model of dynamic knowledge creation," *Long range planning*, vol. 33, no. 1, pp. 5–34, 2000.

A. Gómez, A. Moreno, J. Pazos, and A. Sierra-Alonso, "Knowledge maps: An essential technique for conceptualisation," *Data & Knowledge Engineering*, vol. 33, no. 2, pp. 169–190, 2000.

M. Fernández-López and A. Gómez-Pérez, "A survey on methodologies for developing, maintaining, evaluating and reengineering ontologies," OntoWeb Consortium, A survey on methodologies for developing, maintaining, evaluating and reengineering ontologies D1.4, 2002. [Online]. Available: http://www.sti-innsbruck.at/fileadmin/documents/deliverables/Ontoweb/D1.4.pdf

C. Roussey, "Guidelines to build ontologies: A bibliographic study," LEMA - Université de Liège, Technical Report 1, 2005. [Online]. Available: http://www.towntology.net/Documents/guidelines.pdf

N. Omrane, A. Nazarenko, and S. Szulman, "From linguistics to ontologies - the role of named entities in the conceptualisation process," in *KEOD 2011 - Proceedings of the International Conference on Knowledge Engineering and Ontology Development, Paris, France, 26-29 October, 2011*, 2011, pp. 249–254.

G. Fauconnier and M. Turner, "Conceptual integration networks," *Cognitive Science*, vol. 22, no. 2, pp. 133–187, Mar. 1998. [Online]. Available: http://doi.wiley.com/10.1207/s15516709cog2202_1

C. Pereira, C. Sousa, and A. L. Soares, "A negotiation approach to support conceptual agreements for ontology content," in *On the Move to Meaningful Internet Systems: OTM 2012 Workshops*. Springer, 2012, pp. 474–483.

C. Wagner, "Breaking the knowledge acquisition bottleneck through conversational knowledge management," *Innovative Technologies for Information Resources Management*, p. 200, 2008.

N. Aussenac-Gilles, B. Biébow, and S. Szulman, "Corpus analysis for conceptual modelling," 2000.

T. McEnery and A. Wilson, *Corpus linguistics*. Edinburgh: Edinburgh University Press, 1996.

G. W. Mineau, R. Missaoui, and R. Godin, "Conceptual modeling using conceptual graphs." in *KRDB*, 2000, pp. 73–86.

S. Staab, M. Erdmann, and A. Maedche, "Semantic patterns," 2000.

C. Sousa, A. Soares, C. Pereira, and R. Costa, "Supporting the identification of conceptual relations in semi-formal ontology development," in *ColabTKR 2012 - Terminology and Knowledge Representation Workshop at International Conference on Language Resources and Evaluation*, Istanbul, Turkey, 2012.

F. Alvarez, A. Vaquero, O. Saenz, M. D. Buenaga, A. Flores, and V. R. Palacios, "Semantic relation modeling and representation for problem-solving ontology-based linguistic resources: Issues and proposals," 2007.

J. F. Sowa, "Conceptual graphs for representing conceptual structures," *Conceptual Structures in Practice*, pp. 101–136, 2009.

F. Baader, "Logic-based knowledge representation," in *Artificial intelligence today*. Springer, 1999, pp. 13–41.

J. D. Heflin, "Towards the semantic web: Knowledge representation in a dynamic, distributed environment," Ph.D. dissertation, College Park, MD, USA, 2001, aAI3024929.

D. Calvanese, M. Lenzerini, and D. Nardi, "Unifying class-based representation formalisms," *J of Artificial Intelligence Research*, vol. 11, no. 199-240, pp. 199–240, 1999.

B. Hjorland, "Semantics and knowledge organization," *Annual review of information science and technology*, vol. 41, no. 1, pp. 367–405, 2007.

C. Khoo and J.-C. Na, "Semantic relations in information science," *Annual Review of Information Science and Technology*, vol. 40, p. 157, 2006. [Online]. Available: http://www.ntu.edu.sg/home/assgkhoo/papers/khoo_na.semantic_relations.ARIST2006.pdf

I. Peters and K. Weller, "Paradigmatic and syntagmatic relations in knowledge organization systems," no. 59, pp. 100–107, 2008. [Online]. Available: http://130.203.133.150/viewdoc/summary;jsessionid=B64F67FFADB2C588435E27637283B4AF?doi=10.1.1.218.7863

W. Stock, "Concepts and semantic relations in information science," *Journal of the American Society for Information Science and Technology*, vol. 61, no. 10, pp. 1951–1969, Oct. 2010. [Online]. Available: http://doi.wiley.com/10.1002/asi.21382

A. Elsayed, "A framework for using semantic relations in conceptual structures," in *2008 Eighth IEEE International Conference on Advanced Learning Technologies*, Santander, Cantabria, Spain, 2008, pp. 1069–1070. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4561917

R. Shams and A. Elsayed, "Development of a conceptual structure for a domain-specific corpus." in *Games Computing and Creative Technologies: Conference Papers (Peer-Reviewed)*, 2008, p. 6.

R. Shams, A. Elsayed, and Q. M. Akter, "A corpus-based evaluation of a domain-specific text to knowledge mapping prototype," *Journal of Computers*, vol. 5, no. 1, pp. 69–80, 2010.

V. C. Storey, "Comparing relationships in conceptual modeling: mapping to semantic classifications," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, no. 11, pp. 1478–1489, 2005.

J. Morris, "Non-classical lexical semantic relations," *IN PROCEEDINGS OF HTL-NAACL WORKSHOP ON COMPUTATIONAL LEXICAL SEMANTICS*, pp. 46–51, 2004. [Online]. Available: http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.79.6344

A. Chiu, P. Poupart, and C. DiMarco, "Learning lexical semantic relations using lexical analogies," *OTT'06*, vol. 1, p. 33, 2007.

J. C. Sager, *A practical course in terminology processing*. John Benjamins Publishing, 1990.

B. N. Madsen, B. S. Pedersen, and H. E. Thomsen, "Defining semantic relations for ontoquery," *A structured overview of semantic relations and suggestions for their application in OntoQuery*, pp. 57–88, 2001.

A. Nuopponen, "Concept relations v2. an update of a concept relation classification," Copenhagen, 2005, pp. 127–138.

E. Wüster and M. T. Cabré, *Introducción a la teoría general de la terminología y a la lexicografía terminológica*, U. P. Fabra, Ed., 1998.

G. Lakoff, *Women, fire, and dangerous things: What categories reveal about the mind*. Cambridge Univ Press, 1990.

M. Johnson and G. Lakoff, "Metaphors we live by," *University of Chicago*, 1980.

E. Rosch and B. B. Lloyd, *Cognition and categorization*. Hillsdale, N.J. : New York: L. Erlbaum Associates ; distributed by Halsted Press, 1978.

N. Riemer, *Introducing semantics*. Cambridge University Press, 2010.

L. Gillam, M. Tariq, and K. Ahmad, "Terminology and the construction of ontology," *TERMINOLOGY*, vol. 11, pp. 55–81, 2005. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.61.5244

C. Yu-Liang, "Elicitation synergy of extracting conceptual tags and hierarchies in textual document," *Expert Systems with Applications*, vol. 32, no. 2, pp. 349–357, Feb. 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417405003441

P. Buitelaar, P. Cimiano, P. Haase, and M. Sintek, "Towards linguistically grounded ontologies," in *The semantic web: research and applications*. Springer, 2009, pp. 111–125.

P. Faber, P. León, and J. A. Prieto, "Semantic relations, dynamicity and terminological knowledge bases," *Current Issues in Language Studies*, vol. 1, no. 1, pp. 1–23, 2009.

G. O. Klein and B. Smith, "Concept systems and ontologies: Recommendations for basic terminology," *Transactions of the Japanese Society for Artificial Intelligence = Jinko Chino Gakkai ronbunshi*, vol. 25, no. 3, pp. 433–441, Jan. 2010. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3034144/

A. Thomasson, "Categories," http://plato.stanford.edu/archives/fall2013/entries/categories/, 2012.

E. J. Lowe, *The Four-Category Ontology*. Oxford University Press, Dec. 2005. [Online]. Available: http://www.oxfordscholarship.com/view/10.1093/0199254397.001.0001/acprof-9780199254392

L. Schneider, "The logic of the ontological square," *Studia Logica*, vol. 91, no. 1, pp. 25–51, Feb. 2009. [Online]. Available: http://link.springer.com/article/10.1007/s11225-009-9165-6

D. Kless, "Ontologies and thesauri - similarities and differences," Berlin, 2011.

U. Schwarz and B. Smith, *Applied Ontology: An Intoduction*. ontos verlag, 2008, ch. Ontological Relations.

G. Guizzardi, *Ontological foundations for structural conceptual models*. CTIT, Centre for Telematics and Information Technology, 2005.

W. Degen and H. Herre, "What is an upper level ontology?" Workshop on 'Ontologies', Vienna, 2001.

A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, and R. Oltramari, "Understanding top-level ontological distinctions," in *In Proceedings of IJCAI-01 Workshop on Ontologies and Information Sharing*. AAAI Press, 2001, pp. 26–33.

A. L. Thomasson, *Fiction and metaphysics*. Cambridge University Press, 1999.

S. T. March and G. F. Smith, "Design and natural science research on information technology," *Decis. Support Syst.*, vol. 15, no. 4, pp. 251–266, Dec. 1995. [Online]. Available: http://dx.doi.org/10.1016/0167-9236(94)00041-2

A. Hevner and S. Chatterjee, "Design science research in information systems," in *Design Research in Information Systems*. Boston, MA: Springer US, 2010, vol. 22, pp. 9–22. [Online]. Available: http://link.springer.com/10.1007/978-1-4419-5653-8_2

R. Winter, “Design science research in europe,” *European Journal of Information Systems*, vol. 17, no. 5, pp. 470–475, Oct. 2008. [Online]. Available: http://www.palgrave-journals.com/doifinder/10.1057/ejis.2008.44

A. R. Hevner, “A three cycle view of design science research,” *Scandinavian journal of information systems*, vol. 19, no. 2, p. 4, 2007.

M. Helfert, B. Donnellan, and L. Ostrowski, “The case for design science utility and quality-evaluation of design science artifact within the,” *Systems, Signs & Actions*, vol. 6, no. 1, pp. 46–66, 2012.

J. Krogstie, G. Sindre, and H. Jørgensen, “Process models representing knowledge for action: a revised quality framework,” *European Journal of Information Systems*, vol. 15, no. 1, pp. 91–102, Feb. 2006. [Online]. Available: http://www.palgrave-journals.com/doifinder/10.1057/palgrave.ejis.3000598

R. Stamper, “Signs, norms, and information systems,” in *Signs at work: Semiosis and information processing in Organisations*, B. Holmqvist, P. Anderson, H. Klein, and R. Posner, Eds. Walter de Gruyter, 1996.

S. Barros, R. Costa, A. L. Soares, and M. Silva, “Integrating terminological methods in a framework for collaborative development of semi-formal ontologies,” in *ColabTKR 2012-Terminology and Knowledge Representation Workshop Workshop Programme*, 2012, p. 7.

M. Chein and M.-L. Mugnier, *Graph-based knowledge representation: computational foundations of conceptual graphs*. Springer, 2008.

J. F. Sowa, “Semantics of conceptual graphs,” in *Proceedings of the 17th annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 1979, pp. 39–44.

