e-Informatica Software Engineering Journal, Volume 12, Issue 1, 2018, pages: 237–263, DOI 10.5277/e-Inf180110

Semantic Knowledge Management System to Support Software Engineers: Implementation and Static Evaluation through Interviews at Ericsson

Ali Demirsoy*, Kai Petersen**

*Borsa Istanbul

**Fachbereich Wirtschaft, Department of Software Engineering, University of Applied Sciences Flensburg, Blekinge Institute of Technology

alidemirsoy@gmail.com, kai.petersen@bth.se

Abstract

Background: In large-scale corporations in the software engineering context information overload problems occur as stakeholders continuously produce useful information on process life-cycle issues, matters related to specific products under development, etc. Information overload makes finding relevant information (e.g., how did the company apply the requirements process for product X?) challenging, which is in the primary focus of this paper.

Contribution: In this study the authors aimed at evaluating the ease of implementing a semantic knowledge management system at Ericsson, including the essential components of such systems (such as text processing, ontologies, semantic annotation and semantic search). Thereafter, feedback on the usefulness of the system was collected from practitioners.

Method: A single case study was conducted at a development site of Ericsson AB in Sweden.

Results: It was found that semantic knowledge management systems are challenging to implement, this refers in particular to the implementation and integration of ontologies. Specific ontologies for structuring and filtering are essential, such as domain ontologies and ontologies distinct to the organization.

Conclusion: To be readily adopted and transferable to practice, desired ontologies need to be implemented and integrated into semantic knowledge management frameworks with ease, given that the desired ontologies are dependent on organizations and domains.

Keywords: knowledge management, information overload, case study, semantic web

1. Introduction

One of the main challenges for large-scale organizations is the high number of stakeholders [1]. They all provide/produce information and knowledge and, as a result, increase the amount of information. Besides, many stakeholders are not known to others as organizations grow; thus, the holders of specific pieces of knowledge are not known. Therefore, a significant problem occurs related to the communication and coordination between these stakeholders [2]. A solution offering assistance in overcoming these problems is knowledge management, i.e. the process of acquiring or creating knowledge, transforming it into a reusable form, and maintaining, finding and reusing it [3, 4]. 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 [5]. However, these search mechanisms do not always satisfy the needs of users in terms of the precision of obtained results [6, 7]. In consequence, people who exchange information with each other face the problem of information overload due to the high number of available documents and information

[8–11], i.e. more relevant information than one can assimilate is available [12].

"Semantic Information Retrieval", also referred to as "Semantic Search" [13], has been proposed to address the information overload issue. Semantic search refers to retrieving information based on the interpretations of the meanings of words [6]. Traditionally, there are classical information-retrieval models [14] that are aimed to find the most relevant document for a given query. The models estimate the relevance of documents and rank them via probabilistic methods, such as the Bayes classifier model [15] and the vector space model [16]. However, these models retrieve textual information based on the words' lexical forms, not their meanings. Hence, there is a problem of many irrelevant search outputs as a result of the ambiguity of 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 [5, 7, 17]. 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 [18]. These methods have been successful in some cases in increasing the hit rate during searching [19]. However, the semantic search goes one step beyond these approaches by enabling complex queries and retrieving extracted knowledge from the processed information sources. This way, users can search for meaningful queries instead of textual strings and, in addition to this, automated tasks can process information with a certain level of understanding [17].

There have been several studies that apply semantic technologies to the software engineering domain to conceptualize and organize the knowledge (e.g., [20–22]). These studies focused on different artefacts of the software development life-cycle (e.g., requirements and architectural assets). However, there are only a few examples that aim at organizing the existing knowledge to enhance knowledge reuse within a knowledge management system, where users share documents for the use of others [23,24]. These systems (e.g., blogs, forums, document repositories) are crucial to software engineers for utilizing the existing information by finding a relevant shared document and overcoming problems related to information overload [8, 25].

There is a lack of information how to implement and adopt semantic knowledge management solutions in an organization with no previous experience. Semantic knowledge management systems integrate different aspects (such as semantic annotation, querying, entity ranking, etc.) into an overall system. However, having an integrated solution also makes one less flexible as it is not so easy to simply exchange/expand ontologies as experienced in our study. Current research focuses on presenting final solutions and the ideas behind them but not the ways to make these solutions work [17, 26, 27]. 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 to understand the advantages, costs, and limitations of these systems. Given the high amount of information in documents, a more precise search possibility as well as a more natural way of annotating information could be useful, which is provided by semantic knowledge management systems. Though, for this to work, it must be feasible to implement and also be perceived as useful, which falls within the scope of this work.

Why to investigate a semantic approach as the information retrieval approach? The semantic approach not only offers solutions for achieving precision (number of relevant results compared to all retrieved results) and recall (number of relevant results compared to the number of results that ideally should have been found), but also provides extracted knowledge from the analysis of the contents of documents [28, 29]. Hence, it differs from all other models where the only aim is to retrieve the most relevant document. Here the objective is to retrieve the necessary knowledge, not the document or documents that contain this knowledge [28]. However, it can also be used to retrieve documents based on the semantics of documents and can be integrated with ranking techniques [7]. For this reason, the semantic web approach seems to be one step ahead of the other models, and the semantic search can be used to solve the current problems in information retrieval. However, for machines to read, interpret and process the information one needs a syntactical model. Requirements on machine readability causes a limitation of the type of information which can be modelled and extracted from documents. The most important factor here is the context and the content of the documents, and also the form of the desired information in the documents. Hence, to use semantic search for solving information overload, the needs of the users concerning their information usage and the content of the documents for their domain have to be investigated and analysed to see if it applies to semantic information retrieval. For instance, using semantic technologies has been observed to be very useful in such areas as biology, since the modelled information in biology is very suitable to represent ontologies [30].

The primary goal of this work is to understand and evaluate the feasibility of the implementation of semantic knowledge management systems. The study makes two contributions:

- **Contribution 1:** After the investigation of the context of the company, a semantic knowledge management system was implemented, which highlighted the limitations of such systems from a feasibility perspective. Understanding the limitations is important as these may hinder the adoption in industry [31]. There is a definite need for solutions whose practice can easily implement and integrate into existing environments for a successful transfer to industry [32]. In the context of search-based software testing, Arcuri et al. [33] highlighted that search-based software testing is not readily transferable if no engineering efforts are taken; hence, to make it easy to integrate it and use with the existing systems in practice, additional engineering efforts are required. The ease of integration into existing solutions was a key factor for the successful transfer of research results to industry. The ease also determines the degree of evaluation which in turn is dependent on the degree of the readiness of the solutions available.
- **Contribution 2:** After that practitioners assessed the system by using it in the context of an interview session. The evaluation conducted was a static evaluation [34]. The tatic evaluation allows to gather early feedback in an exploratory fashion and to capture essential issues and needed corrections before further spreading and developing a solution. The evaluation provided valuable qualitative feedback on the potential of semantic knowledge management systems and about their strengths and weaknesses. This research presents a single case study [35] at the development site of Ericsson.

The research comprises three phases. First, the authors focused on understanding the research context. Second, they implemented a solution for the semantic knowledge management system and reflected on their experiences. Third, they conducted a static evaluation [34] gathering qualitative feedback on the solution proposed to identify the most crucial improvement suggestions.

The remainder of the article is structured as follows. Section 2 presents related work. Section 3 describes the research method. The results are presented in Section 4. The conclusions in Section 5 provide the answers to our research questions.

2. Related work

First basic terms concerning data, knowledge, and information are presented. After that, integrated semantic knowledge management frameworks are shown. The subsequent sections explain essential components (e.g., for information retrieval).

2.1. Data, knowledge, and information

Different definitions exist for data, information, and knowledge. According to Thierauf [36], data constitutes raw facts and figures. Data becomes information through contextualization and categorization. Documented experience and know-how already represent knowledge. Hence, documents produced in companies, such as the case company, may contain knowledge if they provide an experience report or a process of how to solve a problem, however, they may also carry only information (e.g., product requirements) or data (e.g., sales figures).

2.2. Semantic knowledge management frameworks

Knowledge management systems are composed of various steps and corresponding tools. It requires a systematic methodology and considerable amount of time and expertise to extract and formalize knowledge from unstructured data and to develop a platform that can find, share and manage information. Hence, the authors will examine the research on knowledge management platforms that provide all these functionalities together. Semantic knowledge management systems introduce structure through ontologies, e.g. enabling faceted search where there is a browsable classification, making the structure of information explicit to the end user.

