Ontology-Based Retrieval of Experts – The Issue of Efficiency and Scalability within the eXtraSpec System

Elżbieta Bukowska, Monika Kaczmarek, Piotr Stolarski, Witold Abramowicz

Ontology-based Retrieval of Experts – the Issue of Efficiency and Scalability within the eXtraSpec System

Elżbieta Bukowska, Monika Kaczmarek, Piotr Stolarski, and Witold Abramowicz

Department of Information Systems,
Faculty of Informatics and Electronic Economy,
Poznan University of Economics
al. Niepodleglosci 10, 61-875 Poznan, Poland
{e.bukowska,m.kaczmarek,p.stolarski,w.abramowicz}
@kie.ue.poznan.pl
http://www.kie.ue.poznan.pl

Abstract. In the knowledge-based economy, organizations often use expert finding systems to identify new candidates or manage information about the current employees. In order to ensure the required level of precision of returned results, the expert finding systems often benefit from semantic technologies and use ontologies in order to represent gathered data. Usage of ontologies however, causes additional challenges connected with the efficiency, scalability as well as the ease of use of a semantic-based solution. Within this paper we present a reasoning scenario applied within the eXtraSpec project and discuss the underlying experiments that were conducted in order to identify the best approach to follow, given the required level of expressiveness of the knowledge representation technique, and other requirements towards the system.

Keywords: expert finding system, expert ontology, reasoning approach

1 INTRODUCTION

In the competitive settings of the knowledge-based economy [OECD, 1996], knowing the skills and expertise of employees as well as conducting an appropriate recruitment process, is a crucial element for the success of an organization. Therefore, organizations turn to IT technology for help [OECD, 1996] and very often take advantage of expert retrieval systems. The traditional expert retrieval systems, being a subset of information retrieval (IR) systems [van Rijsbergen, 1995] , face the same problems as the latter ones. These problems are caused by application of different keywords and different levels of abstraction by users when formulating queries on the same subject or using different words and phrases in the description of a phenomenon, based on which indexes are created. In order to address these issues, very often semantics is applied. There are many initiatives aiming at the development of expert retrieval systems supported by

semantics. One of such initiatives is the eXtraSpec project [Abramowicz et al., 2010]. Its main goal is to combine company's internal electronic documents and information sources available on the Internet in order to provide an effective way of searching experts with competencies in the given field. The eXtraSpec system needs not only to be able to acquire and extract information from various sources, but also requires an appropriate information representation supporting reasoning over person's characteristics. In addition, the reasoning and querying mechanism should, on the one hand, allow to precisely identify required data, and, on the other hand, be efficient and scalable.

The main goal of this paper is to present various reasoning approaches considered within the eXtraSpec project given the required level of expressiveness of the knowledge representation technique, and to discuss the underlying motivation, which led to the development of a semantic-based mechanism to retrieve experts in its current state. The work conducted encompassed both research and practical related aspects. On the one hand, the aim was to contribute to a general understanding of the problem, and on the other hand, the aim was to develop a system that could not only be used as a proof for testing, but also could constitute a fully fledged tool to be used by users. Thus, the System Development Method (SDM) was utilized [Burstein, 2002]. According to Burstein – SDM "allows the exploration of the interplay between theory and practice, advancing the practice, while also offering new insights into theoretical concepts". The approach followed consisted out of three main steps.

First, the concept building phase took place, which resulted in the theoretical concepts presented in the next sections. The next step was system building encompassing development of a system based on the theoretical concepts established. The system development was guided by a number of identified querying strategies that an employer may use in order to discover a potential candidate. The last step was the system evaluation together with the evaluation of three different approaches and the discussion of the obtained results.

In order to meet the defined goal, the paper is structured as follows. First the related work in the relevant research area is shortly discussed. Then, the identified requirements are presented. Next, we focus our attention on the considered reasoning scenarios and the experiments performed in order to select the most appropriate one. The paper concludes with final remarks.

2 RELATED WORK

Following [McDonald and Ackerman, 2000] expert finding systems may aim at expertise identification trying to answer a question: who is an expert on a given topic?, or aim at expertise selection trying to answer a question: what does X know? Within our research, we focus on the first aspect, i.e., on identifying a relevant person given a concrete need.

First systems focusing on expertise identification relied on a database like structure containing a description of experts' skills (e.g., [Yimam-Seid and Kobsa, 2003]). However, such systems faced many problems, e.g., how to ensure precise

results given a generic description of expertise and simultaneously fine-grained and specific queries [Kautz et al., 1996], or how to guarantee the accuracy and validity of stored information given the static nature of a database and volatile nature of person's characteristics. To address these problems other systems were proposed focusing on automated discovery of up-to-date information from specific sources such as, e.g., e-mail communication [Campbell et al., 2003], Intranet documents [Hawking, 2004] or social networks [Michalski et al., 2011] [Metze et al., 2007].