J. Sowa, “Conceptual graphs - bridging the gap between language and logic,” 2003. [Online]. Available: http://www.jfsowa.com/cg/cgonto.htm

A. De Moor, M. Keeler, and G. Richmond, “Towards a pragmatic web,” in *Conceptual Structures: Integration and Interfaces*. Springer, 2002, pp. 235–249.

C. Laudy, J.-G. Ganascia, and C. Sedogbo, “High-level fusion based on conceptual graphs,” in *Information Fusion, 2007 10th International Conference on*. IEEE, 2007, pp. 1–8.

M. Chein and M. laure Mugnier, “Conceptual graphs: fundamental notions,” *Revue d’Intelligence Artificielle*, vol. 6, pp. 365–406, 1992.

J.-F. Baget, “Simple conceptual graphs revisited: hypergraphs and conjunctive types for efficient projection algorithms,” in *Conceptual Structures for Knowledge Creation and Communication*. Springer, 2003, pp. 229–242.

M. Chein and M. Mugnier, “Conceptual graphs are also graphs,” LIRMM (CNRS and Universite de Montpellier II), 161, rue Ada, 34392 Montpellier Cedex 5, Tech. Rep., 1995.

J. F. Sowa, *Conceptual Structures: Information Processing in Mind and Machine*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1984.

M. Montes-y Gómez, A. Gelbukh, and A. López-López, "Comparison of conceptual graphs," in *MICAI 2000: Advances in Artificial Intelligence*. Springer, 2000, pp. 548–556.

C. Bezerra, F. Freitas, and F. Santana, "Evaluating ontologies with competency questions," in *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on*, vol. 3. IEEE, 2013, pp. 284–285.

L. Zemmouchi-Ghomari and A. R. Ghomari, "Translating natural language competency questions into sparql queries: A case study," in *WEB 2013, The First International Conference on Building and Exploring Web Based Environments*, 2013, pp. 81–86.

P. C. B. Fernandes, R. S. Guizzardi, and G. Guizzardi, "Using goal modeling to capture competency questions in ontology-based systems," *Journal of Information and Data Management*, vol. 2, no. 3, p. 527, 2011.

Y. Malheiros and F. Freitas, "A method to develop description logic ontologies iteratively based on competency questions: an implementation." in *ONTOBRAS*, 2013, pp. 142–153.

Y. Ren, A. Parvizi, C. Mellish, J. Z. Pan, K. van Deemter, and R. Stevens, "Towards competency question-driven ontology authoring," in *The Semantic Web: Trends and Challenges*. Springer, 2014, pp. 752–767.

T. Gruber, "Collective knowledge systems: Where the social web meets the semantic web," *Web semantics: science, services and agents on the World Wide Web*, vol. 6, no. 1, pp. 4–13, 2008.

J. F. Sowa, "Conceptual graphs as a universal knowledge representation," *Computers & Mathematics with Applications*, vol. 23, no. 2, pp. 75–93, 1992.

M. Uschold, "Knowledge level modelling: concepts and terminology," *The knowledge engineering review*, vol. 13, no. 01, pp. 5–29, 1998.

D. Shotton, C. Catton, and G. Klyne, "Ontologies for sharing, ontologies for use," *Ontogenesis*, 2010.

T. H. Cao and A. H. Mai, "Ontology-based understanding of natural language queries using nested conceptual graphs," in *Conceptual Structures: From Information to Intelligence*. Springer, 2010, pp. 70–83.

S. Sarawagi, "Information extraction," *Found. Trends databases*, vol. 1, no. 3, pp. 261–377, Mar. 2008. [Online]. Available: http://dx.doi.org/10.1561/1900000003

M. Baroni, B. Murphy, E. Barbu, and M. Poesio, "Strudel: A corpus-based semantic model based on properties and types," *Cognitive Science*, vol. 34, no. 2, pp. 222–254, 2010.

M. A. Rodriguez and P. Neubauer, "Constructions from dots and lines," *Bulletin of the American Society for Information Science and Technology*, vol. 36, no. 6, pp. 35–41, 2010.

J. Davies, A. Duke, and Y. Sure, "Ontoshare - an ontology-based knowledge sharing system for virtual communities of practice," vol. 10, no. 3, pp. 262–283, mar 2004, |http://www.jucs.org/jucs_10_3/ontoshare_an_ontology_based|.

C. Ghidini, B. Kump, S. Lindstaedt, N. Mahbub, V. Pammer, M. Rospocher, and L. Serafini, "Moki: The enterprise modelling wiki," in *The Semantic Web: Research and Applications*. Springer, 2009, pp. 831–835.

C. Sá, C. Sousa, C. Pereira, and A. L. Soares, “Supporting collaborative conceptualisation tasks through a semantic wiki,” in *11th Conferência da Associa\ccão Portuguesa de Sistemas de Informa\ccão*, 2011. [Online]. Available: http://pascal.iseg.utl.pt/~capsi2011/documentos/LivroCAPSI.pdf

P. Reason and H. Bradbury, *The Sage handbook of action research: participative inquiry and practice*, 2nd ed. London ; Thousand Oaks, Calif: SAGE Publications, 2008.

R. L. Baskerville, “Investigating information systems with action research,” *Commun. AIS*, vol. 2, no. 3es, Nov. 1999. [Online]. Available: http://dl.acm.org/citation.cfm?id=374468.374476

U. Flick, “Qualitative research - state of the art,” *Social Science Information*, vol. 41, no. 1, pp. 5–24, Mar. 2002. [Online]. Available: http://ssi.sagepub.com/cgi/doi/10.1177/0539018402041001001

R. Baskerville and M. D. Myers, “Special issue on action research in information systems: making is research relevant to practice-foreword,” *MIS Q.*, vol. 28, no. 3, pp. 329–335, Sep. 2004. [Online]. Available: http://dl.acm.org/citation.cfm?id=2017227.2017229