OntoShare: OntoShare is an organizational knowledge management system that promotes sharing of information between people who have mutual concerns or interests [37]. It is an ontology-based tool that places the profiles of the users at the centre of attention. That is, the interests of each user are modelled by an ontology and this information is extracted from the activities of a user. Every time the 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 which is strong enough, then the system emails the corresponding user to inform about the new document shared. Moreover, the content of the document is also compared to the author's interests in order to add new interests if necessary. 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 existing documents and also search for people that are interested in a certain area.

Knowledge and information management framework (KIM): KIM [27] is a platform for semantic annotation and semantic search over several kinds of information sources. It is used for information extraction from data pools based on an ontology and a knowledge base [27].

KIM comes with an upper-level ontology called PROTON which has about 300 classes and 100 properties in OWL Lite[1]. This ontology covers most general concepts, such as names of people, locations and organizations along with numbers and dates. It also has the KIM World Knowledge Base (WKB) which has about 200,000 entity descriptions to provide background knowledge for commonly known entities. KIM keeps the ontologies and the knowledge bases in the SESAME based Owlim[2] RDF(S) repository.

Moreover, KIM uses the GATE framework for information extraction processes and Lucene from Apache as a retrieval engine [38]. Lucene has been adapted so that it allows indexing by entity types and measure the relevance with respect to entity types.

KIM not only provides full-automatic semantic annotation, but also allows retrieving information based on the metadata that has been created. This brings a new perspective to information retrieval, as the user is able to define a "pattern search". That is, a semantic query can contain entities that are known or extracted before, relations between the entities and attributes of these entities [27]. This means the 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, an organization would be an entity, a location and an 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 used by large organizations to share all kinds of information and can be used for knowledge man-

[1]OWL Lite: https://www.w3.org/TR/owl-features/
[2]Owlim: http://graphdb.ontotext.com/documentation/7.0/enterprise/using-graphdb-with-the-sesame-api.html

agement. A Wiki is a hypertext environment that provides the collaborative editing possibilities of Web pages. Wikis emphasize openness, ease-of-use and modification [39].There are some limitations of Wikis that prevent them from being used as a knowledge management tool. Wikis do not provide structured access to data and do not support knowledge reuse [40]. A semantic Wiki provides annotation capabilities to create formal descriptions, retrieval mechanisms for semantic search, and semi-automatic meta-data extraction system to simplify the annotation process.

Active: The project Active [41] aims to increase the productivity of knowledge sharing via prioritizing the information and knowledge delivery through understanding the current context of a knowledge worker [42]. That is, a filtering mechanism provides 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 behaviours. The idea is based on the fact that users are generally busy with several different tasks during the day and they constantly have to switch and concentrate on a different one.

2.3. Solutions to find relevant information

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 with these tools is the difficulty of finding 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 for. In the 1960s information retrieval was defined as "a field concerned with the structure, analysis, organization, storage, searching and retrieval of information" [43]. Since then the area evolved into many different techniques and models in order to adapt to changing needs, such as exact match models [44], vector space models [45], and probabilistic approaches [18]. The latest approach is based on semantic approaches.

Storing and querying semi-structured data: In order to utilize heterogeneous and incomplete information data research and practice aimed at a semi structured format that is flexible and also appropriate for querying. Approaches for dealing with semi-structured data are 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 [44]. Lately RDF has gained the attention of researchers since it provides much more flexibility compared to XML by not enforcing a hierarchical structure, but supporting any kind of relations between data items.

Semantic Web technologies are the new generation of presenting and sharing data in various application areas. They started to be used in web platforms as well as tools that are in a way related to managing and providing important data [3]. The idea of a Semantic Web is to give information a well-defined representation so that it will be available in a more meaningful, structured and reusable way, which will enable humans and computers to work in cooperation to retrieve data from the Web [46].

In ontology-based Semantic Web applications, information is presented at a semantic level with ontologies, independent of the data structure and implementation, with a set of concepts and relationships between them [23]. This idea emerged from the need to enable some tasks to automatically understand the concepts in order to find relevant information, combine and share it with different resources. The representation of information with ontologies provides a common format between different systems and applications in order to share, understand and use knowledge [47]. This common format is standardized by W3C with the Web Ontology Language (OWL), Resource Description Framework (RDF), etc.

With the use of ontologies, a query is composed of entities from the ontology and their relations. This allows users to set the context of the input query. Moreover, usually in this kind of data retrieval 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 the synonymy problem, as the synonyms of the words already exist in this database and are used during retrieval. Other than solving these two main problems in information retrieval, this method is also useful for extracting key knowledge from document sources. The query results are not only listed as 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.

2.4. Ontologies in software engineering

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 [48], examples are ontologies for software processes [49], requirements [50], software architecture [51] and domains [52], and document ontologies [21].

All these ontologies are being used to improve software development. Their aim is to help software engineers to manage and understand large amounts 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 some improvements in Semantic Web technologies. The drawbacks for now are that constructing ontologies and implementing a Semantic Web enabled tool require a high investment of time. However, after the definition of ontologies, it is very flexible and easy to modify it according to the changing needs of an organization [37]. This also means that a dedicated person may be needed to maintain the semantic systems and their ontologies.

Although there are several studies that focus on developing ontologies related to software engineering processes, there are only few attempts to build an ontology that covers all software engineering knowledge. The most important among them is the work done to create a software engineering ontology based on the Software Engineering Body of Knowledge (SWEBOK) [53]. In SWEBOK a software engineering discipline is categorized into 10 knowledge areas. All these knowledge areas have their own processes and concepts. The proto-ontology, which was created based on SWEBOK, conceptualized all information in over 4000 concepts along with 400 relations and 1200 facts [54].

There are similar projects, such as Onto-SWEBOK, which are designed based on the 2004 Guide to the Software Engineering Body of Knowledge (SWEBOK) [55, 56]. However, none of them is released or publicly available because of unfinished projects due to the complexity, required time and human resources [55].

Another attempt to create a software engineering domain ontology is OntoGLOSE which is a light-weight global ontology [57]. This project uses the Glossary of Software Engineering Terminology published by the IEEE Computer Society [58]. The IEEE Glossary contains 1300 terms and their definitions that are related to the 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 the 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 the IEEE Glossary, which was built in 1980 and updated in 2002, which means that it is out-to-date considering the amount of advances in the last 10 years. Moreover, the fact that it does not have any hierarchy, it is not the ideal way to structure information.

2.5. Tools to support ontology-based knowledge management systems

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

The first step for a KM system is knowledge acquisition, and to acquire information from an unstructured text, several frameworks that can process plain text and extract concepts are used. GATE (General Architecture for Text Engineering)[3] is one of the most commonly used frameworks and has several plug-ins and integration capabilities [59]. It has many flexible language processing components that rely on finite state algorithms and the Java Annotation Patterns Engine (JAPE) language. It is widely used due to its precision for entity recognition and suitability for research as it is open source software. Moreover, it is commonly used in the semantic world because it offers full support for ontology integration. It has been utilized in ontology-based information extraction projects such as Multiflora, hTechSight and MIAKT [60].

IBM produced the UIMA[4] framework, which is an enterprise semantic search tool, but it does not provide full integration and support for ontologies [61]. Another tool is OpenNLP[5] from Apache, it supports many NLP tasks such as tokenization, segmentation, named entity recognition. However, it accomplishes these tasks via its built-in tools, not via any external ontology integration.

When it comes to Knowledge Representation, there are many tools to create, manage and edit ontologies. Protégé[6] is one of the most common open source ontology editors used by developers, researchers and corporations. 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 the Protégé tool.

Uren, et. al [28] provide a comprehensive work on the analysis of different annotation tools and frameworks, and offer a comparison of them.

3. Method

This section illustrates the research method that was used based on the guidelines by Runeson and Höst [35]. In order collaborate with the industry, it was essential to first conduct a qualitative study to learn about the strengths and weaknesses of the solution (semantic knowledge management system), and obtain feedback from practitioners in the context. This also allowed the practitioners to learn about the semantic knowledge management system. The qualitative information could also be useful later to explain the reasons for quantitative results. In this sense, the study is of exploratory nature with a focus on qualitative data.

3.1. Research questions

In this study the following research questions were defined:

- RQ1 (Contribution 1): How to implement semantic knowledge management systems, and which challenges and impediments are observed?
- RQ2 (Contribution 2): How useful is the semantic knowledge management system perceived by software engineering practitioners?