When it comes to the algorithms applied to assess whether a given person is suitable to a given task, at first, standard information retrieval techniques were applied [Ackerman et al., 2002] [Krulwich and Burkey, 1996]. Usually, expertise of a person was represented in a form of a term vector and a query result was represented as a list of relevant persons. If matching a query to a document relies on a simple mechanism checking whether a document contains the given keywords then, the well-known IR problems occur: low precision of returned results, low value of recall and a large number of documents returned by the system the processing of which is impossible. Therefore, a few years ago, the Enterprise Track at the Text Retrieval Conference (TREC) was started in order to study the expert-finding topic. It resulted in further advancements of the expert finding techniques and the application of numerous methods, such as probabilistic techniques or language analysis techniques, to improve the quality of finding systems (e.g., [Balog et al., 2006] [Petkova and Croft, 2006] [Fang and Zhai, 2007] [Serdyukov and Hiemstra, 2008]).

As the Semantic Web technology [Berners-Lee et al., 2001] is getting more and more popular [Shadbolt et al., 2006], it has been used to enrich descriptions within experts finding systems. Semantics in the search systems may be used for analysing indexed documents or queries (query expansion [Navigli and Velardi, 2003]) or operating on semantically described resources with the use of reasoners (e.g., operating on contents of RDF (Resource Description Framework [W3C, 2012]) files and ontologies represented in e.g., OWL (Web Ontology language [OWL, 2012])). Within the expert finding systems, both approaches have been applied as well as a number of various ontologies used to represent competencies and skills were developed, e.g., [Gmez-Prez et al., 2007] [Dorn et al., 2007] [Aleman-Meza et al., 2007].

There are many initiatives that use reasoning over ontologies, e.g., [Goczya et al., 2006] [Haarslev and Mller, 2003]. In [Dentler et al., 2011] authors provide comprehensive comparison of Semantic Web reasoners, considering several characteristics and not limiting it only to reasoning method or computation complexity, but also they analysed supported interfaces or the operating platform. The survey shows that despite significant variety among reasoners, reasoning over complex ontology is still time and resource-consuming.

The problem tackled within this paper is related to the semantic-based expert finding. The eXtraSpec system acquires information from outside and assumes that one can build a profile of a person based on the gathered information. It is important for the users of an expert finding system that the system oper-

ates on a large set of experts. More experts imply bigger topic coverage and increased probability of a question being answered. However, it simultaneously causes problems connected to the heterogeneity of information as well as low values of both the precision and recall measures of the system.

In order to address these issues, the eXtraSpec system benefits from the already developed technologies and tools. However, it offers an added value through their further development and creation of new artefacts. For the needs of the system, a distinct set of ontologies (tailored to the needs of the Polish market as well as taking into account additional non-hierarchical relations) together with a distinct normalization and reasoning (with the pre-reasoning stage) approach have been designed, adjusted to the specific needs of a system.

Within next section we discuss the requirements and show various scenarios considered.

3 REQUIREMENTS

The eXtraSpec system is to support three main scenarios: finding experts with desired characteristic, defining teams of experts and verifying data on a person in question. In order to identify the requirements towards the persons' characteristics, the scope of information needed to be covered by ontologies, as well as the querying and reasoning mechanism developed within the eXtraSpec system, first, exemplary searching scenarios a user looking for experts may be interested in, were considered. The scenarios have been specified based on the conducted studies of the literature and interviews with employers conducting recruitment processes. Six most common searching goals are as follows:

1. To find an expert with some experience at a position/role of interest.
2. To find an expert having some specific language skills on a desired level.
3. To find an expert having some desired competencies.
4. To find students who graduated recently/will graduate soon in a given domain of interest.
5. To find a person having expertise in a specific domain.
6. To find a person with specific education background, competencies, fulfilled roles, etc. Although the enumerated goals (1-5) sometimes are used separately, usually they constitute building blocks of more complex scenarios within which they are freely combined using various logical operators.

The above querying goals imposed some requirements on the information on experts that should be available (e.g., information on the history of employment, certificates), and in consequence, also ontologies that needed to be developed for the project's needs, as well as reasoning and querying mechanism.

