Using Semantic Knowledge Management Systems To Overcome Information Overload Problems In Software Engineering

Ali Demirsoy
This thesis is submitted to the School of Engineering at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering. The thesis is equivalent to 20 weeks of full time studies.

**Contact Information:**

Author:  
Ali Demirsoy  
E-mail: alidemirsoy@gmail.com

University advisor:  
Dr. Kai Petersen  
School of Computing, BTH  
E-mail: kai.petersen@bth.se

School of Computing  
Blekinge Institute of Technology  
SE-371 79 Karlskrona  
Sweden  

Internet: www.bth.se/com  
Phone: +46 455 38 50 00  
Fax: +46 455 38 50 57


**ABSTRACT**

**Context.** Information overload is an increasingly important problem of our age where the amount of data we have is expanding drastically with the use of digital communication. Information retrieval models are developed to help overcoming this problem with computerized tools. Semantic information retrieval, which means retrieving information based on the interpretations of meanings of the words, is one of these models and started to be used commonly to handle large amount of data in the Internet and in enterprises to overcome information overload problems.

**Objectives.** In this study we investigate different information retrieval models for using with knowledge management systems in large-scale organizations from the perspective of software engineers. To this end, we aim at identifying existing issues and needs about information overload and then assessing different solutions against these needs. Afterwards, we analyze the chosen solution, which is semantic search, and define and carry out an implementation process to reflect on it. Finally, the usefulness and feasibility of this type of solutions to overcome the specified information overload problems in software engineering is studied and discussed.

**Methods.** We performed a literature review to extract the existing knowledge, technology, and the problems and solutions in the defined context. Then a case study was conducted at a development site of Ericsson AB in Sweden. Case study involved unstructured and semi-structured interviews for data collection, and an implementation attempt for a simple semantic knowledge management system. Thematic Coding Analysis method is used for qualitative data analysis.

**Results.** We identified 23 codes that are categorized under 8 themes from the opinions of company practitioners about semantic knowledge management systems. They are mainly about the existing problems, arguments for using semantic system for solving them, and suggestions and challenges.

**Conclusions.** We conclude that semantic knowledge management systems have a very high potential to solve information overload problems in software engineering if the necessary measures are taken. We found that the problems are related to search engine and the document structure of the tools; usefulness of semantic system is the capability of ontology based retrieval to filter out irrelevant documents and extract hidden data and people’s skills and interests; and finally the challenge is the necessary endeavor to elicit and satisfy all the needs.

**Keywords:** Semantic information retrieval, knowledge management, ontology, information overload, unstructured information
Acknowledgement

First and foremost, I would like to express my sincere gratitude to my university supervisor Dr. Kai Petersen for his continuous support, patience, and invaluable feedback on my work. I feel lucky to have him as my supervisor.

Besides this, Ericsson AB is gratefully acknowledged for providing me this opportunity. Special thanks to all the interviewees and tutors for their patience in providing very informative responses, which added great value to the thesis.

Finally, I would like to thank all my close friends and dear ones, for helping me get through the difficult times by encouraging and motivating me. And of course, I can’t express my gratitude for my family in words. Without their support and understanding it would have been impossible to accomplish this work.
## CONTENTS

1 INTRODUCTION ............................................................................................................... 1

2 BACKGROUND .................................................................................................................. 4  
   2.1 INFORMATION OVERLOAD ....................................................................................... 4  
      2.1.1 Solving Information Overload Problem ............................................................... 4  
      2.1.2 Technological Solution Alternatives ................................................................. 5  
   2.2 SEMANTIC WEB AND INFORMATION RETRIEVAL .................................................. 10  
      2.2.1 Introduction to Information Retrieval ................................................................. 10  
      2.2.2 Semantic Web ..................................................................................................... 10  
      2.2.3 Knowledge Representation in Semantic Web .................................................... 11  
      2.2.4 Semantic Knowledge Acquisition .................................................................... 15  
      2.2.5 Semantic Annotation ......................................................................................... 15  
      2.2.6 Semantic Information Retrieval ....................................................................... 16  

3 RESEARCH METHOD ....................................................................................................... 17  
   3.1 RESEARCH QUESTIONS, AIMS AND OBJECTIVES ............................................... 17  
   3.2 SELECTION OF RESEARCH METHODS ................................................................ 18  
      3.2.1 Literature Review ............................................................................................... 18  
      3.2.2 Overview of Empirical Research Methods and the Choice of Method ............... 20  
   3.3 CASE STUDY DESIGN ............................................................................................... 22  
      3.3.1 Objective ........................................................................................................... 22  
      3.3.2 The Case and Unit of Analysis ......................................................................... 23  
      3.3.3 Data Collection and Data Analysis Procedures ................................................ 23  
      3.3.4 Threats to Validity ............................................................................................. 29  

4 RELATED WORK ............................................................................................................. 31  
   4.1 USE OF ONTOLOGIES IN SOFTWARE ENGINEERING ........................................... 31  
   4.2 KNOWLEDGE MANAGEMENT AND SEMANTICS ............................................... 33  
      4.2.1 Inhibitors in Implementing Ontology-based Knowledge Management Systems .... 35  
      4.2.2 Related Tools for Semantic Web ....................................................................... 36  
      4.2.3 Ontology-based Knowledge Management Systems ......................................... 38  
      4.2.4 Application of Ontology-based Knowledge Management in Different Domains .. 42  
   4.3 EXPLICIT, IMPLICIT AND TACIT KNOWLEDGE .............................................. 45  

5 RESULTS .......................................................................................................................... 47  
   5.1 PROBLEMS AND USAGE SCENARIOS (RQ1) ....................................................... 47  
   5.2 IMPLEMENTING A SEMANTIC KNOWLEDGE MANAGEMENT SYSTEM (RQ2 AND RQ3) .................................................................................................................. 49  
      5.2.1 Ontology Building .............................................................................................. 49  
      5.2.2 Developing a Prototype Semantic Knowledge Management System ................... 51  
   5.3 QUALITATIVE DATA ANALYSIS (RQ1, RQ2 AND RQ3) .......................................... 56  
      5.3.1 Extracted Themes and Codes .............................................................................. 56  
      5.3.2 Usage Issues (RQ1) ........................................................................................... 58  
      5.3.3 Finding Information Issues (RQ1) ..................................................................... 60  
      5.3.4 Gathering Information Face-to-face (RQ1) ......................................................... 62  
      5.3.5 Usefulness of the Semantic System (RQ3) .......................................................... 63
5.3.6 Improvement Suggestions (RQ1 and RQ3) ................................................................. 65
5.3.7 Ontology and Filtering (RQ2) .................................................................................. 67
5.3.8 Concerns (RQ3) ...................................................................................................... 70
5.3.9 Thematic Network and Relationships between the Codes ........................................ 71

6 DISCUSSION .................................................................................................................. 75
6.1 Information Overload (RQ1) ................................................................................... 75
6.2 Structuring Information with Ontologies (RQ2) ...................................................... 77
6.3 Usefulness of Semantic System (RQ3) ....................................................................... 79

7 CONCLUSIONS ............................................................................................................. 82

8 REFERENCES .................................................................................................................. 85

APPENDIX A - INTERVIEW PROTOCOL ........................................................................... 92
APPENDIX B - MINDMAP ............................................................................................... 94
**LIST OF FIGURES**

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Semantic Web Layer Stack</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>Example of an RDF Graph</td>
<td>13</td>
</tr>
<tr>
<td>3</td>
<td>SPARQL Example</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>Search Strategy</td>
<td>19</td>
</tr>
<tr>
<td>5</td>
<td>Knowledge Map for the Coverage of the Related Work</td>
<td>31</td>
</tr>
<tr>
<td>6</td>
<td>Architecture for Semantic Knowledge Management Systems</td>
<td>35</td>
</tr>
<tr>
<td>7</td>
<td>Ontological Structure in OntoShare</td>
<td>39</td>
</tr>
<tr>
<td>8</td>
<td>KIM Architecture</td>
<td>40</td>
</tr>
<tr>
<td>9</td>
<td>Architecture of SemanticWiki</td>
<td>41</td>
</tr>
<tr>
<td>10</td>
<td>ACTIVE Knowledge Workspace user interface</td>
<td>42</td>
</tr>
<tr>
<td>11</td>
<td>ONTOCLOSE Software Engineering Ontology</td>
<td>50</td>
</tr>
<tr>
<td>12</td>
<td>Annotation template for &quot;Requirements Phase&quot;</td>
<td>51</td>
</tr>
<tr>
<td>13</td>
<td>Structure Search from KIM</td>
<td>54</td>
</tr>
<tr>
<td>14</td>
<td>PROTON Ontology</td>
<td>55</td>
</tr>
<tr>
<td>15</td>
<td>Business Process Framework for Telecom</td>
<td>69</td>
</tr>
<tr>
<td>16</td>
<td>Thematic Network of the Extracted Data</td>
<td>74</td>
</tr>
<tr>
<td>17</td>
<td>Mindmap from the Evaluation Interviews</td>
<td>94</td>
</tr>
</tbody>
</table>
LIST OF TABLES

TABLE 1: THESIS OUTLINE .................................................................................................................. 3
TABLE 2: THE DESCRIPTIONS AND DIFFERENCES OF THE THREE VERSIONS OF OWL .......................................................... 14
TABLE 3: KEYWORDS USED IN ACCORDANCE WITH EACH RESEARCH QUESTION ................................................................. 19
TABLE 4: SUMMARY TABLE FOR THE MAIN ASPECTS OF THE THESIS ...................................................................................... 18
TABLE 5: SUMMARY OF SELECTION OF INTERVIEWEES ............................................................................................................. 26
TABLE 6: COMPARISON OF ANNOTATION TOOLS ......................................................................................................................... 37
TABLE 7: QUANTITY OF ELEMENTS IN SWEBOK PROTO-ONTOCOLY ........................................................................................... 44
TABLE 8: THE SUMMARY OF THE ELICITED PROBLEMS AND POSSIBLE SOLUTIONS ........................................................................ 47
TABLE 9: SELECTION OF TOOLS FOR EACH STEP ....................................................................................................................... 52
TABLE 10: SUMMARY OF OBSERVATIONS FROM THE APPLICATION OF KIM ON A DISCUSSION BOARD CORPUS ................................................. 56
TABLE 11: LIST OF THEMES AND CODES AND THEIR RELATIONS WITH THE RESEARCH INTERESTS .................................................. 57
TABLE 12: OVERVIEW OF THE USAGE ISSUES IN COLLABORATION TOOLS .............................................................................. 59
TABLE 13: OVERVIEW OF THE FINDING INFORMATION ISSUES IN COLLABORATION TOOLS .............................................................. 61
TABLE 14: OVERVIEW OF THE USAGE SCENARIOS IN GATHERING INFORMATION FACE-TO-FACE .................................................... 63
TABLE 15: OVERVIEW OF THE BENEFITS OF SEMANTIC SYSTEMS ........................................................................................................ 64
TABLE 16: OVERVIEW OF THE IMPROVEMENT SUGGESTIONS ON THE EXISTING AND THE SEMANTIC SYSTEMS ................................. 66
TABLE 17: OVERVIEW OF THE OPINIONS ABOUT THE CHOICE OF ONTOLOGY ............................................................................. 69
TABLE 18: OVERVIEW OF THE CONCERNS OF THE SUBJECTS ABOUT THE SEMANTIC SYSTEM .................................................................. 71
TABLE 19: LIST OF ALL IDENTIFIED CODES AND THEIR DESCRIPTIONS .............................................................................. 71
1 INTRODUCTION

During the last few decades, the importance of the challenges faced in software development in large-scale organizations is being noticed and researched. One of the main problems for this kind of organizations is the high number of stakeholders [1]. There might be several stakeholders involved for a specific product and these stakeholders are usually distributed due to the organizational structure. Therefore, a significant problem occurs related to the communication and coordination between these stakeholders [5, 9]. To overcome this problem large-scale international companies use knowledge management systems of which the efficiency is open to discussion. Knowledge management is the process of acquiring or creating knowledge, transforming it into a reusable form, and maintaining, finding and reusing it [10, 11]. Most of the current knowledge management systems use keyword based search models that rely on words’ lexical forms rather than the meanings of the words [116]. However these search mechanisms do not always satisfy the needs of the user in terms precision of the results [12, 13]. As a result, people that exchange information with each other face the problem of information overload due to the high number of available documents and information [122, 123, 124]. This problem corresponds to the latter part of Butcher’s definition of information overload among many others in the literature [120, 125, 126]. He states that it can mean having more relevant information than one can assimilate or it might mean being burdened with a large supply of unsolicited information, some of which may be relevant [120].

In parallel with this problem, the focus of this thesis is on the use of information retrieval technologies in knowledge management in software engineering domain; and in particular on “Semantic Information Retrieval” or in other words “Semantic Search”. Semantic search refers to retrieving information based on the interpretations of the meanings of the words [12]. Traditionally, there are classical information retrieval (IR) models that are aimed to find the most relevant document for a given query. These models are mainly based on estimating the relevance of the documents and ranking them via probabilistic methods, Bayes classifier model [32], vector space model [33] or several others. However, these models retrieve textual information based on word’s lexical forms not the meanings. Hence, there is a problem of many irrelevant search outputs as a result of ambiguity of the words. A word can have more than one meaning or many words can describe the same meaning. In these cases the results might be either irrelevant or insufficient [116, 13, 14]. There are also statistical approaches such as classifying and clustering, which are aimed to overcome these problems by relying on the statistical occurrences of the words [127]. These methods have been successful in some cases to increase the hit rate when searching [128]. However, semantic search goes one step beyond these approaches by enabling complex queries and retrieve extracted knowledge from the processed information sources. This way, the users are able to search with meaningful queries instead of textual strings and moreover automated tasks can process information with a certain level of understanding [14].

Semantic technologies and ontologies have been used in several fields like biology, finance and tourism in order to manage and structure the domain knowledge [34, 35, 37, 38, 102, 103, 104]. Moreover, there have been several studies that apply semantic technologies to software engineering domain in order to conceptualize and organize the knowledge [17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 45, 46]. These studies are focused on different processes of software development lifecycle such as analysis, design, implementation and testing. That is, these applications are directly used in developing and maintaining software. However, there are only a few examples that aim at organizing the existing knowledge in order to enhance knowledge reuse within a knowledge management system where users share documents for the use of others [45, 78]. These systems are crucial to software engineers for utilizing the existing information via finding a relevant shared document and overcome problems related to information overload [99, 108, 122]. Hence, there is a gap in the research about applying and evaluating such systems
in the field of software engineering. Semantic systems have been empirically evaluated in case studies in different areas but not in the software engineering domain [98, 99].

Another research gap here is that, we have not found any study that focuses on the problems and needs of software engineers in the context of knowledge management and information overload within knowledge management systems. Till now, the focus has usually been on managerial point of view when it comes to knowledge management but not from software engineering [129, 130]. That means, we do not know what software engineers need when it comes to managing and reusing knowledge. We cannot simply assume that managers and software engineers have similar problems and wishes as the information and artifacts they are dealing with are not the same. Therefore, there is a need to identify the needs and characteristics of software engineers in the area of knowledge management and information retrieval. Need identification has been shown as one of the most critical steps in overcoming information overload [129].

On the other hand, another research gap related to using semantic knowledge management systems is the lack of information about how to implement and adopt these solutions in an organization with no previous experience. Current research focuses on presenting the final solutions and the ideas behind them but not the ways to make these solutions work [14, 22, 47, 63, 80, 96]. Hence, there is a need to study the process of adopting semantic systems as the experience gathered from here would be valuable for similar adopters in order to understand the advantages, costs and limitations of these systems.

To sum up, the identified research gaps for this study are as follows:

- **Gap 1**: Lack of evaluations and applications of semantic knowledge management systems in software engineering domain.
- **Gap 2**: Lack of understanding and analyzing the needs of software engineers in the context of information retrieval and information overload.
- **Gap 3**: Lack of information about the adoption of semantic solutions in software organizations.

The goal of this thesis work is to understand and evaluate usefulness and feasibility of ontologies and semantic information retrieval technologies in order to overcome information overload problems and enhance knowledge reuse in knowledge management systems of large-scale software organizations.

In this context, an assessment of different solution strategies will be made and afterwards usage of ontologies in software engineering domain and application of them in knowledge management systems will be analyzed. In order to implement a useful system, the needs of the software engineers will be investigated and identified. Based on this knowledge, an ontology-based semantic knowledge management system will be implemented. We aim for implementing such a system and reflect on the implementation process, as similar implementation experiences will be faced by others who intent to use these solutions. The final aim is to evaluate the benefits of such a system to software engineers in gathering implicit or explicit knowledge that they need during development. Overall, we will study semantic solutions in the context of organizations that have not used such knowledge systems before. This includes understanding the needs, implementing and evaluating the solution. This study will reflect what most organizations will experience during their adoption of semantic solutions.

As a result of this thesis work in parallel with the identified research gaps, the novel contributions can be summarized as follows:

- **Contribution Related to Gap 1:**
Evaluation of usefulness of semantic knowledge management systems in gathering implicit and explicit knowledge in software engineering.

- **Contributions Related to Gap 2:**
  - Identifying and gaining an in-depth understanding of the most important issues and needs about information overload in software engineering knowledge management.
  - Specification of ontology and search scenario requirements of software engineers in the context of semantic information retrieval.
  - Assessment of different information retrieval methods and evaluation of them against the identified needs.

- **Contribution Related to Gap 3:**
  - Feasibility analysis and reflection on the defined implementation process of the solution.

In order to address the identified research gap and the goal of the study, this thesis presents an empirical investigation in a development site of Ericsson to identify the existing problems and challenges in knowledge management systems and explore the usefulness of ontology-based semantic knowledge management systems. To this end, an interpretive case study is conducted that consists of design and implementation of a knowledge management system, initial interviews to understand the problems and requirements, and final interviews to identify the needs and evaluate the system. The qualitative data gathered from the final interviews are analyzed with Thematic Coding Analysis method, following the guidelines defined by Robson [16].

The case study is designed as an interpretive study where the subjective truth is extracted from the context and interpreted objectively to provide results for the cases in similar contexts [143, 144]. The context of the case study is arranged according to the main problem addressed in this thesis. That is, the context illustrates a demo of real time environment with sufficient size in order to enable information overload and hence reveal the needs and problems and on the other hand an organization without semantic solution is chosen so that it reflects other similar organizations.

The outline of this thesis is as follows:

### Table 1: Thesis Outline

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Background</strong></td>
<td>Includes all the background information related to overall coverage of the thesis. Information overload problems, alternative solutions to semantic approach and motivation for chosen method are stated in this section. All the terminology and technology that needs to known about Semantic Web is provided.</td>
</tr>
<tr>
<td><strong>Research Method</strong></td>
<td>Includes detailed design, motivation and aims for every research step applied in this study (i.e. literature review, case study and data analysis). Threats to validity associated with these research steps are presented. Research questions, objectives and outcomes are also defined in this section.</td>
</tr>
<tr>
<td><strong>Related Work</strong></td>
<td>Covers previous studies about using and building ontologies, tools and architectures to implement semantic systems and knowledge management systems.</td>
</tr>
</tbody>
</table>
Results

Contains the results that are gathered from the empirical part of this thesis. Results from initial interviews, implementation process and the evaluation interviews are presented with a structure. Answers the research questions are presented.

Discussion

Covers the synthesis and critical analysis of results with respect to application of these results and research questions. The results are analyzed from a practical and academic perspective. Applicability and limitations of alternative solutions are discussed in detail.

Conclusions

The summary of all the work performed and the contributions made within this thesis are presented.

2 BACKGROUND

2.1 INFORMATION OVERLOAD

In an empirical study among 124 managers from various backgrounds and industries, the meaning of information overload was denoted in several ways [129]. The most frequently cited meanings were excessive volume of information (79%), difficulty of managing information (62%), irrelevance or unimportance of most of the information (53%), lack of time to understand it (32%) and multiple sources of it (16%). There are many external and internal causes of information overload in organizations such as number of emails, documents, minutes from meetings and the changing nature of the work, etc [129, 131]. However, one of the most important internal sources of information overload is having unclear requirements to design sort and search interfaces that can satisfy information needs of corporate users [129, 132]. That is, system designers are not aware of business needs of the knowledge workers and hence all the redundant, useless, conflicting data are presented to the users. Hence identifying these needs for user groups from different organizations and domains is one of the important steps in solving the information overload problem.

2.1.1 Solving Information Overload Problem

Information overload problems are being researched extensively in the recent years with the huge data flow in the communication age. In order to understand the importance of solving these problems, the possible consequences of not solving can be discussed. The same survey mentioned above reveals that the most common negative effects of information overload on the knowledge workers were as follows [129]:

- Loss of time (72%)
- Poor quality of work (40%)
- Poor efficiency (16%)
- Frustration, tiredness and stress (16%)
- Poor decision quality (13%)

As can be seen from this case study, information overload severely effects the quality and efficiency of the organizations. Hence solving these problems is crucial. There are several approaches to solving these problems that vary from organizational to technological changes. Some of the approaches can be listed as follows:

- **Organizational strategies**: Discusses the role of the organization to remove the causes of information overload via altering the structure and processes in the organization [129, 133, 134].
• **Individual information management strategies:** Argues personal skills and personal time and load management techniques such as filtering and focusing [130, 135, 136].

• **Technological solutions:** Rely on using software to store and distribute information and knowledge. Usually by means of a decision support system or a knowledge management system [131, 133].

• **Human agent intervention:** Refers to using an intermediate human agent that reprocesses and reroutes the information according to the needs of individuals [131].

Within the scope of this thesis, the problem solving strategy will be based on technological approaches that can facilitate better management of the information in order to reduce the information overload. It has been shown that the knowledge workers’ top solution proposal is filtering the information according to their needs and interests [51, 127, 129]. As the definitions of information overload suggest, filtering out unimportant or irrelevant information among excessive volumes of sources is one of the most important problems at hand. There are many ways that have been researched and developed in order to handle large volumes of information and filter sources according the needs of an individual.

### 2.1.2 Technological Solution Alternatives

To manage and store information sources in business organizations, it is a common practice to utilize document repository or knowledge management tools that facilitate sharing, reusing and managing information between employees. The problem about these tools is the difficulty of finding the relevant information once it is shared in the system. The research area of information retrieval covers the approaches in order to successfully find the document or the information that is being searched. In 1960s information retrieval was defined as “a field concerned with the structure, analysis, organization, storage, searching and retrieval of information” [51]. Since then the area evolved into many different techniques and models in order to adapt to the changing needs.

**Classical Information Retrieval Techniques**

There are certain models that are used to define different approaches of information retrieval that are useful for different scenarios. To have an overview of these models, the major ones can be listed as follows:

• **Exact match models:** This model is based on retrieving documents based on the exact matching of the query and the documents. The documents are retrieved without any ranking.

  o **The Boolean model:** It is the first model of information retrieval and based on retrieving documents that exactly match the query terms. A query can contain logical operators like AND, OR and NOT and each document either matches the given query or not [137].

    The only advantage of the Boolean model is that it provides users a complete control over the system and the search. It is very clear why a document is retrieved or not. However the biggest disadvantage is that it does not provide any ranking of retrieved results as each document is either completely relevant or not relevant. Hence it makes it difficult to use this model in larger sets of documents [138].

• **Vector Space Model:** Due to the limitations of Boolean models, statistical models that are aimed to rank the documents based on their relevancy were developed. In vector space model, documents and queries are represented as vectors in a multidimensional vector space and the relevance of each document is calculated as a cosine similarity between the query and document vectors [137]. Hence the documents are not directly considered as relevant or not relevant, they are ranked based on their similarity value and presented in the result set. Statistical methods dealing with the frequency of occurrence of terms within and between documents are used to create the mentioned vector representations [138].
Despite the obvious advantage with being able to rank documents, there are still some limitations and challenges with this approach. For instance, long documents are poorly presented in vector forms and precision problems occur due to possible use of different vocabularies for the same context and so on.

However there are different solution proposals to overcome these limitations. **Query expansion** and **query reformulation** are methods that are aimed to retrieve relevant results with a better coverage. This is the case when the keywords in the query is too narrow or specific and that causes overseeing some relevant documents that did not match the query string. In these approaches the query string is modified with synonyms or more general words in order to increase the coverage of results [138]. An external database or a thesaurus can be used to achieve this scenario as well.

The above proposals aim to solve the problem of “synonymy”, one of the fundamental problems in information retrieval, which means different words can refer to the same meaning. Another issue is the “polysemy” problem, where one word can have more than one meaning. In such a case, the results from a given query can consist of many different contexts and meanings of which only some of them are relevant to the user. In order to overcome this challenge, there is the **relevance feedback** method that aims to retrieve results with a few iterations based on the feedback received from the user [127]. In particular, the initial set of results are provided to the user and the user marks some of the documents as relevant or not, which is used to refine the search and retrieve a revised set of results based on the feedback from the user.