The research process is conducted in three phases (see Fig. 1). The detailed phases are described in Section 3.3.

3.2. The case and unit of analysis

The case studied was 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 telecommunications and multimedia domain. The company products are used in more than 180 countries in the world. Currently the company has more than 100.000 employees.

[3]GATE: http://gate.ac.uk
[4]Framework UIMA: http://uima.apache.org
[5]Framework OpenNLP: http://opennlp.apache.org
[6]http://protege.stanford.edu

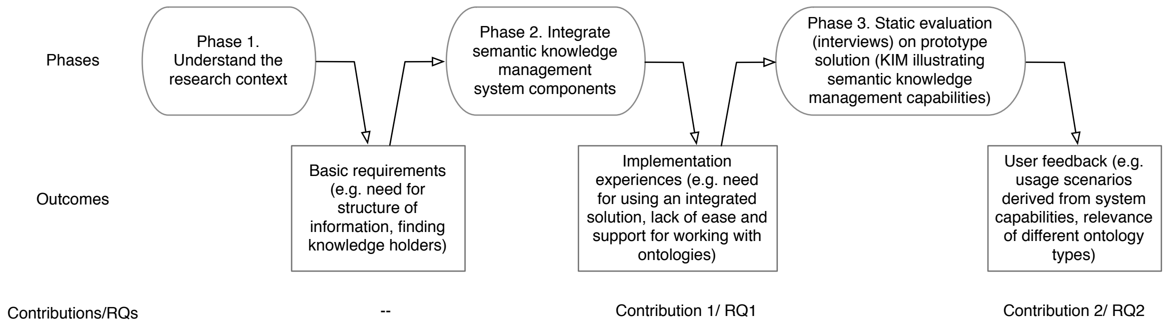

Figure 1. Research process

As far as units of analysis are concerned, internal knowledge management systems and the documentation they entail were defined as the unit of analysis. The case study design can be classified as a single case holistic design [62]. Ericsson uses a set of in-house knowledge management tools. These were platforms where everybody can share all sorts of information. They supported uploading documents and files; sharing blog posts and creating groups and discussion boards.

3.3. Data collection

The study was divided into three phases, understanding the challenges, implementation of the solution and its application in the company, and evaluation interviews.

3.3.1. Phase 1: Understanding the context of the organization

At the beginning, interviews were held with two key stakeholders from the organization to identify their needs and problems, they persons were the key contact persons and gatekeepers. The interviews were unstructured and were aimed to kick-off the project and achieve initial understanding. That is, the purpose was to get to know the company, project, problems in information retrieval, and responsible people. This was not a core part of the research (i.e. it is not reflected in the research questions), though contextual information was highlighted as essential when interpreting findings from case studies [35].

In order to elicit the requirements and issues, three separate meeting sessions were conducted. The first two meetings were held with the industrial contact who was supporting this study in the organization. He was a system level manager with over 20 years of experience. The meeting lasted an hour.

For the third meeting, two experienced software managers from Ericsson, who were responsible for innovation and had technical backgrounds in software engineering, were also invited. The goal was to see what was available in the literature and discuss the applicability of the desired solutions during the interview.

The following topics were discussed during these meeting:

– Introduction of the company and the responsible people to the student.
– Existing challenges related to finding information in the organization.
– Deficiencies of current internal collaboration tools.
– Requirements of a new solution.
– Possible usage scenarios about accessing relevant information.

The interviewer took notes during these interviews. Moreover, bi-weekly workshops were organized to discuss the findings, solution alternatives and status updates with the industrial contact. Hence, the data collected from the initial meetings was validated in these workshops. These meetings were important due to the possibility to obtain constant feedback from problem owners and also analyse the impact of the solution proposals on the company.

3.3.2. Phase 2: Development of a simple semantic knowledge management system

This phase required considerable time and effort in comparison to the other activities. After choosing the solution strategy in the initial interviews, an example system was created and applied to real world data to allow the participants to understand what the application of semantic technologies means in their context. The idea was not to implement a complete system that can replace the existing one, but rather to have a prototype which was sufficient to evaluate the usefulness of semantic systems in general.

Four different components to be supported by a complete semantic knowledge management system were defined and executed in this study. This section describes the components in general, while the details of the actual implementation are provided in Section 4.2.

Text processing (knowledge acquisition): The purpose of a semantic system was to extract knowledge from sets of unstructured information. Hence, the first step was to analyse and process these unstructured documents using the Natural Language Processing (NLP). The NLP technology has evolved to gain many capabilities in order to process the syntax and semantics of a text.

Ontology & knowledge base (knowledge representation): Ontology was one of the most important factors in information extraction as it provides conceptualization to the content of documents and was used for text processing. The ontology must be suitable to the contents of the information sources that are to be processed. Hence, there was a need to make a suitable ontology choice depending on the context of the domain.

Semantic annotation & ontology population (knowledge acquisition) & representation: When NLP tools parse the unstructured text, the entities found there should be annotated and mapped to the ontology. Therefore, the ontology could be populated with the extracted knowledge in the RDF or OWL format. This was the most significant step in information extraction as it was the phase where the relations between entities were defined. There were several platforms and ways to accomplish this step. Since this step was both depended on the NLP tool and the choice of ontology, it was crucial to choose a suitable system to integrate and work efficiently.

Semantic search (knowledge use): Once the ontology was populated with the instances and relations extracted from the text; the only step left was using a query language that was created for the Semantic Web in order to retrieve relevant information. A query engine needed to be chosen and should be supported by a graphical user interface. Users should be able to perform search with semantic capabilities, navigate between sources according to their semantic relations.

The details of the implementation and experiences made are presented in Section 4.2.

3.3.3. Phase 3: Evaluation interviews

The final evaluation and analysis was done by means of interviews with several company employees. This phase provided information about current challenges, obstacles about accessing information and possible improvements, suggestions and critique for the proposed new system. The system usefulness and users' experience with the system with semantic capabilities were evaluated. This time interviews were semi-structured. The prepared questions constituted a checklist of topics that should be covered during the interview.

Selection of interviewees: Knowledgeable practitioners should be chosen to conduct the interviews. Convenience sampling with diversity in mind was applied [63]. The interviews were conducted with employees with experience ranging from 3 to 25 years and with diverse roles, such as project manager, software architect, software developer, R&D specialist, solution architect.

As a result, eight employees were interviewed as can be seen in Table 1 below, which is believed to provide a sufficient amount of information to contribute to the literature and industry.

Interview guide: The interviews were related to the usage of internal collaboration tools of the organization, such as frequency of use,

Table 1. Interviewees

Role	Experience	Responsibilities
Project Manager	10 years in Ericsson 20+ in total	Project management, process improvement, process management
Software Architect	12 years in Ericsson 15+ in total	Software design, development, innovation
Senior Specialist R&D	20 years in Ericsson 25+ in total	Next generational rating and charging, information and business modeling
Solution Architect	7 years in Ericsson 10 years in total	Charging and mediation
Software Developer	2 years in Ericsson 3+ in total	Software customization center
Software Engineer	2 years in Ericsson 7 years in total	Software customization center
Solution Architect	19 years in Ericsson 20+ in total	Telecommunication services
Software Engineer	2 years in Ericsson 5 years in total	Proof of concept integration, Machine-to-machine applications

usage scenarios, satisfaction of the current version and suggested improvements. Later, the new semantic knowledge management system was presented to the users, and they were asked to explore the new system by using it. After they had gained an idea about the system, similar questions to the ones asked at the beginning were repeated and their opinions were collected and compared. The interview was structured as follows and the detailed guide is presented in Appendix A.:

– *Warm up:* First, the interviewer presented himself, the background of the project and the reason for making the interviews. Then the interviewee was asked general questions about their role, experience and current projects. This part of the interview was conducted mainly to build knowledge on the people and situation.
– *Information related to the usage of collaboration tools and problems:* It is important to know how and for which purposes people used the company's tools during their daily work. This part was devoted to figure out how often they used the current systems, how satisfied they were with the system (KIM, see Section 4.2.5) and what they would like to change in these tools. Basically more usage scenarios, requirements and problems with finding information were elicited.
– *Implicit knowledge:* It was important to learn how the employees gathered knowledge when they could not find what they looked for or when they were not satisfied with the findings they obtained. The authors tried to establish if they felt the need to talk to an expert and if so how they found out who the expert or responsible person was 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, an overview of the Semantic Web technologies was given and the information about the usage and goals of the Semantic Knowledge Management Systems were presented. Then the new system was presented as a prototype and the functionalities coming with the Semantic Web were explained. The interviewees were allowed to browse in the system documents for a while in order to make sure they were aware of the differences with the existing traditional knowledge management systems.
– *Satisfaction and evaluation of the proposed system:* The interviewees were asked to compare this system with the existing one. They were also requested to state whether they would use this system more often and if it would help them to make better decisions or reach implicit knowledge more easily. The point of the question was to capture the interviewees' attitude related to the evaluated system (KIM), as this was an important indicator for adoption and the possibility for a solution transfer from academia to industry. The actual decision quality could not be evaluated in this context.
– *Recommendations:* Finally the questions about possible different options for creating the Semantic Knowledge Management

Systems were presented and the interviewees were asked about preferences related to ontologies. Also, suggestions of improvements related to the proposed system were captured.