3.1 Requirements on ontology and its expressiveness

The creation of ontology for the needs of the eXtraSpec project was preceded by thorough analysis of requirements resulting from the scenarios supported by the system:

1. The ontology MUST represent a *is-a* hierarchy of different positions and jobs allowing for their categorization and reasoning on their hierarchical relations.
2. The ontology MUST represent languages certificates (*is-a* hierarchy) together with information on the language and the proficiency level, mapped to one scale.
3. The ontology MUST represent skills and competencies and their hierarchical dependencies as well as some additional relations as appropriate.
4. The ontology MUST provide a hierarchy of educational organizations allowing for their categorization and reasoning on their hierarchical dependencies.
5. The ontology SHOULD provide information on organizations allowing for their categorization (*is-a* relation) as well as provide information on the domains they operate in.
6. Requirements on ontologies are the same as in scenarios 1-5.

Once, the requirements have been identified, the consequences of applying various formalisms and data models for the ontology modelling and its further application, were investigated. In consequence, three assumptions were formulated: only few relations will be needed and thus, represented; developed ontologies should be easy to translate into other formalisms; expressiveness of used ontology language is important, however, the efficiency of the reasoning mechanism is also crucial. As the result of the conducted analysis of different formalisms and data models, the decision was taken to use the OWL language as the underlying formalisms and the SKOS model as a data model. The criteria that influenced our choice were as follows (for details see [Abramowicz et al., 2012]): relatively easy translation into other formalisms; simplicity of representation; expressiveness of used ontology language, and finally, efficiency of the reasoning mechanism.

The basic element of the eXtraSpec system is an already mentioned profile of an expert. Each expert is described with series of information, for example: name and family name, history of education, career history, hobby, skills, obtained certificates. For the needs of the project, a data structure to hold all that information was designed. To make the reasoning possible, a domain knowledge for each of those attributes is needed. The domain knowledge is represented by the ontology. An ontology, according to the definition provided by Gruber [Gruber, 1995], is a formal, explicit specification of a shared conceptualization. It provides a data model, i.e., shared vocabulary that may be used for describing objects in the domain (their type, properties and relations). The important part of every ontology are the instances forming a knowledge base. Instances refer to a concrete object being an instantiation of an object type represented by the ontology. While annotating texts, the ontology is populated: each word or text snippet may be assigned a proper type from the ontology. During annotation process an instance of ontology is assigned to a given object.

Ten attributes from the profile of an expert were selected to be a 'dictionary reference', i.e., the attributes, which values are references to instances from an ontology. Those attributes are, e.g., Educational organization (name of organization awarding the particular level of education or educational title), Skill (an

ability to do an activity or job well, especially because someone has practiced it) or Scope of education (the domain of education (for example: IT, construction, transportation)) (for the full list see [Abramowicz et al., 2011]). While building the ontology for the needs of the eXtraSpec system, a wide range of taxonomies and classifications has been analyzed in order to identify best practices and solutions. As the eXtraSpec system is a solution designed for the Polish language, so is also the developed ontology.

Performed analysis of the requirements imposed on the ontology for the needs of reasoning, concluded with the definition of a set of relations that should be defined: *hasSuperiorLevel* – to represent hierarchical relations between concepts; *isEquivalent* – to represent substitution between concetps; *isLocatedIn* – to represent geographical dependencies; *isLocatedInCity* – to represent geographical dependencies; *isLocatedInVoivodeship* – to represent geographical dependencies; *provesSkillDegree* – connection between skill and certificate; *worksInLineOfBusiness* – to represent dependencies between organizations and lines of business; *isPartOf* – for representation of composition of elements, for example: ability of using MSWord is a part of ability of using MSOffice (however, knowing MSWord does not imply that a person knows the entire MSOffice suit). Additionally, set of SKOS relations have been used: *broader*, *hasTopConcept*, *inScheme*, *narrower* and *topConceptOf*.

3.2 Requirements towards the reasoning and querying mechanism

One of the most important functionalities of the eXtraSpec system is the identification of persons having the desired expertise. The application of the Semantic Web technologies in order to ensure the appropriate quality of returned results implies application of a reasoning mechanism to answer user queries. The mentioned reasoning mechanism should fulfill the following requirements:

1. The querying and reasoning mechanism MUST be able to integrate experience history (e.g., add the length of duration from different places, but gained on the same or similar position) and then reason on a position's hierarchy (i.e., taking into account narrower or broader concepts).
2. If the information is not explicitly given, the querying and reasoning mechanism SHOULD be able to associate different certificates with languages and proficiency levels.
3. The querying and reasoning mechanism SHOULD be able to operate not only on implicitly given competencies, but also reason on jobs and then on connected competencies. Thus, the querying and reasoning mechanism SHOULD tackle also other relations than *is-a*.
4. The querying and reasoning mechanism MUST be able to reason on the hierarchy of educational organizations, on dates and results.
5. The querying and reasoning mechanism SHOULD be able to associate organizations with domains they operate in.
6. The querying and reasoning mechanism MUST be able to combine results from various querying strategies using different logical operators.