- **Probabilistic Approaches:** Other than the statistical approaches that are mentioned above, there are probabilistic models that utilize the probability of a document’s being relevant for the user’s needs. In order to accomplish this, the information needs of the users over a document collection should be defined in advance [127]. The needs are translated to query representations and the documents are converted to document representations. Based on these two, it is determined how well the documents satisfy the information needs.

**Mining Large Databases for Extracting Information**

Data mining and information retrieval are actually close fields where the difference is the absence of a user query in data mining. Data mining is defined as “the use of sophisticated data analysis tools to discover previously unknown, valid patterns and relationships in large data sets” [139]. As can be in from the definition, data mining aims to extract useful information from datasets without too much user interaction. Information retrieval and data mining can be used hand in hand in order to improve the retrieval process and increase user satisfaction. Mainly data mining methods can be divided to three categories:

- **Classification:** Classification is the idea of labeling documents with pre-defined classes in order to set a context to the search when retrieving information [127, 140]. It is also commonly used for identifying spam emails based on the content. Machine learning approaches are commonly used in text classification in order to automatically detect to which class a document belongs. To accomplish this, there is a need for human intervention to use training data for the learning process. That is, a subset of documents that are already associated with classes is used in the beginning for statistical analysis. Then learnt data gathered from these documents are used to automatically assign classes to the rest of the documents. Since the classes are defined by humans and there is a training period in the beginning where documents should manually be classified, this process is usually referred as a supervised method or supervised learning.

- **Clustering:** Clustering algorithms group a set of documents into clusters or classes, which are not explicitly defined or categorized [138, 141]. That is, unlike classification techniques there is no human supervision that assigns documents to certain classes. The labels of the clusters and the
assignment of the documents to clusters are automatically detected based on the distribution and makeup of the data [127].

There are many clustering algorithms and methods that are used in information retrieval. Hierarchical clustering creates a hierarchy of clusters whereas flat clustering simply does not relate clusters to each other. In hard clustering each document is a member of exactly one cluster whereas in soft clustering a document’s assignment is a distribution over all clusters [127].

As the methods for clustering varies, the application of these methods on information retrieval is diverse as well. Some of these applications are as follows:

- **Search Result Clustering**: This refers to presentation of results in response to a query made by the user. Unlike the default presentation simple list of results, the result set is clustered and presented to the user in a way that similar documents appear together [127]. This might be useful to solve polysemy problem where words can refer to different meanings and contexts. When a particular term is searched such as the name of a brand, the clusters from different contexts that are related to that word will be shown to the user and the user will be able to select the specific cluster that refers to the brand so that all the other irrelevant results will be filtered out with a simple step. Vivisimo¹ is a search engine that utilizes this application in order to improve recall in search results.

- **Scatter-Gather**: This technique aims to improve the user interface by enabling iterative clustering. First of all the whole collection is clustered without any query input where the user selects some of these clusters and these clusters are merged and clustered again for the second iteration. Iterations are repeated until the user finds a cluster of interest [127]. This cluster-based navigation is particularly useful when the users are unsure about which terms to search and prefer browsing to searching with query.

- **Collection Clustering**: An alternative to scatter-gather method where clustering is dynamic based on human mediation is collection clustering where clustering is hierarchical and static and is not influenced by user interactions [127]. This approach is commonly used in Google News and similar systems where the user is not exactly making a search but trying to follow interesting articles about recent stories.

- **Language Modeling**: This approach is used to solve synonymy problems in information retrieval where a term can be defined by various other words and the documents that do not contain the specific word in the query are left out although they are relevant for the user. To solve this problem, when a query is made, the initial set of documents that match the query string are provided to the user along with other documents that are from the same cluster. In the end, these documents that did not match the search query but are from the same cluster are included in the result set [127]. For instance, when the query contains the word *car* and several documents are retrieved from a cluster about automobiles, then the documents from this cluster, which use terms like *automobile* or *vehicle* instead of *car*, are included in the results.

- **Cluster-based retrieval**: Clustering can also be used side-by-side with other retrieval techniques such as vector space retrieval to speed up the search. Since some of these models calculate the similarity of each document to a given query, it can take a lot of time for large collections. With clustering approach integrated, this calculation might be applied to the documents in certain clusters that match the search query. By using this much more smaller subset compared to the whole collection, the computation and ranking can be made at a much higher speed [127].

¹ Vivisimo Search, http://vivisimo.com
• **Pattern Mining**: This technique is used for detecting patterns and associations from data, which cannot be detected easily with human effort [138, 142]. This process is also unsupervised as there is no human intervention. The very typical example for this method is the analysis of supermarket products sales. All the history of purchase data is analyzed in order to detect group of items that are often sold together in order to organize the shelves in the supermarket. This method can be used to fight with information overload problems in large sets of information via analyzing and extracting interesting information.

**Storing and Querying Semi-structured Data**
Relational databases that are fully structured are commonly used in business organizations to store related information. However, these structured databases do not always satisfy the changing needs as the existing data are not always structured and they are spread over different sources in different formats. Hence, in order to utilize this heterogeneous and incomplete information the research area started to shift to storing the data in a semi structured format that is more flexible and also appropriate for querying. Most common approaches for dealing with semi-structured data is XML and RDF and their query languages XPath and XQuery for XML and SPARQL for RDF. Especially XML is widely used in a variety of environments for managing and sharing loosely structured data that are represented in a hierarchical manner [138]. Lately RDF has gained the attention of researchers since it provides much more flexibility compared to XML by not enforcing hierarchical structure but supporting any kind of relations between data items.

The use of RDF and hence storing and querying semi-structured data has lately been considered in a whole new research area named semantic web and semantic information retrieval. Semantic Web technologies are the new generation of presenting and sharing data in various application areas. It has started to be used in web platforms as well as tools that are in a way related to managing and providing important data [10]. The idea of Semantic Web is to give information a well-defined representation so that it will be available in a more meaningful, structured and reusable way that will enable humans and computers to work in cooperation to retrieve data from the Web [47]. In ontology-based Semantic Web applications, information is presented at a semantic level with ontology, independent from data structure and implementation, with a set of concepts and relationships between them [45]. This idea emerged from the need to enable some tasks to automatically understand the concepts in order to find the right information and combine and share it with different resources. Representation of information with ontology provides a common format between different systems and applications in order to share, understand and use knowledge [48]. This common format is standardized by W3C with Web Ontology Language (OWL) [30], Resource Description Framework (RDF) [31], etc. OWL is a knowledge representation language to specify an ontology and RDF is a language to describe a data model for resources and relations between them. Other than these, Extensive Markup Language (XML) provides syntax for documents that have a format that is both human-readable and machine-readable, XML Schema is a vocabulary to structure RDF resources [48].

With the use of ontologies, the query is composed of entities from the ontology and their relations. This allows users to set the context of the input query, which solves polysemy problem mentioned before. Moreover, in this kind of data retrieval usually an external knowledge base is used to process the documents and the query. This knowledge base is used not only for text processing but also for solving synonymy problem, as the synonyms of the words already exist in this database and used during retrieval. Other than solving these two main problems in information retrieval, this method is also useful for extracting key knowledge from the document sources. With this method, the query results are not only list of documents but also pure knowledge that is extracted from these documents. The information that is available in various documents and sources can be merged and brought to the user according to the query. The details about retrieving knowledge with semantic retrieval will be given in the upcoming sections.
Motivation for the Choosing Semantic Approach as the Retrieval Method

As seen from the models that are described above, there have been various approaches to solve different issues in the area of information retrieval. The most important problems of search tools are mentioned as ranking, precision (polysemy) and recall (synonymy). All the retrieval models that are described, aim to solve one or more of these existing problems in the field. However, these models are not necessarily alternatives to each other. Moreover, they differ in the amount of human effort needed to apply to models to existing document collections [139, 127].

The latest model, semantic approach, not only offers solutions for precision and recall but also provides extracted knowledge from the analysis of the contents of the documents [49, 50]. Hence, it differs from all other models where the only aim is to retrieve the most relevant document. Here the aim is to retrieve the necessary knowledge, not the document or documents that contain that knowledge [49]. But, it can also be used to retrieve documents based on the semantics of documents and integrated with ranking techniques [13]. For this reason, semantic web approach seems to be one step ahead of the other models and semantic search can be used to solve the common problems in information retrieval.

However, the knowledge that can be extracted from documents has to be systematically modeled so that the machines can read, interpret and process the information. This causes a limitation for the type of information that can be modeled and extracted from documents. The most important factor here is the context and the content of the documents and the type of the desired information in the documents. Hence, in order to use semantic search for solving information overload, the needs of the users with respect to their information usage and the contents of the documents with respect to their domain have to be investigated and analyzed to see if it is applicable to semantic information retrieval. For instance, using semantic technologies have been seen to be very useful in areas like biology as the modeled information in biology is very suitable to represent with ontologies [34, 35, 36, 37].

With the context of this thesis, the advantages, possibilities and drawbacks of using semantic technologies in knowledge management systems in software engineering will be investigated. The aim is to share information more efficiently, extract the right information as quick as possible, gain the right amount of knowledge before making decisions and all in all improve the software development.

This study focuses on the challenges in the organizations about sharing information efficiently to reduce the time to gather knowledge and improve communication. In a very large project, there are different development sites involved and all these sites keep their documentation and notes in certain places [3]. Hence, a clustered set of documents exists and is distributed around organization. All the valuable knowledge gained from the experience of previous projects, which can be quite useful for the new ones, is stored and available in these documentation pools. However, it is quite a burden to find the right information from a stack of all these documents. As a consequence of this problem, stakeholders sometimes make poor decisions as they cannot access the right information and it might lead to wrong improvements in the development of the project [4, 6, 7].

Semantics and ontology bring an understanding to sources and enable processing and merging information via using the concepts and relationships among them [14]. One advantage of ontologies other than reducing the effort is the flexibility of them. Information from different sources can easily be combined and the ontology can be extended without a major effort when needed [8].

More detailed background information and technologies about Semantic Web and information retrieval will be provided in the rest of this chapter. This information is necessary for understanding the vision of Semantic Web and for being able to implement a semantic knowledge management system in the case study.
2.2 **SEMANTIC WEB and INFORMATION RETRIEVAL**

In this section, the latest technologies and developments about Semantic Web and semantic information retrieval will be presented.

### 2.2.1 Introduction to Information Retrieval

Information retrieval has been a popular research area since the amount of documents in the web has increased remarkably during the last decades. There are billions of documents available on the World Wide Web (WWW) right now and this means a massive pool of information. However, the size of the content does not necessarily mean that it is useful as it is [51]. The same challenge applies to corporate data as well. There are several different documents and personal posts related to different projects in large companies which make it really troublesome to extract the right knowledge. Information retrieval systems enable people or software agents to find the right information within a reasonable time [51].

In the classical sense information retrieval is composed of three main phases: indexing, query processing and searching & ranking [52]. All the content is processed in advance and indexed to speed up the search process. The user enters a query according to his needs, which is usually a search string or in other cases image or sound. Then the matched information is found via searching and brought up according to a certain ranking which aims to show the most relevant documents at the top. In the upcoming sections we will present how it works in semantically enhanced information retrieval methods.

### 2.2.2 Semantic Web

As the content and the range of the web is growing and growing, the needs of people are evolving and getting more complex. Although today’s search engines made a remarkably successful job in finding information on the web, recently web is advancing through a new era what is called web 3.0 or in other words Semantic Web [53].

The reason that brought up this development on web varies in different applications. One of them is the necessity to make more complex searches that can bring up aggregated result from various sources [54]. That is, the search should be able extract information from one document and merge with another one in order to present the desired results. Furthermore, traditional search engines are not capable of making certain filtrations. For example, retrieving the list of blonde celebrities who are over 30 years old is not possible with any of the keyword-based search engines on the web, unless there is a specific article about that. To accomplish this, the search mechanism has to narrow down from all the celebrities to these specific ones.

Another reason that brings this change was the inability to specify the context of the search. A search string can be a person’s name and at the same time an organization’s name or a product. In this case the user has to deal with a lot of irrelevant information to gain the right knowledge he wants [55].

On the other hand, probably one of the most important reasons that helped the development of Semantic Web is the need to reach, process, integrate and share information without human intervention [56]. That is, automated tools must be able to understand the content and process it without any manual help. As good as it sounds; this involves so many problems which are being researched since understanding the meaning of the documents on the web, which are mostly unstructured text or images or videos is not an easy task at all, especially considering how wide is WWW.

However, it would be easier to apply this idea to narrower, more closed application areas such as internal corporate systems, which is our motivation. Before coming to that point, we will present some technical
information, which is strongly coupled with the advance of semantic web technologies to understand how processing the content of the web with its meanings is possible.

The following sections will explain in detail the main concepts in semantic technologies starting from how to represent the knowledge to how to acquire and annotate it. These concepts will constitute a more intelligent, advanced and capable way of processing and interconnection of data.

2.2.3 Knowledge Representation in Semantic Web

Semantic Knowledge Representation refers to the study of how to represent the data in a way that it can be processed automatically, and explicit objects and relationships between them can be defined [48]. We can define four different representations that differ in capability and complexity in an ascending order [57]:

- **Tags**: Tags are simply uncategorized words that are used to describe the area or the content of the page without any rule or grammar. It is commonly used by Web 2.0 community in order to categorize content such as in personal blogs and photograph sharing websites [48].

- **Taxonomies**: Taxonomy can be considered as a set of categories that have a hierarchy between them. Daconta defines taxonomy as “The classification of information entities in the form of a hierarchy, according to the presumed relationships of the real-world entities that they represent” [58]. The simplest example would be the classification of creatures in biology.

- **Thesaurus**: Thesaurus can be considered as a taxonomy that has relationships between the concepts along with the hierarchy. However, these relationships are pre-defined and cannot be modified [59]. The ANSI/ISO Monolingual Thesaurus standard defines the word thesaurus as: “A controlled vocabulary arranged in a known order and structured so that equivalence, homographic, hierarchical, and associative relationships among terms are displayed clearly and identified by standardized relationship indicators that are employed reciprocally” [48].

- **Ontology**: Ontology, in this context, would be a much more flexible thesaurus where one can define arbitrary relations and rules related to these relations. Ontology term brought several new technologies and concepts to software engineering, so we will discuss what ontology is and related topics in a separate section below.

2.2.3.1 What is Ontology?

The most commonly used definition of ontology in the context of software engineering comes from Tom Gruber as “explicit and formal specification of a shared conceptualization” [60]. Conceptualization refers to a partial abstract representation of the world that is created for a purpose. This could be a conceptualization of a certain domain with its main terms, relations and restrictions among them. Another definition of ontology is made by W3C: “Ontology defines the terms used to describe and represent an area of knowledge. It includes computer-usable definitions of basic concepts in a domain and the relationships among them” [117].

Ontologies provide a shared understanding to the domains to solve problems related to terminology differences [54]. This shared understanding provides the web to process and interpret the contents of the resources without manual interference. Because ontology offers a structure that can be read and understood by computer agents [61]. However, current web is designed to be viewed by humans only. HTML or even XML is not sufficient to enable a wide-range computer interpretation. Because they do not have semantic modeling, they are only used for physical structure.

Moreover, ontologies are used to improve the searching and information retrieving experiences. Since ontology conceptualizes all content with classes, properties and restrictions; search query can be based on
these terms instead of arbitrary keywords. This way, the user or the automated agents can make semantically meaningful searches in order to extract the right knowledge from the content rather than retrieving the related documents [62].

To provide these capabilities, there are many technologies developed and standardized by W3C in order to formally represent the semantic knowledge [118]. They are called ontology description languages and we will explain the main ones briefly in the following sections.

2.2.3.2 Resource Description Framework (RDF) and RDF Schema (RDFS)

If we think semantic web as a stack of different technologies from the most simple to the most powerful and expressive, RDF is in the medium level just above XML and XML Schema as in the next figure below [29]:

RDF is a language for creating data model for expressing statements about objects and their relations. Statements are defined by triples that are composed of subject, predicate and value. Triples are used to store data and make it easier for machines to process and understand the data. Subject refers to a resource and predicate denotes the relationship between the subject and the object, where object is the value [54, 55].

![Figure 1: Semantic Web Layer Stack](image)
A set of triples is called an RDF Graph, which can be seen in Figure 2 above. In this graph, we have a triple that states “Bob Marley has performed No woman no cry”. Here, Bob Marley is the subject, has_performed is the predicate and No woman no cry is the value. Moreover, rdf:type is a special predicate which is defined by RDF specification and it defines a class-instance relationship [56]. That is, Bob Marley is an instance of the class Singer.

RDF-Schema is a vocabulary description language that extends RDF in order to include some basic features for defining application specific classes and properties. It enables to define sub classes, sub properties and domain and range restrictions on properties [54]. For instance in Figure 2, Singer is a sub class of Artist, which means Singer is a kind of Artist and all Singers are also Artists. Furthermore, we can define the domain and range of the properties. In our example, we can define that has_performed property has a domain Human, which means that only humans can perform artwork.

However, RDFS lacks more advanced capabilities in defining the relationships. For example, it does not provide to set cardinality, equality, disjointedness, etc. [64]. These capabilities will emerge in the semantic web world with the advance of Web Ontology Language by W3C, which will be explained below.

2.2.3.3 Web Ontology Language (OWL)

Due to the limitations of RDF, the community needed a more expressive ontology language through the end of 1990s. Until 2004, there were several proposals for the new language such as Simple HTML Ontological Extensions (SHOE), the Ontology Inference Layer (OIL) and DAML+OIL [54]. Finally, W3C launched the standard for a Web Ontology Language that is called OWL\(^2\). They expanded the earlier work of OIL and improved the integration of it with RDF. OWL solves the deficiencies of RDFS via providing additional vocabulary like relations between classes (e.g. disjointedness), conjunction of classes, property characteristics (e.g. symmetry), cardinality (e.g. one or more, at most one), etc. [64].

\(^2\) OWL, Web Ontology Language, [www.w3.org/2004/OWL/]
OWL has three versions that vary on capabilities and flexibility. Table 2 below demonstrates the basic differences and functionalities of these three [28].

**Table 2: The Descriptions and Differences of the Three Versions of OWL**

<table>
<thead>
<tr>
<th>Description</th>
<th>Constructs</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OWL Lite</strong></td>
<td>Simplicity version of OWL that provides all the basic features. Supports hierarchy and simple constraints</td>
<td>- Class</td>
</tr>
<tr>
<td><strong>OWL DL</strong></td>
<td>Provides maximum expressiveness while retaining computational completeness (all conclusions are computable) and decidability (all computations finish in finite time)</td>
<td>- owl:oneOf</td>
</tr>
<tr>
<td><strong>OWL Full</strong></td>
<td>Provides maximum power and freedom. Does not give any computational guarantee.</td>
<td>- All</td>
</tr>
</tbody>
</table>
2.2.3.4 SPARQL Protocol of RDF Query Language (SPARQL)

After the improvements in Semantic Web during the last decade, there was a need to create a query language to process data that are stored in RDF format, as XML query languages did not satisfy the needs [54]. Although there were other query languages before, in 2008 W3C announced the standard for ontology query language, which is SPARQL [27]. It is a simple query language that resembles SQL and extracts information from RDF graphs. A query can consist of triple patterns that the RDF graph is composed of, and conjunctions and disjunctions [27].

![SPARQL Example](image)

Figure 3 above is a small example of a SPARQL query, which is supposed to return all singers in the ontology and their corresponding songs.

2.2.4 Semantic Knowledge Acquisition

Acquiring semantic knowledge is an expensive process especially in the case of ontologies, as it requires labor of experts in both ontology engineering and the target domain [48]. Ontology-based semantic knowledge refers to storing the data in ontologies, which requires constructing an ontology and populating it with the extracted information.

Although there are semi-automatic ways to construct an ontology, recent research shows that extending a given ontology or building a new ontology gives much better results than automatic ontology construction methods [55].

Ontology population is the task of identifying instances and their relationships corresponding to concepts and relations defined in an ontology. This is also a costly process that requires major understanding of engineering. There are two main automatic ontology population approaches:

- **Pattern-based approaches**: These approaches are based on looking for phrases that might refer to the relation between concepts in a sentence [65]. That is, the verbs in the sentences are analyzed and checked if it corresponds to a relation in the ontology.

- **Wrapper-based approaches**: These approaches are based on analyzing the structure of the documents. In these cases, there are no proper sentences but structured with a certain context [66]. For instance, the information provided in eBay for a certain object is structured in a way that one can see the price, color, specifications, etc. This approach can extract these properties of the object as a relation in the ontology.

2.2.5 Semantic Annotation

In order to translate the unstructured information to a clear formal representation, metadata is used to formally describe the parts of the actual data. Metadata is usually described as data about data [54]. In this
case, it is the machine understandable information about the data. Annotation means the link between the data and the metadata [67].

Metadata and annotations are used to make the information readable by agents and also remove the ambiguity of the words [55]. That is, the same word in a document can refer to a person’s name or an organization’s name, which is not machine understandable. By using annotations and metadata, extracted words will be connected to concepts in the ontology and this ambiguity will be removed.

There are many tools developed for providing automatic, semi-automatic or manual annotation of the documents. These vary depending on the human intervention needed during the annotation process. More about these tools will be discussed in the “Related Work” chapter.

### 2.2.6 Semantic Information Retrieval

As mentioned earlier, the whole idea behind semantic search comes from the need to improve the search capabilities and automate information retrieval process. For both of these goals, building search engines that understand the meanings of data is necessary. For instance, for the automation of information retrieval, we need agents that can process the sources and understand the meanings so that they can use, exchange and share them without any human intervention. If the agents do not know the meanings of the data they cannot know how to utilize it. On the other hand, existing capabilities of search engines has started not to cover the needs of human users [67]. People usually look for information and knowledge when they are making a search but current search systems only return relevant documents and pages, which might have the information; the user cannot seek the knowledge itself. Hence, this burdens users an extra effort and time to gain the right knowledge.

However, achieving semantics for different kinds of information resources is not an easy task and requires many prerequisites. First of all, the data should be abstracted explicitly in a way that agents can interpret. Ontology is the most semantically expressive way for this purpose, which is used to conceptualize and annotate the information resources [57]. For the agents to process information, the data need to have metadata that describe what the data are about [68].

The focus of this thesis work will be on usage of ontology-based semantic information retrieval in knowledge management systems of business corporations and its relations with software engineering and information overload problems. Details of these concepts and the existing work from the literature are presented in the following sections.
3 RESEARCH METHOD

This section will illustrate all the research methods to be conducted for this thesis work and the approach to perform these methodologies. Moreover, the motivation for choosing each research method will also be discussed. The methods will be conducted to accomplish the objectives of this study and answer the research questions.

3.1 RESEARCH QUESTIONS, AIMS and OBJECTIVES

The main goal of this study is defined as follows, based on the template proposed in [74]:

- To analyze ontologies, semantic information retrieval and Semantic Web for the purpose of evaluation,
- with respect to usefulness and feasibility in overcoming information overload problems and enhancing knowledge reuse
- from the point of view of software engineers,
- in the context of knowledge management systems in large-scale organizations.

In order to accomplish this goal, the following objectives are defined which will constitute a roadmap for reaching the overall result:

- Find out the existing problems and needs about information overload and knowledge reuse in existing knowledge management systems.
- Find out how ontologies were previously used to structure information to manage large body of information and what ontology alternatives exist.
- Implement a simple semantic knowledge management system to analyze challenges and applicability.
- Apply the semantic system to the existing platforms in the organization in order to evaluate its usefulness for finding information with practitioners working on software engineering.

Within this perspective the following research questions are defined:

- **RQ1**: What are the existing problems about information overload and finding the right information in large-scale organizations?
- **RQ2**: How were semantics and ontologies used in software engineering domain and how can they be utilized to structure information?
- **RQ3**: How useful are semantic knowledge management systems gathering implicit and explicit knowledge in software engineering?

With the answers to these research questions, first we will be able to identify the existing problems and needs related to information overload in large-scale software development. Then we will collect information about the use of ontologies in software engineering domain in general to see if we can find relevant studies that can be used in this case study. However, the main focus will be on semantic search and hence semantic knowledge management systems to find out the possible benefits of the use of semantics and ontologies to software engineers and software development.