3.4. Data analysis

The qualitative data from the interviews was analysed using Thematic Coding Analysis (TAC). The authors followed the guidelines described by Robson [64] who describes TAC as a generic approach to analyse qualitative data, highlighting its flexibility, ease of application, and efficiency. The process was based on open coding and the identification of themes. The open coding was done manually on papers using color-coding, open codes belonging together were grouped during axial coding (referred to as themes).

3.5. Validity threats

We analysed the validity threats and mitigating factors in our case study following the descriptions given by Yin [62]:

Construct validity: Construct validity is concerned with the extent to which what was intended to be measured was actually measured [35].

- Selection of the Interviewees: The selection process was managed with the help of practitioners from the company. The selection process was a combination of diversity and convenience sampling. As far as convenience sampling is concerned, the selection was made based on the knowledge and availability of the employees. There is a risk that practitioners can choose people who support ideas similar to theirs. The usage of diversity sampling mitigated this threat by selecting employees with more diverse roles and experiences. At 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 was partially reduced as a practitioner from the company was the gatekeeper who made the contact with interview candidates and helped build a trust relationship between the researcher and the interviewees.
- Correct Data: The correctness of the data aggregated by the interviews refers to the researcher's interpretation of what the interviewee actually said. To ensure this, all the interviews were recorded after taking permission from the interviewee so that any misunderstandings due to incomplete interview notes would not occur. Moreover, the interpretations of the interview transcriptions were sent back to the interviewees to obtain their validation feedback (member checking).
- Duration of the usage of the system: The practitioners used the system but only for a limited period of time. The practitioners know the existing system very well. Because the interviewees used the system themselves, they could, for example, understand its capability for different ways of searching (e.g. with regards to filtering specific entities that wpuld show only then and were unambiguously identified, see Section 4.2.5). Even though they did not have long-term experience, it was evident from their responses that they understood the concepts (and hence the opportunities) clearly, evidenced by the very informed feedback regarding Ontologies and Filtering (Sec. 4.2.3).

External validity: External validity is the ability to generalize the findings in a way that they will be interesting for other people representing other interest areas [35].

A single case company has been investigated. The results of single case studies in comparison to, for example, a survey have limited generalizability. However, the benefit are detailed explanations and a profound understanding of the situation that could be obtained. To reduce the effects the authors interviewed employees from different organizational parts of the company. Moreover, the context of the case study was described in detail (see Section 4.1), which allowed to map the findings to other large-scale organizations that are involved in software development.

Reliability: Reliability refers to the issue of finding the same results when the same study is replicated in the same setting [35]. The main threat to reliability are possible misunderstandings about the questions that were asked to interviewees, as they might have misunderstood the questions and hence provide answers different from the intended ones. An attempt was made to reduce this threat by keeping the questions as simple as possible. Open-ended questions were preferred so that the participants were encouraged to talk and express their opinions openly.
Internal validity: Internal validity concerns the validity of causal relations in explanatory case studies. It is related to the unconsidered factors that might have an impact on the relation [35]. The analysis of the usefulness of semantic knowledge management systems can be biased because of the employees' opinions about the existing systems. For instance, if the existing system had a search engine that is as powerful as the one Google applied to their documentation, the findings could potentially change.

4. Results

First the research context is presented, it is followed by the answers to the two research questions.

4.1. Phase 1: Context

A multinational large-scale organization like Ericsson has thousands of employees all around the world and hundreds of projects running in parallel. Considering the increasing amount of globally distributed projects in the 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. To share knowledge people used blogs, Wikis, discussion boards, project contents and documents. Since all Ericsson employees, i.e. more than 100.000 people, use these tools; there are large amounts of documents. All these documents and information are not stored in a structured way and, hence, it is necessary to find ways of managing this large volume of unstructured data. It is imperative to investigate how to overcome these problems. All the interviewees mentioned that the existing search facility does not satisfy existing needs and so a more intelligent solution should be found. In particular semantic knowledge management and ontologies allow to bring structure to the information stored, which was one of the motivations for the company to participate in the study.

The following challenges and needs of the organization were raised during the initial meetings to understand the context:

- The practitioners defined usage scenarios that are common, in particular active search based on queries, passive search, analysis of contributors (users), and the analysis of trends. Overall, the practitioners identified scenarios that are common and well understood in the knowledge management community.
- Structure of information was a common issue, which is not specific to the company. Bringing structure to information is well supported by ontologies, making them interesting for the company. Performance issues and formatting were specific for the company and could be easily improved.
- The search engine used at the company is perceived as a poor quality one.
- Filtering of search results and complicated structures have been highlighted, which is also a good motivation for annotating documents and mapping them to an ontology in the context of a semantic knowledge management system.
- A challenge was also finding an expert, which is recognized as a key challenge in literature, too [65].

In summary, the conclusion was that the search should be improved, and that semantic knowledge management systems could be a potentially useful solution.

4.2. Phase 2: Development of a Simple Semantic Knowledge Management System (RQ1)

This is the phase where it was necessary to make a comprehensive research and spend time and effort on the development of a new knowledge management system, which took a total of four person months. One of the reasons for the effort needed was the absence of information about how to implement a semantic knowledge management system in the literature. Although the final solutions were presented in some studies, the way to implement them 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.

It is important to point out that the attempt to build the semantic knowledge management system based on components was not successful for the above mentioned reason. The best working solution was to utilize an integrated solution (KIM), which is described in Section 4.2.5. As KIM is easier to use, a transfer to the software industry for knowledge management purposes is more likely. Thus, KIM is used in the subsequent steps of the study (i.e. Phase 3). The principle architecture of semantic knowledge management systems and how it relates to KIM is shown in Figure 2. The details of the KIM platform are further elaborated in Popov et al. [27].

4.2.1. Ontology building

The first step to build an ontology is determining the domain and the scope [66]. In this case, the domain are all kinds of knowledge that can be shared in Ericsson software projects. That is, the ontology should cover aspects from generic software engineering domain to the company domain. The latter can be considered as the projects, characteristics of projects, employees and terms related to the telecommunication domain. However, in the scope of this work, the focus will be more on the concepts that are directly related to software engineering. The specific terminology of the company will be left for future research. 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, one can say that the ontology should only be sufficient to cover the topics that organizational members can possibly share or mention. Hence, the ontology should provide answers to such questions as people's interests, expertise, projects, locations of projects and people.

The second step in building an ontology is considering reusing existing ontologies instead of creating a new one [66].

There have been several studies about building ontologies in software engineering. Most of these attempts focused on specific phases of software engineering, such as requirements, architecture, implementation, testing, maintenance [20–22, 50, 67]. However, 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 aimed to adopt the SWEBOK Guide as a formal ontology. Such an ontology would be a good choice for the scope of this research as it would cover all the content and terminology in the software engineering domain. Unfortunately these attempts have not yet been successful or completed due to its complexity and required effort [54–56].

As a result, a decision was made to work with the only successfully released global ontology OntoGLOSE, which is based on the IEEE's global terminology for software engineering [57]. Although there are certain drawbacks of this ontology, such as the lack of coverage and the fact that it is outdated and primitive; utilizing this lightweight ontology would still be sufficient for the scope of this study to reach the current research goals.