Appendix A

CRRM Ontology

CRRM ontology codified in OWL

```
<?xml version="1.0"?>


<!DOCTYPE Ontology [
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY xml "http://www.w3.org/XML/1998/namespace" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22 rdf syntax ns#" >
]>


<Ontology xmlns="http://www.w3.org/2002/07/owl#"
     xml:base="http://dionisio.inescn.pt/ontologies/crrm"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf schema#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdf="http://www.w3.org/1999/02/22 rdf syntax ns#"
     xmlns:xml="http://www.w3.org/XML/1998/namespace"
     ontologyIRI="http://dionisio.inescn.pt/ontologies/crrm">
    <Prefix name="" IRI="http://www.w3.org/2002/07/owl#"/>
    <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#"/>
    <Prefix name="rdf" IRI="http://www.w3.org/1999/02/22 rdf syntax ns#"/>
    <Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#"/>
    <Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf schema#"/>
    <Prefix name="swrl" IRI="http://www.w3.org/2003/11/swrl#"/>
    <Prefix name="query" IRI="http://swrl.stanford.edu/ontologies/built ins
        /3.3/query.owl#"/>
    <Prefix name="swrlb" IRI="http://www.w3.org/2003/11/swrlb#"/>
    <Declaration>
        <Class IRI="#AmbiguousRelations"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Any"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Cause"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Cause Effect"/>
    </Declaration>
    <Declaration>
        <Class IRI="#CauseEffect_Template"/>
    </Declaration>
    <Declaration>
        <Class IRI="#ConceptualRelation"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Constituition_and_Containment"/>
```

```
    </Declaration>
    <Declaration>
        <Class IRI="#Container"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Containment"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Containment_Template"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Content"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Effect"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Ends"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Generic"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Generic  Dependence"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Generic  Specific"/>
    </Declaration>
    <Declaration>
        <Class IRI="#GenericSpecific_Template"/>
    </Declaration>
    <Declaration>
        <Class IRI="#LinkingPhrase"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Local"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Means"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Part"/>
    </Declaration>
    <Declaration>
        <Class IRI="#PartHood"/>
    </Declaration>
    <Declaration>
        <Class IRI="#PartHood_Template"/>
    </Declaration>
```

```
    <Declaration>
        <Class IRI="#Participation"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Participation_Template"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Spatial"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Spatial_Template"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Specific"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Template"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Temporal"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Temporal_Template"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Term"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Time Space_Dependence"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Usage"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Usage_Template"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Whole"/>
    </Declaration>
    <Declaration>
        <Class IRI="#defaultLinkingPhrase"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#OWLObjectProperty_154bc8e4_72d7_48d2_a89a_9141494a8914"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#OWLObjectProperty_198c01e3_d615_4f5f_b2bf_cc80c55c100b"/>
```

```
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#
            OWLObjectProperty_4c4b67c7_35ec_4fec_ab9b_eb4293d4f30d"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#
            OWLObjectProperty_51251faa_5158_44d7_beba_6b1eb77ad196"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#
            OWLObjectProperty_96c9f883_5dff_4bf9_a96b_4d1f5ccb258b"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#
            OWLObjectProperty_e66f24b5_f9c5_49f8_8f31_9105044711eb"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#designates"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#follows"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#hasConceptualRelation"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#hasPart"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#hasSource"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#hasTarget"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#hasTemplate"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#hasTerm"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#hasType"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#incompatibleTypes"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#isCauseOf"/>
```

```
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#isCausedBy"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#isDesignatedBy"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#isLocatedIn"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#isLocationOf"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#isPartOf"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#isTypeOf"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#isUsedBy"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#precedes"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#relatesTo"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#uses"/>
    </Declaration>
    <Declaration>
        <DataProperty IRI="#hasCompetencyQuestion"/>
    </Declaration>
    <Declaration>
        <DataProperty IRI="#hasIntent"/>
    </Declaration>
    <Declaration>
        <NamedIndividual IRI="#OWLNamedIndividual_058cd8c9_d267_4b33_9dcf_3171ba1aa410"/>
    </Declaration>
    <Declaration>
        <NamedIndividual IRI="#OWLNamedIndividual_11d5abc7_b602_433f_9457_86a93b73c256"/>
    </Declaration>
    <Declaration>
        <NamedIndividual IRI="#OWLNamedIndividual_266ef55e_f579_4481_98ea_61aa09ed0e61"/>
    </Declaration>
```

```
    <Declaration>
        <NamedIndividual IRI="#
            OWLNamedIndividual_361e575c_0955_458f_89d9_d525d2accaa9"/>
    </Declaration>
    <Declaration>
        <NamedIndividual IRI="#
            OWLNamedIndividual_368ad93e_19f7_40f3_bcf3_06677f41cc30"/>
    </Declaration>
    <Declaration>
        <NamedIndividual IRI="#
            OWLNamedIndividual_573934a2_1786_4ec2_b42d_a5cc41ed0935"/>
    </Declaration>
    <Declaration>
        <NamedIndividual IRI="#
            OWLNamedIndividual_6593e364_31d5_4157_9238_9fe5eafccd25"/>
    </Declaration>
    <Declaration>
        <NamedIndividual IRI="#
            OWLNamedIndividual_6c4f1230_6123_4f7e_b23c_e41156e6ac8f"/>
    </Declaration>
    <Declaration>
        <NamedIndividual IRI="#
            OWLNamedIndividual_979dc71e_d533_47d9_b96f_39cb499fcdff"/>
    </Declaration>
    <Declaration>
        <NamedIndividual IRI="#
            OWLNamedIndividual_a8870049_ca0c_477a_ab0d_9d1bb581787b"/>
    </Declaration>
    <Declaration>
        <NamedIndividual IRI="#
            OWLNamedIndividual_abe3dc92_2f2f_46c3_903a_56f45f283c1a"/>
    </Declaration>
    <Declaration>
        <NamedIndividual IRI="#
            OWLNamedIndividual_c2ab4650_8bc3_40e1_bd93_612e9d8551e7"/>
    </Declaration>
    <Declaration>
        <NamedIndividual IRI="#
            OWLNamedIndividual_d0c7d18a_b27c_4de7_a3ef_b7c5fb847824"/>
    </Declaration>
    <Declaration>
        <NamedIndividual IRI="#
            OWLNamedIndividual_d9013d84_84d1_4a8b_91d6_9c2840554593"/>
    </Declaration>
    <Declaration>
        <NamedIndividual IRI="#
            OWLNamedIndividual_f0506f52_8089_4b1c_bd55_ffe69efc88f8"/>
    </Declaration>
    <Declaration>
```

```
        <NamedIndividual IRI="#OWLNamedIndividual_f56f88d6_07be_4ca1_8ba9_38a3f85151ed"/>
    </Declaration>
    <Declaration>
        <NamedIndividual IRI="#OWLNamedIndividual_fda53d39_9eea_409f_88b8_b4e45b591b39"/>
    </Declaration>
    <Declaration>
        <NamedIndividual IRI="#OWLNamedIndividual_ff873cd4_6496_44a4_8b48_0a01fd936729"/>
    </Declaration>
    <Declaration>
        <NamedIndividual IRI="#is_part_of"/>
    </Declaration>
    <EquivalentClasses>
        <Class IRI="#Cause"/>
        <ObjectIntersectionOf>
            <Class IRI="#Term"/>
            <ObjectSomeValuesFrom>
                <ObjectProperty IRI="#isCauseOf"/>
                <Class IRI="#Term"/>
            </ObjectSomeValuesFrom>
        </ObjectIntersectionOf>
    </EquivalentClasses>
    <EquivalentClasses>
        <Class IRI="#Container"/>
        <ObjectIntersectionOf>
            <Class IRI="#Term"/>
            <ObjectSomeValuesFrom>
                <ObjectProperty IRI="#OWLObjectProperty_e66f24b5_f9c5_49f8_8f31_9105044711eb"/>
                <Class IRI="#Term"/>
            </ObjectSomeValuesFrom>
        </ObjectIntersectionOf>
    </EquivalentClasses>
    <EquivalentClasses>
        <Class IRI="#Content"/>
        <ObjectIntersectionOf>
            <Class IRI="#Term"/>
            <ObjectSomeValuesFrom>
                <ObjectProperty IRI="#OWLObjectProperty_51251faa_5158_44d7_beba_6b1eb77ad196"/>
                <Class IRI="#Term"/>
            </ObjectSomeValuesFrom>
        </ObjectIntersectionOf>
    </EquivalentClasses>
    <EquivalentClasses>
        <Class IRI="#Effect"/>
        <ObjectIntersectionOf>
```

```
            <Class IRI="#Term"/>
            <ObjectSomeValuesFrom>
                <ObjectProperty IRI="#isCausedBy"/>
                <Class IRI="#Term"/>
            </ObjectSomeValuesFrom>
        </ObjectIntersectionOf>
    </EquivalentClasses>
    <EquivalentClasses>
        <Class IRI="#Ends"/>
        <ObjectIntersectionOf>
            <Class IRI="#Term"/>
            <ObjectSomeValuesFrom>
                <ObjectProperty IRI="#uses"/>
                <Class IRI="#Term"/>
            </ObjectSomeValuesFrom>
        </ObjectIntersectionOf>
    </EquivalentClasses>
    <EquivalentClasses>
        <Class IRI="#Generic"/>
        <ObjectIntersectionOf>
            <Class IRI="#Term"/>
            <ObjectSomeValuesFrom>
                <ObjectProperty IRI="#isTypeOf"/>
                <Class IRI="#Term"/>
            </ObjectSomeValuesFrom>
        </ObjectIntersectionOf>
    </EquivalentClasses>
    <EquivalentClasses>
        <Class IRI="#Local"/>
        <ObjectIntersectionOf>
            <Class IRI="#Term"/>
            <ObjectSomeValuesFrom>
                <ObjectProperty IRI="#isLocationOf"/>
                <Class IRI="#Term"/>
            </ObjectSomeValuesFrom>
        </ObjectIntersectionOf>
    </EquivalentClasses>
    <EquivalentClasses>
        <Class IRI="#Means"/>
        <ObjectIntersectionOf>
            <Class IRI="#Term"/>
            <ObjectSomeValuesFrom>
                <ObjectProperty IRI="#isUsedBy"/>
                <Class IRI="#Term"/>
            </ObjectSomeValuesFrom>
        </ObjectIntersectionOf>
    </EquivalentClasses>
    <EquivalentClasses>
        <Class IRI="#Part"/>
```

```
        <ObjectIntersectionOf>
            <Class IRI="#Term"/>
            <ObjectSomeValuesFrom>
                <ObjectProperty IRI="#isPartOf"/>
                <Class IRI="#Term"/>
            </ObjectSomeValuesFrom>
        </ObjectIntersectionOf>
    </EquivalentClasses>
    <EquivalentClasses>
        <Class IRI="#Specific"/>
        <ObjectIntersectionOf>
            <Class IRI="#Term"/>
            <ObjectSomeValuesFrom>
                <ObjectProperty IRI="#hasType"/>
                <Class IRI="#Term"/>
            </ObjectSomeValuesFrom>
        </ObjectIntersectionOf>
    </EquivalentClasses>
    <EquivalentClasses>
        <Class IRI="#Whole"/>
        <ObjectIntersectionOf>
            <Class IRI="#Term"/>
            <ObjectSomeValuesFrom>
                <ObjectProperty IRI="#hasPart"/>
                <Class IRI="#Term"/>
            </ObjectSomeValuesFrom>
        </ObjectIntersectionOf>
    </EquivalentClasses>
    <SubClassOf>
        <Class IRI="#AmbiguousRelations"/>
        <Class IRI="#ConceptualRelation"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Any"/>
        <Class IRI="#Term"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Cause"/>
        <Class IRI="#Term"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Cause Effect"/>
        <Class IRI="#ConceptualRelation"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#CauseEffect_Template"/>
        <Class IRI="#Template"/>
    </SubClassOf>
    <SubClassOf>
```

```
        <Class IRI="#CauseEffect_Template"/>
        <ObjectAllValuesFrom>
            <ObjectProperty IRI="#hasConceptualRelation"/>
            <Class IRI="#Cause Effect"/>
        </ObjectAllValuesFrom>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#CauseEffect_Template"/>
        <ObjectAllValuesFrom>
            <ObjectProperty IRI="#hasTerm"/>
            <ObjectUnionOf>
                <Class IRI="#Cause"/>