The summary of all the relations between the objectives, research questions, outcomes and the methods are given in Table 3 below.
### Table 3: Summary Table for the Main Aspects of the Thesis

<table>
<thead>
<tr>
<th>Aim</th>
<th>Research Question</th>
<th>Outcome</th>
<th>Method</th>
<th>Research Gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find out the deficits of current knowledge management systems in order to analyze which of them can be solved by semantic systems</td>
<td>RQ1</td>
<td>An analysis of usage scenarios and problems with the existing knowledge management systems</td>
<td>Literature Review &amp; Case Study</td>
<td>Gap 2</td>
</tr>
<tr>
<td>Find out usages of different ontologies to structure information</td>
<td>RQ2</td>
<td>An analysis of the usage of ontologies in software engineering and in other domains</td>
<td>Literature Review</td>
<td>Gap 1</td>
</tr>
<tr>
<td>Investigate the most appropriate ontology options for using in semantic KM systems</td>
<td>RQ2</td>
<td>An analysis of different ontologies that can be used to overcome information overload problems</td>
<td>Literature Review &amp; Case Study</td>
<td>Gap 1</td>
</tr>
<tr>
<td>Apply an ontology based semantic knowledge management system to Ericsson</td>
<td>RQ3</td>
<td>A platform which provides semantic search on the existing knowledge repository of the organization</td>
<td>Case Study</td>
<td>Gap 3</td>
</tr>
<tr>
<td>Evaluate the new system and observe the improvements in accessing knowledge</td>
<td>RQ3</td>
<td>An analysis of the applicability of semantic knowledge management systems to software engineering domain</td>
<td>Case Study</td>
<td>Gap 1 &amp; 3</td>
</tr>
</tbody>
</table>

### 3.2 SELECTION OF RESEARCH METHODS

This section will present the discussion about choosing the right research methods that suit the scope and the goals of this thesis.

#### 3.2.1 Literature Review

The research for this thesis work will be built upon a traditional literature review to reveal the existing studies about information overload, information retrieval, Semantic Web and knowledge management. Firstly, the aim is to find out problems, challenges and needs about information overload and retrieval in the academia and in the industry. In addition, the goal is to extract technical information about how to use semantic technologies and to see how ontologies have been used in knowledge management and information retrieval in software engineering and in other domains. The knowledge that will be acquired by this step will constitute a basis for the next research methods. As for this reason, traditional literature review is chosen instead of systematic review since traditional reviews are more suitable for gathering basic knowledge about an area [2]. Traditional literature review is generally used for summarizing results of several reviews or giving broad background information on a certain subject [2]. Since we are not aiming to aggregate any evidence and we do not want to reveal all the existing knowledge about the area,
we do not need to perform a thorough systematic review. A literature review would be sufficient to find the relevant studies in order to get an overview of the information overload, semantic information retrieval and knowledge management.

General information about the design for the literature review will also be provided under this section:

**Search Strategy**: As a method to perform this research, the process depicted in Figure 4 is followed. First of all, several trial searches are made in Google Scholar to find general information related to research questions. Among the results, the most cited papers, books and relevant thesis samples are selected and read through. This process is important, as the author of the thesis is not very familiar to the research area.

The background information gathered from these results is used to refine the existing keywords and derive new keywords that will serve for the goals of this study. In the end, several keywords were created to search for articles in different databases.

![Figure 4: Search Strategy](image)

After the keywords are refined, the following search strings, shown in Table 4, are used to search for articles in specified databases.

<table>
<thead>
<tr>
<th><strong>Research Question</strong></th>
<th><strong>Search String</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RQ1</strong></td>
<td>(“information overload” OR “finding information” OR “information retrieval”) AND (“knowledge reuse” OR “knowledge management”) AND (problems OR challenges OR issues)</td>
</tr>
<tr>
<td><strong>RQ2</strong></td>
<td>(semantic OR ontolog*) AND (“software engineering” OR “software development”) (semantic or ontolog*) AND (“knowledge management” OR “structuring...”</td>
</tr>
<tr>
<td><strong>RQ3</strong></td>
<td>(“information overload” OR “finding information” OR “information retrieval”) AND (“knowledge reuse” OR “knowledge management”) AND (semantic OR ontology*) AND (benefits OR usefulness OR advantages) AND (software engineering) AND (“knowledge management” AND (tacit OR implicit))</td>
</tr>
</tbody>
</table>

**Study Selection Criteria**

The aim of the literature review is to reveal the existing problems, applications and technologies in general in order to gain a basic knowledge that will be sufficient to perform the case study. Therefore, a large set of different studies is included in the study selection criteria.

- Both studies conducted in industry and in an academic environment are included.
- Systematic reviews that summarize different usage scenarios of ontologies and semantic technologies in software engineering are included primarily.
- Architecture suggestions for a general knowledge management system are not included unless they are related to Semantic Web or ontologies.
- Moreover, articles that argue the existing problems in knowledge management are included.
- However, papers that focus on how to build a Semantic Web and challenges about migrating towards Semantic Web are discarded, as the focus of the thesis is knowledge management.
- On the other hand, studies that focus on traditional information retrieval models and algorithms are also not considered. These decisions are made in parallel to the objectives of the research questions.

**3.2.2 Overview of Empirical Research Methods and the Choice of Method**

This section will cover the basics of different empirical research methods, description and goals of this particular study and finally the motivation for choosing each research method.

To begin with there are two main approaches to conduct an empirical study: Qualitative and quantitative research [69]. Qualitative research is related to interpreting a phenomenon based on opinions of related people. It is concerned with understanding the problems and discovering causes on a certain situation. Quantitative research deals with numerical data and tries to find a cause-effect relationship or compare different groups. It promotes numerical comparisons and statistical analysis.

As part of this thesis work, the empirical study aims to understand the existing problems based on the subjective opinions of the people and analyze the possible solutions and their usefulness. Since we do not have any specific numerical data and the situation is open to interpretation of the subjects, qualitative research is chosen to be the most appropriate method to carry out this study.

Depending on the purpose and the conditions of the investigation, there are five main types of strategies to conduct an empirical study:

**Experiments:** Experimenting is a method where a statistical analysis is performed to see the effects of a change in a highly controlled environment. To see the actual effects, one or more variables are manipulated and all other variables are controlled at fixed levels. They are usually concerned with a limited scope and run in a laboratory setting [69].
**Case study**: Case study is composed of an observation of a real world situation in a natural setting. It enables an in-depth investigation of a phenomenon, project or an activity. Data is collected for a specific purpose during the investigation and can be analyzed to establish relationships between attributes [69, 70].

**Survey**: Surveys are for studying a phenomenon in a large population by selecting a sample that represents the whole. The data collection is performed via questionnaires or interviews and the results are analyzed to derive explanatory and descriptive conclusions [69, 71].

**Post-mortem analysis**: It refers to the analysis of the whole or any part of the project retrospectively. The post-mortem can be performed by investigating project documentation or by interviewing the people involved in the project [69].

**Action research**: In action research the objective is to introduce an intervention in a real-world setting and observe its effects on the target situation. The researcher should actively be involved in all steps of action research including implementing the intervention, collection of data and their interpretation. Action research requires an extensive effort from the researcher, as he needs to actively take part in the organization in the implementation phase [72, 73].

**Motivation for choice of research method**: The aim of this study is to find out the possibilities of applying semantic technologies to organization’s internal collaboration tools in order to solve problems related to sharing information and to improve knowledge extraction process. Hence we need a deep understanding about people’s problems and needs that will be sufficient for us to come up with solutions. Semantic Web based solutions will be presented to the practitioners and an analysis will be made to find out how much of the problems will come to a solution.

The choice for the research method in this thesis is case study. The reason why case study is chosen as the empirical research methodology is that, it is usually used for exploring or testing a certain phenomenon and enabling deep understanding [14]. In our case, we will identify the problems and explore if semantics is applicable to the system and understand the needs of the employees and usefulness of semantic systems in the context (RQ1 and RQ3). Although action research seems also appropriate for this situation, we do not wish to call it that way, as we do not intend to solve the whole problem. The idea is more about exploring the possibility of a solution that will be a base for further studies. Due to the extensive effort needed for action research and the complexity of the studied case, a complete solution in a real-world setting will not be implemented.

On the other hand, in order to identify the problems and explore the solutions we need to construct a real world context and make observations, as these problems are highly human-centric and hence subjective. As for this reason, the case study has also an interpretive approach as this approach is used in situations where subjective truth cannot be separated from the context [144]. The aim with this interpretive approach is to come up with objective interpretations from subjective data gathered from a certain case.

Besides, controlled experiment is not considered either, since designing an experiment with controlled variables is hard to achieve in this case. A complex and human based process like gathering knowledge cannot be controlled with concrete attributes. Moreover, the solution, opinions obtained with this study cannot be replicated in a lab environment. Moreover, controlled experiments are objectivist studies and are used in situations where the context is not open to interpretations.

Surveys are not chosen either as they are used for getting an overall overview of a situation in a large population and generalizing it. However we need an in-depth understanding rather than generalizing results or analysis (RQ1). Since we already have an agreed company and practitioners who are available...
and willing to participate in interviews in order to provide deep understanding, case study is a better choice.

Post-mortem analysis is not considered since we do not have a specific project that we want to analyze. We want to investigate a case and understand it and look for the applicability of a proposed solution (RQ3).

Overall, practical part of this research will form the main contribution to this study and this will be by means of a case study. The study will be conducted at a development site of Ericsson AB located in Sweden.

Within the case study two sub-methods will be used for data collection and data analysis: Interviews and thematic coding analysis. Initial interviews will be conducted to identify the challenges and come up with the usage scenarios that specify how to utilize the system. According to the outcomes, possible solutions will be researched and the most appropriate semantics enabled alternative will be developed and applied to the system of the company within the case study. Final interviews will be used to evaluate the success of the solution via observing the satisfaction of the end users and thematic coding analysis will be used to analyze the data from the interviews. Details of these methods and motivations for selections are given in Case Study Design section below.

### 3.3 Case Study Design

Large-scale globally distributed organizations need to utilize collaboration tools in order to maintain and manage the internal communication between different development sites, departments and team members. As part of this thesis work, we will analyze the platforms that Ericsson has been using and evaluate the applicability of Semantic Web technologies to them.

This chapter is composed of objectives and the definition of the case study, unit of analysis, data collection and analysis methods and the threats to the validity of this thesis. Design and planning instructions defined by Runeson and Martin in their paper about case study guidelines will be followed [15]. To follow a protocol for the case study is substantial, since it will serve as a guide during the execution of the study, the existence of a protocol provides a formulation for the case study design and it makes it easier for the other researchers to read and review the study [15].

#### 3.3.1 Objective

We define this case study as “exploratory” and “improving” according to the classification of Robson in his book about research methodologies [16]. That is, it is a study that aims to find out what is happening in a certain situation or phenomenon and explore new ideas, possibilities and generate solutions in order to improve the current situation. When the solution is applied, it will be explored whether it has the potential to lead any improvements or not. This exploration and improvement is conducted within an interpretive approach that includes subjective opinions of humans. The subjective truth gathered during the study is interpreted objectively in order to create conclusions that can reflect to other cases in similar contexts [143, 144].

In our case, the goal is to explore how useful it would be if we apply Semantic Web technologies, including ontologies and semantic search, to internal knowledge management of large-scale organizations in terms of improving the efficiency of information exchange and possible benefits to enhance knowledge reuse.
Traditional knowledge management systems, collaboration tools and information sharing platforms provide classical keyword-based search capabilities [49]. As a result of the search the users face a set of matching documents and they try to extract the required knowledge from these documents. The goal of this case study is to analyze how semantic information retrieval techniques can reduce the time to extract information from large sets of documents. Knowledge and experience about previous or existing other projects is a key factor to achieve success and access to right information is an important part of it especially in larger organizations that store huge amount of previous knowledge.

### 3.3.2 The Case and Unit of Analysis

The case being studied is a development site of Ericsson AB located in Sweden. The company is one of the leading telecommunication companies in the world and develops software in telecommunication and multimedia domain. The company’s products are used in more than 180 countries in the world. The company currently has more than 100,000 employees.

As far as units of analysis are concerned, internal knowledge management systems and the documentation they entail is defined as the unit of analysis. Hence this study has a single case holistic design [70]. Ericsson uses a set of in-house knowledge management tools. These are platforms where everybody can share all sorts of information. They support uploading documents and files; sharing blog posts and creating groups and discussion boards.

The problem comes from the pile of information sources from different development sites from all around the world, which makes it hard to track the contents and find the right information. Currently these systems support only keyword-based search and this does not yield satisfactory results for the company employees. For this reason, people have started not to use these systems efficiently, which can cause a lack of communication in between teams. As a result, they are looking for different ways to extract knowledge from these large pools of information. These issues are in parallel with the recent studies about information overload and information retrieval and form the baseline for our research gaps.

To overcome these challenges, we aim at applying new information retrieval technologies to these enterprise collaboration tools in order to improve data access capabilities. We will find out the ways about how this technology should be used in this context. To achieve this, we will implement a basic tool and reflect on this implementation process in terms of state-of-the-art, challenges and lesson learnt so that this experience will be a basis for the organizations that have not used such systems before and intent to implement it. In the end, we will evaluate the tool with company practitioners in order to judge its usefulness. This whole process will constitute a guideline and keystone for further attempts to overcome information overload problems in software development in large-scale organizations.

### 3.3.3 Data Collection and Data Analysis Procedures

This section will explain in detail how each procedure will be performed in order to manage the conduct of the research. The empirical part of this work is composed of three main phases:

1. **Data Collection by Initial Interviews**: Interviews will constitute an important part of this research in the first and the final phases of the thesis. When we need to be in close contact with the problem owners, usually the easiest and fastest way to extract information about the needs of customers is conducting interviews [76]. In the beginning, some interviews are held with the tutors from the organization to identify their needs and problems. These are unstructured interviews, which are aimed to kick-off the project and have an initial understanding.

   Among many interview techniques like unstructured, structured, semi-structured, collaborative and so on; we decided to conduct unstructured interviews for the initial phase of the study [75]. That is, we
will not necessarily have a set of questions to ask to the interviewee but the flow of the conversation will shape the content. The purpose of these interviews is simply to get to know the domain, problems and the environment. Hence, we do not need to prepare a structure for the interview, rather we want to have an introductory meeting and have an understanding about the issues.

In order to elicit their requirements and issues, three separate interview sections will be conducted. The first two interviews are in the first half of the May, 2012 with the industrial tutor who is assisting this study in the organization. He is a system level manager with over 20 years of experience. The interviews lasts an hour and the academic supervisor Dr. Kai Petersen from BTH will join to all of the initial interviews. Participation of the supervisor is important as he and the practitioner in charge are more familiar with the subject area and they conduct a brainstorm during the elicitation process.

For the third interview, two experienced software managers from Ericsson, who are responsible for innovation and have technical backgrounds in software engineering, will also join the session. This one is conducted in June after the literature review is started and initial findings are extracted. The goal is to see what is available in the literature and argue the applicability of the alternative solutions during the interview.

The following topics are discussed during these interviews:
• Introduction of the company and the domain to the student
• The existing challenges in the organization about finding information
• The deficiencies of current internal collaboration tools
• Requirements of a new solution
• Possible usage scenarios about accessing the right information

Interviewee will take notes during these interviews and the collected data will be analyzed and synthesized within the supervision of the academic advisor. Moreover, bi-weekly workshops are organized to discuss the findings, solution alternatives and status updates with the industrial tutor. Hence the data collected from the initial interviews will be validated in these workshops. These meetings are significant due to getting constant feedback from the problem owners and also analyze the impact of the solution proposals to the company.

2. Development of a Simple Semantic Knowledge Management System: This is one of the most important phases of the thesis and requires considerable time and effort. After choosing the solution strategy in the initial interviews, it is important for the credibility of the case study to create an example system and apply it to real world data for the participants to have an idea about the semantic technologies. The idea is not to implement a complete system that can replace the existing one but rather have something simple that is sufficient to evaluate the usefulness of semantic systems in general.

Building a semantic knowledge management system consists of many intermediary steps that require different technologies and knowledge. Based on the general architecture proposed by Davies, which is presented in Section 4.2 and in Figure 6, we define four different components that a complete semantic knowledge management system has to support:

2.1. Text Processing (Knowledge Acquisition): The purpose of semantic system is to extract knowledge from large sets of unstructured information. Hence, the first step is to analyze and process these unstructured documents that might contain text or different formats like image, video, xml and so on. In the scope of this thesis only text processing will be considered.
Natural Language Processing (NLP) technology has evolved to gain many capabilities in order to process the syntax and semantics of a text. The tools for processing text use some language resources like lexicons, corpora or ontologies so that they can identify the words that exist in a sentence. Parsers look up all the sentences in order to parse the syntax of the sentence and associate it with the language resources [77].

2.2. Ontology & Knowledge Base (Knowledge Representation): Ontology is one of the most important factors in information extraction as it provides a conceptualization to the resources and will be used for text processing. The ontology must be suitable to the contents of the information sources that are to be processed. Hence there is a need to make the right ontology choice depending on the context of the domain.

There are general (upper-level) ontologies available in the literature, which can be utilized for the general concepts like names of cities, people, organizations, etc. These ontologies are not domain-specific, so they do not contain a detailed taxonomy. Depending on the usage scenarios we might feel that an upper-level ontology is sufficient for our needs or there might be a need to design a new ontology or edit an existing one according to the concepts in software engineering domain.

On the other hand, along with ontologies, knowledge bases can be used to store the already known data in machine understandable format. Knowledge bases include instances with respect to the entities and relations of the ontology. To perform semantic information extraction, NLP tools usually come up with a general knowledge base, which means it has only general information that a person knows by common sense. That is, the list of well-known people, organizations, locations and similar entities are stored in the knowledge base so that they can be used during text processing. There are several databases available online, which serve this purpose such as Wordnet\(^3\) and DBpedia\(^4\).

However, a specialized system on a certain enterprise will require specific terminology and taxonomy. For instance, Ericsson uses several domain specific terms that are related to telecommunications and they are commonly known among people in this area. Moreover, the names of the people, roles and positions, project names and abbreviations that exist within Ericsson organization should exist in the knowledge base so that they can be recognized during text processing.

2.3. Semantic Annotation & Ontology Population (Knowledge Acquisition & Representation): When the NLP tools parse the unstructured text, found entities should be annotated and mapped to the ontology. So the ontology will be populated with the extracted knowledge in the RDF or OWL format. This is the most significant step on information extraction as this is the phase where the relations between entities are defined. There are several platforms and ways to accomplish this step. Since this step is both depended on the NLP tool and the choice of ontology, it is crucial to choose the right system to integrate and work efficiently.

2.4. Semantic Search (Knowledge Use): Once the ontology is populated with the instances and relations extracted from the text; the only thing left is using a query language that is created for Semantic Web in order to retrieve the right information. A query engine needs to be chosen among many and should be supported by a graphical user interface in order to provide an easy to use system. Users should be able to perform search with semantic capabilities, navigate between sources according to their semantic relations and so on.

---

\(^3\) WordNet, [http://wordnet.princeton.edu](http://wordnet.princeton.edu)

\(^4\) DBpedia, [http://dbpedia.org/About](http://dbpedia.org/About)
3. **Data Collection by Evaluation Interviews**: The final evaluation and analysis will be by means of interviews with several company employees. This phase will constitute the main data collection process of this thesis and will provide information about the current challenges, obstacles about accessing information and possible improvements, suggestions and critics for the proposed new system. The system’s usefulness and users’ experience with the system with semantic capabilities will be evaluated.

This time interviews will be semi-structured. That is, there will be a set of questions but they are not necessarily asked in the same order as they are planned. The flow of the interview will shape the order of the questions [15]. The prepared questions will constitute a sort of checklist of topics that should be mentioned during the interview. The reason for the choice of semi-structured interviews this is the fact that, by the time of evaluation there will be a certain set of questions and issues that need to be specifically discussed unlike the case for the unstructured interviews in the initial phase. On the other hand, it is desired that interview structure should allow exploration of possible new issues and improvisation during the flow of the interview [75]. For this reason, fully structured interview is not chosen.

a. **Selection of Interviewees**: Interviews are the most important phase of the data collection for exploring the situation and for evaluation of the proposed system. Hence the right practitioners should be chosen to conduct the interviews. For the selection method, a combination of different approaches is taken into consideration. First of all, maximum diversity of the roles and experiences of the interviewees is aimed so that reliable information can be collected from different sources. Hence, sampling for diversity method is applied [119]. The interviews will be conducted with employees with experience from 3 to 25 years and with diverse roles such as project manager, software architect, software developer, R&D specialist, solution architect and so on.

On the other hand, gathering maximum amount of useful information, suggestions and opinions within a restricted time is a very important factor when selecting interview subjects. For this reason, it is considered useful to take some help from experts so that they can point out the right people and roles that can possibly have valuable knowledge in parallel with the research goals of these interviews. So, suggestions of the tutor in the organization are taken into consideration and some employees that can have interest in usage of knowledge management systems, innovation and semantic technologies are selected as candidates. Among the suggested ones, convenience sampling method is applied, to be able complete the data collection interviews as soon as possible (before their Christmas holidays) [119]. Hence, practitioners who are easier to access and have more availability are chosen for the interviews.

As a result, 8 employees will be interviewed as seen in the Table 5 below, which we believe will provide sufficient amount of information to contribute to the literature and industry.

<table>
<thead>
<tr>
<th>Role</th>
<th>Experience</th>
<th>Department / Responsibilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Manager</td>
<td>10 years in Ericsson 20+ in total</td>
<td>Project management, process improvement, process management</td>
</tr>
<tr>
<td>Software Architect</td>
<td>12 years in Ericsson 15+ in total</td>
<td>Software design, development, innovation</td>
</tr>
</tbody>
</table>
b. **Interview Guide:** The interviews will basically be related to the usage of internal collaboration tools of the organization, such as frequency of use, usage scenarios, satisfaction of the current version, suggested improvements and so on. Later, the new semantic knowledge management system will be presented to the users, and they will be asked to explore the new system. After they have an idea about the system, similar questions to the ones asked in the beginning will be repeated and their opinions will be taken and compared to the previous ones. We can formulate the execution of interviews as follows and the complete interview guide can be seen in Appendix A:

- **Warm up:** First, the interviewer will present himself, a little bit about the background of the project and the reason for making the interviews. Then general questions to the interviewee about his role, experience and current projects will be asked. This phase is mainly for getting to know the people and situation and also for breaking the ice before starting the actual interview.

- **Information related to usage of collaboration tools and problems:** It is important to know how and for which purposes people use these tools during their daily work. This phase is for figuring out how often they use these systems, how satisfied they are with the current version and how they would prefer these tools to be. Basically more usage scenarios, requirements and problem about finding information will be elicited in this phase.

- **Implicit knowledge:** We want to know how the employees gather knowledge when they cannot find what they look for or they are not satisfied with the findings they get. We will try to learn if they feel the need to talk to an expert and if so how they find out who the expert or responsible is in that area and so on. These questions are based on the data collected in the initial interviews.

- **Presentation of the prototype of the new system:** In this phase, we will give an overview of the Semantic Web technologies and provide information about the usages and goals of Semantic Knowledge Management Systems. Then we will present the new system as a prototype and explain the functionalities coming with Semantic Web. We will let the interviewee to browse in the system between the documents for a while in order to make sure the interviewee is aware of the differences with the existing traditional knowledge management systems.

- **Satisfaction and evaluation about the proposed system:** We will ask the employees to compare this system with the existing one and ask if they find it useful or not. We will try to know if they
would use this system more often and if it would help them to make better decisions or reach implicit knowledge more easily.

- **Recommendations:** Finally we will present questions about possible different options for creating Semantic Knowledge Management Systems and ask for preferences of the interviewees about ontologies. Also suggestions about the improvements about the proposed system will be appreciated.

c. **Data Analysis Guidelines:** As a result of the interviews, a reasonable amount of qualitative data will be collected. Although these data completely consist of words, some procedures and helpful routines are defined to analyze the data more systematically and efficiently. They are, however, less technical compared to quantitative data and closer to codified common sense [16]. Furthermore, there is usually a need for an expert to guide and support the process. During this process supervisor of this thesis will give constant feedback about the processes to be conducted.

We have chosen Thematic Coding Analysis as the most appropriate method for data analysis in this thesis. We will apply the guidelines that are thoroughly described by Robson in his study named Real World Research [16]. He describes it as a generic approach to analyze large amounts of qualitative data. The reason we chose this method is also based on the observations of Robson, which are consistent with our case:

- **Flexible:** This approach can be applied to all types of qualitative data.
- **Easy to apply:** There is relatively less need for experience of the researcher in qualitative research.
- **Quick:** The time and effort to understand and apply the method is relatively less than the other methods such as Grounded Theory where a complete application should take about 12 months [43].

There are other options to analyze qualitative data such as Grounded Theory, pattern-matching, explanation-building and time-series analysis. However, they do not support understanding the nature and complexity of the process taking place except for Ground Theory [70]. The reason why we did not choose Grounded Theory is because we do not look for developing a new theory as in the case of this method. We might as well apply only Open-Coding Analysis, which is a part of this method, but we believe finding the relations between themes in thematic approach will be more useful for us to interpret results.