4.2.2. Text processing

For processing an unstructured text, it was 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 (A Nearly-New Information Extraction System). Using ANNIE's components such as tokenizer, gazetteer and sentence splitter; one can extract

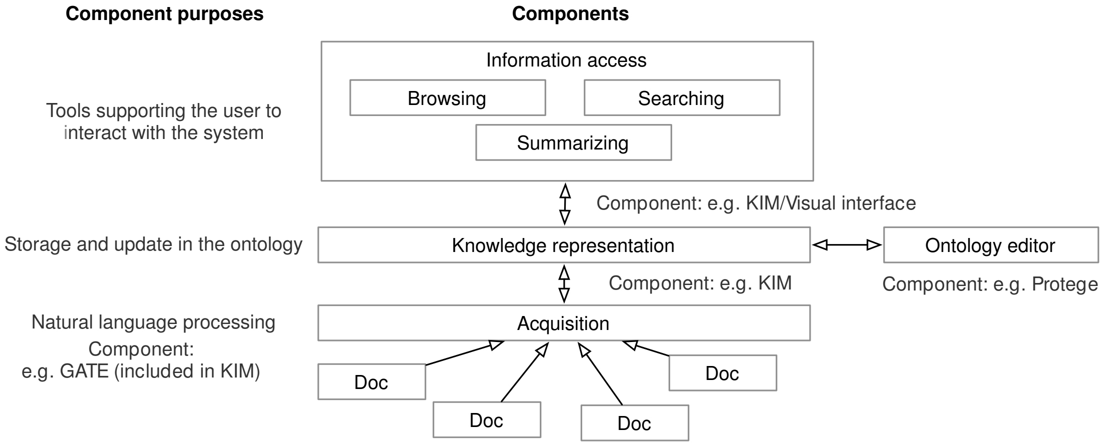

Figure 2. Semantic Knowledge Management Architecture

generic information from the corpora of the unstructured text. GATE can find the names of well-known organizations, names of people, locations, numbers, etc.

When GATE and ANNIE were applied to some documents from the knowledge management system of the company, it could be observed that the recognized entities were not enough to cover the content and the context of the technical documents that were used, such as domain specific terms that were relevant for the practitioners, but not highlighted.

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

4.2.3. Ontology and 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.

In the process of building and managing ontologies, Protégé was selected as an ontology editor for several reasons. First of all, Protégé is an open source research project which is extensively used in the academic world. Moreover, the authors had previous experience in using this tool in another academic project. Finally and possibly 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 Protégé was used to manage the existing ontologies as described in the previous section.

4.2.4. Semantic annotation and ontology population

A fully automatic semantic annotation tool is needed to apply it and evaluate directly on the corpus of the organization's knowledge management systems. Manual annotation tools require user intervention, so their usefulness cannot be directly evaluated without manually populating them with information.

Finally, the decision was to use GATE also for semantic annotation as it supports the automatic annotation of documents. Therefore, it was used to make an initial attempt to annotate a company document with the built ontology. At the end, GATE was used for NLP and semantic annotation and Protégé was used for building ontology.

After exploring the tool for a while and gaining the understanding of how it worked, it became clear that adapting the processing resources of GATE was 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 could not be done manually within the time and resources provided by the company and the research project. Choosing an external knowledge base and integrating it would also mean a need for a substantial amount of time,. Moreover, even though the knowledge base can be integrated, the GATE annotation system 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 NLP expertise and a 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 a significant amount of time as well. Considering the time constraint, a decision was made to make a mind switch and look for alternative solutions. The authors looked for integrated platforms that use GATE and also provide semantic search facilities with ontologies.

4.2.5. Integrated semantic knowledge management platform (KIM – Knowledge and Information Framework)

The decision to use the KIM platform (see Section 2) was made because it met the requirements of this study and the defined usage scenarios. Some reasons why the other platforms could not be used encompassed the fact that 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 for solving search problems. KIM supports the fully-automatic semantic annotation of documents and comes with an upper-level ontology and a semantic search engine. KIM is based on GATE for NLP purposes. It comes with an ontology named PROTON[7] 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 with KIM according to the needs of the domain. The stated requirements were analysed and compared with what KIM can offer and, in consequence, the following results were achieved:

- KIM's general ontology covers most of the aspects defined in the scope of the ontology for the purpose of this study. There is no need for numerous changes in the ontology design such as classes and relations. It is possible to integrate the OntoGLOSE domain ontology and this will enable KIM to recognize domain specific concepts. There is no need for very specific relations between classes as our usage scenarios are only based on extracting who is talking about what topic, either. As long as the topic is recognized, it would be sufficient to satisfy the specified requirements.
- If the domain ontology is not enough to cover all the aspects, as it does not have any concepts developed in the last 10 years and many other concepts about the company domain, the KIM knowledge base can be extended with an external knowledge base. For instance, KIM supports integrating KIM with DBpedia[8] which is a structured knowledge base containing all Wikipedia entries. Considering the fact that Wikipedia contains all the terminology that we need for software engineering as well as the telecommunications domain, integrating DBpedia would be a convenient solution.
- KIM provides "Boolean Search" which is a keyword-based search and corresponds to "Active Search" in 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 for the same scenario. "Timeline" search, which shows the popularity of selected entities over a period of time, can be used for the "Trends" scenario defined by the authors. On the other hand, KIM also provides navigation between documents according to their relations, which enables "Passive Search". The KIM search frame and the "Structure" search menu can be seen in

[7]PROTON: http://proton.semanticweb.org
[8]DBpedia: http://dbpedia.org/

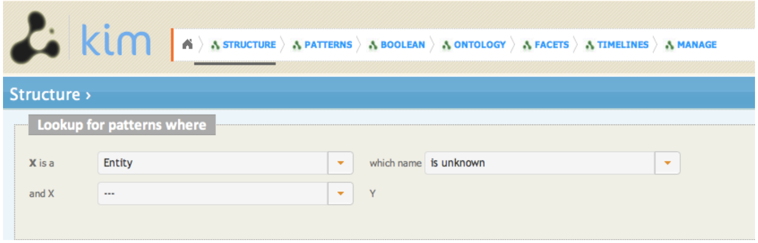

Figure 3. Structure Search from KIM

Figure 4. PROTON ontology

Figure 3. KIM has the capability of detecting persons in the text through the basic PROTON ontology. That is, if the same person for example always appears in blog posts, discussions or other documents in relation to a specific product, then this person could be an interesting contact. This can be captured as a semantic query can find relations between entities, such as people and topics, hence supporting the "Tribe scenario".

First of all, an attempt was made to integrate DBpedia with KIM. To be able to use the DBpedia instances, it was necessary to to integrate the DBpedia ontology with PROTON, which is the generic ontology of KIM. However, the whole DBpedia ontology be mapped to PROTON as it would cause too much complexity. Therefore, Person, the Organization and Abstract classes of DBpedia were taken and mapped to PROTON, so that the names of all well-known people, organizations and also abstract topics which contain the software engineering related topics are included. Figure 4 represents a part of the PROTON ontology and its Person and Organization classes.

However, due to poor documentation and the lack of available external support and expertise, it was not possible to successfully integrate DBpedia to the KIM knowledge base. Integrating DBpedia consists of many steps, such as 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 them.

Therefore, it was decided to integrate the software engineering domain ontology OntoGLOSE. Although it did not satisfy all our needs, it was be a good starting point for a further study to modify and extend its coverage.

After integrating this domain ontology, it was established that the system still did not to recognize the entities in this ontology. As a backup solution, a manual integration of this ontology to the actual PROTON ontology was conducted via Protégé Ontology Editor. Since OntoGLOSE does not have any hierarchy, it was easy to manually copy its classes to the other ontology. However, the relations were neglected as they were not interesting for this context. However, even after these steps were taken, KIM still did not manage 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 two e-mail exchanges with them, which was not enough to fix the problems.

Therefore, the system was evaluated during the interviews as it was. The discussion board pages from one of Ericsson collaboration tools were downloaded manually and loaded to KIM as a corpus. There was no quantitative measure, though too much information was in the system to easily search it. Thus, the corpus was sufficient to evaluate the usefulness from an end-user perspective during the interviews as they could experience the main concepts of the semantic knowledge management system.

The key findings for RQ1 are presented below.

Key findings and observations for RQ1:

i) It is time intensive to build a semantic knowledge management system, in particular setting up the ontology is a great challenge which required the majority of the effort.
ii) Rather than integrating different parts of a semantic knowledge management system, it is recommended to use an integrated platform as it is easier and hence more likely transferable to industry. Thus, KIM was used in Phase 3 of the study.
iii) Different types of searches (in particular pattern making use of the ontology) are possible with KIM, hence making explicit use of ontologies.
iv) KIM does not allow to easily integrate ontologies other than PROTON, which is a limitation. Beyond that KIM is easy to use.