In addition to the requirements mentioned, the following requirements for the querying and reasoning mechanism also need to be considered:

- building queries in a structured way (i.e., feature: desired value);
- supporting definition of desired values of attributes in a way suitable to the type of data stored within the given feature;
- joining a subset of selected criteria within the same category into one complex requirement using different logical operators;
- formulating a set of complex requirements within one category with different logical operators;
- joining complex requirements formulated in various profile categories into one criteria with different logical operators.

The logical operators between different set of criteria and criteria themselves, include such operators as: must, should, must not.

In addition, the developed mechanism is to be used within the settings of large companies or for the needs of the employment market dealing with thousands of people registered and hundreds of queries to be answered. That is why, the reasoning mechanism itself should, on the one hand, support precise identification of required data (it should be ensured by application of the semantic technologies), however, on the other hand, needs to be efficient and scalable.

The next section presents the test-bed and experiments that were conducted in order to identify the best scenario to fulfil the identified requirements (precision and recall on the one hand, and efficiency and scalability on the other), taking into account the identified scenarios and the desired level of expressiveness of the knowledge representation language.

4 EXPERIMENT TESTBED AND RESULTS

The main process in the eXtraSpec system flows as follows. The eXtraSpec system acquires automatically data from dedicated sources, both company external (e.g., LinkedIn portal) and internal ones. As the eXtraSpec system was developed for the needs of Polish market, it operates on the Polish language. The content of an HTML page is parsed and the relevant building blocks are identified. Then, within each block, the relevant information is extracted. The extracted content is saved as an extracted profile (PE), which is an XML file compliant with the defined structure of an expert profile based on the European Curriculum Vitae Standard. Therefore, it consists of a number of attributes, such as, e.g., education level, position, skill, that are assigned to different profile's categories such as, e.g., personal data, educational history, professional experience

Next, data in PE is normalized using the developed ontology (see previous section). As a result of the normalization process, the standardized profile is generated (PN). An important assumption is: one standardized profile describes one person, but one person may be described by a number of standardized profiles (e.g., information on a given person at different points in time or information acquired from different sources). Thus, normalized profiles are analysed and then

aggregated, in order to create an aggregated profile (PA) of a person (i.e., one person is described by one and only one PA).

Finally, the reasoning mechanism is fed with the created aggregated profiles and answers user queries on experts. The queries are formulated with the help of specially developed Graphical User Interface (GUI).

4.1 Considered scenarios

Given the defined requirements from the previous section as well as the already implemented system flow, three possible scenarios of using the reasoning mechanism were considered.

The *first* scenario involves using the *fully-fledged semantics* by expressing all expert profiles as instances of an ontology during normalization phase, formulating queries using the defined ontology, and then, executing a query using the reasoning mechanism. This approach involves the need to load all ontologies into the reasoning engine and representing all individual profiles as ontology instances (see fig. 1).

The *second* scenario relies on the *query expansion* using ontology, i.e., adding keywords to the query by using an ontology to narrow or broaden the meaning of the original query. Thus, each user query needs to be normalized and then expanded using ontology, therefore, application of a reasoner is necessary (see fig. 2).

The *third* scenario called *pre-reasoning* involves two independent processes: (1) creation of enriched profiles (indexes), to which additional information reasoned from the ontology is added and saved within the repository as syntactic data; (2) formulating query with the help of the appropriate GUI using the defined ontology serving as a controlled vocabulary. Then, the query is executed directly on a set of profiles using the traditional mechanisms of IR (e.g., Lucene). There is no need to use the reasoning engine while executing a query (see fig. 3).

In order to make an informed decision, we have run a set of experiments to check the performance and the fulfilment of the identified requirements.

4.2 Experiment design

The implementations of the experiments were preceded with the conceptualization stage. The effect of the conceptualization is presented on the pseudo-UML flow diagrams. Each diagram represents one experiment and is strictly coupled with the implementation given in details below. Note that classes and methods names are not included into conceptual charts. Instead the conceptual operations are only present.

We decided to build a general-purpose framework that will enable to run all experiments. All experiments encompassed two main phases: data preparation and running live experiments.