Robson defines 5 phases for applying the Thematic Coding Analysis [16]:

1) **Familiarizing with the data:** This is the first phase when one reads and comprehends the collected data. It involves first transcribing the data from the audio record and then getting familiar with the data.

2) **Generating Initial Codes:** Codes refer to the most basic segment or element of information that can be interpreted in a meaningful way related to the situation. This phase of generating codes aims to organize the data into meaningful groups. This process will constitute a basis for the next step. For instance codes for a study that analyzes environmental pollution can consist chemicals, gases, ozone, vehicles, wastes, recycling, etc.

3) **Identifying Themes:** This phase refers to grouping codes into different categories so that all relevant data will be gathered together. Each category will be called as “theme”. For example themes for a study that analyzes environmental pollution can consist air pollution, water pollution, thermal pollution, etc.
4) **Constructing Thematic Networks:** This is the process in order to group the themes and create a map or network with them. There can be more than 1 map if some themes are not related to others at all. These maps can also constitute a hierarchic structure for the themes extracted from the raw data.

5) **Integration and Interpretation:** This is the phase where the actual analysis of the data is made. Tables or diagrams shall be used in order to make comparative and relational exploration. The patterns and results from the collected information should be described, summarized and interpreted to demonstrate the quality of analysis.

### 3.3.4 Threats to Validity

Validity is an important factor in empirical studies. It refers to the creditableness of the results in the sense that to what extent they are true and not biased [15]. In a case study, possible elements that can threaten the validity of the research should be identified and analyzed carefully during the design phases so that possible actions to remove or reduce the impacts can be taken. We will analyze the validity threats and mitigating factors in our case study following the descriptions given by Yin [70]:

- **Construct Validity:** Construct validity is concerned with the extent to which was intended to be measured was actually measured [15]. That is, the questions that the researcher has in mind should be interpreted the same by the participants of the empirical study. To mitigate this kind of threats, we have taken the following actions [72]:
  - **Selection of the Interviewees:** The selection process was managed by the help of practitioners from the company and the supervisor who works partly in the company. Selection process was a combination of diversity and convenience sampling. As far as convenience sampling is concerned, selection was made based on the knowledge and availability of the employees. Here there is a risk that practitioners can choose people who support similar ideas with them. The usage of diversity sampling mitigated this threat by selecting employees with more diverse roles and experiences. In the end, the interviewee selection formed quite a diverse and potentially useful list of organization members.
  - **Reactive Bias:** This one refers to the risk that the interviewees might be affected by the presence of the researcher and give biased answers that would influence the outcome of the study. This threat is partially reduced, as the tutor from the company was the gatekeeper who made the contact with interview candidates and aided building trust relationship between the researcher and the interviewees.
  - **Correct Data:** The correctness of the data aggregated by the interviews refers to the interpretation of the researcher about what the interviewee actually said. To ensure this, all the interviews will be recorded taking permission from the interviewee so that any misunderstandings due to incomplete interview notes will not occur. Moreover, the interpretations of the interview transcriptions were sent back to interviewees for validation and receiving feedback.

- **External Validity:** External validity is the ability to generalize the findings in a way that they will be interested for other people from different cases [15]. We have identified the following threat and the action to mitigate the effect:
  - **A single company:** One of the risk factors that make it hard to generalize the findings is that the study has been conducted in only one company. Due to the time and effort constraints it was not possible to include other companies. Because we had to create a real environment to understand the real problems and usefulness in an interpretive study where users experience leads to results. Creating real environments in different domains and companies are costly and time consuming. However, to minimize the effects we have interviewed with employees from different units and...
different nationalities. Moreover, the context of the case study has been described in detail and it shows that the findings can easily be generalized and mapped to other large-scale organizations that are involved in software development.

- **Specific in-house knowledge management tools:** The analysis about usefulness of semantic knowledge management systems can be biased according to the employees’ opinion about the existing systems. They might have made compared the proposed system with the existing one and provided biased results. For instance, if the existing system would have a search engine that is as powerful as Google, we could have had a more healthy analysis about usefulness of semantic search.

**Reliability:** Reliability refers to the issue about finding the same results in case of a replication of the same study in the same setting. That is, the collected data and analysis should not be dependent on the specific researcher [15]. The following threat was identified related to reliability:

- **Data Interpretation:** Researcher's interpretation of the data gathered from the interviews is a threat that is related to subjective interpretation of the researcher. To cope with this threat, the interviews will be recorded and the participants will be asked to validate the interpretations after the analysis of the researcher. Hence, this threat will be mitigated and the collected data in case of a replication will lead to parallel results.

- **Correct Data:** Another threat to reliability is possible misunderstandings about the questions that are asked to interviewees. Interviewees might have misunderstood the questions and hence provide different answers. This threat is tried to be reduced by keeping the questions as simple as possible. Open-ended questions are preferred so that the participants are encouraged to talk and express their opinions openly.

**Internal Validity:** Internal validity is concerned about the validity of causal relations in explanatory case studies. It is related to the unconsidered factors that might have an impact on the relation. Since we are not conducting explanatory but exploratory study and we do not have causal relations, this validity threat is irrelevant for us. However, there can be inferences about the design of semantic solutions and information retrieval methods that are overseen and hence stay as an open threat.
4 RELATED WORK

This section is composed of previous studies that have been made in the area of this thesis. We will present the related work with the focus on semantic knowledge management systems as seen in the knowledge map in Figure 5. The main focus of the related work chapter will be semantic knowledge management systems, which is presented as the biggest bubble in the figure. The sizes of the bubbles represent relatively how much focus will be given to the corresponding concept. The distance of the bubbles represent how much of the relationship between two bubbles will be discussed in this study. Relationships between bubbles that are relatively further will be covered in less detail compared to others.

Semantic knowledge management systems are related to various topics that are demonstrated in the map and these topics will somehow be covered here. Moreover, the aim is to have a software engineering approach while conducting this research. In order to have an idea about the relation between software engineering and semantics, we will first summarize the studies that correlate these two and specify the position of this study.

![Knowledge Map for the Coverage of the Related Work](Image)

4.1 USE OF ONTOLOGIES IN SOFTWARE ENGINEERING

In this section, we will present several usages of ontologies in different phases and aspects of software engineering. Software engineering is composed of several processes that are really complex and contain vast amount of information and knowledge. Hence the same problem with the Web exists also here, which
is the difficulty of finding the right knowledge or keeping track of information. Hence Semantic Web technologies have been applied to different processes of software engineering in order to formalize information, improve access from different physical locations, improve universal information retrieval and allow checking and pairing different concepts and information [46].

Here a classification of the usage of ontologies in various software engineering activities is presented [45]:

1. **Software Process Ontology**: Software projects are developed according to a certain software process in order to manage complexity and communication. A software process ontology can be used for automation, formal reasoning, searching for specifications, ensuring process conformance, etc. [26]. The ontology will define activities, phases, artifacts and relations between them. For instance, in a waterfall model, requirements engineering would be a phase and it will have activities like elicitation, specification, verification, etc. Each of these activities might have artifacts and can be traced and utilized with semantic technologies [25].

2. **Requirement Ontology**: Software requirements specifications can be mapped to an ontology via mapping functional requirements to different concepts such as events, actions and their relationships. These requirements can be viewed as behaviors of the system as a reaction to different scenarios and meaningfully queried by semantic query languages [24].

3. **Software Architecture Ontology**: Many diagrams that are used to describe software architecture and design can be defined by ontology. An architecture document specifies the components of the system and their interactions and constraints. This information can be kept in the concepts and relations of an ontology [23]. With a similar approach; ontologies can be used for flow, state and UML diagrams to describe application logic or software design [21, 22].

4. **Domain Ontology**: Domain ontologies are used to specify the structure, scope and all the concepts of a certain area. They can be used for knowledge reuse in software engineering such as documents, source code, etc. [20]. Reusing the existing information can save time and reduce the effort.

5. **Implementation Ontologies**: Implementation phase produces several sources of different information, which is hard to track and retrieve. Ontologies can be used to organize and manage these artifacts. For instance, object-oriented source code can be mapped to an ontology with its packages, classes, methods, attributes, etc. [19]. This will improve the developers understanding of the code and also improve the information retrieval facilities. Even more, the source code can be automatically generated via a transformation from the ontology [17].

6. **Documentation Ontologies**: To communicate and share knowledge in large-scale organizations, documentation is one of the most significant mediums. Plenty of information is stored in these documents, which is usually forgotten or lost in a very large pool. Hence loads of valuable information stays unreached and this costs redundancy and time loss. A documentation ontology can be designed which would cover various concepts that might appear in the software documents [18]. This ontology can be utilized in order to extract knowledge directly from the pool of documents instead of searching for a document and finding the right information inside the document via reading it.

7. **Document Ontologies**: There are different types of documents that are created for storing different kinds of information. There can be documents about customer related contents, software artifacts, guidelines, business processes, products and so on. To organize the produced documents according to their types, a document ontology can be used [45]. For instance, a document
ontology created for Semantic Wikis, models the document types and their relationships with other documents [78]. The relationships can be with templates of documents, other document types or previous versions of the same document.

Document ontologies aim to organize documents whereas documentation ontologies aim at organizing information stored inside the documents.

8. **Other Ontologies:** Semantic technologies can be applied to many other activities in software engineering to create a formalization and conceptualization. Many activities that can be structured as concepts and relationships have been modeled with ontologies such as software maintenance, quality, testing and defect [45].

All these ontologies are being used to improve software development. Their aim is to help software engineers to manage and better understand large amount of information in a shorter period of time. Although there are good examples of the usage of these ontologies, the area is still evolving and the usage of semantic technologies in software engineering will increase in the coming years with the improvements in Semantic Web technologies. The drawbacks for now are that, constructing ontologies and implementing a Semantic Web enabled tool are heavy tasks that require serious work. However, after the definition of ontologies, it is very flexible and easy to modify according to the changing needs of the organization [55].

This focus of this study is more related to documentation ontologies as we are trying to extract knowledge from unstructured documents with the help of ontology based Semantic Web technologies. Moreover, domain ontology can also be considered to cover all the knowledge in software engineering, which automatically covers the documentation ontology. In the following sections, we will go deeper in Knowledge Management in software engineering and its usefulness with ontologies.

4.2 **KNOWLEDGE MANAGEMENT AND SEMANTICS**

Enterprise knowledge management (KM) refers to formally managing information resources so that information access and knowledge reuse will be facilitated [79]. It has attracted the attention of researchers more and more with the exponentially increase in the number of documents and information on the Web and in corporations. This led to many changes and challenges in Information Science. The first breakthrough was accomplished in information retrieval engines by the development of relevance ranking systems instead of full-text indexing [80]. However these developments do not satisfy the current needs of large-scale organizations in effectively managing, sharing and understanding information.

Eppler defines Knowledge Management as a systematic approach that regards implicit and explicit knowledge as a key resource for handling the knowledge at individual, team, organization and inter-organizational level in order to improve quality, time and cost effectiveness and innovation [81]. The reason why we need another breakthrough in KM systems is that the enterprises are still not aware of what they actually know due to the deficits of existing KM systems [82]:

- **Expert workers spend too much time looking for necessary information:** Experienced employees who are highly paid lose a lot of time searching for a piece of information that is needed to finish their task.
- **Essential knowledge is only available is some people’s heads:** Not every kind of knowledge can be documented and be made available to other employees. In these situations individual communication to transfer know-how might be necessary. In that case, it is not always easy to know who the “brain” of that area is in the company.
• **Rich information is buried in a stack of documents and data:** Although the knowledge and experiences are documented, it is not easy to find the relevant document and extract the right knowledge from various irrelevant data.

• **Repeated decision mistakes due to ignoring previous experiences:** In current systems, there is no easy way to gather necessary experiences from previous projects for not doing the same mistakes again. It is simply impossible to scan all documents about a project and extract right lessons.

• **Delays in deliverables and drops in product quality:** Since the flow of information does not work smoothly due to the problems with sharing and managing, it causes problems with managing the time and quality.

The motivations that lead us to solve these issues and overcome information overload problems are behind the importance and usefulness of knowledge reuse in corporations. Corporate memory can be reused:

• to avoid loss of know-how of an expert after his retirement or resign,
• to exploit the experience obtained from previous projects,
• to exploit the knowledge map of the company for organizational strategy,
• to improve information flow and communication in the enterprise,
• to improve learning of the new employees,
• to integrate different know-how of an organization [121].

Although corporate memory is substantial for the reason mentioned above, most of the knowledge in the organizations is available either as unstructured text or multimedia [55]. Hence, it is a big challenge for companies to manage and maintain information in their intranets.

The vision with Semantic Web involves at this point. It aims to turn all available information to machine-readable format by annotating documents with meta-data. The usage of meta-data along with ontologies will constitute an explicit formalization of the information and allow more advanced search mechanisms and reasoning systems. Ontologies have an important role here as they form a shared understanding of the domain, which provides agents to process information and communicate [83].

Semantic Knowledge Management systems include many different subsystems. Davies provides a general architecture as in the figure below, which consists of 4 main stages [55]:

1. **Knowledge Acquisition:** This is the phase where the natural language processors extract information from large amounts of unstructured or semi-structured documents. Knowledge is acquired not only from documents but also from humans in order to create ontologies.

2. **Knowledge Representation:** The acquired knowledge is stored in the created ontology. OWL is the latest ontology language to represent these data.

3. **Knowledge Maintenance:** Ontology middleware is used to support, maintain and use knowledge bases.

4. **Knowledge Use:** Information access tools provide end users to find, share, browse and organize knowledge extracted from the pile of documents.
4.2.1 Inhibitors in Implementing Ontology-based Knowledge Management Systems

In recent years, ontologies are increasingly used as a key technology in sharing and reusing knowledge components. Ontologies are widely used knowledge engineering, artificial intelligence, natural language processing, information retrieval, etc. [83].

However, several challenges still exist related to applying ontologies in real-world settings. Some critical issues related to working with ontologies in real-world enterprise applications are presented below:

- **Building Ontologies**: It is a difficult, time-consuming and expensive process to build thorough ontologies [84]. First of all, ontologies require a consensus in the community where members might have different visions for the domain. The solution is either to merge a number of lightweight ontologies that are built by different groups of people or use rigorous formal ontologies developed by a consortium or a standards organization [84]. However, merging ontologies will emerge various problems of consistency and building a standard ontology for a domain requires considerable collaborative effort and analysis [86].

- **Managing Multiple Ontologies**: It is almost impossible to place all the knowledge required for an organization in a single ontology [85]. Hence, it will require integration mechanisms for using multiple ontologies. However, this goal requires a huge effort due to size and frequent problems about mapping or integrating ontologies [86].

- **Managing Evolving Ontologies**: An organization’s knowledge structure would often change and evolve. Hence, these changes should be reflected to the ontologies constantly [85]. Having a protocol about this process is an important challenge [55, 87]. For instance, new functionalities
might be added to the existing systems, there can be changes in the domain and so on. Adapting the ontologies to the changes might lead to problems about consistency of the ontologies.

- **Scalability**: Enterprises usually do not centrally manage knowledge management systems [85]. Individual departments or units might create smaller system for their own needs. Furthermore, a large amount of information might already exist outside the knowledge management systems such as databases, file systems and so on. Creating an ontology-based system that can centrally manage all the information might be problematic.

- **Metadata Creation**: Creating metadata related to the entities in the ontology requires an effort [88]. There are various semantic annotation tools to automate this process. The most suitable one that is closer to satisfying the needs should be chosen and adapted to accomplish successful creation of metadata.

### 4.2.2 Related Tools for Semantic Web

There are numerous tools that are developed in the vision of Semantic Web. This section provides an overview of the tools that might be related to developing a Semantic Knowledge Management System.

1. **Natural Language Processing Frameworks**: The first step for a KM system is the knowledge acquisition and to acquire information from unstructured text, there are several frameworks that can process plain text and extract concepts out of it. **GATE** (General Architecture for Text Engineering)\(^5\) is one of the most commonly used frameworks and has several plug-ins and integration capabilities [77]. It is built by Sheffield University Natural Language Processing (NLP) group. It has many flexible language processing components that rely on finite state algorithms and the Java Annotation Patterns Engine (JAPE) language. It is widely common with its precision for entity recognition and suitable for research as it is open source software. Moreover it is commonly used in the semantic world due to its full support for ontology integration. It has been utilized in ontology-based information extraction projects such as Multiflora, hTechSight and MIAKT [89].

   On the other hand there are other frameworks like **OpenPipeline**\(^6\) and **Smila**\(^7\) but there are is not much of a research about them in the academia. IBM has produced **UIMA**\(^8\) framework, which is an enterprise semantic search tool, but it does not provide full integration and support for ontology [90]. Another tool is **OpenNLP**\(^9\) from Apache and it supports many NLP tasks such as tokenization, segmentation, named entity recognition, etc. However it accomplishes these tasks via its built-in tools not via integration of an external ontology.

2. **Ontology Editors**: When it comes to Knowledge Representation, there are many tools to create, manage and edit ontologies. **Protégé**\(^10\) is one of the most commonly used open source ontology editors by developers, researchers and corporations. It is being developed by Stanford University and it provides a user-friendly interface to build ontologies, knowledge-based tools and applications thanks to its support for plug-in extensions. GATE also has integration support for Protégé tool.

---

\(^5\) GATE (General Architecture for Text Engineering), [http://gate.ac.uk](http://gate.ac.uk)
\(^6\) OpenPipeline, [http://openpipeline.com](http://openpipeline.com)
\(^8\) UIMA Unstructured Information Management Applications, [http://uima.apache.org](http://uima.apache.org)
\(^9\) OpenNLP, [http://opennlp.apache.org](http://opennlp.apache.org)
\(^10\) Protégé, [http://protege.stanford.edu](http://protege.stanford.edu)
TopBraid Composer\textsuperscript{11} is another powerful ontology editor that is compliant with W3C standards. However, they do not provide full functionality in their free version of the framework. There are many other tools in the market such as Neon Toolkit, Hozo, Knooldl, and so on, but Protégé is the most extensive and research-oriented one based on our research.

3. Semantic Annotation Tools: There are quite a lot of semantic annotation tools and frameworks available in the market. They differ in terms of their standard format, ontology support, supported document format, design, ontology learning, and so on. Some of these tools provide manual automation and some are semi-automatic or automatic.

Uren, et al provide a comprehensive work on the analysis of different annotation tools and frameworks and comparison of them \cite{49}. Different tools are specialized on different aspects of semantic technologies and serve different usage scenarios. In the scope of this thesis, we will not go into details of each of these tools but rather present a summary of the analysis with Table 6 provided below which is aggregated in a doctoral thesis work by Fernandez \cite{48}.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Amaya</td>
<td>RDF(S), XLink, XPointer</td>
<td>Annotation Server</td>
<td>HTML, XHTML, XML</td>
<td>XPointer</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Mangrove</td>
<td>RDF</td>
<td></td>
<td>HTML, email</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vannotea</td>
<td>XML</td>
<td></td>
<td>MPEG-2, JPEG2000, Direct3D</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OntoMat</td>
<td>DAML+OIL, OWL, SQL, XPointer</td>
<td>OntoBroker Annotation Server</td>
<td>HTML, DeepWeb</td>
<td>XPointer, Pattern Matching</td>
<td>Yes</td>
<td>Supervised Learning</td>
</tr>
<tr>
<td>M-OntoMat-Annotizer</td>
<td>XML, RDF(S), DOLCE</td>
<td>Ontology Server</td>
<td>MPEG-7</td>
<td>Yes</td>
<td>Genetic Algorithm</td>
<td></td>
</tr>
<tr>
<td>SHOE Knowledge Annotation</td>
<td>SHOE</td>
<td>Ontology Server</td>
<td>HTML</td>
<td>Yes</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>SMORE</td>
<td>RDF(S)</td>
<td>Ontology Server and Ontology Editing</td>
<td>HTML, text, email and images</td>
<td>Yes</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Open Ontology Forge</td>
<td>RDF(S), XML, XLink, XPointer, DublinCore</td>
<td>Local Editable Ontologies</td>
<td>HTML, text, images (SVG)</td>
<td>XPointer</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>COHSE annotator</td>
<td>DAML+OIL</td>
<td>Ontology Server</td>
<td>HTML (via DOM)</td>
<td>XPointer</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Lixto</td>
<td>Wrappers</td>
<td></td>
<td>Stores Annotated Page</td>
<td>Yes</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>MnM</td>
<td>RDF(S), DAML+OIL,</td>
<td>Ontology Server</td>
<td>HTML, text</td>
<td>Yes</td>
<td>Supervised Learning</td>
<td></td>
</tr>
</tbody>
</table>

\textsuperscript{11} TopBraid Composer, \url{http://www.topquadrant.com/products/TB_Composer.html}
<table>
<thead>
<tr>
<th>Tool</th>
<th>Ontology/Document Format</th>
<th>Learning Method</th>
<th>Supervision Needed</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Melita</td>
<td>RDF(S), DAML+OIL</td>
<td>Local Editable Ontologies</td>
<td>Yes</td>
<td>Supervised Learning</td>
</tr>
<tr>
<td>Parmenides</td>
<td>XML(CAS)</td>
<td>Clustering to suggest additions</td>
<td>Yes</td>
<td>Unsupervised Learning</td>
</tr>
<tr>
<td>Armadillo</td>
<td>RDF(S)</td>
<td>HTML</td>
<td>Yes</td>
<td>Unsupervised Learning</td>
</tr>
<tr>
<td>KnowItAll</td>
<td>HTML</td>
<td></td>
<td>Yes</td>
<td>Unsupervised Learning</td>
</tr>
<tr>
<td>SmartWeb</td>
<td>RDF, RDF(S), OWL</td>
<td></td>
<td>Yes</td>
<td>Unsupervised Learning</td>
</tr>
<tr>
<td>Pankow</td>
<td>HTML</td>
<td></td>
<td>Yes</td>
<td>Unsupervised Learning</td>
</tr>
<tr>
<td>AeroSWARM</td>
<td>OWL</td>
<td>Local Ontologies</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>SemTag</td>
<td>RDF(S)</td>
<td>HTML</td>
<td>Yes</td>
<td>Unsupervised Learning</td>
</tr>
<tr>
<td>KIM</td>
<td>RDF(S), OWL</td>
<td>KIMO</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Rainbow Project</td>
<td>RDF, WSDL/SOAP</td>
<td>Shared Upper-level Ontology</td>
<td>Yes</td>
<td>Supervised Learning</td>
</tr>
<tr>
<td>h-TechSight</td>
<td>DAML+OIL</td>
<td>Ontology Editor</td>
<td>HTML</td>
<td>No</td>
</tr>
<tr>
<td>WICKOffice</td>
<td>Microsoft Smart Documents</td>
<td>Microsoft Office</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>AktiveDoc</td>
<td>HTML RDF</td>
<td>HTML</td>
<td>Yes</td>
<td>Supervised and Unsupervised Learning</td>
</tr>
<tr>
<td>SemanticWord</td>
<td>DAML+OIL</td>
<td>Word</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Mappie</td>
<td>HTML OCML</td>
<td>HTML</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Thresher</td>
<td>RDF</td>
<td>Ontology Personalization</td>
<td>HTML</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### 4.2.3 Ontology-based Knowledge Management Systems

As explained in the previous sections, KM systems are composed of various different steps and corresponding tools. It requires a systematic methodology and considerable amount of time and expertise to extract and formalized knowledge from unstructured data and to develop a platform that can find, share and manage information. Hence, we will have a look at the research on knowledge management platforms that provide all these functionalities together.

#### OntoShare

OntoShare is an organizational knowledge management system that promotes sharing of information between people who have mutual concerns or interests [55]. It is an ontology-based tool that places the profiles of the users as the center of attention. That is, the interests of each user are modeled with an ontology and this information is extracted from the activities of the user. Every time a user shares some information, the system first performs a text analysis in order to extract the theme of the document, which will constitute a brief summary of the content. Then the system scans all other users’ profiles in order to look for a strong match between the content of the document and the users’ interests. When there is a relation strongly enough, then the system emails the corresponding user to inform about the new
As seen in the ontology, general information about the employees and the documents are kept as RDF data and the relations of the concepts, tools and languages with these employees and documents are stored and available to extract. OntoShare provides many semantic search capabilities as well as a keyword-based search supported semantically by the concepts and user profiles. The user can search for documents that they might be interested in, modify annotations of the existing documents and also search for people that are interested in a certain area.

**KIM (Knowledge and Information Management Framework)**

KIM is a platform for semantic annotation and semantic search over several kinds of information sources. It is used for information extraction from large-scale data pools based on an ontology and a huge knowledge base [92].