                <Class IRI="#Effect"/>
            </ObjectUnionOf>
        </ObjectAllValuesFrom>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#CauseEffect_Template"/>
        <ObjectMaxCardinality cardinality="1">
            <ObjectProperty IRI="#hasTerm"/>
            <Class IRI="#Cause"/>
        </ObjectMaxCardinality>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#CauseEffect_Template"/>
        <ObjectMaxCardinality cardinality="1">
            <ObjectProperty IRI="#hasTerm"/>
            <Class IRI="#Effect"/>
        </ObjectMaxCardinality>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#ConceptualRelation"/>
        <ObjectAllValuesFrom>
            <ObjectProperty IRI="#isDesignatedBy"/>
            <Class IRI="#LinkingPhrase"/>
        </ObjectAllValuesFrom>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#ConceptualRelation"/>
        <DataSomeValuesFrom>
            <DataProperty IRI="#hasCompetencyQuestion"/>
            <Datatype abbreviatedIRI="xsd:string"/>
        </DataSomeValuesFrom>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#ConceptualRelation"/>
        <DataExactCardinality cardinality="1">
            <DataProperty IRI="#hasIntent"/>
            <Datatype abbreviatedIRI="xsd:string"/>
```

```
        </DataExactCardinality>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Constituition_and_Containment"/>
        <Class IRI="#ConceptualRelation"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Container"/>
        <Class IRI="#Term"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Containment"/>
        <Class IRI="#Constituition_and_Containment"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Containment_Template"/>
        <Class IRI="#Template"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Containment_Template"/>
        <ObjectAllValuesFrom>
            <ObjectProperty IRI="#hasConceptualRelation"/>
            <Class IRI="#Containment"/>
        </ObjectAllValuesFrom>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Containment_Template"/>
        <ObjectAllValuesFrom>
            <ObjectProperty IRI="#hasTerm"/>
            <ObjectUnionOf>
                <Class IRI="#Part"/>
                <Class IRI="#Whole"/>
            </ObjectUnionOf>
        </ObjectAllValuesFrom>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Containment_Template"/>
        <ObjectMaxCardinality cardinality="1">
            <ObjectProperty IRI="#hasTerm"/>
            <Class IRI="#Part"/>
        </ObjectMaxCardinality>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Containment_Template"/>
        <ObjectMaxCardinality cardinality="1">
            <ObjectProperty IRI="#hasTerm"/>
            <Class IRI="#Whole"/>
        </ObjectMaxCardinality>
    </SubClassOf>
```

```
    <SubClassOf>
        <Class IRI="#Content"/>
        <Class IRI="#Term"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Effect"/>
        <Class IRI="#Term"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Ends"/>
        <Class IRI="#Term"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Generic"/>
        <Class IRI="#Term"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Generic Dependence"/>
        <Class IRI="#ConceptualRelation"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Generic Specific"/>
        <Class IRI="#Generic Dependence"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#GenericSpecific_Template"/>
        <Class IRI="#Template"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#GenericSpecific_Template"/>
        <ObjectAllValuesFrom>
            <ObjectProperty IRI="#hasConceptualRelation"/>
            <Class IRI="#Generic Specific"/>
        </ObjectAllValuesFrom>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#GenericSpecific_Template"/>
        <ObjectAllValuesFrom>
            <ObjectProperty IRI="#hasTerm"/>
            <ObjectUnionOf>
                <Class IRI="#Generic"/>
                <Class IRI="#Specific"/>
            </ObjectUnionOf>
        </ObjectAllValuesFrom>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#GenericSpecific_Template"/>
        <ObjectMaxCardinality cardinality="1">
            <ObjectProperty IRI="#hasTerm"/>
```

```
            <Class IRI="#Generic"/>
        </ObjectMaxCardinality>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#GenericSpecific_Template"/>
        <ObjectMaxCardinality cardinality="1">
            <ObjectProperty IRI="#hasTerm"/>
            <Class IRI="#Specific"/>
        </ObjectMaxCardinality>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#LinkingPhrase"/>
        <ObjectSomeValuesFrom>
            <ObjectProperty IRI="#designates"/>
            <Class IRI="#ConceptualRelation"/>
        </ObjectSomeValuesFrom>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#LinkingPhrase"/>
        <ObjectAllValuesFrom>
            <ObjectProperty IRI="#hasSource"/>
            <Class IRI="#Term"/>
        </ObjectAllValuesFrom>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#LinkingPhrase"/>
        <ObjectAllValuesFrom>
            <ObjectProperty IRI="#hasTarget"/>
            <Class IRI="#Term"/>
        </ObjectAllValuesFrom>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Local"/>
        <Class IRI="#Term"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Means"/>
        <Class IRI="#Term"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Part"/>
        <Class IRI="#Term"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#PartHood"/>
        <Class IRI="#Constituition_and_Containment"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#PartHood_Template"/>
```

```
        <Class IRI="#Template"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#PartHood_Template"/>
        <ObjectAllValuesFrom>
            <ObjectProperty IRI="#hasConceptualRelation"/>
            <Class IRI="#PartHood"/>
        </ObjectAllValuesFrom>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#PartHood_Template"/>
        <ObjectAllValuesFrom>
            <ObjectProperty IRI="#hasTerm"/>
            <ObjectUnionOf>
                <Class IRI="#Part"/>
                <Class IRI="#Whole"/>
            </ObjectUnionOf>
        </ObjectAllValuesFrom>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#PartHood_Template"/>
        <ObjectMaxCardinality cardinality="1">
            <ObjectProperty IRI="#hasTerm"/>
            <Class IRI="#Part"/>
        </ObjectMaxCardinality>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#PartHood_Template"/>
        <ObjectMaxCardinality cardinality="1">
            <ObjectProperty IRI="#hasTerm"/>
            <Class IRI="#Whole"/>
        </ObjectMaxCardinality>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Participation"/>
        <Class IRI="#ConceptualRelation"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Participation_Template"/>
        <Class IRI="#Template"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Spatial"/>
        <Class IRI="#Time Space_Dependence"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Spatial_Template"/>
        <Class IRI="#Template"/>
    </SubClassOf>
```

```
    <SubClassOf>
        <Class IRI="#Spatial_Template"/>
        <ObjectSomeValuesFrom>
            <ObjectProperty IRI="#hasTerm"/>
            <Class IRI="#Local"/>
        </ObjectSomeValuesFrom>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Spatial_Template"/>
        <ObjectAllValuesFrom>
            <ObjectProperty IRI="#hasConceptualRelation"/>
            <Class IRI="#Spatial"/>
        </ObjectAllValuesFrom>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Spatial_Template"/>
        <ObjectMaxCardinality cardinality="2">
            <ObjectProperty IRI="#hasTerm"/>
            <Class IRI="#Term"/>
        </ObjectMaxCardinality>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Template"/>
        <ObjectAllValuesFrom>
            <ObjectProperty IRI="#hasConceptualRelation"/>
            <ObjectIntersectionOf>
                <Class IRI="#ConceptualRelation"/>
                <Class IRI="#defaultLinkingPhrase"/>
            </ObjectIntersectionOf>
        </ObjectAllValuesFrom>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Template"/>
        <ObjectAllValuesFrom>
            <ObjectProperty IRI="#hasTerm"/>
            <Class IRI="#Term"/>
        </ObjectAllValuesFrom>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Temporal"/>
        <Class IRI="#Time Space_Dependence"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Temporal_Template"/>
        <Class IRI="#Template"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Time Space_Dependence"/>
        <Class IRI="#ConceptualRelation"/>
```

```
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Usage"/>
        <Class IRI="#ConceptualRelation"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Usage_Template"/>
        <Class IRI="#Template"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Usage_Template"/>
        <ObjectAllValuesFrom>
            <ObjectProperty IRI="#hasConceptualRelation"/>
            <Class IRI="#Usage"/>
        </ObjectAllValuesFrom>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Usage_Template"/>
        <ObjectAllValuesFrom>
            <ObjectProperty IRI="#hasTerm"/>
            <ObjectUnionOf>
                <Class IRI="#Ends"/>
                <Class IRI="#Means"/>
            </ObjectUnionOf>
        </ObjectAllValuesFrom>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Usage_Template"/>
        <ObjectMaxCardinality cardinality="1">
            <ObjectProperty IRI="#hasTerm"/>
            <Class IRI="#Ends"/>
        </ObjectMaxCardinality>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Usage_Template"/>
        <ObjectMaxCardinality cardinality="1">
            <ObjectProperty IRI="#hasTerm"/>
            <Class IRI="#Means"/>
        </ObjectMaxCardinality>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Whole"/>
        <Class IRI="#Term"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#defaultLinkingPhrase"/>
        <Class IRI="#LinkingPhrase"/>
    </SubClassOf>
    <DisjointClasses>
```

```
        <Class IRI="#Cause Effect"/>
        <Class IRI="#Containment"/>
        <Class IRI="#Generic Specific"/>
        <Class IRI="#PartHood"/>
        <Class IRI="#Participation"/>
        <Class IRI="#Spatial"/>
        <Class IRI="#Temporal"/>
        <Class IRI="#Usage"/>
    </DisjointClasses>
    <DisjointClasses>
        <Class IRI="#CauseEffect_Template"/>
        <Class IRI="#Containment_Template"/>
        <Class IRI="#GenericSpecific_Template"/>
        <Class IRI="#PartHood_Template"/>
        <Class IRI="#Participation_Template"/>
        <Class IRI="#Spatial_Template"/>
        <Class IRI="#Temporal_Template"/>
        <Class IRI="#Usage_Template"/>
    </DisjointClasses>
    <DisjointClasses>
        <Class IRI="#CauseEffect_Template"/>
        <Class IRI="#GenericSpecific_Template"/>
    </DisjointClasses>
    <DisjointClasses>
        <Class IRI="#Constituition_and_Containment"/>
        <Class IRI="#Generic Dependence"/>
    </DisjointClasses>
    <DisjointClasses>
        <Class IRI="#LinkingPhrase"/>
        <Class IRI="#Term"/>
    </DisjointClasses>
    <ClassAssertion>
        <Class IRI="#PartHood"/>
        <NamedIndividual IRI="#is_part_of"/>
    </ClassAssertion>
    <ClassAssertion>
        <Class IRI="#defaultLinkingPhrase"/>
        <NamedIndividual IRI="#is_part_of"/>
    </ClassAssertion>
    <DataPropertyAssertion>
        <DataProperty IRI="#hasCompetencyQuestion"/>
        <NamedIndividual IRI="#is_part_of"/>
        <Literal datatypeIRI="&xsd;string">Are terms a component/constituent or attached to each other?</Literal>
    </DataPropertyAssertion>
    <DataPropertyAssertion>
        <DataProperty IRI="#hasCompetencyQuestion"/>
        <NamedIndividual IRI="#is_part_of"/>
```

```
        <Literal datatypeIRI="&xsd;string">Are the terms physically engaged?</Literal>
    </DataPropertyAssertion>
    <DataPropertyAssertion>
        <DataProperty IRI="#hasCompetencyQuestion"/>
        <NamedIndividual IRI="#is_part_of"/>
        <Literal datatypeIRI="&xsd;string">Are terms nested?</Literal>
    </DataPropertyAssertion>
    <DataPropertyAssertion>
        <DataProperty IRI="#hasIntent"/>
        <NamedIndividual IRI="#is_part_of"/>
        <Literal datatypeIRI="&xsd;string">Used to represent relations between concepts in which a concept has another concept as its constituent forming a whole, which could be dependent or independent from its parts.</Literal>
    </DataPropertyAssertion>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#OWLObjectProperty_154bc8e4_72d7_48d2_a89a_9141494a8914"/>
        <ObjectProperty IRI="#relatesTo"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#OWLObjectProperty_198c01e3_d615_4f5f_b2bf_cc80c55c100b"/>
        <ObjectProperty IRI="#relatesTo"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#OWLObjectProperty_4c4b67c7_35ec_4fec_ab9b_eb4293d4f30d"/>
        <ObjectProperty IRI="#relatesTo"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#OWLObjectProperty_51251faa_5158_44d7_beba_6b1eb77ad196"/>
        <ObjectProperty IRI="#relatesTo"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#OWLObjectProperty_96c9f883_5dff_4bf9_a96b_4d1f5ccb258b"/>