4.3. Phase 3: Evaluation interviews (RQ2)

In Phase 3 the reflections of the practitioners on the usefulness of KIM, the ontology and filtering as well as possible improvements to the knowledge-based system, are discussed.

4.3.1. Usefulness of KIM

All of the interviews confirmed that the overall approach that comes with the semantic systems seems very useful. Although they all remarked that their current search engine was totally incapable and the proposed one (the new one?) cannot even be compared to the existing one, they pointed out some strong points of the semantic search.

Finding documents and faceted search: All of them found it useful to search for documents with their relation to people, topics and authors. However, they suggested different ontology alternatives, which will be discussed in the "Ontology and Filtering" section below.

Two interviewees found "Faceted search" the most useful, as it starts broader and narrows down based on the results of 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.

Finding eople and their position, roles and locations: Most of the interviewees (6 of 7) 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 mentioned that they did not need this functionality because they knew everybody he needed . Others stated that finding experts and knowledgeable people was quite a common scenario in Ericsson as there are experts in almost every area and their knowledge is indispensable. One of them added that, "Finding the right person was a common practice in Ericsson. It is a large organization. Not everyone knows everything but you can find an expert in almost every area. However, sometimes you don't know who they are. You should be very active in forums, etc., but it needs spending time on them regularly. So this facet search is very, very useful." They all agreed that the correct recognition of software engineering and telecommunication terms by the NLP 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 this information is actually stored somewhere in the company database. However, they would like to integrate this database, which is not directly accessible for employees, to this semantic system so that they could utilize organizational data while searching.

Extracting statistical data and decision making: Another point that the interviewees mentioned was the statistical data that could be gathered by means of this new system, which is similar to the "Trends" in usage scenarios. By analysing what people talked 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 communicate about can be extracted with this system to make an organizational decision. Another example given by an interview subject was as follows: "If we have a lot of people working with GUI in a unit, or the majority of graphical people in Ericsson work in this city, maybe we should set up a centre there. This will mean that 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 could be useful for sales people who go to customers. The connection to software engineering is not immediately evident. Though, in the context of continuous integration, customer relations in the organization are tightly coupled with software development, e.g. to enable continuous releases. Also, information from and to sales/customers is essential and becomes a part of guiding development and testing effort, as well as giving input to requirements engineering. In particular, from the point of view lean software development perspective, it is important to take an end to end perspective, from inception of an idea to sales and deployment.

4.3.2. Ontology and filtering

Practitioners were generally excited about the use of ontologies and making structural searches with respect to the ontology. However, none 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 the Telecom domain.

Ontology complexity and structure: A practitioner mentioned his concerns about the use of ontologies as an ontology can become quite big and have a lot of branches, which makes it too complex. Repeated breaking down the information to branches might make people lose track and become confused. He stated that "Although the usage of taxonomies is good for a 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. Another interviewee foresaw this and suggested gaining the ability 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 was no complete tree structure. This interviewee suggested keeping the ontology very general and focusing on the tagging system.

Usefulness of the SWEBOK ontology: When it comes to the choice of ontologies, interviewees were asked if they would like the see knowledge areas based on SWEBOK in the ontology structure so that they could use them to extract and filter information. However, all of them stated that they did not really need that kind of queries and one subject stated that these knowledge areas and lifecycle phases were not very clear when you used agile development. They declared their own choice of ontology would be useful for them.

Document type ontology: Document types and domains were the most desired ontologies by the interviewees. Three subjects specified that they would like to see the document types in the ontology so that they could filter the documents according to type. All the interviewees were asked to discuss their usage scenarios for these collaboration tools and the type of documents they dealt with. For the document types they gave the following examples: product description documents, project planning documents (requirements, user stories), design documents, business process modelling 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 problems related to the document type by stating that "The problem with document types is that there is no common structure about where to place these documents in the project repository. It can be anywhere." Hence, the partcipants could not easily find a specific document for a certain project or product. One interviewee denoted that if the semantic system could recognize the type of the document automatically by processing the content of the document, it would be a benefit for them.

Telecom domain ontology: Another common suggestion was a domain ontology based on telecom operations and services. Four interviewees mentioned that when they searched for a term, the results came from all different domains that were not interesting for them. When they were asked about what exactly they meant when they sais domain, one interviewee only stated that he would like to see only the results from the network (technical) domain or from the business domain. The other three participants were slightly 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), etc.

They suggested using eTOM[9] (Enhanced Telecom Operations Map) which is a guidebook that defines the most common standards for business processes in the telecommunications industry.

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

Organization-specific ontology: Another subject proposed the Ericsson project management framework PROPS-C as an alternative to the classical lifecycles defined in SWEBOK. This framework includes the business readiness, sales and project management processes. They are all composed of such phases as analysis, planning, monitoring, execution, contract management, etc. The interviewee suggested searching for documents according to these defined phases.

The same subject proposed to have the Ericsson Product Catalogue domain in the ontology. He said 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 catalogue 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 some place in the product catalogue, 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, however, the information about these projects ia lost in local repositories. Hence, relations between projects and the products from the catalogue can be useful for finding documents.

[9] eTOM: http://www.tmforum.org/BestPracticesStandards/BusinessProcessFramework/6637/Home.html

4.3.3. Improvements for the knowledge-based system

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 mentioned some improvements that can be applied in the system.

Search mechanisms: First of all, one interviewee stated that they do not want to be locked into a set of predefined queries when making 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 the ability to search for entities that do not satisfy the relation specified in the search pattern. For instance, searching for people who talk or do not talk about a certain topic should be available. He explained his concern by stating that "For example if competitors in our knowledge base haven't talked about something, it means that we don't have any understanding about what they are doing. Because they must talk about it."

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" at the usage scenarios in the beginning of the case study.

Tagging: All the interviewees at some point mentioned tags and they pointed out the importance of an intelligent tagging system. They indicated that tags are very useful for understanding the context and content of a document and a search engine should consider tags in a smart way in the search algorithm. However, they all agreed that tags in the current system were not used efficiently at all. One interviewee stated that people did not know the purpose of tags so they just wrote something or left it empty. Another interviewee mentioned 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 tags. One interviewee said that "In the case of an 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, the interviewees opposed to the introduction of a fully automatic solution. 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. However, 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 then the tagging is fully automatic. Moreover, one interviewee suggested binding tags with entities in the ontology which are 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 because recognized entities already have their descriptions.

Results presentation: Furthermore, some participants suggested improvements in the representation of the results. For example, one of the subjects wanted to see the tags or the summary of the document directly in the search results so that it can help them 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 one can have traceable trees based on location, product, etc.

The key findings for research question RQ2 are stated below.

Key findings and observations for RQ2:

i) The ontologies related to software engineering were not of the main interest to practitioners.

They were more interested in domain-specific ontologies and document ontologies (recognizing a document type).

ii) The practitioners were positive about the different search options in KIM, in particular the Facet search and the Structural search. Being able to see extracted information without making a query is of great interest, however, it is not provided by traditional search tools. This also facilitates easy filtering, which was important to them.

iii) It is important to have simple ontologies to be still understandable.

iv) There should be a possibility to filter a search query by the domain, document type, and organizational structure.

v) The costs of implementation, migration, and maintenance have been raised as an important factor.

vi) In summary, the interview subjects denoted diverse opinions about the use of ontologies and what type of ontology they would like to see. However, the domain and documentation seem to be most dominant ones.

5. Conclusion

In this work, 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 be aware of. The problems are related to the search engine and the structure of the existing tools, the technology is able to 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. The research questions asked can be answered as follows.

RQ1: How to implement semantic knowledge management systems? First individual components were implemented and an attempt was made to integrate them. This was a considerable effort, and the use of an already integrated solution (here KIM) was preferred. Still, the difficulty of integrating and updating new ontologies was high. It was found that practitioners need tailored ontologies, which is a hindrance for technology transfer. In general, the KIM system should reuse existing components (e.g. GATE) and ontologies as much as possible. However, the difficulty was to actually work and integrate the components. Even with the integrated solution, it was difficult to add and modify ontologies.

RQ2: How useful are semantic knowledge management systems in finding relevant knowledge in software engineering? The key part of a semantic knowledge management system is the ontology to be used, as the most beneficial structure has to be found. So far, we could not find any completed and released software engineering ontology that covers all the knowledge in the domain. Yet, the case study revealed that this was not necessarily needed. It was found that the 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 observed that the business domain of the organization was equally if not more important, the practitioners indicated that the information they reuse or search is often related to domain specific knowledge, solutions, products, business processes, etc. 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 catalogue of the company.