KIM comes with an upper-level ontology called PROTON, which has about 300 classes and 100 properties in OWL Lite. This ontology covers most general concepts like names of people, locations and organizations along with number, dates and so on. It has also a KIM World Knowledge Base (WKB) that has about 200,000 entity descriptions to provide background knowledge for the commonly known entities. KIM keeps the ontologies and the knowledge bases in the SESAME based Owlim RDF(S) repository.

Moreover, KIM uses GATE framework for information extraction processes and Lucene from Apache as a retrieval engine [93]. Lucene has been adopted so that it allows indexing by entity types and measure relevance with respect to entity types. The following figure shows architecture of a KIM-based system.

---

KIM not only provides full-automatic semantic annotation but also allows retrieving information based on the metadata that have been created. This brings a new perspective to information retrieval, as the user is able to make a “pattern search”. That is, a semantic query can contain entities that are known or extracted before, relations between entities and attributes of these entities [80]. That means, a user can for example find out the names of the organizations in a specific location that have more than 100 employees in one single query. In this case, organization would be an entity, location and employee number would be a relation and that specific location and the number 100 would be the attributes.

**Semantic Wikis**

Wikis are also a way for large organizations to share all kinds of information and can be used for knowledge management. A Wiki is a hypertext environment that provides collaborative editing of Web pages. The main focuses of Wikis are openness, ease-of-use and modification [94]. Studies show that characteristics of Wikis show that they can be used for knowledge sharing. However there are some limitations of Wikis that prevents them from usage as a knowledge management tool. Wikis do not provide structured access to the data and do not support knowledge reuse [95].

Semantic Wiki idea has occurred from the idea of combining the Semantic Web technologies with Wikis in order to solve the mentioned problems. In summary, a Semantic Wiki provides the following functionalities:

- Annotation of pages to create formal descriptions of resources.
- A retrieval mechanism that allows semantic search capabilities to make queries not only with keywords but also with patterns, topics and semantic relations.
- An automatic or semi-automatic meta-data extraction system to simplify the annotation process.
Figure 9 presents an example of the architecture of a Semantic Wiki that consists of 5 components [95]:

- The **user interface** is usually a Web server where users can enter text and annotations and browse terms from the content of the shared ontology.
- The **page server** manages the versioning, access control and handles the conversion of the pages to Wiki interchange format (WIF).
- **Parser** is used for parsing the text for annotating and transferring this information in the Wiki interchange format WIF.
- **Content store** contains the RDF triples and enables information exchange between agents.
- **Data analyzer** is used for finding the relations between different resources in order to browse between related pages.

Semantic Wikis are useful for information collection and offer much more improved navigation and search capabilities compared to ordinary Wikis. However, it is open to discussion whether they would satisfy the needs of an organizational Semantic KM system or not. Surely, they can be used knowledge sharing in organizations but not as the only means of sharing system due to the diversity of information resources in large-scale enterprises.

**ACTIVE**

Another EU funded large-scale research project is **ACTIVE**[^13], which is coordinated by British Telecommunications. It is a 3 years project that was held from March 2008 to February 2011 with a budget of €11.9 Million [96]. Active aims to increase the productivity of knowledge sharing via prioritizing the information and knowledge delivery through an understanding of the current context of a knowledge worker [97]. That is, a filtering mechanism provides to the user only the information that is contextually related to the user’s current task or project. The users are involved in creating and shaping their context of work via creating tags manually or automatically by their behaviors. The idea is based on the fact that users are generally busy with several different tasks during the day and they have to switch and concentrate on one another constantly.

Active knowledge software has been applied and evaluated in 3 case studies in the consulting, telecommunications and engineering industry. Most of these cases are focusing on customer support units of large-scale organizations. They aim to reduce the response time to the customers and get better and quicker in taking advantage of sales opportunities. However, it can be used to overcome the challenge of information overload in a software development environment. Active Knowledge Workspace (AKWS) is a software program that is designed to integrate the components that provide various functionalities developed through Active project [98]. AKWS can work with standard office automation tools like Microsoft Word, PowerPoint and Outlook as can be seen in Figure 10.

[^13]: ACTIVE, [http://www.active-project.eu](http://www.active-project.eu)
There is also a semantic wiki platform developed under this project that is called Semantic MediaWiki [99]. It provides wiki’s general collaborative features such as enabling easy information sharing focusing on the content and also it lets users to make structured semantic search based on the relations in the ontology such as products and expertise.

4.2.4 Application of Ontology-based Knowledge Management in Different Domains

Organizations from different domains feel the need to use some sort of knowledge management in order to systematically structure, maintain and reuse information [100]. This way, they can build a general understanding about how a certain system works or see the big picture about the sources and structure of information. Ontology is one of the most important necessities to develop an advanced KM system. It provides a shared vocabulary to retain the knowledge. There have been several studies to create ontologies for conceptualizing different domains such as biomedical, information sciences, tourism and so on. In this section we will try to present some of the works that have been done in creating domain ontologies.

- **Biology Domain:** Biology is one of the most advanced domains in building ontologies to structure information. Several scientists utilize ontologies to formally define the biological terms systematically. It is specifically appropriate to define with ontologies due to the hierarchical structure of biological data. For instance, they have developed Gene Ontology and Cell Cycle Ontology to model the biological components and processes related to molecular biology and cell cycle research [34, 35].

There are many projects that make use of Semantic Web technologies to structure and integrate information in this domain. BioGateway forms a knowledge base that integrates data from various public sources and allow querying with a graphical user interface [36]. Bio2RDF project similarly
collects documents from public bioinformatics databases to form a global data warehouse [37]. There are many more projects such as CardioSHARE, LarKC, RDFScape, SEMMAS and so on, which aim to integrate and structure different information sources in this domain. While many of them achieved to satisfy their goals in data integration, some of them failed due to unsuccessful user interface designs, ambiguous objectives or poor agreement on semantics [100].

Due to the high number of ontologies developed in this domain, they have built ontology repositories in order to facilitate access to several ontologies easily. UMLS (Unified Medical Language System) is one of the biggest with over 140 ontologies and OBO (Open Biomedical Ontology) is another ontology library publicly available [38, 39]. However, the variety of ontologies creates another problem about mapping different ontologies to each other to aggregate information and problems related to diversity of ontology languages [101].

**Financial Domain:** Ontologies have also been used in financial domain in order to structure the domain information and improve data extraction and analysis. For instance, Cheng et al. have proposed an ontology-based business intelligence application to improve the data mining and the statistical analysis processes [102]. They intended to use the application for decision support claiming that knowledge management is an important aspect of decision-making.

They have created a very simple ontology that can satisfy their needs. The ontology had classes for the data, models, parameters and reports. The system supported manipulation of these concepts in order to come up with the best model that fits the business policies. This architecture happened to be a successful prototype that can be used by enterprise corporations to increase their business intelligence capabilities like decision-making. However, this does not constitute an ontology that covers all the concepts in financial domain. It has specifically been developed for a certain use case.

**Tourism Domain:** Tourism industry utilizes heterogeneous resources related to accommodation, travel information, weather, security and so on. Hence, they need to structure this information and find an efficient way to access and retrieve related knowledge. There have been many studies on building an ontology for tourism domain. For instance, Harmonise Ontology was developed by EU Project Harmonise and it aims to provide a shared vocabulary and structure for the tourism organizations to exchange data [103]. Moreover, World Tourism Organization’s thesaurus was mapped to an ontology named Mondeca Tourism Ontology and it has all concepts in tourism domain [103]. There are many other similar ontologies that are focused on different aspects about tourism [104].

**Software Engineering Domain:** Although there are several studies that focus on developing ontologies related to software engineering processes, there are only few attempts to build a thorough ontology that covers all software engineering knowledge. The most important among them is the work to create a software engineering ontology based on Software Engineering Body of Knowledge (SWEBOK) [40]. SWEBOK is a comprehensive study where almost all researchers in software engineering community agree on the content. In SWEBOK software engineering discipline is categorized in 10 knowledge areas such as requirements, testing, quality, etc. All these knowledge areas have their own processes and concepts. The SWEBOK team defined 5 objectives to establish this project and these objectives are consistent with the reason why there is a need for software engineering domain ontology [41]:

- To specify the contents of the software engineering domain
- To provide conceptual access to software engineering
- To define a worldwide comprehensive view to software engineering
○ To set the boundaries of software engineering compared to computer science, computer engineering, project management, etc.
○ To provide a basis for curriculum development and certification material

The proto-ontology, which was created based on SWEBOK, conceptualized all the information in over 4000 concepts and 400 relations and 1200 facts [83]. The overview of the quantity of the defined elements in this ontology is provided in Table 7 below. As it can be understood from this table, proto-ontology is a very comprehensive study that covers almost all software engineering terminology.

Table 6: Quantity of Elements in SWEBOK Proto-ontology

<table>
<thead>
<tr>
<th></th>
<th>Relationships</th>
<th>Concepts</th>
<th>Facts</th>
<th>Principles</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWEBOK Main Structure</td>
<td>4</td>
<td>48</td>
<td>55</td>
<td>0</td>
</tr>
<tr>
<td>Ch 01 Introduction</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Ch 02 Software</td>
<td>24</td>
<td>240</td>
<td>72</td>
<td>0</td>
</tr>
<tr>
<td>Requirements</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ch 03 Software</td>
<td>44</td>
<td>307</td>
<td>211</td>
<td>2</td>
</tr>
<tr>
<td>Design</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ch 04 Software</td>
<td>21</td>
<td>214</td>
<td>63</td>
<td>0</td>
</tr>
<tr>
<td>Construction</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ch 05 Software</td>
<td>96</td>
<td>1001</td>
<td>165</td>
<td>7</td>
</tr>
<tr>
<td>Testing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ch 06 Software</td>
<td>44</td>
<td>706</td>
<td>140</td>
<td>0</td>
</tr>
<tr>
<td>Maintenance</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ch 07 Software</td>
<td>31</td>
<td>85</td>
<td>46</td>
<td>0</td>
</tr>
<tr>
<td>Configuration</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ch 08 Software</td>
<td>33</td>
<td>72</td>
<td>46</td>
<td>0</td>
</tr>
<tr>
<td>Engineering</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ch 09 Software</td>
<td>45</td>
<td>587</td>
<td>134</td>
<td>1</td>
</tr>
<tr>
<td>Engineering</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ch 10 Software</td>
<td>19</td>
<td>263</td>
<td>62</td>
<td>0</td>
</tr>
<tr>
<td>Engineering</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tools and Methods</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ch 11 Software</td>
<td>34</td>
<td>447</td>
<td>61</td>
<td>5</td>
</tr>
<tr>
<td>Quality</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ch 12 Related</td>
<td>12</td>
<td>171</td>
<td>32</td>
<td>0</td>
</tr>
<tr>
<td>Disciplines of</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engineering</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TOTAL 407 4141 1087 15
There are similar projects like Onto-SWEBOK, which are designed based on the 2004 Guide to the Software Engineering Body of Knowledge (SWEBOK) [105, 106, 107]. However none of them is released or publicly available because of unfinished projects due to complexity that requires huge amount of time and human resources [106].

Another attempt to create a software engineering domain ontology is OntoGLOSE which is a light-weight global ontology [86]. This project uses Glossary of Software Engineering Terminology published by the IEEE Computer Society [42]. IEEE Glossary contains 1300 terms and their definitions that are related to software engineering domain. The created ontology is composed of 1521 classes where each class has a unique meaning. Moreover 329 relationships between classes were extracted using a semantic and linguistic analysis of the text in the glossary. As a result, OntoGLOSE is the only publicly available global ontology for the software engineering domain. The ontology does not have hierarchical classification; it rather forms a simple vocabulary and relationships among them that can be used for semantic annotation. The drawback of this ontology is that it is based on IEEE Glossary which was built in 1980 and updated in 2002 which makes it not up-to-date considering the amount of advances in the last 10 years. Moreover, the fact that it does not have any hierarchy is not the ideal way to structure information.

4.3 EXPLICIT, IMPLICIT AND TACIT KNOWLEDGE

Nonaka classifies knowledge as explicit and tacit knowledge [114]. Explicit knowledge is the information that can be expressed in words or numbers and shared or transmitted formally and systematically between individuals. This kind of knowledge can easily be documented and maintained in a knowledge management system and be reused by software engineers. On the other hand, tacit knowledge is much more personal and less formalized and is based on one’s actions, experience, ideas and values. Tacit knowledge is composed of both technical and cognitive dimensions. Technical dimension refers to skills, know-how and crafts on a specific context. Cognitive dimension consists of beliefs, ideas, viewpoints, etc. That is, it shapes the way we perceive the world. Tacit knowledge usually cannot be shared via written or even verbal communication. It can be transferred via joint activities such as spending time, being in the same team, etc [114].

There is also another classification that adds implicit knowledge somewhere between these two definitions. Frappaolo defines this body of knowledge as the tacit knowledge that can be transformed into explicit knowledge with dedicated efforts and methods [115]. He focuses on this aspect as the majority of the organizational knowledge is presumed to be tacit but most of the knowledge management tools and methods focus on the explicit knowledge, as it is easier to manage and share.

Although the value of the implicit knowledge is very high, the steps to gather this knowledge require a lot of effort. First of all, the right sources should be identified and the tacit knowledge should be analyzed to decide whether it is suitable for codifying to implicit knowledge via some kind of mining and translation process. The knowledge transfer from the tacit to the explicit knowledge through implicit knowledge can be carried out with a structured interviewing approach to extract the key elements of the human thought [115].

Looking at all these classifications and sharing methods for both tacit and implicit knowledge, identification of the right person to gather the information stands as the major problem to initiate the solution. Although there are several studies about transforming this kind of knowledge to explicit knowledge, few discuss the problems related to how people find the person to contact and what kind of information they seek in these situations [11]. With the practical part of this thesis work, the possible
benefits of using Semantic Web technologies in knowledge management tools in order to find the right person with the right knowledge will be investigated and the scenarios about what they look for and when they need it will be analyzed.
5 RESULTS

This section illustrates the results gathered from all the empirical studies conducted as part of this thesis. First, the data collected from the initial interviews such as usage scenarios and summary of requirements are presented. Then, the procedures in the process of development of a simple Ontology-based Knowledge Management System are explained in detailed. Finally, the qualitative data analysis will be presented for the information gathered from the final interviews.

5.1 PROBLEMS and USAGE SCENARIOS (RQ1)

A multinational large-scale organization like Ericsson has thousands of employees all around the world and hundreds of projects running around. Considering the increasing amount of globally distributed projects in software engineering domain, communication between team members is an essential part of software development. To increase the efficiency in communication, enterprises use knowledge management tools for enabling employees to find and share knowledge digitally.

Interviewees pointed out that the problems about information overload in Ericsson are based on the inadequacy of their current knowledge management systems used. These are platforms where people can share all sorts of information like blogs, wikis, discussion boards, project contents and documents. Since all Ericsson employees, which is more than 100,000 people, use these tools; there are vast amount of documents laying somewhere in these storages where no one can know and track. They said that, all these documents and the information laying inside are not stored in a structured way and they are looking for ways to manage huge amount of unstructured data. They want to investigate how they can overcome these problems. All the interviewees mentioned that the existing search facility does not satisfy people’s needs and they need a more intelligent solution.

In addition, they wanted this intelligent system to facilitate extracting hidden data from what people share. For instance, they mentioned about finding experts or even the “brain” in a certain area. Brain refers to the most knowledgeable person about a practice, a technology or a subject. The reason why they need to find experts is that, not all the knowledge can be explicitly documented in these tools, or the contents of the related documents may not be enough to comprehend the meaning of the document. One interviewee suggested that this kind of statistical information could be used to extract summaries about what people talk about and what the trending topics are.

On the other hand, they gave the news portal example such as BBC\textsuperscript{14}, where related articles are suggested in a frame while reading another article. This feature might be useful for people to browse around and build knowledge about an area instead of having multiple searches.

All the findings and solution proposals elicited in these interviews are summarized Table 8 below.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Required Software Solution</th>
<th>Proposed Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>The data in the system are too diverse and unstructured. It’s time consuming to find the right information.</td>
<td>An easy way to browse around and search within unstructured information</td>
<td>Using alternative information retrieval methods to bring a structured approach to searching</td>
</tr>
<tr>
<td>There is too much pollution of data,</td>
<td>Search facility should provide</td>
<td>Improved search options</td>
</tr>
</tbody>
</table>

\textsuperscript{14} BBC World Cup 2010, \url{http://www.bbc.co.uk/worldcup/}
as these tools are used globally by all company employees. | advanced filtering mechanisms | Classifying and clustering approach or advanced tagging systems or semantic annotation of documents in order to browse between relevant pages like in a news portal  
There is no easy way to browse between related documents about a certain topic. Hence, to extract all knowledge about a specific topic, one has to make several searches and scan all matched documents. | Semantically related documents should be grouped together and made available for the user to browse  
| The required information is not always explicitly documented. Sometimes you need to find the right person to gather the knowledge you need. | The system should help the users to find the related people or the experts in an area  
| The search facility does not return successful results. It takes too much effort and time to find the right document among all irrelevant data. | More advanced search mechanisms that can increase the efficiency to return the right data  
| The people tend to quit using these tools due to all these problems and it leads to poor knowledge reuse that negatively affects efficiency. | Developing a new system or improve the existing one considering the needs of employees to motivate them  
| Moreover we have defined some usage scenarios that will help us to design and decide in the upcoming phases:  
• **Active search**: The users should be able to search for documents using queries. We have defined this as “Active search” where the user manages the search himself. In this scenario, the user usually uses a search bar to enter the search strings and use advanced options if necessary.  
• **Passive search**: Passive search refers to suggestion of relevant pages to the one user is reading. When the user is on a certain page, semantically related pages that can attract the interest of the user should be provided in a separate frame. The aim of this usage scenario is to speed up the finding information process via enabling the user to navigate between similar pages easily. Online news portals are given as example to this scenario where related news articles are suggested depending on the content or tags.  
• **Finding the Tribe**: Tribe in our case refers to the group of people who are working on or interested in a certain topic. Sometimes it is impossible to find the necessary information documented explicitly. This kind of implicit information can only be gathered via direct contact. However, it is a challenge to find the right person who has knowledge in that area. The shared knowledge in the system should be analyzed so that people’s interests and knowledge areas can be extracted according to the shared information. The user should be able to search for people who are related to a certain topic, search for the history of the people about what they have written and where they are subscribed, etc. Furthermore, user should be able to find the “brain” (e.g. the expert) in an area, not just the people who are participating.
- **Trends:** Due to the huge amount of information available, it is hard for the employees to extract aggregated knowledge from the knowledge management tools. Users want to find a way to see the recent trends and popular topics about what people are talking about and working on. They want to know what people are discussing mostly in a certain subject.

Considering all these problems and needs in the organization and the usage scenarios, semantic information retrieval approach has been chosen as a solution method and applied in the rest of the case study.

### 5.2 IMPLEMENTING A SEMANTIC KNOWLEDGE MANAGEMENT SYSTEM (RQ2 and RQ3)

This is the phase where we needed to make a comprehensive research and spent quite a time and effort for developing a new knowledge management system (3 to 4 person months). One of the reasons for that is the absence of information in the literature about how to implement a semantic knowledge management system. Although final solutions were presented in some studies, the way to implement to implement the solution was barely mentioned. For this reason, this section will illustrate the steps to accomplish this goal and the results gathered during the process. An important detail about the following two sections is that, they are not necessarily sequential processes; ontology building was performed simultaneously when the development attempt was made.

#### 5.2.1 Ontology Building

Building an ontology is the first and one of the most important steps in knowledge engineering. We need to have a domain ontology so that we can structure information and have a shared understanding for analyzing and enabling reuse of the domain knowledge [44]. There are many ontology development methodologies in the literature of which none of them can be considered correct or wrong [112, 55, 44]. However due to its convenience and the time constraint of this thesis work, Noy and McGuinness’s knowledge engineering methodology was followed [44].

**First step** to build an ontology is determining the domain and the scope. In our case, the domain is all kinds of knowledge that can be shared in software projects of Ericsson. That is, the ontology should cover aspects from generic software engineering domain as well as the company domain. The company domain can be considered as the projects, characteristics of projects, employees and terms related to telecommunication domain. However in the scope of this thesis, we will focus more on the concepts that are directly related to software engineering. The specific terminology of the company will be left as a future work.

The usage purpose of this ontology is to categorize all the necessary information about software engineering that might be shared in collaboration tools. Considering the usage scenarios that are defined in the previous section, we can say that the ontology should only be sufficient to cover the topics that organizational members can possibly share or mention. Hence, ontology should provide answers to questions like people’s interests, expertise, projects, locations of projects and people, etc.

**The second step** of building an ontology is considering reusing existing ontologies instead of creating a new one. It is very often that someone has done a similar job and refining and extending existing ontologies according to the needs is always a reasonable option. Furthermore, building an ontology from scratch requires a huge amount of time and expertise as ontology engineering can be seen as a separate discipline and has its own practices, methodologies and processes [113].

There have been several studies about building ontologies in software engineering. Most of these attempts were focusing on specific phases of software engineering such as requirements, architecture,
implementation, testing, maintenance, etc. [17, 18, 19, 21, 24]. However we need a universal ontology for the whole software engineering domain that can cover all the aspects. In the first sight, integrating all these ontologies from different sub-domains seems like a good idea to build a complete domain ontology. Unfortunately, it is not the case in reality, as it requires a huge effort due to its size and more importantly due to frequent issues related to integration and merging of ontologies [86].

On the other hand, there are not many projects that try to develop ontologies that fully conceptualize all the knowledge in the field of software engineering. The major efforts to achieve this goal are aiming to adopt the SWEBOK Guide as a formal ontology. Such an ontology would be a perfect choice for the scope of this thesis as it would cover all the contents and terminology in software engineering domain. Unfortunately these attempts have not yet been successful or completed due to its complexity and required effort [83, 105, 106, 107].

As a result, we decided to work with the only successfully released global ontology OntoGLOSE, which is based on IEEE’s global terminology for software engineering [86]. Although there are certain drawbacks of this ontology such as lack of coverage, being outdated and primitive; utilizing this lightweight ontology would still be sufficient for the scope of this study to reach the actual research goals.

Finally, we did not need to proceed with the next steps of ontology building such as creating classes, hierarchy, properties of the classes, instances and so on. Accomplishing all these steps requires a comprehensive study and effort, which would not fit to the scope and duration of this thesis.

Figure 11 below is a screenshot of a part of this ontology from the Protégé Ontology Editor. Definition of each class from the glossary of IEEE is annotated as a comment as seen in Figure 12.
5.2.2 Developing a Prototype Semantic Knowledge Management System

As described in section 3.3.3 under Case Study Design chapter, developing a semantic knowledge management system consists of many steps. This section will cover the discussion about the selection of tools and implementation trials and aggregated results.

Text Processing
For processing unstructured text, we have decided to use GATE due to its common usage in semantic web research and support for ontology based information extraction. GATE comes with an information extraction system called ANNIE\textsuperscript{15} (A Nearly-New Information Extraction System). Using ANNIE’s components such as tokeniser, gazetteer, sentence splitter and so on; one can extract generic information from the corpora of unstructured text. GATE can find the names of the well-known organizations, names of people, locations, numbers, etc.

When GATE and ANNIE was applied to some documents from the knowledge management system’s of the company, it was seen that the recognized entities were not enough to cover the contents and the context of the technical documents that were used.

Hence, to extract the information related to software engineering, a suitable ontology should be integrated as a language resource and necessary changes to the processing resources of GATE should be reflected.

Ontology & Knowledge Base
After understanding how to use GATE, the next step was investigating how ontologies can be involved in text processing. The decision was building a simple ontology in order to have an initial idea about the usage of ontologies.

To build and manage ontologies, Protégé was our choice of ontology editor due to several reasons. First of all, Protégé is an open source research project, which is extensively used in the academic world. Moreover, the author of this thesis has previous experience in using this tool in another academic project. Finally and maybe the most importantly, GATE and Protégé support integration for each other and support many other tools and extensions.

For this initial phase, a very simple ontology that already covers some of the content of the document was built and used for testing text processing and annotation. Later on Protégé is used to manage existing ontologies as described in the previous section.

Semantic Annotation & Ontology Population
In order to try the created prototype ontology, a semantic annotation tool needed to be chosen. Among the many argued in Section 4.2.2, it is understood that each of those tools are specialized in certain usages for certain applications and users. Some of them are already outdated such as OntoMat, SHOE Knowledge

\textsuperscript{15}GATE’s ANNIE System, \url{http://gate.ac.uk/ie/annie.html}
and Melita; Vannotea and M-Ontomat are specialized in videos and images; and some support only Microsoft Office documents, such as WICKOffice and SemanticWord. Moreover, applicability of the tools to our case was also considered. For instance, Magpie is only applicable to webpages and hence discarded. On the other hand quite a number of these tools like Mangrove, Vannotea, OntoMat, Shoe Knowledge annotator, Open Ontology Forge and Cohse support manual annotation of documents, which is not desired for this thesis, as the selected tool will directly be presented during the interviews. A full automatic semantic annotation tool is needed in order to apply and evaluate directly on the corpus of organization’s knowledge management systems. Manual annotation tools require user intervention, so they cannot directly be evaluated.