In the data preparation phase the XML-based profile (being in fact a set of XML files) is being converted into either SQL or OWL (depending on the

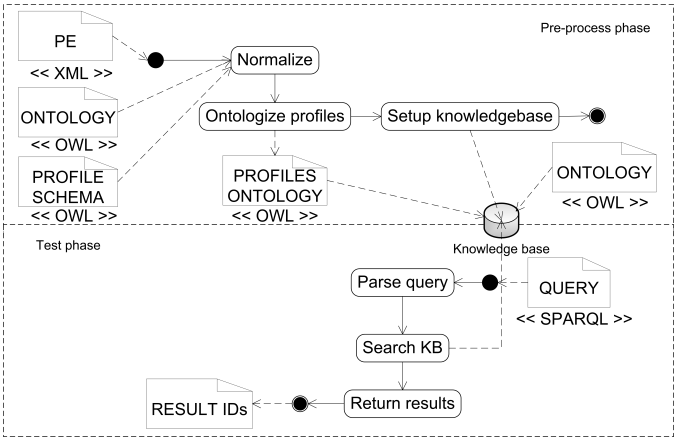

Fig. 1. First scenario – fully-fledged semantics.

scenario). Both conversions are made using the generic *XMLConversionFactory* class. The factory uses the concretizations of *AbstractXMLConverter*, which is either *XMLToOWLConverter* or *XMLToSQLConverter* to perform the actual task.

The OWL conversion employs the XSLT style-sheet prepared with the Java XML2OWL Mapping Tool[1] experimental software by Toni Rodrigues and Pedro Costa. As a consequence, the conversion limits itself to running the style-sheet transform engine targeted at each profile XML file. The OWL schema was prepared manually by an knowledge architect using the Protege OWL editor[2], based on the XML schema definition of an expert profile.

The SQL Lite 3 relational database system[3] has been chosen as an SQL engine. SQL Lite is well known free software, which delivers an out-of-the-box, simple, yet powerful tool. The decision to use SQL Lite was influenced by the fact that other modules of the eXtraSpec project are using this SQL engine. Moreover, it is useful as it works on single files as storage units, which makes management of the databases easy. Similarly to OWL, the SQL schema reflecting the profile information, was prepared manually by a human expert.

We needed to tailor the *XMLToSQLConverter* class in order to produce the proper SQL statements in the SQL Lite flavour. This is normal as almost any SQL engine has some deviations from the standard[4]. In contrast to the OWL conversion, this time the converter class produces JAXB[5] instances of

[1] http://jxml2owl.projects.semwebcentral.org/
[2] http://protege.stanford.edu/
[3] http://www.sqlite.org/
[4] http://www.contrib.andrew.cmu.edu/ shadow/sql/sql1992.txt
[5] http://www.oracle.com/technetwork/articles/javase/index-140168.html

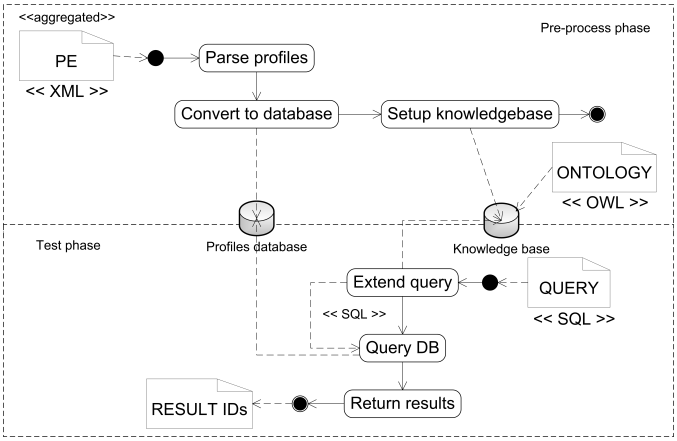

Fig. 2. Second scenario – query expansion.

ProfileExtracted class - a native eXtraSpec artifact. Then, the content of the instance is serialized into series of INSERT statements.

In the case of OWL life-cycle, before the live experiment phase one more step had to be done in advance. The *OntologiesMerger* instance is being used in order to merge all the generated and required ontologies. This includes: OWL profile schema, OWL-converted profile instances, as well as eXtraSpec thesauri and ontologies; the latter being in fact the SKOS vocabulary.

We have chosen the Pellet 2.2 ontology reasoning engine[6] to manage the knowledge bases as well as a SPARQL end-point. The rationale behind the decision is that the engine is provided with moderately good documentation and code examples and it is free to use. Finally, it offers three approaches to internally represent the knowledge base, one being a RDF graph. The representation as RDF is needed when joining SKOS and OWL together.

The *IExperiment* interface contracts only one operation: *runExperiment*. By calling the method on every class which implements *IExperiment* we start the test cycles.

AbstractQuererer class has two concrete subclasses that do the task of firing queries. The *SPARQLQuererer* instance is able to query the Pellet inference engine with the SPARQL language whereas *SQLQuererer* do the same for SQL Lite end-point. The *AbstractQuererer* provides the solution to consume single embedded queries, as well as fire a list of queries taken from the text file. As the result the effects are stored into results file.