        <ObjectProperty IRI="#relatesTo"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#OWLObjectProperty_e66f24b5_f9c5_49f8_8f31_9105044711eb"/>
        <ObjectProperty IRI="#relatesTo"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#follows"/>
        <ObjectProperty IRI="#relatesTo"/>
```

```
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#hasPart"/>
        <ObjectProperty IRI="#relatesTo"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#hasType"/>
        <ObjectProperty IRI="#relatesTo"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#isCauseOf"/>
        <ObjectProperty IRI="#relatesTo"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#isCausedBy"/>
        <ObjectProperty IRI="#relatesTo"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#isLocatedIn"/>
        <ObjectProperty IRI="#relatesTo"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#isLocationOf"/>
        <ObjectProperty IRI="#relatesTo"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#isPartOf"/>
        <ObjectProperty IRI="#relatesTo"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#isTypeOf"/>
        <ObjectProperty IRI="#relatesTo"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#isUsedBy"/>
        <ObjectProperty IRI="#relatesTo"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#precedes"/>
        <ObjectProperty IRI="#relatesTo"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#uses"/>
        <ObjectProperty IRI="#relatesTo"/>
    </SubObjectPropertyOf>
    <InverseObjectProperties>
        <ObjectProperty IRI="#
            OWLObjectProperty_154bc8e4_72d7_48d2_a89a_9141494a8914"/>
```

```
        <ObjectProperty IRI="#OWLObjectProperty_198c01e3_d615_4f5f_b2bf_cc80c55c100b"/>
    </InverseObjectProperties>
    <InverseObjectProperties>
        <ObjectProperty IRI="#OWLObjectProperty_96c9f883_5dff_4bf9_a96b_4d1f5ccb258b"/>
        <ObjectProperty IRI="#OWLObjectProperty_4c4b67c7_35ec_4fec_ab9b_eb4293d4f30d"/>
    </InverseObjectProperties>
    <InverseObjectProperties>
        <ObjectProperty IRI="#OWLObjectProperty_e66f24b5_f9c5_49f8_8f31_9105044711eb"/>
        <ObjectProperty IRI="#OWLObjectProperty_51251faa_5158_44d7_beba_6b1eb77ad196"/>
    </InverseObjectProperties>
    <InverseObjectProperties>
        <ObjectProperty IRI="#isDesignatedBy"/>
        <ObjectProperty IRI="#designates"/>
    </InverseObjectProperties>
    <InverseObjectProperties>
        <ObjectProperty IRI="#precedes"/>
        <ObjectProperty IRI="#follows"/>
    </InverseObjectProperties>
    <InverseObjectProperties>
        <ObjectProperty IRI="#hasPart"/>
        <ObjectProperty IRI="#isPartOf"/>
    </InverseObjectProperties>
    <InverseObjectProperties>
        <ObjectProperty IRI="#isTypeOf"/>
        <ObjectProperty IRI="#hasType"/>
    </InverseObjectProperties>
    <InverseObjectProperties>
        <ObjectProperty IRI="#isCausedBy"/>
        <ObjectProperty IRI="#isCauseOf"/>
    </InverseObjectProperties>
    <InverseObjectProperties>
        <ObjectProperty IRI="#isLocatedIn"/>
        <ObjectProperty IRI="#isLocationOf"/>
    </InverseObjectProperties>
    <InverseObjectProperties>
        <ObjectProperty IRI="#isUsedBy"/>
        <ObjectProperty IRI="#uses"/>
    </InverseObjectProperties>
    <FunctionalObjectProperty>
        <ObjectProperty IRI="#isCauseOf"/>
    </FunctionalObjectProperty>
    <FunctionalObjectProperty>
        <ObjectProperty IRI="#isCausedBy"/>
    </FunctionalObjectProperty>
```

```
    <SymmetricObjectProperty>
        <ObjectProperty IRI="#relatesTo"/>
    </SymmetricObjectProperty>
    <TransitiveObjectProperty>
        <ObjectProperty IRI="#follows"/>
    </TransitiveObjectProperty>
    <TransitiveObjectProperty>
        <ObjectProperty IRI="#hasPart"/>
    </TransitiveObjectProperty>
    <TransitiveObjectProperty>
        <ObjectProperty IRI="#hasType"/>
    </TransitiveObjectProperty>
    <TransitiveObjectProperty>
        <ObjectProperty IRI="#isPartOf"/>
    </TransitiveObjectProperty>
    <TransitiveObjectProperty>
        <ObjectProperty IRI="#isTypeOf"/>
    </TransitiveObjectProperty>
    <TransitiveObjectProperty>
        <ObjectProperty IRI="#precedes"/>
    </TransitiveObjectProperty>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#designates"/>
        <Class IRI="#LinkingPhrase"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#follows"/>
        <Class IRI="#Term"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#hasConceptualRelation"/>
        <Class IRI="#Template"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#hasPart"/>
        <Class IRI="#Term"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#hasSource"/>
        <Class IRI="#ConceptualRelation"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#hasSource"/>
        <Class IRI="#LinkingPhrase"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#hasTarget"/>
        <Class IRI="#ConceptualRelation"/>
    </ObjectPropertyDomain>
```

```
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#hasTarget"/>
        <Class IRI="#LinkingPhrase"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#hasTemplate"/>
        <Class IRI="#ConceptualRelation"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#hasTerm"/>
        <Class IRI="#Template"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#hasType"/>
        <Class IRI="#Term"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#incompatibleTypes"/>
        <Class IRI="#Term"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#isCauseOf"/>
        <Class IRI="#Term"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#isCausedBy"/>
        <Class IRI="#Term"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#isLocatedIn"/>
        <Class IRI="#Term"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#isLocationOf"/>
        <Class IRI="#Term"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#isPartOf"/>
        <Class IRI="#Term"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#isTypeOf"/>
        <Class IRI="#Term"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#isUsedBy"/>
        <Class IRI="#Term"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
```

```
        <ObjectProperty IRI="#precedes"/>
        <Class IRI="#Term"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#relatesTo"/>
        <Class IRI="#Term"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#uses"/>
        <Class IRI="#Term"/>
    </ObjectPropertyDomain>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#designates"/>
        <Class IRI="#ConceptualRelation"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#follows"/>
        <Class IRI="#Term"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#hasConceptualRelation"/>
        <Class IRI="#Term"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#hasPart"/>
        <Class IRI="#Term"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#hasSource"/>
        <Class IRI="#Term"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#hasTarget"/>
        <Class IRI="#Term"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#hasTemplate"/>
        <Class IRI="#Template"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#hasTerm"/>
        <Class IRI="#Term"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#hasType"/>
        <Class IRI="#Term"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#incompatibleTypes"/>
```

```
        <Class IRI="#Term"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#isCauseOf"/>
        <Class IRI="#Term"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#isCausedBy"/>
        <Class IRI="#Term"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#isLocatedIn"/>
        <Class IRI="#Term"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#isLocationOf"/>
        <Class IRI="#Term"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#isPartOf"/>
        <Class IRI="#Term"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#isTypeOf"/>
        <Class IRI="#Term"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#isUsedBy"/>
        <Class IRI="#Term"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#precedes"/>
        <Class IRI="#Term"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#relatesTo"/>
        <Class IRI="#Term"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#uses"/>
        <Class IRI="#Term"/>
    </ObjectPropertyRange>
    <DataPropertyDomain>
        <DataProperty IRI="#hasCompetencyQuestion"/>
        <Class IRI="#ConceptualRelation"/>
    </DataPropertyDomain>
    <DataPropertyDomain>
        <DataProperty IRI="#hasIntent"/>
        <Class IRI="#ConceptualRelation"/>
```

```
    </DataPropertyDomain>
    <DataPropertyRange>
        <DataProperty IRI="#hasCompetencyQuestion"/>
        <Datatype abbreviatedIRI="xsd:string"/>
    </DataPropertyRange>
    <DataPropertyRange>
        <DataProperty IRI="#hasIntent"/>
        <Datatype abbreviatedIRI="xsd:string"/>
    </DataPropertyRange>
    <DLSafeRule>
        <Body>
            <ClassAtom>
                <Class IRI="#Cause Effect"/>
                <Variable IRI="urn:swrl#x"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#LinkingPhrase"/>
                <Variable IRI="urn:swrl#x"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#a"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#b"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#c"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#d"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#e"/>
            </ClassAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#follows"/>
                <Variable IRI="urn:swrl#e"/>
                <Variable IRI="urn:swrl#a"/>
            </ObjectPropertyAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#hasSource"/>
                <Variable IRI="urn:swrl#x"/>
                <Variable IRI="urn:swrl#a"/>
            </ObjectPropertyAtom>
```

```
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#hasTarget"/>
                <Variable IRI="urn:swrl#x"/>
                <Variable IRI="urn:swrl#b"/>
            </ObjectPropertyAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#isLocatedIn"/>
                <Variable IRI="urn:swrl#a"/>
                <Variable IRI="urn:swrl#c"/>
            </ObjectPropertyAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#isLocatedIn"/>
                <Variable IRI="urn:swrl#b"/>
                <Variable IRI="urn:swrl#d"/>
            </ObjectPropertyAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#precedes"/>
                <Variable IRI="urn:swrl#e"/>
                <Variable IRI="urn:swrl#b"/>
            </ObjectPropertyAtom>
        </Body>
        <Head>
            <ClassAtom>
                <Class IRI="#AmbiguousRelations"/>
                <Variable IRI="urn:swrl#x"/>
            </ClassAtom>
        </Head>
    </DLSafeRule>
    <DLSafeRule>
        <Body>
            <ClassAtom>
                <Class IRI="#Cause Effect"/>
                <Variable IRI="urn:swrl#x"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#LinkingPhrase"/>
                <Variable IRI="urn:swrl#x"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#y"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#z"/>
            </ClassAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#hasSource"/>
                <Variable IRI="urn:swrl#x"/>
```

```
                <Variable IRI="urn:swrl#y"/>
            </ObjectPropertyAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#hasTarget"/>
                <Variable IRI="urn:swrl#x"/>
                <Variable IRI="urn:swrl#z"/>
            </ObjectPropertyAtom>
        </Body>
        <Head>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#isCauseOf"/>
                <Variable IRI="urn:swrl#y"/>
                <Variable IRI="urn:swrl#z"/>
            </ObjectPropertyAtom>
        </Head>
    </DLSafeRule>
    <DLSafeRule>
        <Body>
            <ClassAtom>
                <Class IRI="#Cause Effect"/>
                <Variable IRI="urn:swrl#y"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#LinkingPhrase"/>
                <Variable IRI="urn:swrl#x"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#LinkingPhrase"/>
                <Variable IRI="urn:swrl#y"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#a"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#b"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#c"/>
            </ClassAtom>
            <ClassAtom>
                <ObjectUnionOf>
                    <Class IRI="#Cause Effect"/>
                    <Class IRI="#Generic Specific"/>
                    <Class IRI="#PartHood"/>
                    <Class IRI="#Participation"/>
                    <Class IRI="#Temporal"/>
```

```
                </ObjectUnionOf>
                <Variable IRI="urn:swrl#x"/>
            </ClassAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#hasSource"/>
                <Variable IRI="urn:swrl#x"/>
                <Variable IRI="urn:swrl#a"/>
            </ObjectPropertyAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#hasSource"/>
                <Variable IRI="urn:swrl#y"/>
                <Variable IRI="urn:swrl#b"/>
            </ObjectPropertyAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#hasTarget"/>