In the end, the decision was to use GATE also for semantic annotation as it supports automatic annotation of documents. Therefore, it was used for having an initial attempt to annotate a company document with the built ontology. In the end, GATE was used for natural language processing and semantic annotation and Protégé was used for building ontology. The summary for the decisions for the tools can be seen in Table 9 below.

**Mind switch:** After exploring the tool for a while and understand how things work, we realized that adapting the processing resources of GATE is not such an easy task and might require a lot of effort. First of all, building a knowledge base, creating instances for each entity of the ontology, which would be sufficient for evaluating the system during the case study cannot be done manually within the time limit of this thesis. Choosing an external knowledge base and integrating it would need an important amount of time either. Moreover, even though the knowledge base can be integrated, annotation system of GATE should be modified so that it can recognize and instantiate the relations between entities. Based on the tutorials and documentation of the framework, this requires advanced natural language processing expertise and considerable amount of research and effort.

In addition, after this step a query engine with a graphical user interface needs to be implemented which would require significant amount of time as well. Considering the time constraint, we decided to make a mind switch and look for alternative solutions. We looked for integrated platforms that use GATE and but also provide semantic search facilities with ontologies.

<table>
<thead>
<tr>
<th>Implementation Steps</th>
<th>Related Tools (See 4.2.2)</th>
<th>Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text Processing</td>
<td>Natural Language Processing Frameworks</td>
<td><strong>GATE:</strong> Selected due to its common usage in Semantic Web research and full support for integrating ontologies. <strong>OpenPipeLine, Smila, OpenNLP, UIMA:</strong> Not chosen as they do not support integrating an external ontology.</td>
</tr>
<tr>
<td>Ontology &amp; Knowledge Base</td>
<td>Ontology Editors</td>
<td><strong>Protégé:</strong> Selected due to its common usage in ontology research, support integration with GATE and author’s familiarity. <strong>TopBraid Composer:</strong> Powerful but commercial product. Not chosen, as it is not free.</td>
</tr>
<tr>
<td>Semantic Annotation &amp; Ontology Population</td>
<td>Semantic Annotation Tools</td>
<td><strong>GATE:</strong> Selected to reduce time spent to learn other tools and its support for ontology-based semantic annotation. <strong>Others:</strong> Discarded due to outdated products, lack of</td>
</tr>
</tbody>
</table>
SEKT Project
Semantically Enabled Knowledge Technologies (SEKT)\textsuperscript{16} project is one of the biggest research efforts today in the area of semantic technologies. It is a 3 years project which was coordinated by major contributors to semantic technologies such as John Davies from British Telecom and research groups from University of Sheffield, University of Karlsruhe and so on. Several components like ontology editors, upper-level ontologies, semantic annotation tools and semantic reasoners were developed under this project, which are commonly used in semantic world right now.

We decided to use KIM platform as it fits to our requirements and usage scenarios that we have defined. Among the others, OntoShare is not available online, Semantic Wiki and ACTIVE cannot be applied to existing knowledge management systems, they need to be built as a new system. Moreover, they do not satisfy the initial requirements about solving search problems.

KIM (Knowledge and Information Framework)
KIM supports a full-automatic semantic annotation of documents and comes with an upper-level ontology and a semantic search engine. KIM is based on GATE for natural language processing purposes. It comes with an ontology named PROTON\textsuperscript{17} that covers the most general concepts such as named entities (people, locations, organizations) and concrete domains (numbers, dates, etc.). However, a more specific ontology can be integrated to KIM according to the needs of the domain.

We have analyzed our requirements and what KIM offers and concluded with the following results:

- KIM’s general ontology covers most of the aspects we defined in the scope of the ontology. We do not need too many changes in the ontology design such as classes and relations. We can integrate the OntoGLOSE domain ontology and we will enable KIM to recognize domain specific concepts. We do not need very specific relations between classes as our usage scenarios are only based on extracting who is talking about what topic. As long as the topic is recognized, it would be sufficient to satisfy our requirements.

- If the domain ontology is not enough to cover all the aspects, as it does not have any concepts from the last 10 years and many other concepts about the company domain, we can extend the knowledge base of KIM with an external knowledge base. For instance, KIM supports integrating KIM with DBpedia\textsuperscript{18} which is a structured knowledge base containing all entries from Wikipedia. Considering the fact that Wikipedia contains all the terminology that we need for software engineering as well as telecommunications domain, integrating DBpedia would be a very convenient solution for the goals of this thesis.

\textsuperscript{16} SEKT, \url{http://www.sekt-project.com}
\textsuperscript{17} PROTON Ontology, \url{http://proton.semanticweb.org}
\textsuperscript{18} DBpedia, \url{http://dbpedia.org/About}
KIM provides “Boolean Search” which is a keyword-based search and corresponds to “Active Search” in the defined usage scenarios. Moreover, it provides “Structure” and “Pattern” search in order to search for the extracted relations which can be used for the “Finding the Tribe” scenario. “Facet search”, which is a relational filtering mechanism, can also be used to for the same scenario. “Timeline” search, which shows the popularity of selected entities over a period of time, can be used for “Trends” scenario that we have defined. On the other hand, KIM also provides navigating between documents according to their relations, which enables “Passive Search”. The search frame of KIM and the “Structure” search menu can be seen from the figure below:

First of all, we tried to integrate DBpedia with KIM. To be able to use the DBpedia instances, we need to integrate DBpedia ontology with PROTON, which is the generic ontology of KIM. However, we cannot take all the DBpedia ontology and map it to PROTON as it would cause too much complexity. Therefore, we took Person, Organization and Abstract classes of DBpedia and mapped it to PROTON so that we will have the names of all well know people, organizations and also abstract topics which contain the software engineering related topics. Figure 14 represents a part of the PROTON ontology and its Person and Organization classes.
However, due to poor documentation and lack of expertise in the area, we could not manage to integrate DBpedia to KIM knowledge base. Integrating DBpedia consists of many steps like mapping of ontologies, adding statements for each entity in DBpedia, setting labels of each entity, setting up gazetteers for each newly added class, adding Jape transducers and so on. Hence, the documentation for such complex tasks should be clear and detailed so that developers with no extensive experience can also accomplish.

Therefore, we decided to integrate the software engineering domain ontology OntoGLOSE. Although it would not be satisfying all our needs, it would be a good starting point for a further study to modify and extend the coverage of it.

After integrating this domain ontology, we figured that the system still does not recognize the entities in this ontology. As a backup solution, we tried to manually integrate this ontology to the actual PROTON ontology via using Protégé Ontology Editor. Since OntoGLOSE does not have any hierarchy, it was easy to manually copy its classes to the other ontology. However, we neglected the relations as they were not interesting for this context. In the end, we still could not manage KIM to recognize these terms. The company was consulted by e-mail, but due to the long delays in getting a reply from the support team, there were only 2 e-mail exchanges with the company, which was not enough to fix the problems.

Therefore, the system was evaluated during the interviews as it is. The discussion board pages from one of the collaboration tools of Ericsson were downloaded manually and loaded to KIM as a corpus. The observations about the system are summarized in Table 10 below.
Table 9: Summary of Observations from the Application of KIM on a Discussion Board Corpus

<table>
<thead>
<tr>
<th>Observation</th>
<th>Possible Improvements</th>
</tr>
</thead>
<tbody>
<tr>
<td>The collaboration tools contain quite a number of abbreviations about the projects and Telecommunications domain, which cannot be recognized by the system.</td>
<td>List of all abbreviations in the company domain and their definitions should be collected and added to the instance base.</td>
</tr>
<tr>
<td>Most of the domain specific words about software engineering are not recognized.</td>
<td>A heavyweight software engineering domain ontology or DBpedia should be integrated.</td>
</tr>
<tr>
<td>Alignment of the annotated texts and comments in the discussion board is not successful. It makes it difficult to read.</td>
<td>The alignment needs to be configured specifically for the structure of the discussion board (nested comments, etc.)</td>
</tr>
<tr>
<td>Not all the names can be recognized by the system. Especially surnames are problematic.</td>
<td>List of all company employees should be collected and added to the instance base.</td>
</tr>
<tr>
<td>The relations about the person and the specific topic he is commenting about cannot be extracted successfully. However, a relation between all the topics in that discussion board and that person is defined.</td>
<td>To extract the actual relation here, a serious work has to be done. A syntactic pattern has to be created according to the structure of the documents and proper rules have to be defined so that satisfactory results will be gained.</td>
</tr>
<tr>
<td>“Structure” and “Pattern” search is not currently useful for finding the topics that people are working on or interested in.</td>
<td>The accuracy of these searches is dependent on the improvement in the previous observation above. If the correct relations can be extracted, these searches will perform much better.</td>
</tr>
<tr>
<td>“Timeline” and “Facet” search provides satisfactory results to find the popularity of a certain topic and finding the tribe. However, there is no general way to search for recent ‘hottest’ topics.</td>
<td>The search bar should enable user to look for popular topics without selecting an entity.</td>
</tr>
</tbody>
</table>

5.3 QUALITATIVE DATA ANALYSIS (RQ1, RQ2 and RQ3)
This section will explain the analysis of the qualitative data collected from the interviews. As the analysis method, thematic coding analysis has been used following the guidelines of Robson [16]. The structure of this section will be based on the codes and themes extracted from the interview data and their relations.

5.3.1 Extracted Themes and Codes
Thematic Coding Analysis method is applied to the transcriptions that are created from the audio records of the interviews. This method is applied based on the guidelines that are described in section 3.3.3 under Case Study Design chapter. A total of 7 main themes have been identified from the interviews and the explanations of each theme are presented below:

- **Usage Issues**: These are the problems that the interviewees mentioned, about the usage of their collaboration tools. It refers to the poor usability of the tools and the way people use these tools without any control and common understanding.

- **Finding Information Issues**: Interview subjects have a lot of problems when it comes to finding documents. It arises from the poor the search engine they currently have, the complex and huge structure of the tools and the problems about choosing the right document from the search results.
• **Gathering Information Face-to-face:** Interviewees consult other people when they cannot find the documents they look for. This theme refers to the two scenarios they do in such cases. They either ask others about how to find the right document or they directly try to gather the knowledge they look for from an expert.

• **Improvement Suggestions:** Interview subjects had certain ideas about how to overcome the identified problems about information overload. This theme is focused on their suggestions to the existing systems and to the proposed semantic system. First of all, they all want a more intelligent and accurate search engine with advanced search facilities. Moreover, they wish to have some additional features and better use of tags.

• **Usefulness of the Semantic System:** The subjects were asked to provide opinions about the usefulness of the new system in regard to solving the mentioned problems. They mostly mentioned about the help of the system to find the right people or experts, finding the right documents and extracting hidden information from the statistics that can be collected from this tool.

• **Ontology and Filtering:** This theme is concerned with employees’ opinions about the use of ontologies and ontology choice. Since, ontology is used as a filter in semantic search; they provided their thoughts about what they would like to see in the ontology so that they can get better hits in the search results. None of them was interested in a complete software engineering domain ontology but they proposed ontologies based on document types, Telecom domain and organization specific frameworks.

• **Concerns:** Interviewees also presented their concerns about the proposed semantic system’s capability to solve the information overload problem. The success of this system cannot be achieved totally unless it is used in the right context, people trust it and there is a common understanding about the usage of these systems. Moreover, the cost to implement and maintain such a system should not be disregarded.

Table 11 below summarizes the themes and codes extracted from the interview texts and their categories based on the topics in the interviews, which also mean the research interests.

**Table 10: List of Themes and Codes and their relations with the research interests**

<table>
<thead>
<tr>
<th>Themes</th>
<th>Topics</th>
<th>Information Overload Problems</th>
<th>Implicit Knowledge</th>
<th>Semantic System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usage Issues</td>
<td>Usability of tools</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Lack of supervision and understanding</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Finding Information Issues</td>
<td>Understanding the context of document</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Complex structure</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Technically incapable search engine</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gathering Information Face-to-face</td>
<td>Finding an expert to consult</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Asking help to find the relevant information</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Improvement Suggestions

<table>
<thead>
<tr>
<th>Usefulness of the Semantic System</th>
<th>A smarter Google-like search engine</th>
<th>Search mechanism</th>
</tr>
</thead>
<tbody>
<tr>
<td>New features to existing tools</td>
<td></td>
<td>Tagging</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Representation of results</td>
</tr>
</tbody>
</table>

#### Usefulness of the Semantic System

- Finding people
- Finding documents
- Extracting statistical and hidden data

#### Ontology and Filtering

<table>
<thead>
<tr>
<th>Usefulness of ontologies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document Type ontology</td>
</tr>
<tr>
<td>Telecom domain ontology</td>
</tr>
<tr>
<td>Organization specific ontology</td>
</tr>
</tbody>
</table>

#### Concerns

<table>
<thead>
<tr>
<th>Agreed procedures and understanding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost to the organization and the users</td>
</tr>
<tr>
<td>Right context</td>
</tr>
<tr>
<td>Trust</td>
</tr>
</tbody>
</table>

### 5.3.2 Usage Issues (RQ1)

When interviewees were asked about their problems about using knowledge management tools efficiently and reusing the existing knowledge that is stored in the document stores, they did not only mention the problems about searching documents but also some general problems about using the tools.

For example one of the interviewee complained about how slow the current collaboration tool is and it restrains them from using it. People just share documents there but nobody reuses those documents. The same interviewee also mentioned that the Wiki page that the existing tool provides is not actually a Wiki page but rather an online word processor where the focus was the formatting and layout instead of the content. He said “It wasn’t usable and formatting was totally messed up. Normally a wiki page has very little formatting possibilities, it focuses on content.” He mentioned that as a result of these formatting problems they had to create their own wiki pages. He also generalized it by stating, “We and probably most of the products in Ericsson have started our own wiki pages that follow the standard wiki format. So there are a lot of wiki pages and hosts for developers.”

The project manager that was interviewed mentioned a lack of top-down structure that starts from a very simple and intuitive page that guides the user about the area and goes deep into different branches with informational hooks. He expressed this by saying, “When there is a fresh employee, I would love to be able to give this person a starting page to have a look, read through, browse around, look at interesting links and build his own knowledge based on informational hooks.” That means search is not the only problem because sometimes you do not have anything specific to search, you just want to gather general knowledge and you need a guide for that. Two other interview subjects also mentioned similar problems about fresh employees and they declared that they provide the links of all necessary documents to newcomers or they send those documents by e-mail.
Three of the interviewees indicated that the problem is not only about the tools but the way people use these tools. The processes, procedures and structures to handle the information set are not well understood by everyone. One interviewee pointed out, “There are sort of frameworks and guideline about how to do this. But we haven’t had time to set it and run in the organization to get all the people understand the way we would like to work. More than understand, accept and take part and building up to this. Because everyone is overly stressed with deliveries of software now.” The interviewee states that stress with the deliveries of software due to time limit is the reason behind this problem. Furthermore, two of the interview subjects mentioned that it is difficult for the old organizations with an older generation to get used to using new trends like discussion boards and networking tools. They are not accustomed to using such portals and as a reason for not using one interviewee stated the following: “We have our own facebook for instance. But we are out of time for that. That’s a lot of things to do. If you have your network internally you can always ask somebody else. You don’t have to ask there.”

On the other hand, one interviewee stated that he can find what he looks for only if it is announced specifically in a global area. If it is not advertised, there is no way to find that information in a local repository of a region or unit because it is between hundreds of other documents without any structure and context. However, not everybody announces his work or project in these global areas. He indicated that it is his biggest problem by saying, “Right now the biggest problem about search is not everyone creates an entry about what they did in this global knowledge object area. If there is an entry there, I can find what I look for.”

One of the interviewees remarked the same issue with a different perspective and indicated that the problem relies on the absence of supervision and control in these systems. People place documents in random locations that are totally irrelevant. One can find irrelevant personal documents in a location where you look for a solution of a certain technical problem. He gave the following example: “We have very diverse knowledge sources like forums, knowledge objects and discussion boards. People create an entry about food recipes in the knowledge object area. There is no supervision mechanism in these tools and hence the quality decreases like this.” As a result of lack of control and governance quality decreases and hence usability of the tools reduces.

The summary of the issues mentioned here is provided below in Table 12:

<table>
<thead>
<tr>
<th>Codes</th>
<th>Problems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usability of Tools</td>
<td>➢ Slow tools</td>
</tr>
<tr>
<td></td>
<td>➢ Poorly formatted Wiki</td>
</tr>
<tr>
<td></td>
<td>➢ No top-down structure that goes from simple to complex that can guide a fresh employee to build up knowledge</td>
</tr>
<tr>
<td></td>
<td>➢ Older generation is not used to use social media</td>
</tr>
<tr>
<td>Lack of Supervision and Understanding</td>
<td>➢ No common understanding and cooperation about how to use the tools</td>
</tr>
<tr>
<td></td>
<td>➢ No common understanding about where to place the documents</td>
</tr>
<tr>
<td></td>
<td>➢ No supervision or control about what people share and where they put them</td>
</tr>
</tbody>
</table>
5.3.3 Finding Information Issues (RQ1)

All the interviewees stated the problem about the lack of knowledge reuse and they believe they have huge amount of information stored in the organizational storage. One interviewee pointed out this problem by saying, “I would say, sometimes it is that complicated to find things so we usually don’t reuse much. There is a lot of information laying around that nobody actually knows about.”

For the knowledge reuse, they expressed the following scenarios for the situations that they need to search for information via a search bar or browsing around to find the right document:

- To gather related knowledge when a task out of the usual domain or area comes.
- To understand the meaning of a drawing of a model when there is no explanatory text or the text is not sufficient to understand the whole concept.
- To check what others did in a similar task when a new task comes that needs to be started from zero.
- To look for similar projects about the same solution, when the solution document at hand is not enough to understand how to apply.

The majority of the interviewees (5 out of 7) strongly agreed on how difficult it is to find the right information when they face these scenarios. The main reason all of them pointed out is the poor quality of the existing search engines.

Two of the interviewees actually stated that they never use the search engine anymore because they cannot find anything. However, these interviewees said they know where things are thanks to their experience and knowledge, so they do not have critical problems about finding information. The other subjects told that they have to use the search engine but they can never get good hits so it takes a lot of time and effort to find what they look for. Sometimes they cannot find it at all so they try to ask other people instead.

About the usage of search engine, one subject gave this important information about how often people use the search bar: “People normally don’t use the search engine here. We did a survey before with 200 people here and the majority of developers are never using the search engine. Those that are using it just use a couple of times per year for things that are not related to work. 1/3 of people didn’t even know about search engine.” This shows an interesting fact about how often developers reuse knowledge.

Three of the interview subjects denoted that they do not believe there is any ranking algorithm behind the search engine and it retrieves randomly wherever the keywords exist in document names or inside documents. One interviewee pointed out the ranking problem as follows: “We have ranking problem when we make a search. The first results are usually irrelevant. You have to scan the next pages of results to find the right document.” Two of them complained about the absence of an advanced search option where they can create filters. Two interviewees specifically signified that when they enter more than one keyword, the search engine retrieves randomly the documents that either of the keywords exists. But the users are actually looking for documents that both terms exist.

In addition, all subjects agreed on how huge and complicated the structure of these tools and only the ones with a significant experience stated that they do not have any problems browsing around this structure. However, they also admitted that browsing with ease in this structure is depended on experience and said that, “The information is in the systems now, but the question is how to find it. Relies very much on experience and previous knowledge to find that information in the system. Even if you don’t have a formal
mentor, you still need to have someone to help you into the system. The threshold for getting to the information is rather high. You need a guy preferably who is built into the information structure itself.”

The subjects who remarked that they cannot find documents if they try to browse in the structure specified that the tree structure is too complex and they do not know in which branch of the tree they should go into while searching. The tree can branch according to region, customer unit, service unit or even project size. If you do not know the size of the project you look for, you cannot find it. Hence, these subjects stated that they do not look for information in this way anymore. To find the location of a project document, they need the exact path or the link to that document. So, they always bookmark and save the links to the documents they want to reuse.

Moreover, since there is no common understanding and control about where to place the documents, browsing around this structure becomes even harder as people might put them anywhere. For instance, some projects’ document repositories are not structured at all and all the documents exist in the same level. The content of each project repository depends on the project.

On the other hand, one of the interviewees specifically focused on understanding the context of the document problem when he is performing searches. When the search bar returns several results, it is very troublesome to understand which the document is a useful one. Sometimes it requires reading most of the document to figure out what it is about. One interviewee gave the following example about understanding the context: “I search for something and I get a document and it looks fine. I try to find out in which context it is written. Sometimes you don’t even get which context it is. Then you have to understand the context. For example, a term called policy. It most of the times means some kind of network rule, quality service. But in the business domain, it is actually a higher specification of business rules. So you have a problem when you try to find something. Are you in the business domain and talking about business policy or are you in the network domain and talking about network policies.” Furthermore, when a special term, acronym or an abbreviation is searched, the results can come from different domains such as business or network. The same term or the abbreviation can have different meanings in different domains and all these documents show up in the search results. It is very difficult to filter out the irrelevant information without reading the documents especially when the user has a mindset about what he looks for and he gets confused about what he reads.

Six interviewees mentioned about the naming and numbering format of the documents according to document type, product and project. This code helps user to understand the type and content of the document. Four interview subjects indicated that it is much easier to find documents when these codes are used. They think coding makes everything very well defined and structured. However, this convention started to disappear later and it is not commonly used in every department. One interviewee explained this as follows: “I really liked before when we had the hierarchy with naming convention for the document. And you can know exactly what you get with the name of the document. So you could search in that naming and numbering format. Then you know what it talks about from the name. That disappeared later. Now it’s not strict. Not everyone does it. It was very easy to say I would like this kind of documentation and search for that.” One interview subject on the other hand, stated that finding documents according to these codes was a big problem before, because that was the only way to search for documents.

The summary of the issues mentioned here is provided below in Table 13:

<table>
<thead>
<tr>
<th>Codes</th>
<th>Problems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understanding the context of document</td>
<td>➢ Filtering out irrelevant search results without</td>
</tr>
<tr>
<td>Complex structure</td>
<td>Having to read the whole document</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td></td>
<td>➢ No general information about the domain and context of the document in the search results</td>
</tr>
<tr>
<td></td>
<td>➢ Problems with acronyms, abbreviations and terms that can mean different things in different domains, groups or time</td>
</tr>
<tr>
<td></td>
<td>➢ No convention for naming and numbering documents anymore</td>
</tr>
<tr>
<td></td>
<td>➢ Huge and messy document stores</td>
</tr>
<tr>
<td></td>
<td>➢ Complicated tree structure that makes it hard to browse</td>
</tr>
<tr>
<td></td>
<td>➢ No common structure about the content of project repositories</td>
</tr>
<tr>
<td></td>
<td>➢ Requires experience to find documents and help people show the right places</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Technically incapable search engine</th>
<th>Requires time and effort to reach the right document from results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>➢ No ranking algorithm</td>
</tr>
<tr>
<td></td>
<td>➢ No intelligent algorithm about which part of the document the keyword exists</td>
</tr>
<tr>
<td></td>
<td>➢ No advanced search criteria</td>
</tr>
<tr>
<td></td>
<td>➢ Inaccurate search results</td>
</tr>
</tbody>
</table>

### 5.3.4 Gathering Information Face-to-face (RQ1)

Knowledge sharing and transfer between employees does not happen only by means of explicit documents. People find themselves in need of consulting other people when they are looking for information. In the scope of this thesis, it has been referred as part of implicit knowledge.

Interview subjects mentioned that there are cases that they go and ask people when they cannot find the information they are looking for in the system or when they do not think that information can be gathered from the documents.

Five interviewees indicated that when they cannot find what they are looking for, they ask help from other people. Three of them denoted that they first ask someone experienced in their group, a project manager or a solution architect, about where to find the related documents. That is, they do not directly ask about their problems but they first try to find documents in the system by their help. If they still cannot reach a solution this way, then they try to find someone who is an expert in the area to spare some time to share his knowledge.

In order to find the right person to ask, interviewees stated different scenarios. Experienced interview subjects indicated that they already know who is experienced in each area. One of them said: “I have been here long now. So you normally meet people from everywhere and you send email, call them, sms them and ask about things and they come back in a day or something most of the time. Or they know someone to forward.” Others stated that they ask their manager to help them to find the right person to ask. One of the
interviewees told that in such a situation they send e-mails to everybody in their working group to find out if anybody has any knowledge about the interested topic.