The *SQLQuererer* may use the *SQLQueryExpander* instance, if there is registered one. The *SQLQueryExpander* class provides mechanism for parsing SQL SELECT statements and enrich the WHERE clauses in such a way that a larger set of results will be returned. The query expansion so far extracts keywords

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

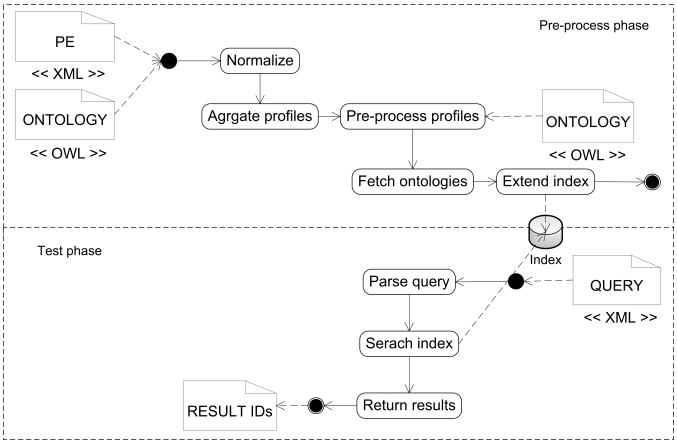

Fig. 3. Third scenario – pre-reasoning.

in the SQL IN conjunction and works on the eXtraSpec SKOS vocabularies in order to find any sub-concepts matching the keyword. If the result set is not empty, then the initial keyword is being replaced with the list of keywords reflecting sub-concepts and the concept itself. The rationale for replacing the IN sets and not parsing the whole WHERE clause with every single condition is the simplicity. On the other hand the IN operator is recognized as of little poorer performance with SELECT queries.

To realize the information retrieval side of the third mechanism, the open-source java library Lucene [Apache, 2012], supported by the Apache Software Foundation, was selected. Instead of searching text documents directly, Lucene searches the previously prepared index. This speeds up the searching process.

Finally, we run all the experiments within the NetBeans 6.9 IDE[7]. The environment is perfect for programming tasks, but above all it provides a ready set of code profilers. The profilers can be thoroughly configured and allow for easy results management together with tools for raw data export into popular data analysis software.

4.3 Results

The test set consisted of nineteen profiles. Test profiles were carefully selected in order to provide sufficient level of reasoning complexity. Profiles were prepared based on real examples extracted from web sources, but enhanced manually in order to cover as many information types as possible. Simultaneously, twelve sample queries have been created. Test queries were prepared to cover as much searching scenarios as possible, including reasoning over different ontologies. We prepared an answer template that combines queries and test profiles that should

[7] http://netbeans.org/

be returned. This template was used to calculate precision and recall of the examined methods. This step was important especially due to the fact that most searching scenarios required reasoning mechanism in order to achieve high recall and precision. Reasoning mechanism had to not only analyze concepts used in queries and profiles, but also super- and sub-concepts from ontologies.

Since we measured system efficiency based on prototype that contains limited resources (i.e., limited number of profiles and branches in ontologies), test queries were executed 100 times in a row which gave a set of 1200 queries. We have obtained the results presented in tab 1, while usage of the memory is presented in fig. 4.

Table 1. Experiments results

	SparqlExperiment	**QueryExpansion**	**Pre-reasoning (native)**
No. of queries	1200	1200	1200
Execution time (ms)	43937	82173	30597
Precision	0,99	0,86	0,98
Recall	0,96	0,63	0,95

Our experiments showed that applying the fully-fledged semantics is a precise, but neither efficient nor scalable solution – in our settings it was not able to fulfill the expected load of the system.

The query expansion provides an increased precision of the results (in comparison to the traditional IR mechanisms) and has better scalability and efficiency than the fully-fledged semantics, however, does not allow to take full advantage of the developed ontologies and existing relations between concepts.

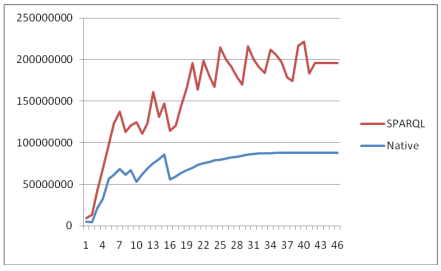

Fig. 4. Memory usage

Only application of the third considered scenario allows taking advantage of the mature IR mechanisms while increasing the accuracy and completeness of the returned results by: introducing a preliminary stage called pre-reasoning in

order to create enriched indexes and the minimum use of the reasoning engine during the search.

The slight decrease in the precision and recall values in the comparison to the fully-fledged semantics was caused by the errors that occurred within the normalization phase, and were not caused by the reasoning mechanism itself. The identified errors in the normalizer component were corrected and the precision and recall values obtained within the new run equalled to the ones of the fully-fledged semantics.