                <Variable IRI="urn:swrl#x"/>
                <Variable IRI="urn:swrl#b"/>
            </ObjectPropertyAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#hasTarget"/>
                <Variable IRI="urn:swrl#y"/>
                <Variable IRI="urn:swrl#c"/>
            </ObjectPropertyAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#relatesTo"/>
                <Variable IRI="urn:swrl#c"/>
                <Variable IRI="urn:swrl#a"/>
            </ObjectPropertyAtom>
        </Body>
        <Head>
            <ClassAtom>
                <Class IRI="#AmbiguousRelations"/>
                <Variable IRI="urn:swrl#y"/>
            </ClassAtom>
        </Head>
    </DLSafeRule>
    <DLSafeRule>
        <Body>
            <ClassAtom>
                <Class IRI="#Cause Effect"/>
                <Variable IRI="urn:swrl#y"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#LinkingPhrase"/>
                <Variable IRI="urn:swrl#x"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#LinkingPhrase"/>
                <Variable IRI="urn:swrl#y"/>
```

```
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#a"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#b"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#c"/>
            </ClassAtom>
            <ClassAtom>
                <ObjectComplementOf>
                    <Class IRI="#Cause Effect"/>
                </ObjectComplementOf>
                <Variable IRI="urn:swrl#x"/>
            </ClassAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#hasSource"/>
                <Variable IRI="urn:swrl#x"/>
                <Variable IRI="urn:swrl#b"/>
            </ObjectPropertyAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#hasSource"/>
                <Variable IRI="urn:swrl#y"/>
                <Variable IRI="urn:swrl#a"/>
            </ObjectPropertyAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#hasTarget"/>
                <Variable IRI="urn:swrl#x"/>
                <Variable IRI="urn:swrl#c"/>
            </ObjectPropertyAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#hasTarget"/>
                <Variable IRI="urn:swrl#y"/>
                <Variable IRI="urn:swrl#b"/>
            </ObjectPropertyAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#isCauseOf"/>
                <Variable IRI="urn:swrl#a"/>
                <Variable IRI="urn:swrl#b"/>
            </ObjectPropertyAtom>
        </Body>
        <Head>
            <ClassAtom>
                <Class IRI="#AmbiguousRelations"/>
                <Variable IRI="urn:swrl#y"/>
```

```
                </ClassAtom>
            </Head>
        </DLSafeRule>
        <DLSafeRule>
            <Body>
                <ClassAtom>
                    <Class IRI="#Cause Effect"/>
                    <Variable IRI="urn:swrl#y"/>
                </ClassAtom>
                <ClassAtom>
                    <Class IRI="#LinkingPhrase"/>
                    <Variable IRI="urn:swrl#x"/>
                </ClassAtom>
                <ClassAtom>
                    <Class IRI="#LinkingPhrase"/>
                    <Variable IRI="urn:swrl#y"/>
                </ClassAtom>
                <ClassAtom>
                    <Class IRI="#Term"/>
                    <Variable IRI="urn:swrl#a"/>
                </ClassAtom>
                <ClassAtom>
                    <Class IRI="#Term"/>
                    <Variable IRI="urn:swrl#b"/>
                </ClassAtom>
                <ClassAtom>
                    <ObjectUnionOf>
                        <Class IRI="#Constituition_and_Containment"/>
                        <Class IRI="#Generic Dependence"/>
                        <Class IRI="#Participation"/>
                        <Class IRI="#Spatial"/>
                    </ObjectUnionOf>
                    <Variable IRI="urn:swrl#x"/>
                </ClassAtom>
                <ObjectPropertyAtom>
                    <ObjectProperty IRI="#hasSource"/>
                    <Variable IRI="urn:swrl#x"/>
                    <Variable IRI="urn:swrl#a"/>
                </ObjectPropertyAtom>
                <ObjectPropertyAtom>
                    <ObjectProperty IRI="#hasSource"/>
                    <Variable IRI="urn:swrl#y"/>
                    <Variable IRI="urn:swrl#a"/>
                </ObjectPropertyAtom>
                <ObjectPropertyAtom>
                    <ObjectProperty IRI="#hasTarget"/>
                    <Variable IRI="urn:swrl#x"/>
                    <Variable IRI="urn:swrl#b"/>
                </ObjectPropertyAtom>
```

```
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#hasTarget"/>
                <Variable IRI="urn:swrl#y"/>
                <Variable IRI="urn:swrl#b"/>
            </ObjectPropertyAtom>
        </Body>
        <Head>
            <ClassAtom>
                <Class IRI="#AmbiguousRelations"/>
                <Variable IRI="urn:swrl#y"/>
            </ClassAtom>
        </Head>
    </DLSafeRule>
    <DLSafeRule>
        <Body>
            <ClassAtom>
                <Class IRI="#Constituition_and_Containment"/>
                <Variable IRI="urn:swrl#x"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#LinkingPhrase"/>
                <Variable IRI="urn:swrl#x"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#PartHood"/>
                <Variable IRI="urn:swrl#x"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#y"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#z"/>
            </ClassAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#hasSource"/>
                <Variable IRI="urn:swrl#x"/>
                <Variable IRI="urn:swrl#y"/>
            </ObjectPropertyAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#hasTarget"/>
                <Variable IRI="urn:swrl#x"/>
                <Variable IRI="urn:swrl#z"/>
            </ObjectPropertyAtom>
        </Body>
        <Head>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#isPartOf"/>
```

```
                <Variable IRI="urn:swrl#y"/>
                <Variable IRI="urn:swrl#z"/>
            </ObjectPropertyAtom>
        </Head>
    </DLSafeRule>
    <DLSafeRule>
        <Body>
            <ClassAtom>
                <Class IRI="#Constituition_and_Containment"/>
                <Variable IRI="urn:swrl#x"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#LinkingPhrase"/>
                <Variable IRI="urn:swrl#x"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#y"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#z"/>
            </ClassAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#hasSource"/>
                <Variable IRI="urn:swrl#x"/>
                <Variable IRI="urn:swrl#y"/>
            </ObjectPropertyAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#hasSource"/>
                <Variable IRI="urn:swrl#x"/>
                <Variable IRI="urn:swrl#z"/>
            </ObjectPropertyAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#hasTarget"/>
                <Variable IRI="urn:swrl#x"/>
                <Variable IRI="urn:swrl#y"/>
            </ObjectPropertyAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#hasTarget"/>
                <Variable IRI="urn:swrl#x"/>
                <Variable IRI="urn:swrl#z"/>
            </ObjectPropertyAtom>
        </Body>
        <Head>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#incompatibleTypes"/>
                <Variable IRI="urn:swrl#y"/>
                <Variable IRI="urn:swrl#z"/>
```

```
            </ObjectPropertyAtom>
        </Head>
    </DLSafeRule>
    <DLSafeRule>
        <Body>
            <ClassAtom>
                <Class IRI="#Containment"/>
                <Variable IRI="urn:swrl#x"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#LinkingPhrase"/>
                <Variable IRI="urn:swrl#x"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#y"/>
            </ClassAtom>
            <ClassAtom>
                <ObjectMinCardinality cardinality="2">
                    <ObjectProperty IRI="#hasTarget"/>
                    <Class IRI="#Term"/>
                </ObjectMinCardinality>
                <Variable IRI="urn:swrl#x"/>
            </ClassAtom>
        </Body>
        <Head>
            <ClassAtom>
                <Class IRI="#AmbiguousRelations"/>
                <Variable IRI="urn:swrl#x"/>
            </ClassAtom>
        </Head>
    </DLSafeRule>
    <DLSafeRule>
        <Body>
            <ClassAtom>
                <Class IRI="#Generic Specific"/>
                <Variable IRI="urn:swrl#x"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#LinkingPhrase"/>
                <Variable IRI="urn:swrl#x"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#y"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#z"/>
```

```
            </ClassAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#hasSource"/>
                <Variable IRI="urn:swrl#x"/>
                <Variable IRI="urn:swrl#y"/>
            </ObjectPropertyAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#hasTarget"/>
                <Variable IRI="urn:swrl#x"/>
                <Variable IRI="urn:swrl#z"/>
            </ObjectPropertyAtom>
        </Body>
        <Head>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#isTypeOf"/>
                <Variable IRI="urn:swrl#y"/>
                <Variable IRI="urn:swrl#z"/>
            </ObjectPropertyAtom>
        </Head>
    </DLSafeRule>
    <DLSafeRule>
        <Body>
            <ClassAtom>
                <Class IRI="#LinkingPhrase"/>
                <Variable IRI="urn:swrl#x"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Spatial"/>
                <Variable IRI="urn:swrl#x"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#y"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#z"/>
            </ClassAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#hasSource"/>
                <Variable IRI="urn:swrl#x"/>
                <Variable IRI="urn:swrl#y"/>
            </ObjectPropertyAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#hasTarget"/>
                <Variable IRI="urn:swrl#x"/>
                <Variable IRI="urn:swrl#z"/>
            </ObjectPropertyAtom>
        </Body>
```

```
        <Head>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#isLocationOf"/>
                <Variable IRI="urn:swrl#z"/>
                <Variable IRI="urn:swrl#y"/>
            </ObjectPropertyAtom>
        </Head>
    </DLSafeRule>
    <DLSafeRule>
        <Body>
            <ClassAtom>
                <Class IRI="#LinkingPhrase"/>
                <Variable IRI="urn:swrl#x"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#a"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#b"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Usage"/>
                <Variable IRI="urn:swrl#x"/>
            </ClassAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#hasSource"/>
                <Variable IRI="urn:swrl#x"/>
                <Variable IRI="urn:swrl#a"/>
            </ObjectPropertyAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#hasTarget"/>
                <Variable IRI="urn:swrl#x"/>
                <Variable IRI="urn:swrl#b"/>
            </ObjectPropertyAtom>
        </Body>
        <Head>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#isUsedBy"/>
                <Variable IRI="urn:swrl#a"/>
                <Variable IRI="urn:swrl#b"/>
            </ObjectPropertyAtom>
        </Head>
    </DLSafeRule>
    <DLSafeRule>
        <Body>
            <ClassAtom>
                <Class IRI="#LinkingPhrase"/>
```

```
                <Variable IRI="urn:swrl#x"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#y"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#z"/>
            </ClassAtom>
            <ClassAtom>
                <ObjectComplementOf>
                    <Class IRI="#Generic Specific"/>
                </ObjectComplementOf>
                <Variable IRI="urn:swrl#x"/>
            </ClassAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#hasSource"/>
                <Variable IRI="urn:swrl#x"/>
                <Variable IRI="urn:swrl#y"/>
            </ObjectPropertyAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#hasTarget"/>
                <Variable IRI="urn:swrl#x"/>
                <Variable IRI="urn:swrl#z"/>
            </ObjectPropertyAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#isTypeOf"/>
                <Variable IRI="urn:swrl#y"/>
                <Variable IRI="urn:swrl#z"/>
            </ObjectPropertyAtom>
        </Body>
        <Head>
            <ClassAtom>
                <Class IRI="#AmbiguousRelations"/>
                <Variable IRI="urn:swrl#x"/>
            </ClassAtom>
        </Head>
    </DLSafeRule>
    <DLSafeRule>
        <Body>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#a"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#b"/>
            </ClassAtom>
```

```
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#OWLObjectProperty_154bc8e4_72d7_48d2_a89a_9141494a8914"/>
                <Variable IRI="urn:swrl#a"/>
                <Variable IRI="urn:swrl#b"/>
            </ObjectPropertyAtom>
        </Body>
        <Head>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#isUsedBy"/>
                <Variable IRI="urn:swrl#b"/>