One interview subject claimed that it is usually better to ask someone instead of searching. The reason for that is firstly about the hideous search engine that takes a lot of time and effort, and secondly it could be more beneficial to talk to someone as they can give different suggestions that can lead to solutions in different directions.

Three interviewees denoted that sometimes the information that they look for cannot be gathered from any document in these tools. They might need to talk to someone for specific operational questions about how and when to do certain things. On the other hand, even if documents exist about the same problem and solution it may not be enough. One interviewee indicated this issue as follows:” That information I ask to people, exists in document form but the thoughts behind it, the things that lead to that solution are not always there. Talking to the guys who did it or thought about it is even more helpful to make the decision.” So they might feel the need to talk to people who contributed to the document.

The summary of the scenarios and actions mentioned here is provided below in Table 14:

<table>
<thead>
<tr>
<th>Codes</th>
<th>Scenarios</th>
<th>Actions</th>
</tr>
</thead>
</table>
| Finding an expert to consult | ➢ After it is not possible to find it by searching in the system  
➢ When that information cannot be gathered from any document (operational information)  
➢ When it is better to ask people to understand the thoughts behind  
➢ Instead of time consuming search | ➢ Ask someone experienced in that area  
➢ Ask someone you know to divert you to an expert  
➢ Ask one level up to your manager to divert you to the responsible  
➢ Send e-mails to your group to find someone with knowledge |
| Asking help to find the relevant information | ➢ When it is not possible to find it by searching in the system | ➢ Ask someone experienced to show the location of documents  
➢ Ask someone who is responsible for the task |

5.3.5 Usefulness of the Semantic System (RQ3)

All of the interviews agreed that the overall approach that comes with the semantic systems seems very useful. Although they all remarked that their current search engine is totally incapable and this one cannot even be compared to the existing one, they pointed out some of the strong points of semantic search. One interviewee explained this comparison by saying, “Currently, we have a very very simple search mechanism. It doesn’t even have ranking, prioritization capabilities. Since we don’t have these basic skills, this is quite a next generation.”

Most of the interview subjects’ initial concern was finding documents that contain the information that they are looking for. Hence, rather than finding information related to any other entity in the ontology, they were focused on the documents. All of them found useful to search for documents with its relation to people, topics and authors. But they suggested different ontology alternatives, which will be discussed in “Ontology and Filtering” section below.
Two interviewees found “Facet search” the most useful, as that search starts broader and narrows down based on the results of the added filters. One of them stated that, “I like the idea of refining the search. Start broader and then based on the result, narrow it down. That’s a good way to search. Because that’s the way you search normally, going from broader to specific.” One interviewee indicated that, being able to see all the extracted information without even making a query is very useful because you can see beforehand if it is worth your time looking into the database.

Most of the interviewees also agreed on the usefulness of this system about finding people, which was previously defined in this study as “Finding the Tribe” in usage scenarios. One subject told that he does not need that functionality because he knows everybody he needs. Others stated that finding experts and knowledgeable people is quite a common scenario in Ericsson as there are experts about almost every area and their knowledge is indispensable. One of them added that, “Finding the right person is used very often in Ericsson. It is a huge organization. Not everyone knows everything but you can find an expert in almost every area. But you don’t know who they are. You should be very active in forums, etc, but it needs spending time on it regularly. But this facet search is very very useful.” They all agreed that correct recognition of software engineering and telecommunication terms by the natural language processing tool is crucial for the success of this search engine. Two interviewees indicated that extracting organizational information about people’s position, roles and locations would not be necessary or useful since that information is actually stored somewhere in company database. However, they would like to integrate that database, which is not directly accessible for employees, to this semantic system so that they can utilize the organizational data while searching.

Another point that the interviewees mention was the statistical data that can be gathered by means of this new system, which is similar to the “Trends” in usage scenarios. By analyzing what people talk about, a significant amount of hidden data might be collected. For instance, people’s skills and interests can be identified by processing the entries they are involved in. Furthermore, a summary of what people talk about can be extracted with this system to make organizational decision. For example, an interview subject gave this example: “If we have a lot of people working with GUI in a unit, or majority of graphical people in Ericsson work in this city, maybe we should set up a center there. That is, the statistics that we need are available directly there. Even if people don’t update their profiles, they write documents so they will be recognized anyway.” Another interviewee suggested that this kind of information about trends and statistics can be useful for sales people who go to customers.

The summary of the benefits, scenarios and search facilities mentioned here is provided below in Table 15:

<table>
<thead>
<tr>
<th>Codes</th>
<th>Usefulness</th>
</tr>
</thead>
</table>
| **Finding Documents** | ➢ *Structure Search*: Useful for searching documents according to people, topics and authors  
➢ *Facet Search*: Useful for starting from broad and narrow down  
➢ *Facet Search*: Useful for seeing all extracted entities before searching |
| **Finding People**   | ➢ *Finding the Tribe*: Useful for finding experts if you do not know a lot of people  
➢ Extracting people’s position, role and location is not needed |
Extracting statistical and hidden data

- **Trends**: Useful for finding what people talk about
- Useful for identifying people’s skills and interests
- Useful for making organizational decision based on statistics

### 5.3.6 Improvement Suggestions (RQ1 and RQ3)

When participants were complaining about their problems with the existing system, they provided solution options for the existing systems and improvement possibilities to the proposed semantic system.

When interviewees were explaining what they need, they all mentioned about a smarter, more intelligent search engine that is similar to the Google and so on. Four of the interviewees expressed ranking as the most important factor, as they want to see the most relevant results at the top of results. They want certain documents prioritized so that they appear first in the search results. One interview subject mentioned about seeing the results based on the strength of the documents, which is similar to the most powerful answer feature in discussion forums. The same subject suggested having various other advanced search options like seeing what other people searched, adding advanced filters to the query and so on.

In addition to this, subjects wanted to have some new features that can be useful for them to find the right information or person. For instance, one interviewee wanted to see the not only the current position of a person but also the previous positions of him so that he can have an idea about the background of the author. He explained the reason as follows: “It would be nice to see the position of the author of the document at the time of creation and also the positions the author had before. Depending on the history of that person and his specialization, he might be the one to pinpoint to get more information. Because they dived into the subject.” The same subject also thought it would be useful to see not only the authors or responsible people for a document but also all the people involved in helping the writer to bring in that knowledge. Another feature suggestion was having a summary part for every document, which would be used like an abstract in an academic paper in order to give an idea to the user what the document is about. Moreover, most of the interviewees found understanding the meanings of acronyms and abbreviations troublesome, and one interviewee suggested having a list for these in the end of every document. He said, “There should be abbreviation and acronyms list in every document. Some acronyms are common knowledge for your own group but not for whole Ericsson. Or it can mean something at a time, but not another time.” One interviewee thought it would be very useful if he could favorite some documents or repositories that he works often so that he can access easily without searching every time and his search results would be based on these areas that can interest him. Furthermore, 4 of the participants wanted to keep and enhance the usage of the coding convention based on the name and number of the document. They find it more structured, hierarchical and easy to search.

As far as the proposed semantic system is concerned, interviewees mainly made comments about the content of the ontology as it shapes the search mechanism. However they expressed some of the improvements that can be applied to the system. First of all, one interviewee stated that he does not want to be locked into a set of pre-defined queries when you make a structured search based on the entities and their relations in the ontology. He would prefer to write a search sentence; the system should semantically process it and if it matches any of the relations in the ontology then results should be retrieved based on that, otherwise it should perform a standard search.

Another suggestion was being able to search for entities that do not satisfy the relation that is specified in the search pattern. For instance, searching for people who talking about and also who is not talking about a certain topic shall be available. He explained his concern by stating, “For example if competitors in our
Moreover, three interviewees suggested jumping to similar documents based on the overall content of the document. The existing system only allows jumping between documents based on a single annotation inside the document. This suggestion was identified as “Passive Search” in the usage scenarios in the beginning of the case study.

Almost all the interviewees at some point mentioned about tags and majority of them pointed out the importance of an intelligent tagging system. They indicated that tags are very useful for understanding the context and content of the document and search engine should consider the tags in a smart way in the search algorithm. However, they all agreed that tags in the current system are not used efficiently at all. One interviewee stated that people do not know the purpose of tags so they just write something or leave it empty. Another interviewee told that people do not have the patience to write proper tags so they do not pay much attention. He says people should not be forced to tag.

Three of the subjects proposed to have a closed solution for the tags. One interviewee said, “In case of open-ended solution, someone will eventually tag in a different way and it will be problematic.” The current system has a tag library and people can choose tags from there but they can also add any tag to the library without any supervision and control. The interviewee found this system messy and not usable.

However, most of the interviewees opposed to a solution that is fully automatic. That is, they want to be able to modify the tags of documents even if they are not the authors and add new tags to the tag library. But the tag library should be very wide and well controlled. Hence, they prefer a semi-automatic tagging system. This also applies to the semantic system proposed as the annotation and tagging is fully automatic. Moreover, one interviewee suggested binding tags with the entities in the ontology and being able to search according to those tags. Currently the semantic system uses the most frequent annotations as tags but it is not possible to modify them. Another interviewee suggested having descriptions for tags. This is possible when the annotations are used as tags as recognized entities have already their descriptions.

Furthermore, some participants suggested improvements in the representation of the results. For example, one of the subjects wanted to see the tags or summary of the document directly in the search results so that it can help him to choose the document with the right context. Another practitioner proposed to have results collapsed according to the ranking and organizational structure. In this way, you can have traceable trees based on location, product and so on.

The summary of all these suggestions made by the interviewees is given below in Table 16:

<table>
<thead>
<tr>
<th>Codes</th>
<th>Suggestions</th>
</tr>
</thead>
</table>
| Smarter Google-alike Search Engine | ➢ Ranking and prioritization  
➢ Advanced search criteria  
➢ Suggestions from what others searched |
| New Features to Existing Systems          | ➢ History of the positions of employees  
➢ List of people provided the knowledge in the document  
➢ Having a summary or abstract in every document  
➢ List of acronyms and abbreviations for every |
<table>
<thead>
<tr>
<th>Search Mechanism</th>
<th>document</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>➢ Being able to favorite some documents and repositories</td>
</tr>
<tr>
<td></td>
<td>➢ Naming and numbering structure for the documents</td>
</tr>
<tr>
<td></td>
<td>➢ Not pre-defined structured search patterns but processing the free text search query</td>
</tr>
<tr>
<td></td>
<td>➢ Searching for entities that match and also search for entities that do not match</td>
</tr>
<tr>
<td></td>
<td>➢ <em>Passive Search</em>: Jumping to related documents based on the content</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tagging</th>
<th>➢ Intelligent tagging system that search engine considers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>➢ Suggestions for tags</td>
</tr>
<tr>
<td></td>
<td>➢ Not completely free text, a closed solution</td>
</tr>
<tr>
<td></td>
<td>➢ Well controlled, wide tag library</td>
</tr>
<tr>
<td></td>
<td>➢ Any user should be able to modify tags (Semi-automatic)</td>
</tr>
<tr>
<td></td>
<td>➢ Binding tags to entities in ontology</td>
</tr>
<tr>
<td></td>
<td>➢ Descriptions of tags</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Representation of Results</th>
<th>➢ Should show tags or summary of the document in the results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>➢ Collapsed view of the results based on ranking and organizational structure</td>
</tr>
</tbody>
</table>

### 5.3.7 Ontology and Filtering (RQ2)

Practitioners were generally excited about the use of ontologies and making structural searches with respect to ontology. However, neither of them was directly interested in seeing a software engineering ontology with all the practices in the domain. They stated that their search scenarios are more about terms in Telecom domain.

A practitioner mentioned his concerns about the use of ontologies as the ontology can quite big and have a lot of branches, which makes the ontology itself complex. Breaking down the information to branches again and again might lead people to lose the track of it and get lost. He stated, “Although usage of taxonomies is good for human brain to understand, people might easily get lost in it if it gets too large.” Hence, creating a complete ontology that has all the information structured in a certain domain would probably be too enigmatic and cause information overload problems anyway. Another interviewee foresaw this and suggested being able to search in the ontology as well. This can prevent people from getting lost in the branches of the ontological structure.

Another point the practitioner mentioned was the fact that there is no complete tree structure. He explained his concern about using ontologies by saying, “Structuring terminology information with an ontology will not solve all the problems. You don’t have a complete tree structure, it’s more like a graph where one term can be related to different nodes not to one parent” This interviewee suggested keeping the ontology very general and focusing on the tagging system.
When it comes to choice of ontologies, interviewees were asked if they would like to see the knowledge areas based on SWEBOK in the ontology structure so that they can use them to extract and filter information. However, all of them stated that they do not really need that kind of queries and one subject stated that these knowledge areas and lifecycle phases are not that clear when you have agile development. However they declared their choice of ontology that would be useful for them.

Document types and domain were the most desired ontologies by the interviewees. 3 subjects specified that they would like to see the document types in the ontology so that they can filter the documents according to type. All the interviewees were asked to tell about their usage scenarios for these collaboration tools and the type of documents they deal with. For the document types they gave the following examples: Product description documents, project planning documents (requirements, user stories), design documents, business process modeling documents, architectural documents, release packages, CPI (customer product information) documents, operational documents, test reports, proposal, pre-sales and after sales documents, installation documents, solution documents, Interface description documents, user guides and so on.

One interviewee mentioned about the problems about document type by stating, “The problem with document types is that there is no common structure about where to place these documents in a project repository. It can be anywhere.” Hence they cannot easily find a specific document for a certain project or product. Interview subject denoted that if the semantic system can recognize the type of the document automatically by processing the content of the document, it will be a huge benefit for them.

Another common suggestion was a domain ontology based on Telecom operations and services. Four interviewees mentioned that when they search for a term, the results come from all different domains that are not interesting for them. When they were asked about what exactly they mean when they say domain, one interviewee only stated that he would like to see only results from network (technical) domain or from business domain. The other three participants were a little bit more specific and they gave the following examples: Operation support systems (OSS), Business Support System (BSS), Charging, Mediation, Service Delivery Platform, Customer Relationship Management (CRM) and so on.

They suggested using eTOM19 (enhanced Telecom Operations Map), which is a guidebook that defines the most common standards for business processes in the telecommunications industry. It defines three major process areas:

- **Strategy, Infrastructure, and Product** that covers planning and lifecycle management
- **Operations** which covers the core of operational management
- **Enterprise Management** covering corporate or business support management.

As it can be seen from Figure 15 below, which details the processes and their categorization, most of the domains that the interviewees mentioned are classified in this business process framework.

---

Interview subjects indicated that they would like to have a combination of domain, document type and organizational structure of the company when they create a search query. Organizational structure refers to the existing structure of the tools, which is based on location, region, unit, project and so on.

Another subject proposed to have Ericsson’s project management framework PROPS-C as an alternative to the classical lifecycles defines in SWEBOK. This framework includes business readiness process, sales process and project management process. These processes are composed of phases such as analysis, planning, monitoring, execution, contract management and so on. The interviewee suggested searching for documents according to these defined phases.

Same subject proposed to have Ericsson’s Product Catalog domain in the ontology. He told that, “There are products and services such as network optimization and project management. When I make a project somewhat related to a product in the catalog domain, I do not enter this project as a product because it is only a small part of it. Normally I put this document as a project under my unit. If I don’t advertise this as a knowledge object or something, nobody can find this project. If I can relate this project to somewhere in the product catalog then it will increase its possibility to be found.” This is important because other people might have similar projects that are related to only some part of the main products, but information about these projects are lost in local repositories. Hence, relations between projects and the products from the catalog can be useful for finding documents.

To sum up, interview subjects denoted diverse opinions about the use of ontologies and what type of ontology they would like to see. However, domain and documentation seem to be most dominant ones. The summary of opinions about ontologies and filtering is provided in Table 17 below:

### Table 16: Overview of the Opinions About the Choice of Ontology

<table>
<thead>
<tr>
<th>Codes</th>
<th>Opinions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usefulness of Ontologies</td>
<td>➢ Too large ontology can get too complex and people</td>
</tr>
</tbody>
</table>
5.3.8 Concerns (RQ3)

Along with all the improvement suggestions and comments about usefulness of the semantic system, some interviewees expressed their concerns about the success of such a system.

One interview subject thought that this system would be more useful when there is no structure at all such as wikis and blogs that are used especially in agile development. He described when it would be more useful by saying, “Provided you have the right context, specifically if you work on wiki or blog based environment, which is the way we are going with agile and lean as people try to get rid of documents.” He said within these conditions, you can extract valuable information from there. However, existing tools have their own structure and existing problems will be solved when people have an understanding about how to utilize this structure and embrace it.

One of the interviewees denoted the possible new problems that can occur by using this system. The costs for implementation, migration and maintenance of this tool should not be discarded and should be analyzed carefully. Moreover if there will be a cost for the end users about the usage of the tool, then that should be considered as well.

Two interviewees remarked that there are two main reasons for the information overload: People and tools. This can solve the tool part of the problem but you still need to build a set of processes, procedures and a common understanding for the people about how to utilize existing information. Otherwise not all the problems will be solved.

Another point that two of the interviewees mentioned was the trust issue. People need to trust this tool so that when they make a search they will believe that they are getting exactly what they want and they are not missing out anything. One interviewee explained this as follows: “It’s just a matter of getting used to it and trusting it to start using it. I have to make sure it brings the exact results. Then I’d use this.” Moreover, the annotations should be created accurately and abbreviations should be recognized in the right meaning.
The summary of all these concerns are provided in Table 18 below:

<table>
<thead>
<tr>
<th>Codes</th>
<th>Details</th>
</tr>
</thead>
</table>
| Right Context | ➢ Totally unstructured environment  
➢ Wiki and blogs in agile development |
| Cost to Organization and Users | ➢ Cost for implementing, migrating and maintaining  
➢ Cost to the end users |
| Agreed Procedures and Understanding | ➢ Problems are based on people and tools  
➢ People should embrace the usage of the system for a complete success |
| Trust | ➢ To be able to rely on results about the validity and completeness of the results  
➢ Accurate annotations |

5.3.9 Thematic Network and Relationships between the Codes

Out of the data collected from eight interviews made with the company practitioners, a total of seven themes and twenty-four codes are identified. The summary of all the identified codes can be seen in Table 19 below. These codes not only cover the problems they mention but also their opinions, suggestions, concerns and usage scenarios. The codes refer to segments of important data that are interpreted from interviewees’ answers. In order to help the coding process, a mind map from the interview data was created and can be seen in Appendix B.

<table>
<thead>
<tr>
<th>Code ID</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>Usability of tools</td>
<td>Complains about usability of the tools that need to be fixed</td>
</tr>
<tr>
<td>C2</td>
<td>Lack of supervision and understanding</td>
<td>Problems about how people use the tools inefficiently</td>
</tr>
<tr>
<td>C3</td>
<td>Understanding the context of document</td>
<td>Problems about difficulty of selecting the right document</td>
</tr>
<tr>
<td>C4</td>
<td>Complex structure</td>
<td>Problems about the existing structure of the tools</td>
</tr>
<tr>
<td>C5</td>
<td>Technically incapable search engine</td>
<td>Problems about the quality of existing search engine</td>
</tr>
<tr>
<td>C6</td>
<td>Finding an expert to consult</td>
<td>Scenario about consulting an expert to gather information</td>
</tr>
<tr>
<td>C7</td>
<td>Asking help to find relevant information</td>
<td>Scenario about consulting someone to find a document</td>
</tr>
<tr>
<td>C8</td>
<td>A smarter Google-like search engine</td>
<td>Suggestions for the searching problem</td>
</tr>
<tr>
<td>C9</td>
<td>New features to existing tools</td>
<td>Suggestions that can help to solve information overload problem</td>
</tr>
<tr>
<td>C10</td>
<td>Search mechanism</td>
<td>Improvement suggestions for the way semantic</td>
</tr>
<tr>
<td>Code</td>
<td>Category</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>----------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>C11</td>
<td>Tagging</td>
<td>Opinions about the usage of tags in the documents</td>
</tr>
<tr>
<td>C12</td>
<td>Representation of results</td>
<td>Improvement suggestions for the way the system returns results</td>
</tr>
<tr>
<td>C13</td>
<td>Finding people</td>
<td>Usefulness of semantic system to find experts</td>
</tr>
<tr>
<td>C14</td>
<td>Finding documents</td>
<td>Usefulness of semantic system to find the right document</td>
</tr>
<tr>
<td>C15</td>
<td>Extracting statistical and hidden data</td>
<td>Usefulness of semantic system to extract trends, skills and statistics</td>
</tr>
<tr>
<td>C16</td>
<td>Usefulness of ontologies</td>
<td>Usefulness of ontologies and software engineering ontology to structure information</td>
</tr>
<tr>
<td>C17</td>
<td>Document type ontology</td>
<td>Suggestion of an ontology that covers document types</td>
</tr>
<tr>
<td>C18</td>
<td>Telecom domain ontology</td>
<td>Suggestion of an ontology that covers processes in telecommunication domain</td>
</tr>
<tr>
<td>C19</td>
<td>Organization specific ontology</td>
<td>Suggestion of an ontology that is specific to organizational structure and frameworks</td>
</tr>
<tr>
<td>C20</td>
<td>Cost to organization and users</td>
<td>Concerns of practitioners about the cost of semantic system as opposed to its benefits</td>
</tr>
<tr>
<td>C21</td>
<td>Right context</td>
<td>Concerns about the context of using a semantic system</td>
</tr>
<tr>
<td>C22</td>
<td>Agreed procedures and understanding</td>
<td>Concerns about the problems that are related to people which may not be solved</td>
</tr>
<tr>
<td>C23</td>
<td>Trust</td>
<td>Trust issues about using a semantic system</td>
</tr>
</tbody>
</table>

The mentioned codes above have some certain relations with each other as shown in Figure 16 below. The groups of codes with the same color represent the identified themes. Codes are in strong relation within themes with regard to their context, and have other relations with the codes outside the theme. The codes that are more commonly mentioned in different themes and hence have more relations with other codes are represented as bigger bubbles. These are the identified relations from the thematic network map:

- C2 causes problems with the usability of tools (C1) because it causes a decrease in the quality of the tools.
- Complex structure of the tools also causes a decrease in the usability (C4 and C1).
- Since the structure is too complicated, the context of the document cannot be extracted from that structure (C3 and C4).
- As the search engine is not efficient and returns random hits from the database, people need to rely more on the structure of the tools to find documents (C4 and C5). Otherwise when the results come randomly, the context of the document cannot be understood (C3 and C5).
- To be able to understand the context easily, new features are needed for the system, which will eventually increase usability (C3 and C9 and C1).
- People would like to see a smarter search engine with ranking and prioritization skills to solve the search engine problems (C5 and C8).
- Problems about C4 and C5 cause people to ask for help from others in order to find the location of the documents in the system (C7).
• When a proper document cannot be found, people look for an expert in the area to gather knowledge (C6 and C7). Semantic system brings a useful solution for this issue (C13).
• Another usefulness of the semantic system is overcoming the problems about previous search engine to find the right documents (C5 and C14).
• From the documents’ content and authors, statistical data and people’s knowledge can be extracted (C13 and C14 and C15).
• To be able to understand the context of the documents (C3), tags should be used efficiently in the search mechanism (C11) and in the representation of results (C12). Other new features should be added to the representation of the results (C9 and C12).
• Usage of ontologies (C16) will lead to improvements in finding documents (C14) and it might be useful to overcome the problems about comprehending the complex structure of the tools (C4).
• To utilize a more useful ontology (C16), document types (C17) and telecommunication domain structure (C18) and organization specific frameworks (C19) shall be considered. The main benefit comes from understanding the type of document or the position of the document in telecommunications domain or the position of the document in the frameworks of organization.
• However, in order to measure the true benefits of ontology, its cost to the organization for creation and maintenance should be considered as well (C16 and C20). Ontologies will be more useful if they are utilized in the right context (C16 and C21).
• If the procedures about how to use the system are embraced by people (C22), then it will create a better solution. Otherwise it will be useful only if there is no structure at all and there is no need for common understanding (C21).
Figure 16: Thematic Network of the Extracted Data
6 DISCUSSION

This section will cover the discussion and synthesis of the results that are presented in the previous chapters. Some implications and comparisons will be provided with accordance to the research questions defined before. The section will be addressed around three main points and each of these subsections will be structured as follows:

- **Observation**: An introduction about the discussion topic and overall observations about the results related to that matter.
- **Related work comparison**: A comparison of the results that are collected from the empirical study and the literature.
- **Practical implications**: The implications and reflections of the results and comparisons to practitioners and researchers.

6.1 Information Overload (RQ1)

One of the aims with this research study was to find out the problems that are related to information overload in large-scale organizations. Understanding the problems is important to analyze and create solutions. In parallel with the first research question about finding information, the interviewees were asked how they face challenges about information overload when they are using their collaboration and knowledge management tools.