Thus, taking into account the formulated requirements and obtained results, we decided to use the third scenario:

- First, creating indexes of profiles – optimized for search, i.e., structured so as to enable a very fast search based on criteria pre-set by a user. The aggregated profile is analysed, divided into relevant sections, and then enriched with additional information using an ontology (pre-reasoning). Any modification of the ontology forces the need to change indexes.
- The second process that needs to be supported is defining the query matching mechanism on the enriched indexes – this process is initiated by the task of a user formulating queries using a graphical interface. An employer, constructing a query, points interesting criteria and values they should meet. In the background of the interface, the desired values of various features from the lists and combo boxes, point to specific elements from the ontology.

5 CONCLUSIONS

The main goal of the eXtraSpec project was to develop a system supporting analysis of company documents and selected Internet sources for the needs of searching for experts from a given field or with specific competencies. The provided system focuses on processing texts written in the Polish language. The obtained information is stored in the system in the form of expert's profiles and may be consolidated when needed.

Within this paper, we have discussed the concept and considered scenarios regarding the implementation of the reasoning mechanism for the needs of the eXtraSpec system. We argue that by introducing the pre-reasoning phase, the application of semantics may be used to achieve precise results when searching for experts and at the same time, ensure the proper performance and scalability. The conducted experiments have shown that the selected scenario constitute a compromise between the expressiveness and efficiency of the developed solution.

Applying semantics undoubtedly offers a way to handle precision, recall, and helps to normalize data, however, application of semantics impacts the performance as well as scalability of the system. Therefore, a design decision needs always to be taken regarding the way the semantics should be applied in order to ensure the required quality of the system, given the expected expressiveness level of the knowledge representation. Thus, semantic technology has undoubtedly many to offer, however, its adoption in real-life scenarios will be hampered, until the set of mature tools is delivered.

References

[Abramowicz et al., 2011] Abramowicz, W., Bukowska, E., Kaczmarek, M., and Starzecka, M. (2011). Semantic-enabled efficient and scalable retrieval of experts. In *International Conference on Information, Process, and Knowledge Management, eKNOW 2011.*

[Abramowicz et al., 2012] Abramowicz, W., Bukowska, E., Kaczmarek, M., and Starzecka, M. (2012). Ontology structure, reasoning approach and querying mechanism in a semantic-enabled efficient and scalable retrieval of experts. *International Journal On Advances in Software.* Accepted for publication.

[Abramowicz et al., 2010] Abramowicz, W., Kaczmarek, T., Stolarski, P., Wecel, K., and Wieloch, K. (2010). Architektura systemu wyszukiwania ekspertów eXtraSpec. In *Technologie Wiedzy w Zarzadzaniu Publicznym.*

[Ackerman et al., 2002] Ackerman, M., Wulf, V., and Pipek, V. (2002). *Sharing Expertise: Beyond Knowledge Management.* MIT Press.

[Aleman-Meza et al., 2007] Aleman-Meza, B., Boj?rs, U., Boley, H., Breslin, J. G., Mochol, M., Nixon, L. J., Polleres, A., and Zhdanova, A. V. (2007). Combining rdf vocabularies for expert finding. In *Proceedings of the 4th European conference on The Semantic Web: Research and Applications*, pages 235–250, Innsbruck, Austria. Springer-Verlag.

[Apache, 2012] Apache (2012). http://lucene.apache.org. last access date: 22.03.2012.

[Balog et al., 2006] Balog, K., L., A., and De. Rijke, M. (2006). Formal models for expert finding in enterprise corpora. In *Proceedings of the ACM SIGIR*, pages 43–50.

[Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. http:www.scientificamerican.comarticle.cfm?id=the-semantic-web; 20 May 2009.

[Burstein, 2002] Burstein, F. (2002). *Research methods for students and professionals: Information management and systems*, volume 2, chapter Systems development in information systems research, pages 147–158. Wagga Wagga, Australia: Centre for Information Studies, Charles Sturt University, 2nd edition.

[Campbell et al., 2003] Campbell, C. S., Maglio, P. P., Cozzi, A., and Dom, B. (2003). Expertise identification using email communications. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 528–321. ACM Press.

[Dentler et al., 2011] Dentler, K., Cornet, R., Ten Teije, A., and De Keizer, N. (2011). Comparison of reasoners for large ontologies in the owl 2 el profile. *Semantic Web Journal to appear Available from httpwwwsemanticwebjournalnet*, 1(2):1–5.

[Dorn et al., 2007] Dorn, J., Naz, T., and Pichlmair, M. (2007). Ontology development for human resource management. In *Proceedings of 4th International Conference on Knowledge Management*, Series on Information and Knowledge Management, pages 109–120.