Overall, when looking at the initial requirements one may reason on their fulfilment.

– Structure of information: The need to structure information and making people aware of this structure was highlighted as very important. A means to do this are ontologies. Given the difficulty of updating and adding new ontologies, the requirement has only been partially fulfilled.

- Finding experts: This also requires the update of the ontology incorporating organization-specific roles and terminology. Hence, only with an easy updating method, this would be achieved.

Future work: A replication 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 necessary to conduct several case studies. On the other hand, another company in the telecommunications domain should also be analysed in order to remove the defined external validity threats. Also experimentation is needed. That is, in future work, the actual time to find information should be measured and also the quality of the decisions should be evaluated. This study may help in formulating research propositions as well as providing explanations for quantitative findings.

Acknowledgments

The work was partially supported by a research grant for the ORION project (reference number 20140218) from The Knowledge Foundation in Sweden.

References

[1] J.L. Krein, P. Wagstrom, S.M. Sutton Jr, C. Williams, and C.D. Knutson, "The problem of private information in large software organizations," in *Proceedings of the 2011 International Conference on Software and Systems Process*. ACM, 2011, pp. 218–222.

[2] E. Carmel and R. Agarwal, "Tactical approaches for alleviating distance in global software development," *IEEE Software*, Vol. 18, No. 2, 2001, pp. 22–29.

[3] J. Grudin, "Enterprise knowledge management and emerging technologies," in *Proceedings of the 39th Annual Hawaii International Conference on System Sciences*, Vol. 3. IEEE, 2006, pp. 57a–57a.

[4] M. Alavi and D.E. Leidner, "Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues," *MIS quarterly*, 2001, pp. 107–136.

[5] C.Y. Yang and S.J. Wu, "Semantic web information retrieval based on the Wordnet." *International Journal of Digital Content Technology & its Applications*, Vol. 6, No. 6, 2012.

[6] J. Mustafa, S. Khan, and K. Latif, "Ontology based semantic information retrieval," in *4th International IEEE Conference Intelligent Systems*, Vol. 3. IEEE, 2008, pp. 22–14.

[7] W. Wei, P.M. Barnaghi, and A. Bargiela, "Semantic-enhanced information search and retrieval," in *Sixth International Conference on Advanced Language Processing and Web Information Technology*. IEEE, 2007, pp. 218–223.

[8] A. Edmunds and A. Morris, "The problem of information overload in business organisations: A review of the literature," *International journal of information management*, Vol. 20, No. 1, 2000, pp. 17–28.

[9] M.J. Eppler and J. Mengis, "The concept of information overload: A review of literature from organization science, accounting, marketing, MIS, and related disciplines," *The information society*, Vol. 20, No. 5, 2004, pp. 325–344.

[10] O.E. Klapp, *Overload and boredom: Essays on the quality of life in the information society*. Greenwood Publishing Group Inc., 1986.

[11] J. Feather, *The information society: A study of continuity and change*. London: Facet Publishing, 2004.

[12] H. Butcher, *Meeting managers' information needs*. London: ASLIB/IMI, 1998.

[13] R. Guha, R. McCool, and E. Miller, "Semantic search," in *Proceedings of the 12th international conference on World Wide Web*. ACM, 2003, pp. 700–709.

[14] N.J. Belkin and W.B. Croft, "Information filtering and information retrieval: Two sides of the same coin?" *Communications of the ACM*, Vol. 35, No. 12, 1992, pp. 29–38.

[15] C.J.V. Rijsbergen, *Information Retrieval*, 2nd ed. Newton, MA, USA: Butterworth-Heinemann, 1979.

[16] G. Salton, A. Wong, and C.S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, Vol. 18, No. 11, 1975, pp. 613–620.

[17] P. Warren, "Building semantic applications with SEKT," in *Integration of Knowledge, Semantics and Digital Media Technology, 2005. EWIMT 2005. The 2nd European Workshop on the (Ref. No. 2005/11099)*. IET, 2005, pp. 429–436.

[18] C.D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*. Cambridge University Press, 2008, Vol. 1.

[19] A.K. Jain, M.N. Murty, and P.J. Flynn, "Data clustering: A review," *ACM computing surveys (CSUR)*, Vol. 31, No. 3, 1999, pp. 264–323.

[20] I.N. Athanasiadis, F. Villa, and A.E. Rizzoli, "Enabling knowledge-based software engineering through semantic-object-relational mappings," in *Proceedings of the 3rd International Workshop on Semantic Web Enabled Software Engineering*, 2007.

[21] R. Witte, Y. Zhang, and J. Rilling, "Empowering software maintainers with semantic web technologies," in *The Semantic Web: Research and Applications*. Springer, 2007, pp. 37–52.

[22] C. Kiefer, A. Bernstein, and J. Tappolet, "Analyzing software with iSPARQL," in *Proceedings of the 3rd ESWC International Workshop on Semantic Web Enabled Software Engineering (SWESE)*, 2007.

[23] Y. Zhao, J. Dong, and T. Peng, "Ontology classification for semantic-web-based software engineering," *IEEE Transactions on Services Computing*, Vol. 2, No. 4, 2009, pp. 303–317.

[24] B. Decker, E. Ras, J. Rech, B. Klein, and C. Hoecht, "Self-organized reuse of software engineering knowledge supported by semantic wikis," in *Proceedings of the Workshop on Semantic Web Enabled Software Engineering (SWESE)*, 2005.

[25] E. Simperl, I. Thurlow, P. Warren, F. Dengler, J. Davies, M. Grobelnik, D. Mladenic, J.M. Gomez-Perez, and C.R. Moreno, "Overcoming information overload in the enterprise: The active approach," *IEEE Internet Computing*, Vol. 14, No. 6, 2010, pp. 39–46.

[26] D. Hyland-Wood, D. Carrington, and S. Kaplan, "Toward a software maintenance methodology using semantic web techniques," in *Second International IEEE Workshop on Software Evolvability*. IEEE, 2006, pp. 23–30.

[27] B. Popov, A. Kiryakov, D. Ognyanoff, D. Manov, and A. Kirilov, "KIM – A semantic platform for information extraction and retrieval," *Natural language engineering*, Vol. 10, No. 3-4, 2004, pp. 375–392.

[28] V. Uren, P. Cimiano, J. Iria, S. Handschuh, M. Vargas-Vera, E. Motta, and F. Ciravegna, "Semantic annotation for knowledge management: Requirements and a survey of the state of the art," *Web Semantics: science, services and agents on the World Wide Web*, Vol. 4, No. 1, 2006, pp. 14–28.

[29] T.R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing?" *International Journal of Human-Computer Studies*, Vol. 43, No. 5, 1995, pp. 907–928.

[30] M. Ashburner, C.A. Ball, J.A. Blake, D. Botstein, H. Butler, J.M. Cherry, A.P. Davis, K. Dolinski, S.S. Dwight, J.T. Eppig *et al.*, "Gene ontology: Tool for the unification of biology," *Nature genetics*, Vol. 25, No. 1, 2000, pp. 25–29.

[31] A. Kanso and D. Monette, "Foundations for long-term collaborative research," in *Proceedings of the 2014 ACM International Workshop on Long-term Industrial Collaboration on Software Engineering (WISE 2014), Vasteras, Sweden, September 16, 2014*, 2014, pp. 43–48.

[32] V. Garousi, K. Petersen, and B. Özkan, "Challenges and best practices in industry-academia collaborations in software engineering: A systematic literature review," *Information & Software Technology*, Vol. 79, 2016, pp. 106–127.

[33] A. Arcuri, "An experience report on applying software testing academic results in industry: We need usable automated test generation," *Empirical Software Engineering, in print*, 2017, pp. 1–23.

[34] T. Gorschek, P. Garre, S. Larsson, and C. Wohlin, "A model for technology transfer in practice," *IEEE Software*, Vol. 23, No. 6, 2006, pp. 88–95.

[35] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical software engineering*, Vol. 14, No. 2, 2009, pp. 131–164.

[36] R.J. Thierauf, *Knowledge management systems for business*. Greenwood Publishing Group, 1999.

[37] J. Davies, D. Fensel, and F. Van Harmelen, *Towards the semantic web: Ontology-driven knowledge management*. John Wiley & Sons, 2003.

[38] A. Kiryakov, B. Popov, I. Terziev, D. Manov, and D. Ognyanoff, "Semantic annotation, indexing, and retrieval," *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 2, No. 1, 2004, pp. 49–79.