**Observation**: Information overload and overcoming the problems that come with it depend on many diverse factors that require to be analyzed together. The main motivation behind trying to solve those problems relies on facilitating knowledge reuse as much as possible. The main motivation behind increasing knowledge reuse relies on enhancing the quality of products and reducing the time spent.

All the interviewees without any exception admitted that they are not reusing the existing knowledge efficiently. They showed different reasons for that and told about their own ways of dealing with it. First of all, the fact that some people stopped using the search bar infers that they are not benefiting from the existing sources enough.

Based on the results, the problems and solution proposals were shaped around four major alternatives:

1. **Problem source**: Only people,
   **Solution**: Fix the people

   The group of participants who were expressing this problem and solution claimed that there are not any major problems with the tools. Tools have their own structure and it is fairly easy to find information within these tools if you are competent on the usage of them.

   They claim that the problem is about the way people use these tools. People are not aware of the processes, procedures and structures about handling the information set. Everyone is overly stressed with deliveries of software and they did not find time to understand how to use the systems. They should not only understand but also accept and take part in this.

   Their solution proposal to this problem does not lay with tools at all. The common understanding and agreement about the processes; the knowledge on who does what and who is responsible for what need to be built in people’s minds so that information overload problem will be solved.
One important detail about this group of participants is that, they have over 20 years of experience of which at least 10 years in this specific company. Hence, they know where things are in these tools and they stated that they can find what they look for without any major problem if everybody puts the documents in the right place.

2. **Problem source**: Only tools,  
**Solution**: Fix the tools

This group of interview subjects remarked the tools as the source of the problems about information overload and reusing existing knowledge. They did not mention about people when they were expressing their issues.

They expressed that the main problem is the poor quality of the knowledge management tools. They mainly complained about how the search engine is not capable to retrieve relevant results, how the structure of the system is too complicated and how usability of the tools is low.

One common observation about the people in this group was the fact that they feel the need to search for documents a couple of times a week whereas the people in the first category stated that they only perform a search when a task out of their usual domain comes, which is not very often.

3. **Problem source**: People and tools,  
**Solution**: Fix the tools

This alternative was an idea of a single interview subject. When it comes to problems, he expressed that the source of problems is depended on people and tools, but as a solution he proposed fixing only tool related problems.

He indicated that lack of supervision and control, about what people write and where they upload documents is a problem for the quality of knowledge management. However, it is not a problem that can be overcome easily. The same could be a problem for Web unless we have efficient search engines like Google. Since Web users can enter any kind of data to Internet in any desired format, Web would turn into a dump without successful search mechanisms. He claimed that the same issue threatens the knowledge management systems nowadays and the proper solution to this is an intelligent search mechanism that can help people to filter out irrelevant data.

The interviewee stated that, there will always be some people who do not follow the agreed procedures. So the solution should better rely on the tools not the people. Since there is no supervision mechanism, semantic search may be the solution to the problem.

4. **Problem source**: People and tools,  
**Solution**: Fix the people and the tools

This option was mentioned by one of the interviewees. He agreed that problems occur due to diverse reasons and he remarked that semantic search can be a solution to find documents in the right context. However, there is still a need for the people to understand how to share knowledge. If people do not share their knowledge with documents, knowledge will not be reused anyway. Moreover, people should also be aware that existing knowledge can be very useful to gain time and effort and they need to try to find it.

**Related Work Comparison**: Information overload has attracted the attention of researchers for a long time now. In a survey that Reuter conducted in 1998 showed that 42% of the participants who were executives from 16 countries said they have information overload [108]. In order to overcome this
problem in the Internet, the focus of the academia was on building better search engines that can help people find what they look for [10]. One of the breakthroughs was succeeded by Google in information retrieval in Web. However, the turn of attention towards searching in enterprise knowledge management was a bit later than this [10].

As described in the background chapter, lately the attention of researchers is turning towards Semantic Web in order to overcome information overload problems in Web. This means, the challenges about finding information have not yet been solved and the evolving technologies are being used to overcome these evolving problems.

Most of the literature that we have addressed in the scope of this thesis focuses on the solution possibilities with the usage of intelligent tools with advanced technologies. Even the ones that see the root of the problems as people aim to solve it by creating new automated solutions that can reduce the people effect [109].

**Practical Implications:** Observations show that there are four alternatives of problem definitions and solution suggestions. The number of participants is not enough to generalize any of these four alternatives. However, they can form basis for a further investigation in the organization about the employees’ needs and problems.

The background and experience of subjects can shape their opinions significantly. For instance, it is quite normal in this case that experienced interviewees who are accustomed to using these do not have serious problems with the knowledge management tools. Furthermore, the organizational role of the subjects and their usage scenarios of these tools should not be discarded. The two employees who did not find any major problem about the tools are the same people who stated that they almost do not use the search bar. One of these subjects was a project manager who expressed that he is dealing with daily operational data more often rather than static information. Therefore, another empirical research that has a more thorough coverage that would consider all these factors can be conducted to generalize these results.

As far as implications to the company are concerned, we can say that the subjects who search for information and use the search bar more often are more concerned with the quality of tools. These subjects were solution architects or software engineers who look up documents to create and apply solutions. Reusing existing knowledge is crucial for them when they deal with developing solutions. Concerns of this kind of people should be prioritized and satisfied.

On the other hand, issues related to the ways people use the systems should not be discarded. A common understanding between the organization members would be a key factor that can bring successful knowledge management and reuse. However, we can expect that a more capable and intelligent system might encourage people to use these tools more efficiently and collaboratively.

All in all, this time enterprise systems should follow the lead of Web in a closer contact and follow the new technologies and trends that are being applied in Web. Considering the fact that, current search facility of the organization is nowhere near the ones that we have for Internet, this will not be an easy process for the company. A step-by-step process is necessary to be followed in order to overcome information overload problems.

### 6.2 Structuring information with ontologies (RQ2)

Another aim of this study was to analyze the needs and the usage of ontologies to structure information and investigate its usefulness to overcome information overload problems in software engineering domain.
During the interviews, the subjects’ opinions about ontologies and the type of ontology they would like to see were collected. They provided diverse comments about the concept.

**Observation:** Interviewees were generally positive with the idea of using ontologies to have a better traceability of the documents and use it to filter out irrelevant information. However, the initial plans for the thesis were not entirely parallel with the results gathered from the data collection. The main focus was creating a software engineering domain ontology and analyzing its usefulness for software engineers while using a semantic knowledge management system.

To cover all the aspects in software engineering within a structure, SWEBOK was thought as the most appropriate way. However when the interviewees were asked about having the knowledge areas from SWEBOK in the ontology such as Software Requirements, Testing, Maintenance, Management and Quality; none of them saw the necessity to structure documents and information based on such an ontology. They said that the problems they have when searching will not necessarily be solved if they use this ontology for filtering information.

However, they wanted to see a document type ontology that could partially be covered by the software engineering domain ontology as it would cover all the software artifacts. The most common preference for the ontology was a Telecom domain ontology that covers business process frameworks in telecommunications domain. Because retrieving documents from a certain domain was a major problem. Hence they suggested a guidebook for Telecom operations (eTOM) instead of the guidebook for software engineering. Other suggestions were having the product catalog of Ericsson and creating relations between projects and product catalog; seeing organizational structure of the company in the ontology; and Ericsson’s project management framework PROPS-C in the ontology.

Overall, we can observe that their choices are mainly shaped by organization specific and business domain specific needs. The only choice that can be generalized to software engineering domain is the document ontology. However, there can still be document types in the document repository that are specific to the organization or telecommunications domain.

There were also concerns about using ontologies. Mapping a huge knowledge to a tree structure might not be very efficient as it can get too complex and cause getting lost in the ontology. Moreover, cost of creating and maintaining ontologies can be very high. On the other hand, most of the interviewees at some point mentioned about tags and about how useful they would be if there are used efficiently. Because, tags provide the initial ideas about the content of the document before reading it. Some suggested correlating tags to the ontology in order to perform the most efficient search.

**Related Work Comparison:** In the related work chapter, various ontology attempts and usages were presented in both software engineering domain and other domains. However, a complete domain ontology that conceptualizes all the knowledge in software engineering could not be found, as those attempts have not been completed yet [83, 105, 106, 107].

It turned out that participants were not really interested in such an ontology. One of the ontology suggestions was a document type ontology, which was also referred, in related work. They want to specify the type of document when they are performing a search. However, not all the documents they mention are software artifacts. There are various types of documents in the system about products, business processes, customers, guidelines and so on.

The ontology suggestion about business process frameworks in telecommunications domain was not included in the related work section. Our main focus was software engineering concepts. After all, the practitioners who are working in software development stated that they prefer to have this ontology in the
system. We realize that, ontologies that are based on eTOM exist in the literature in order to facilitate Semantic Web Services and Service Oriented Architecture [110, 111].

Knowledge representation models were presented in “Background” chapter from the most primitive and less semantically representative to the most complete in terms of semantic knowledge representation: Tags, Taxonomies, Thesauri, Ontologies. From the results collected, it can be seen that tags still constitute an important role in finding information. Hence, we can say that using a combination of tags and ontologies is also possible and those models are not necessarily alternatives.

**Practical Implications:** The most important result from the collected data is the absence of the need for a software engineering domain ontology in knowledge management systems. Considering the time and effort needed to implement such a comprehensive ontology, it is a positive implication for the organizations in case they intend to implement a semantic knowledge management system. Moreover, the possible problems that might occur by using such a huge ontology will not be an issue.

However, it can create other kinds of burdens to the organizations, as they might need to build their own ontology that satisfies the needs of employees. They cannot simply use a general software engineering ontology that might exist in the literature or in the market.

Modifying an existing document ontology and eTOM ontology according to the needs of the organization might be a rational solution considering the cost and return of investment. The combination of these two ontologies was the mostly desired choice for the practitioners.

Organizational structure already exists in upper-level ontologies but there would still be modifications there. The most important point here is that, this knowledge about organization’s employees, regions, units is already stored in a database of the company. This information should directly be retrieved and populated to the ontology. As an alternative to this, the existing system can be modified so that there would be an advanced search option to filter based on the internal structure of the organization and structure of the document repositories. Semantic solution is not an obligation here.

The suggestions about project management framework PROPS-C and the product catalog of Ericsson came from only one interviewee. The usefulness of such ontologies should be further investigated. Nevertheless, the rational given about using the product catalog of the company seems that it would be beneficial for searching for certain projects that are about a specific product.

On the other hand, even if there is no need for a complete software engineering domain ontology based on SWEBOK or so on, all the terminology of software engineering and similar disciplines should be included in the knowledge base of the ontology so that the annotations will be successful.

All in all, we saw that for investigating the ontology preferences of software engineers, the business domain of the organization should not be discarded or could even be given privileges based on the results of this study. Moreover, using tags along with ontologies is also important for structuring information. People sometimes feel the need to categorize documents in a way that they understand and think it would be more useful compared to placing it inside a pre-defined taxonomy.

### 6.3 Usefulness of semantic system (RQ3)

The main goal of the thesis was to analyze the usefulness of semantic knowledge management systems in order to solve information overload problems and enhance knowledge reuse in software engineering. Within this perspective, an attempt to implement a semantic system was made and opinions of practitioners were collected to discuss and evaluate such a system.
**Observation:** The main expectation of the interviewees was being able to filter out irrelevant data as efficient as possible. In order to create advanced filters that help to retrieve the most relevant information, creating queries based on the entities in ontologies and relations among them is the edge of semantic search. Participants were thrilled with the possibility of creating such complex, meaningful search queries.

The vision with the semantic search is to retrieve aggregated and extracted knowledge without a need for human intervention from the information sources. However, this is not exactly the feature that the practitioners were excited about. Their main goal is to find the right document among many irrelevant ones, not retrieving the extracted knowledge from processing text documents. The reason is, they need to read the entire document to understand the solution strategy, understand how to apply the solution in which kinds of situations and so on. They said that, sometimes that is not enough either and they need to go and talk to the person who wrote the document to comprehend the ideas and thoughts behind that solution.

Hence, the most useful scenario for them when performing a structured search is to choose “document” entity from the ontology and filter according to relations of that entity with domain, document type, project, product, people, location and so on. The result set would be a set of documents from which they need to figure out which one is the most appropriate document.

Nevertheless, most of the practitioners found it useful to be able to extract hidden data about interests and skills of people, statistics for the use of decision makers and so on.

The most important observation about the implementation of semantic knowledge management systems is that it requires important amount of time and resources in order to build a suitable solution. It is a process that needs to be handled systematically and requires a lot of experience and expertise. Even if there are many existing tools that can be utilized and integrated to create a semantic system, there is still a lot of work to do in order to modify these tools to specific cases. Text processing, ontologies, query engine should all be adapted according to the particular needs of the organization. They cannot be applied as it is, otherwise results will not be satisfactory.

**Related Work Comparison:** One of the visions of Semantic Web was to make the information in the documents machine understandable so that it will allow making meaningful searches that aggregate results from different sources. In our case, the main benefit of semantic capabilities is to make the system understand the context of the document by processing the text in it. Hence, people can filter the documents according to their type, domain and relations with other entities although this information is not explicitly available.

The observation presented above is consistent with the existing knowledge in semantic information retrieval. According to Finin et al., Semantic Web would produce with two kinds of information sources: First is conventional text documents enriched by annotations and second is statements that can be interpreted by machines that capture some of the meanings of the content of the documents [62]. In this case, the usefulness of the semantic system comes from those statements that are extracted based on the content of documents. However, the statements that are useful can be considered as high level, as they are the relations between the author or document itself with the overall context of the document. More detailed extractions from processing each sentence of the document are not needed. Hence we can say that extracting the context of documents was the most important benefit of semantic systems.

**Practical Implications:** Putting aside all the technical results and implications, overall it can be stated that semantic knowledge management systems could solve most of the practitioners’ problems about gathering knowledge by finding documents and experts.
However, the success of this solution strongly depends on the quality of semantic capabilities. The quality and efficiency of each process in a semantic system constitutes an equal significance for the overall success of solving information overload problems. That is, natural language processing, semantic annotation, ontology population, search mechanism should all be carefully implemented and adapted to the requirements of the organization and the existing knowledge management structure.

For instance, since the existing systems are not all plain text and have already some kind of structure in both directory and document level, text processors should be able to understand these structures and create relation accordingly. Moreover, all the terminology that is of interest to software engineers in the organization should be recognized and annotated with the right context and meaning. Subsequently, if the ontology is created according to the needs that are identified here and if it can be populated successfully, the only operation left will be the usage of accurate search mechanism for semantic information retrieval. However, it is this author’s opinion that the result sets can still be too large and cause waste of time if the semantic search does not have some kind of ranking and prioritization algorithm.

All these steps are crucial for building the trust that is needed for the users to count upon the results that this intelligent semantic system would bring. If there happens to be a problem with any of these steps, people will lose their faith to the results brought by this system and will continue using traditional approaches for searching.

Nonetheless, we saw in the implementation process that achieving these quality requirements demands substantial amount of research and effort, which should not be overlooked. Hence, the costs and return of investment (ROI) for such a system should carefully be analyzed. However, it is almost impossible and unrealistic to expect an exact measure on the ROI in knowledge management. It is not possible to quantify the time, revenue and reputation that can be lost with the decisions and developments based on outdated, incomplete or incorrect information [63].
7 CONCLUSIONS

In this thesis, the main contribution was the analysis of the usefulness and applicability of ontology-based semantic information retrieval technologies in knowledge management systems in the context of software engineering in large-scale organizations. To perform this analysis from all perspectives, we identified the existing problems, available technology, useful aspects and challenges that the organizations should bear in mind. The problems are related to the search engine and structure of the existing tools, the technology is able process documents to extract the knowledge inside, useful aspects are related to filtering out irrelevant documents and extracting people’s skills and interests, and the challenge is the necessary effort to satisfy all the needs.

In order to determine the usefulness of semantic search, a literature review was conducted, which was followed by a case study in a large-scale international telecommunications company. Case study involved interviews with a total number of 11 practitioners from the organization and implementation attempt for a simple semantic system. The outcomes of this thesis can be used as guidelines and a basis for the organizations in similar context, which are intending to build a semantic knowledge management system or looking for solutions for the information overload and knowledge reuse problems.

As a result, conclusions can be grouped in three branches according to research questions:

Problems about information overload:

RQ1: What are the existing problems about finding the right information in large-scale organizations?

Problems related to information overload are mainly about the structure of document repositories and inadequate search engine. These two issues cause practitioners to get lost in between the documents. They cannot figure out which is the right document, as the structure does not help them to understand the context and content of the documents and the search engine is not capable of specifying the overall context.

The problem about the structure is its complexity or absence. When there is a structure, it happens to be too complex due to the size of organization and people cannot follow it. When there is no structure, people cannot find the document they need, as hundreds of different documents are placed together. These two scenarios cause chaos and employees start to place documents carelessly and stop searching for existing knowledge.

The problem about the search engine is the traditional keyword-based information retrieval methods and lack of intelligence. Without any semantic filtering mechanism, the search query brings results from various domains, information sources and document types, where the keywords exist. This causes retrieving loads of irrelevant data to the sight of users. Finding the intended document among these is utterly irritating and time consuming if not impossible.

Using ontologies to structure information

RQ2: How were semantics and ontologies used in software engineering domain and how can they be utilized?

Usage of ontologies is very wide in handling the complexity of different types of information. The most commonly used area of ontologies is biology domain due to the appropriateness of taxonomic structure of biological information sources. However, ontologies are already used in several applications of software engineering to structure and manage large amount of information. They are mostly aimed at directly
involving to software development phases such as software processes, requirements, architecture, implementation and testing.

However, our contribution was not about improving development directly but about using ontologies in software engineering to structure the knowledge and experiences available and hence improve the development by enhancing reusing existing knowledge. So far, we could not find any completed and released software engineering ontology that covers all the knowledge in the domain. Yet, case study revealed that it was not necessarily needed. We found out that, practitioners mostly need a document ontology so that they can filter documents by their type and content.

Moreover, when it comes to reusing knowledge, it was seen that the business domain of the organization is equally important, if not more. Practitioners indicated that the information they reuse or search is often about domain specific knowledge, solutions, products, business processes and so on. Hence, the ontology should cover these aspects so that they can filter the documents accordingly. They proposed ontologies that cover business process frameworks for telecommunications (eTOM), organizational structure of the corporation, project management framework of the organization (PROPS-C) and the product catalog of the company.

**Usefulness and feasibility of semantic systems**

**RQ3: How useful are semantic knowledge management systems gathering implicit and explicit knowledge in software engineering?**

The main usefulness of semantic systems and semantic information retrieval comes from the possibility of solving the issues that practitioners are facing in traditional keyword searches. That is, the main focus of software engineers is always about finding documents as quickly as possible by filtering out the irrelevant ones and semantic IR provides this by using the meaning of documents. Practitioners were generally very excited about using semantic systems in order to solve these problems.

However, finding documents is not the only usage scenario for semantic search, as we mentioned before. Extracting knowledge from the contents of the information sources is the main difference compared to traditional search approaches. We concluded that, these capabilities about analyzing the meanings of documents, finding people, extracting skills and interests are generally found useful by the software engineers. Nevertheless, we can easily say that their priority was about finding relevant documents rather than extracting this kind of knowledge.

As an overall conclusion about usefulness of semantic systems, we can say that it is very promising to solve information overload problems if the focus is given to the specific needs of users. Primary necessity of users is to understand the context of document without reading it, so that they can select the right one when searching. The more they can filter out irrelevant documents, the more efficient they will be able to work. Hence, semantic search should be focused on this aspect of knowledge extraction. Moreover, extracted knowledge about the context of the document shall be used as tags, so that users will have initial idea about the document without a need to read it whole.

On the other hand, the overall conclusion about feasibility of semantic systems in large-scale enterprises is the importance of necessary effort for development. Implementing the ambitious goals and requirements that are specified by the help of practitioners is difficult to achieve and some parts might even be too optimistic. For instance, building the suggested ontologies may be very costly considering the size and complexity of the business domain. However, extracting the knowledge about people’s skills and interests does not require that complex ontology, a lightweight ontology along with an online lexical database will be enough to extract this kind of information.
Future Work:
Considering the observations discussed in Chapter 6, it is seen that the interviewees' opinions about information overload problems and solution proposal are highly effected by their age and experience. Hence, a further research can be conducted with a broader population considering the background of the practitioners so that the results can be generalized.

In addition, the case study can be conducted in another large-scale company that operates in a domain other than telecommunications. The comparison of the two would yield important results about interviewees’ ontology choice. It is essential to see if their main ontology choice is also based on the business domain of the corporation. To generalize the needs of software engineers about ontologies, it is a must to conduct several case studies. On the other hand, another company in telecommunications domain should also be analyzed in order to remove the defined external validity threats.

Moreover, the ontology that was used in the case study can slightly be modified and a lexical database can be integrated as intended so that the presented tools would be more usable and the practitioners can have the opportunity to try using the system alone instead of guidance of the interviewer. Then surveys can also be used for data collection with a broader population.
REFERENCES


[31] “RDF – Semantic Web Standards.” Internet: [http://www.w3.org/RDF](http://www.w3.org/RDF), [Jan. 18, 2013].


APPENDIX A - Interview Protocol

Introduction
- Present yourself
- Ask about recording and confidentiality

The subject of the thesis is about Semantic-Web based Enterprise Knowledge Management systems. The focus is on improving information retrieval capabilities in knowledge management systems. That is, we want to explore the benefits semantic search in enterprise environment. What we mean by semantic search is using meaningful, complex queries instead of traditional keyword based search platforms (e.g. Google) and retrieving aggregated knowledge from different sources. The result set in semantic search is actually extracted knowledge instead of set of documents that contain the search string.

The reason why we’d like to conduct interviews is to understand how employees of Ericsson gather implicit and explicit knowledge during their daily work and the role of internal collaboration tools in this process. That is, we want know if these tools can satisfy the needs of people to find out existing knowledge.

The focus is on how you cope with problems related to information overload and finding information.

The data that we will collect in this interview will be very important for understanding the problems about the current situation and the usefulness of the proposed system to solve the existing problems. We believe it will be a huge benefit for the organization if we can reduce the time spent to find the right information and hence reduce the redundancy of sharing information.

General Questions about Background and Communication
1. Could you please tell me about your roles and responsibilities? (also current projects, previous experiences, etc.)
2. Can you tell me about how you share information or documents in your projects between team members and with other related departments, units, etc.?
   2.1. How would you classify the types of information you share?
   2.2. What kind of tool do you use for each type of information?
3. What kind of problems do you face about sharing or finding each type of information?
   3.1. In which of these information types do you feel there is information overload and people spend too much time to access information?

Scenarios and Problems about Collaboration Tools
4. How often do you use collaboration tools of Ericsson (give examples)? (scale: daily, weekly)
   • For what purposes do you use? What kind of information do you look for or do you share? (possible scenarios)
   • Do you easily accomplish your goals in these scenarios
5. Can you give me example search scenarios from your daily work?
   • Do you find documents by browsing around? In which cases?
   • Search string examples?
How would you like to filter?
- SWEBOK knowledge areas and practices
- Software lifecycle phases
- Document types
- Organizational structure (based on projects, products)
- Domain

6. How would you evaluate your satisfaction with the search facilities in these tools? WHY

7. What do you suggest should be changed, improved when it comes to searching?

**Implicit Knowledge**

8. What do you do if you cannot find the information you are looking for in these tools?

9. How often do you need to go and talk to a person with expertise or experience, in order to gather knowledge (even if it’s an abbreviation that you don’t know the meaning)?

   9.1. In what kind of situations does this happen? What kind of information?

10. How do you find the person to ask?

11. When you need to ask a question, do you first perform a search if someone already shared that information? If so, do you usually find it or not?

**Demo and Evaluation**

Present the semantic tool with its functionalities and show search scenario examples.

12. What do you think about the presented tool? How would you rate its usefulness? Why?

13. Do you think your search experience will be different with this tool?


14. Do you think the speed of finding information can change with this technology? If so how much would it change if they have to rate on a scale?

   14.1. For which type of scenarios and information types?

15. What improvements do you think can be made?

16. Would you use it to find the related people to ask your questions, (to gain implicit knowledge)?

17. Would you prefer to add tags manually for every information you share for more accurate results, or you would prefer it automatic like this?

18. What about a software engineering ontology, would you search based on software engineering processes, artifacts?

19. If you have to rate on a scale, what would you say about using a Semantic system like this over the existing systems you have? Would you prefer this version? Why?

20. Final question, do you think we have missed something important that we can mention? Do you have anything else to add?
APPENDIX B - Mindmap

Figure 17: Mindmap from the Evaluation Interviews