                <Variable IRI="urn:swrl#a"/>
            </ObjectPropertyAtom>
        </Head>
    </DLSafeRule>
    <DLSafeRule>
        <Body>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#a"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#b"/>
            </ClassAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#OWLObjectProperty_4c4b67c7_35ec_4fec_ab9b_eb4293d4f30d"/>
                <Variable IRI="urn:swrl#a"/>
                <Variable IRI="urn:swrl#b"/>
            </ObjectPropertyAtom>
        </Body>
        <Head>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#isUsedBy"/>
                <Variable IRI="urn:swrl#b"/>
                <Variable IRI="urn:swrl#a"/>
            </ObjectPropertyAtom>
        </Head>
    </DLSafeRule>
    <DLSafeRule>
        <Body>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#a"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#b"/>
```

```
            </ClassAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#isCauseOf"/>
                <Variable IRI="urn:swrl#a"/>
                <Variable IRI="urn:swrl#b"/>
            </ObjectPropertyAtom>
        </Body>
        <Head>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#precedes"/>
                <Variable IRI="urn:swrl#a"/>
                <Variable IRI="urn:swrl#b"/>
            </ObjectPropertyAtom>
        </Head>
    </DLSafeRule>
    <DLSafeRule>
        <Body>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#a"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#b"/>
            </ClassAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#isPartOf"/>
                <Variable IRI="urn:swrl#a"/>
                <Variable IRI="urn:swrl#b"/>
            </ObjectPropertyAtom>
        </Body>
        <Head>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#precedes"/>
                <Variable IRI="urn:swrl#a"/>
                <Variable IRI="urn:swrl#b"/>
            </ObjectPropertyAtom>
        </Head>
    </DLSafeRule>
    <DLSafeRule>
        <Body>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#a"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#b"/>
            </ClassAtom>
```

```
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#isTypeOf"/>
                <Variable IRI="urn:swrl#a"/>
                <Variable IRI="urn:swrl#b"/>
            </ObjectPropertyAtom>
        </Body>
        <Head>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#precedes"/>
                <Variable IRI="urn:swrl#b"/>
                <Variable IRI="urn:swrl#a"/>
            </ObjectPropertyAtom>
        </Head>
    </DLSafeRule>
    <DLSafeRule>
        <Body>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#x"/>
            </ClassAtom>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#y"/>
            </ClassAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#isPartOf"/>
                <Variable IRI="urn:swrl#x"/>
                <Variable IRI="urn:swrl#y"/>
            </ObjectPropertyAtom>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#isPartOf"/>
                <Variable IRI="urn:swrl#y"/>
                <Variable IRI="urn:swrl#x"/>
            </ObjectPropertyAtom>
        </Body>
        <Head>
            <ObjectPropertyAtom>
                <ObjectProperty IRI="#incompatibleTypes"/>
                <Variable IRI="urn:swrl#y"/>
                <Variable IRI="urn:swrl#x"/>
            </ObjectPropertyAtom>
        </Head>
    </DLSafeRule>
    <DLSafeRule>
        <Body>
            <ClassAtom>
                <Class IRI="#Term"/>
                <Variable IRI="urn:swrl#x"/>
            </ClassAtom>
```

```
                <ClassAtom>
                    <Class IRI="#Term"/>
                    <Variable IRI="urn:swrl#y"/>
                </ClassAtom>
                <ObjectPropertyAtom>
                    <ObjectProperty IRI="#isTypeOf"/>
                    <Variable IRI="urn:swrl#x"/>
                    <Variable IRI="urn:swrl#y"/>
                </ObjectPropertyAtom>
                <ObjectPropertyAtom>
                    <ObjectProperty IRI="#isTypeOf"/>
                    <Variable IRI="urn:swrl#y"/>
                    <Variable IRI="urn:swrl#x"/>
                </ObjectPropertyAtom>
            </Body>
            <Head>
                <ObjectPropertyAtom>
                    <ObjectProperty IRI="#incompatibleTypes"/>
                    <Variable IRI="urn:swrl#y"/>
                    <Variable IRI="urn:swrl#x"/>
                </ObjectPropertyAtom>
            </Head>
        </DLSafeRule>
        <DLSafeRule>
            <Body>
                <ClassAtom>
                    <Class IRI="#Term"/>
                    <Variable IRI="urn:swrl#x"/>
                </ClassAtom>
                <ClassAtom>
                    <Class IRI="#Term"/>
                    <Variable IRI="urn:swrl#y"/>
                </ClassAtom>
                <ObjectPropertyAtom>
                    <ObjectProperty IRI="#isUsedBy"/>
                    <Variable IRI="urn:swrl#x"/>
                    <Variable IRI="urn:swrl#y"/>
                </ObjectPropertyAtom>
                <ObjectPropertyAtom>
                    <ObjectProperty IRI="#isUsedBy"/>
                    <Variable IRI="urn:swrl#y"/>
                    <Variable IRI="urn:swrl#x"/>
                </ObjectPropertyAtom>
            </Body>
            <Head>
                <ObjectPropertyAtom>
                    <ObjectProperty IRI="#incompatibleTypes"/>
                    <Variable IRI="urn:swrl#y"/>
                    <Variable IRI="urn:swrl#x"/>
```

```
                </ObjectPropertyAtom>
            </Head>
        </DLSafeRule>
        <AnnotationAssertion>
            <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
            <IRI>#AmbiguousRelations</IRI>
            <Literal datatypeIRI="&rdf;PlainLiteral">Defined as a containment relation, which cannot ensure one to one relationships btween parts and wholes</Literal>
        </AnnotationAssertion>
        <AnnotationAssertion>
            <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
            <IRI>#Containment</IRI>
            <Literal datatypeIRI="&rdf;PlainLiteral">Nested. One to one correspondence. Implies ownership</Literal>
        </AnnotationAssertion>
        <AnnotationAssertion>
            <AnnotationProperty abbreviatedIRI="rdfs:label"/>
            <IRI>#OWLNamedIndividual_058cd8c9_d267_4b33_9dcf_3171ba1aa410</IRI>
            <Literal xml:lang="en" datatypeIRI="&rdf;PlainLiteral">type of</Literal>
        </AnnotationAssertion>
        <AnnotationAssertion>
            <AnnotationProperty abbreviatedIRI="rdfs:label"/>
            <IRI>#OWLNamedIndividual_11d5abc7_b602_433f_9457_86a93b73c256</IRI>
            <Literal xml:lang="en" datatypeIRI="&rdf;PlainLiteral">type of</Literal>
        </AnnotationAssertion>
        <AnnotationAssertion>
            <AnnotationProperty abbreviatedIRI="rdfs:label"/>
            <IRI>#OWLNamedIndividual_266ef55e_f579_4481_98ea_61aa09ed0e61</IRI>
            <Literal xml:lang="en" datatypeIRI="&rdf;PlainLiteral">type of</Literal>
        </AnnotationAssertion>
        <AnnotationAssertion>
            <AnnotationProperty abbreviatedIRI="rdfs:label"/>
            <IRI>#OWLNamedIndividual_361e575c_0955_458f_89d9_d525d2accaa9</IRI>
            <Literal xml:lang="en" datatypeIRI="&rdf;PlainLiteral">type of</Literal>
        </AnnotationAssertion>
        <AnnotationAssertion>
            <AnnotationProperty abbreviatedIRI="rdfs:label"/>
            <IRI>#OWLNamedIndividual_368ad93e_19f7_40f3_bcf3_06677f41cc30</IRI>
            <Literal xml:lang="en" datatypeIRI="&rdf;PlainLiteral">type of</Literal>
        </AnnotationAssertion>
        <AnnotationAssertion>
            <AnnotationProperty abbreviatedIRI="rdfs:label"/>
            <IRI>#OWLNamedIndividual_573934a2_1786_4ec2_b42d_a5cc41ed0935</IRI>
```

```
        <Literal xml:lang="en" datatypeIRI="&rdf;PlainLiteral">type of</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#OWLNamedIndividual_6593e364_31d5_4157_9238_9fe5eafccd25</IRI>
        <Literal xml:lang="en" datatypeIRI="&rdf;PlainLiteral">type of</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#OWLNamedIndividual_6c4f1230_6123_4f7e_b23c_e41156e6ac8f</IRI>
        <Literal xml:lang="en" datatypeIRI="&rdf;PlainLiteral">used by</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#OWLNamedIndividual_979dc71e_d533_47d9_b96f_39cb499fcdff</IRI>
        <Literal xml:lang="en" datatypeIRI="&rdf;PlainLiteral">type of</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#OWLNamedIndividual_a8870049_ca0c_477a_ab0d_9d1bb581787b</IRI>
        <Literal xml:lang="en" datatypeIRI="&rdf;PlainLiteral">can solve</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#OWLNamedIndividual_abe3dc92_2f2f_46c3_903a_56f45f283c1a</IRI>
        <Literal xml:lang="en" datatypeIRI="&rdf;PlainLiteral">used by</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#OWLNamedIndividual_c2ab4650_8bc3_40e1_bd93_612e9d8551e7</IRI>
        <Literal xml:lang="en" datatypeIRI="&rdf;PlainLiteral">type of</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#OWLNamedIndividual_d0c7d18a_b27c_4de7_a3ef_b7c5fb847824</IRI>
        <Literal xml:lang="en" datatypeIRI="&rdf;PlainLiteral">used to solve</Literal>
    </AnnotationAssertion>
    <AnnotationAssertion>
        <AnnotationProperty abbreviatedIRI="rdfs:label"/>
        <IRI>#OWLNamedIndividual_d9013d84_84d1_4a8b_91d6_9c2840554593</IRI>
```

```
            <Literal xml:lang="en" datatypeIRI="&rdf;PlainLiteral">used by</Literal>
        </AnnotationAssertion>
        <AnnotationAssertion>
            <AnnotationProperty abbreviatedIRI="rdfs:label"/>
            <IRI>#OWLNamedIndividual_f0506f52_8089_4b1c_bd55_ffe69efc88f8</IRI>
            <Literal xml:lang="en" datatypeIRI="&rdf;PlainLiteral">part of</Literal>
        </AnnotationAssertion>
        <AnnotationAssertion>
            <AnnotationProperty abbreviatedIRI="rdfs:label"/>
            <IRI>#OWLNamedIndividual_f56f88d6_07be_4ca1_8ba9_38a3f85151ed</IRI>
            <Literal xml:lang="en" datatypeIRI="&rdf;PlainLiteral">type of</Literal>
        </AnnotationAssertion>
        <AnnotationAssertion>
            <AnnotationProperty abbreviatedIRI="rdfs:label"/>
            <IRI>#OWLNamedIndividual_fda53d39_9eea_409f_88b8_b4e45b591b39</IRI>
            <Literal xml:lang="en" datatypeIRI="&rdf;PlainLiteral">addressed by</Literal>
        </AnnotationAssertion>
        <AnnotationAssertion>
            <AnnotationProperty abbreviatedIRI="rdfs:label"/>
            <IRI>#OWLNamedIndividual_ff873cd4_6496_44a4_8b48_0a01fd936729</IRI>
            <Literal xml:lang="en" datatypeIRI="&rdf;PlainLiteral">type of</Literal>
        </AnnotationAssertion>
        <AnnotationAssertion>
            <AnnotationProperty abbreviatedIRI="rdfs:label"/>
            <IRI>#OWLObjectProperty_154bc8e4_72d7_48d2_a89a_9141494a8914</IRI>
            <Literal datatypeIRI="&rdf;PlainLiteral">concludedBy</Literal>
        </AnnotationAssertion>
        <AnnotationAssertion>
            <AnnotationProperty abbreviatedIRI="rdfs:label"/>
            <IRI>#OWLObjectProperty_198c01e3_d615_4f5f_b2bf_cc80c55c100b</IRI>
            <Literal datatypeIRI="&rdf;PlainLiteral">concludes</Literal>
        </AnnotationAssertion>
        <AnnotationAssertion>
            <AnnotationProperty abbreviatedIRI="rdfs:label"/>
            <IRI>#OWLObjectProperty_4c4b67c7_35ec_4fec_ab9b_eb4293d4f30d</IRI>
            <Literal datatypeIRI="&rdf;PlainLiteral">startedBy</Literal>
        </AnnotationAssertion>
        <AnnotationAssertion>
            <AnnotationProperty abbreviatedIRI="rdfs:label"/>
            <IRI>#OWLObjectProperty_51251faa_5158_44d7_beba_6b1eb77ad196</IRI>
            <Literal datatypeIRI="&rdf;PlainLiteral">isContainedBy</Literal>
        </AnnotationAssertion>
        <AnnotationAssertion>
            <AnnotationProperty abbreviatedIRI="rdfs:label"/>
```

```
            <IRI>#OWLObjectProperty_96c9f883_5dff_4bf9_a96b_4d1f5ccb258b</IRI>
            <Literal datatypeIRI="&rdf;PlainLiteral">starts</Literal>
        </AnnotationAssertion>
        <AnnotationAssertion>
            <AnnotationProperty abbreviatedIRI="rdfs:label"/>
            <IRI>#OWLObjectProperty_e66f24b5_f9c5_49f8_8f31_9105044711eb</IRI>
            <Literal datatypeIRI="&rdf;PlainLiteral">contains</Literal>
        </AnnotationAssertion>
        <AnnotationAssertion>
            <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
            <IRI>#Template</IRI>
            <Literal datatypeIRI="&xsd;string">The instances of a template are
                basic conceptual structures.</Literal>
        </AnnotationAssertion>
        <AnnotationAssertion>
            <AnnotationProperty abbreviatedIRI="rdfs:label"/>
            <IRI>#follows</IRI>
            <Literal datatypeIRI="&rdf;PlainLiteral">follows</Literal>
        </AnnotationAssertion>
        <AnnotationAssertion>
            <AnnotationProperty abbreviatedIRI="rdfs:label"/>
            <IRI>#precedes</IRI>
            <Literal datatypeIRI="&rdf;PlainLiteral">precedes</Literal>
        </AnnotationAssertion>
</Ontology>



<!  Generated by the OWL API (version 3.4.2) http://owlapi.sourceforge.net  >
```