[39] S. Schaffert, F. Bry, J. Baumeister, and M. Kiesel, "Semantic wikis," *IEEE Software*, Vol. 25, No. 4, 2008, pp. 8–11.

[40] E. Oren, M. Völkel, J.G. Breslin, and S. Decker, "Semantic wikis for personal knowledge manage-

ment," in *Database and Expert Systems Applications.* Springer, 2006, pp. 509–518.
[41] P. Warren, J.M. Gómez-Pérez, and C.R. Moreno, "ACTIVE – enabling the knowledge-powered enterprise," in *International Semantic Web Conference (Posters & Demos)*, 2008.
[42] V. Ermolayev, C.R. Moreno, M. Tilly, E. Jentzsch, J.M. Gomez-Perez, and W.E. Matzke, "A context model for knowledge workers," in *Proceedings of the Second Workshop on Context, Information and Ontologies*, V. Ermolayev, J.M. Gomez-Perez, P. Haase, and P. Warren, Eds., 2010.
[43] R. Baeza-Yates, B. Ribeiro-Neto *et al.*, *Modern information retrieval.* ACM press New York, 1999, Vol. 463.
[44] T. Calders, G.H. Fletcher, F. Kamiran, and M. Pechenizkiy, "Technologies for dealing with information overload: An engineer's point of view," *Information Overload: An International Challenge for Professional Engineers and Technical Communicators*, 2012, pp. 175–202.
[45] D. Hiemstra, "Information retrieval models," *Information Retrieval: Searching in the 21st Century*, 2009, pp. 2–19.
[46] T. Berners-Lee, J. Hendler, O. Lassila *et al.*, "The semantic web," *Scientific American*, Vol. 284, No. 5, 2001, pp. 28–37.
[47] M. Fernández, I. Cantador, V. López, D. Vallet, P. Castells, and E. Motta, "Semantically enhanced information retrieval: An ontology-based approach," *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 9, No. 4, 2011, pp. 434–452.
[48] H.J. Happel and S. Seedorf, "Applications of ontologies in software engineering," in *2nd International Workshop on Semantic Web Enabled Software Engineering (SWESE 2006)*, 2006. [Online]. https://km.aifb.kit.edu/ws/swese2006/final/happel_full.pdf
[49] J. Scott Hawker, H. Ma, and R. Smith, "A web-based process and process models to find and deliver information to improve the quality of flight software," in *The 22nd Digital Avionics Systems Conference*, Vol. 1. IEEE, 2003, pp. 3–B.
[50] J. Caralt and J.W. Kim, "Ontology driven requirements query," in *40th Annual Hawaii International Conference on System Sciences.* IEEE, 2007, pp. 197c–197c.
[51] P. Inostroza and H. Astudillo, "Emergent architectural component characterization using semantic web technologies," in *Proc. Second International Workshop Semantic Web Enabled Software Eng.* Citeseer, 2006. [Online]. https://km.aifb.kit.edu/ws/swese2006/final/inostroza_full.pdf
[52] B. Antunes, P. Gomes, and N. Seco, "SRS: A software reuse system based on the semantic web," in *3rd International Workshop on Semantic Web Enabled Software Engineering (SWESE)*, 2007.
[53] A. Abran, P. Bourque, R. Dupuis, and J.W. Moore, *Guide to the software engineering body of knowledge – SWEBOK.* IEEE Press, 2001.
[54] C. Calero, F. Ruiz, and M. Piattini, *Ontologies for software engineering and software technology.* Springer Science & Business Media, 2006.
[55] P. Wongthongtham, E. Chang, T. Dillon, and I. Sommerville, "Development of a software engineering ontology for multisite software development," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 21, No. 8, 2009, pp. 1205–1217.
[56] O. Mendes, A. Abran, and H.K.M. Québec, "Software engineering ontology: A development methodology," *Metrics News*, Vol. 9, 2004.
[57] J.R. Hilera and L. Fernández-Sanz, "Developing domain-ontologies to improve sofware engineering knowledge," in *Fifth International Conference on Software Engineering Advances (ICSEA).* IEEE, 2010, pp. 380–383.
[58] J. Radatz, A. Geraci, and F. Katki, *IEEE standard glossary of software engineering terminology*, The Institute of Electrical and Electronics Engineers, Inc. Std. 610.12-1990(R2002), 1990.
[59] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan, "GATE: An architecture for development of robust HLT applications," in *Proceedings of the 40th annual meeting on association for computational linguistics.* Association for Computational Linguistics, 2002, pp. 168–175.
[60] K. Bontcheva, V. Tablan, D. Maynard, and H. Cunningham, "Evolving GATE to meet new challenges in language engineering," *Natural Language Engineering*, Vol. 10, No. 3-4, 2004, pp. 349–373.
[61] D. Ferrucci and A. Lally, "UIMA: An architectural approach to unstructured information processing in the corporate research environment," *Natural Language Engineering*, Vol. 10, No. 3-4, 2004, pp. 327–348.
[62] R.K. Yin, *Case study research: Design and methods.* SAGE Publishing, 2013.
[63] W.G. Cochran, *Sampling techniques.* John Wiley & Sons, 2007.
[64] C. Robson, *Real world research*, 2nd ed. Oxford: Blackwell Publishing, 2002.
[65] I. Rus and M. Lindvall, "Guest editors' introduction: Knowledge management in software engi-

neering," *IEEE Software*, Vol. 19, No. 3, 2002, pp. 26–38.

[66] N.F. Noy and D.L. McGuinness, *Ontology Development 101: A Guide to Creating Your First Ontology*, Stanford University, (2002). [Online]. https://protege.stanford.edu/publications/ontology_development/ontology101.pdf

[67] U. Dinger, R. Oberhauser, and C. Reichel, "SWS-ASE: Leveraging web service-based software engineering," in *International Conference on Software Engineering Advances*. IEEE, 2006, pp. 26–26.

Appendix A. Interview guide

A.1. Introduction

- Present yourself
- Ask about recording and confidentiality

The subject of the research is Semantic-Web based Enterprise Knowledge Management system. The focus is on improving information retrieval capabilities in knowledge management systems. That is, we want to explore the benefits of semantic search in enterprise environments. 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 the semantic search is actually extracted knowledge instead of a set of documents that contain the search string. The reason why we would like to conduct interviews is to understand how Ericsson employees gather implicit and explicit knowledge during their daily work and specify 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 the 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 benefit for the organization if we can reduce the time spent on finding relevant information and hence reduce the redundancy of sharing information.

A.2. 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 how you share information or documents in your projects with team members and with other related departments, units, etc.?
   - How would you classify the types of information you share?
   - 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? In which of these information types do you think there is information overload and people spend too much time to access information?
4. How often do you use collaboration tools/information/documentation of Ericsson (give examples)? (scale: daily, weekly) What purposes do you use them for? 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?
6. How would you like to filter?
   - SWEBOK knowledge areas and practices,
   - Software lifecycle phases,
   - Document types,
   - Organizational structure (based on projects, products),
   - Domain.
7. How would you evaluate your satisfaction with the search facilities in these tools? WHY?
8. What do you suggest should be changed or improved when it comes to searching?
9. What do you do if you cannot find the information you are looking for in these tools?
10. How often do you need go and talk to a person with expertise or experience, in order to gather knowledge (even if it is simply an abbreviation that you don't know the meaning of). In what kind of situations does this happen? What kind of information?
11. How do you find the person to ask about a given issue?
12. When you need to ask a question, do you first perform a search if someone already shared this information? If so, do you usually find it or not?

A.3. Demo and evaluation

Present the semantic tool with its functionalities and show search scenario examples based on the loaded discussion forum pages within the system. Illustrate different search types (such as faceted search, browsing the ontology, filtering).

1. What do you think about the presented tool? How would you rate its usefulness? Why?
2. How is the experience different from what you are currently using? Why?
3. Do you think the speed of finding information can change with this technology? If so how much would it change if you had to rate them on a scale?
4. For which type of scenarios and information types?
5. What improvements do you think can be made?
6. Would you use it to find the related people to ask your questions (to gain implicit knowledge)?
7. Would you prefer to add tags manually for every information you share for more accurate results, or you would prefer it automatic like this?
8. What about a software engineering ontology, would you search based on software engineering processes, artefacts?
9. If you have to rate on a scale, what would you say about using a semantic system like this in comparison with the existing systems you have? Would you prefer this version? Why?
10. Do you think we have missed anything important that we can mention? Do you have anything else to add?