[Fang and Zhai, 2007] Fang, H. and Zhai, C. (2007). Probabilistic models for expert finding. In *Proceedingsof the ECIR*, pages 418–430.

[Goczya et al., 2006] Goczya, K., Grabowska, T., Waloszek, W., and Zawadzki, M. (2006). The knowledge cartography a new approach to reasoning over description logics ontologies. In Wiedermann, J., Tel, G., Pokorn, J., Bielikov, M., and tuller, J., editors, *SOFSEM 2006: Theory and Practice of Computer Science*, volume 3831 of *Lecture Notes in Computer Science*, pages 293–302. Springer Berlin / Heidelberg.

[Gruber, 1995] Gruber, T. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computation Studies*, 43:907928.

[Gmez-Prez et al., 2007] Gmez-Prez, A., Ramrez, J., and Villazn-Terrazas, B. (2007). An ontology for modelling human resources management based on standards. In Apolloni, B., Howlett, R. J., and Jain, L. C., editors, *KES (1)*, volume 4692 of *Lecture Notes in Computer Science*, pages 534–541. Springer.

[Haarslev and Mller, 2003] Haarslev, V. and Mller, R. (2003). Racer: An owl reasoning agent for the semantic web. In *Proc. Int'l Wkshp on Applications, Products and Services of Web-based Support Systems (Held at 2003 IEEE/WIC Int'l Conf. on Web Intelligence)*, pages 91–95. Society Press.

[Hawking, 2004] Hawking, D. (2004). Challenges in enterprise search. In *Proceedings of the 15th Australasian database conference – Volume 27*, ADC '04, pages 15–24, Darlinghurst, Australia, Australia. Australian Computer Society, Inc.

[Kautz et al., 1996] Kautz, H., Selman, B., and Milewski, A. (1996). Agent amplified communication. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, pages 3–9.

[Krulwich and Burkey, 1996] Krulwich, B. and Burkey, C. (1996). Contactfinder agent: answering bulletin board questions with referrals. In *Proceedings of the National Conference on Artificial Intelligence*, pages 10–15.

[McDonald and Ackerman, 2000] McDonald, D. W. and Ackerman, M. S. (2000). Expertise recommender: a flexible recommendation system and architecture. In *CSCW'00: Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 231–240. ACM Press.

[Metze et al., 2007] Metze, F., Bauckhage, C., and Alpcan, T. (2007). The "spree" expert finding system. In *Proceedings of the First IEEE International Conference on Semantic Computing (ICSC 2007)*, pages 551–558. IEEE Computer Society.

[Michalski et al., 2011] Michalski, R., Palus, S., and Kazienko, P. (2011). Matching organizational structure and social network extracted from email communication. In Abramowicz, W., editor, *BIS*, volume 87 of *Lecture Notes in Business Information Processing*, pages 197–206. Springer.

[Navigli and Velardi, 2003] Navigli, R. and Velardi, P. (2003). An analysis of ontology-based query expansion strategies. In *Workshop on Adaptive Text Extraction and Mining, (Cavtat Dubrovnik, Croatia, Sept 23)*.

[OECD, 1996] OECD (1996). The knowledge-based economy. Retrieved from http://www.oecd.org/dataoecd/51/8/1913021.pdf. GENERAL DISTRIBUTION, OCDE/GD(96)102.

[OWL, 2012] OWL (2012). http://www.w3.org/TR/2004/REC-owl-features-20040210/. last access date: 22.03.2012.

[Petkova and Croft, 2006] Petkova, D. and Croft, W. (2006). Hierarchical language models for expert finding in enterprise corpora. In *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intel-ligence*, pages 599–608.

[Serdyukov and Hiemstra, 2008] Serdyukov, P. and Hiemstra, D. (2008). Modeling documents as mixtures of persons for expert finding. In *Proceedings of the ECIR*, pages 309–320.

[Shadbolt et al., 2006] Shadbolt, N., Berners-Lee, T., and Hall, W. (2006). The semantic web revisited. *IEEE Intelligent Systems*, 21(3):96–101.

[van Rijsbergen, 1995] van Rijsbergen, C. J. (1995). Information retrieval and information reasoning. In *Computer Science Today*, pages 549–559.

[W3C, 2012] W3C (2012). http://www.w3.org/RDF/. last access date: 22.03.2012.

[Yimam-Seid and Kobsa, 2003] Yimam-Seid, D. and Kobsa, A. (2003). Expert finding systems for organizations: Problem and domain analysis and the demoir approach. In *Journal of Organizational Computing and Electronic Commerce*, 13(1), pages 1–24.