...

Appendix B

FORSYS Exploratory Survey

FORSYS Exploratory Survey

SCOPE
Current survey was built in the scope of FORSYS case study, whose objective is to reach, collaboratively, a common understanding around the DSS concept.

GOAL
Competency assessment in information management and knowledge representation (KR) practices.

Introductory questions

1. **How comfortable are you with the topic of DSS in the context of FORSYS project?**
   Mark only one oval.

   - Not Comfortable
   - Somewhat Comfortable
   - Comfortable
   - Highly Comfortable

2. **Are you comfortable about the current shared view on the FMDSS concept?**
   Mark only one oval.

   - Not Comfortable
   - Somewhat Comfortable
   - Comfortable
   - Highly Comfortable

On information / knowledge management engagement

3. **During your daily tasks, do you find the need to sort, classify and code information/knowledge?**
   Mark only one oval.

   - Almost Never
   - Occasionally
   - Frequently
   - Highly Comfortable

4. **Do you participate in activities of classification and organisation of information?**
   Mark only one oval.

   - Almost Never
   - Occasionally
   - Frequently
   - Highly Comfortable

5. **How often are you using tools to research, identify and compile information for sharing purposes?**
Mark only one oval.
- Almost Never
- Occasionally
- Frequently
- Highly Comfortable

6. **In your collaborative activities, do you find the need to define policies to record and disseminate information?**
Mark only one oval.
- Almost Never
- Occasionally
- Frequently
- Highly Comfortable

7. **In your collaborative activities, do you recommend or implement strategies to information and knowledge management?**
Mark only one oval.
- Almost Never
- Occasionally
- Frequently
- Highly Comfortable

8. **In your collaborative activities, it is your concern to achieve a common understanding about the domain?**
Mark only one oval.
- Almost Never
- Occasionally
- Frequently
- Highly Comfortable

On knowledge representation engagement

9. **How often do you use textual statements to describe and expose your knowledge about a specific subject?**
Mark only one oval.
- Almost Never
- Occasionally
- Frequently
- Highly Comfortable

10. **How often do you use tables to describe and expose your knowledge about a specific subject?**
Mark only one oval.

- Almost Never
- Occasionally
- Frequently
- Highly Comfortable

11. **How often do you use indented lists to describe and expose your knowledge about a specific subject?**
Mark only one oval.

- Almost Never
- Occasionally
- Frequently
- Highly Comfortable

12. **How often do you use graphical representations to describe and expose your knowledge about a specific subject?**
Mark only one oval.

- Almost Never
- Occasionally
- Frequently
- Highly Comfortable

On knowledge representation engagement

13. **How often do you use simple building block diagrams to describe and expose your knowledge about a specific subject?**
Example of a building block diagram could be the Semantic Web Stack, available at http://en.wikipedia.org/wiki/Semantic_Web_Stack
Mark only one oval.

- Almost Never
- Occasionally
- Frequently
- Almost Always

14. **Do you consider that simple building block diagrams could be applied to describe and expose knowledge from any domain field?**
Mark only one oval.

- Almost Never
- Occasionally
- Frequently
- Almost Always

15. **Do you consider that simple building block diagrams could be a fast and/or systematic way to describe and expose knowledge from a specific domain field?**

Mark only one oval.

- Almost Never
- Occasionally
- Frequently
- Almost Always

16. **How often do you use flowchart to describe and expose your knowledge about a specific subject?**

Mark only one oval.

- Almost Never
- Occasionally
- Frequently
- Almost Always

17. **Do you consider that flowchart could be applied to describe and expose knowledge from any domain field?**

Mark only one oval.

- Almost Never
- Occasionally
- Frequently
- Almost Always

18. **Do you consider that flowchart could be a fast and/or systematic way to describe and expose knowledge from a specific domain field?**

Mark only one oval.

- Almost Never
- Occasionally
- Frequently
- Almost Always

19. **How often do you use concept maps to describe and expose your knowledge about a specific subject?**

Mark only one oval.

- Almost Never
- Occasionally
- Frequently
- Almost Always

20. **Do you consider that concept maps could be applied to describe and expose knowledge from any domain field?**

Mark only one oval.

- Almost Never
- Occasionally
- Frequently
- Almost Always

21. **Do you consider that concept maps could be a fast and/or systematic way to describe and expose knowledge from a specific domain field?**

Mark only one oval.

- Almost Never
- Occasionally
- Frequently
- Almost Always

22. **How often do you use mind maps to describe and expose your knowledge about a specific subject?**

Mark only one oval.

- Almost Never
- Occasionally
- Frequently
- Almost Always

23. **Do you consider that mind maps could be applied to describe and expose knowledge from any domain field?**

Mark only one oval.

- Almost Never
- Occasionally
- Frequently
- Almost Always

24. **Do you consider that mind maps could be a fast and/or systematic way to describe and expose knowledge from a specific domain field?**

Mark only one oval.

- Almost Never
- Occasionally
- Frequently
- Almost Always

On describing the contents of knowledge

When using graphical representations to express knowledge, how often do you care about:

25. **a) describing concepts/categories/classes?**
Mark only one oval.

- Almost Never
- Occasionally
- Frequently
- Almost Always

26. **b) describing relations among concepts/categories/classes?**
Mark only one oval.

- Almost Never
- Occasionally
- Frequently
- Almost Always

27. **c) describing attributes/properties?**
Mark only one oval.

- Almost Never
- Occasionally
- Frequently
- Almost Always

28. **d) describing constraints?**
Mark only one oval.

- Almost Never
- Occasionally
- Frequently
- Almost Always

29. **e) describing definitions?**
Mark only one oval.

- Almost Never
- Occasionally
- Frequently
- Almost Always

When using graphical representations to express knowledge,

how often, each of the following items do you consider a "bottleneck"?

hint: you can consider how easily is for you to come up with a proper designation or definition

30. **a) representing concepts/categories/classes**

Mark only one oval.

- Almost Never
- Occasionally
- Frequently
- Almost Always

31. **b) representing relations among concepts/categories/classes**

Mark only one oval.

- Almost Never
- Occasionally
- Frequently
- Almost Always

32. **c) representing attributes/propriedades**

Mark only one oval.

- Almost Never
- Occasionally
- Frequently
- Almost Always

33. **d) representing constraints**

Mark only one oval.

- Almost Never
- Occasionally
- Frequently
- Almost Always

34. **e) representing definitions**

Mark only one oval.

- Almost Never
- Occasionally
- Frequently
- Almost Always

Appendix C

Analysis of the "Has" Conceptual Relation for the FORSYS case study

ANALYSIS OF THE "HAS" CONCEPTUAL RELATION FOR THE FORSYS CASE STUDY

LIST OF CONCEPTUAL STRUCTURES WITH THE "HAS" RELATION			ANSWERS FROM THE DIFFERENT GROUPS INVOLVED			
SOURCE CONCEPT	CONCEPTUAL RELATION	TARGET CONCEPT	MENING OF THE "has" RELATION FOR GROUP I	MENING OF THE "has" RELATION FOR GROUP II	MENING OF THE "has" RELATION FOR GROU III	MENING OF THE "has" RELATION FOR GROUP IV
Decision support technique	*has*	MCDM method	d-type of, non mandatory	d) "generic-specific" relation	b) Compound relation	c) Containment relation
Decision support technique	*has*	Optimisation algorithm	d-type of, non mandatory	d) "generic-specific" relation	b) Compound relation	c) Containment relation
Decision support technique	*has*	Ucertainity evaluation	d-type of, non mandatory	d) "generic-specific" relation	a) Member field relation	a) Member field relation
Decision support technique	*has*	Other model	d-type of, non mandatory	d) "generic-specific" relation	b) Compound relation	c) Containment relation
Support of knowledge management	*has*	KM process type	d-type of, non mandatory	a) Member field relation	a) Member field relation	c) Containment relation
Social participation aspect	*has*	Participatory planning task supported	b) is composed by	b) Compound relation	a) Member field relation	c) Containment relation
DSS development	*has*	Knowledge management technique	a) Member field relation	c) Containment relation	a) Member field relation	b) Compound relation
DSS development	*has*	Software development methodology	uses	b) Compound relation	a) Member field relation	b) Compound relation
DSS	*has*	DSS development	b) Compound relation	b) Compound relation	a) Member field relation	a) Member field relation
DSS	*has*	Decision support technique	b) Compound relation	c) Containment relation	a) Member field relation	a) Member field relation
DSS	*has*	Social participation aspect	(provides support to)	a) Member field relation	a) Member field relation	a) Member field relation
DSS	*has*	support of knowledge management software	(provides support to)	c) Containment relation	a) Member field relation	a) Member field relation
Decision support technique	*has*	Ecological model	d-type of, non mandatory	b) Compound relation	c) Containment relation	d) Generic-specific relation
Decision support technique	*has*	Forest production model	d-type of, non mandatory	b) Compound relation	c) Containment relation	d) Generic-specific relation
Decision support technique	*has*	Social model	d-type of, non mandatory	b) Compound relation	c) Containment relation	d) Generic-specific relation
Forest management problem type	*has*	DSS	is supported by	c) Containment relation	b) Compound relation	a) Member field relation
Forest management problem type	*has*	Country	a) Member field relation	c) Containment relation	b) Compound relation	a) Member field relation
Forest management problem type	*has*	Decision support technique	is supported by	c) Containment relation	b) Compound relation	a) Member field relation
Forest management problem type	*has*	Support of knowledge management	is supported by	c) Containment relation	b) Compound relation	a) Member field relation
Forest management problem type	*has*	FORSYS problem dimension	a) Member field relation	a) Member field relation	d) Generic-specific relation	a) Member field relation
Lesson learned	*has*	DSS	results from	c) Containment relation	c) Containment relation	a) Member field relation
Lesson learned	*has*	DSS development	N/A	c) Containment relation	c) Containment relation	a) Member field relation
Lesson learned	*has*	country	a) Member field relation	c) Containment relation	c) Containment relation	a) Member field relation
Lesson learned	*has*	Case study	results from	c) Containment relation	c) Containment relation	a) Member field relation
Lesson learned	*has*	Decision support technique	relates to/ refer to	c) Containment relation	c) Containment relation	a) Member field relation
Lesson learned	*has*	Decision stage	N/A	a) Member field relation	a) Member field relation	a) Member field relation
Lesson learned	*has*	DSS domain	a) Member field relation	a) Member field relation	d) Generic-specific relation	a) Member field relation
Lesson learned	*has*	KM process type	relates to/ refer to	a) Member field relation	c) Containment relation	a) Member field relation
Lesson learned	*has*	Knowledge management technique	relates to/ refer to	c) Containment relation	c) Containment relation	a) Member field relation
Lesson learned	*has*	Actor perspective	a) Member field relation	a) Member field relation	a) Member field relation	a) Member field relation
Lesson learned	*has*	FORSYS problem dimension	a) Member field relation	a) Member field relation	a) Member field relation	a) Member field relation
Lesson learned	*has*	Social participation aspect	relates to/ refer to	a) Member field relation	c) Containment relation	a) Member field relation
Lesson learned	*has*	Source	a) Member field relation	b) Compound relation	a) Member field relation	a) Member field relation
Lesson learned	*has*	Consequence	a) Member field relation	b) Compound relation	d) Generic-specific relation	a) Member field relation
FORSYS problem dimension	*has*	Temporal scale	b) Compound relation	a) Member field relation	a) Member field relation	b) Compound relation
FORSYS problem dimension	*has*	Spatial context	b) Compound relation	a) Member field relation	a) Member field relation	b) Compound relation
Case study	*has*	FORSYS problem dimension	b) Compound relation	a) Member field relation	a) Member field relation	a) Member field relation
Case study	*has*	country	a) Member field relation	c) Containment relation	a) Member field relation	a) Member field relation
Case study	*has*	Organisation	a) Member field relation	c) Containment relation	a) Member field relation	a) Member field relation
Organisation	*has*	Country	a) Member field relation	c) Containment relation	a) Member field relation	a) Member field relation
Software	*has*	SFM data	has input/output	c) Containment relation	d) Generic-specific relation	a) Member field relation
Softwae	*has*	User-friendliness of GUI	a) Member field relation	b) Compound relation	d) Generic-specific relation	a) Member field relation