International Journal of Software Engineering and Knowledge Engineering
© World Scientific Publishing Company

ASSISTED KNOWLEDGE BASE GENERATION, MANAGEMENT AND COMPETENCE RETRIEVAL

ANDREA BELLANDI, PIERFRANCESCO BELLINI, ANTONIO CAPPUCCIO,
PAOLO NESI, GIANNI PANTALEO, NADIA RAUCH

*Distributed Systems and Internet Technology, Department of Systems and Informatics, University of Florence,
Firenze, Italy, tel: +39-0554796523, fax: +39-055-4796363
{bellandi, pbellini, cappuccino, nesi, pantaleo, rauch}@dsi.unifi.it*

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

Despite the presence of many systems for developing and managing structured taxonomies and/or SKOS models for a given domain for which small documents set are accessible, the production and maintenance of these domain knowledge bases is still a very expensive and time consuming process. This paper proposes a solution for assisting expert users in the development and management of knowledge base, including SKOS and ontologies modeling structures and relationships. The proposed solution accelerates the knowledge production by crawling and exploiting different kinds of sources (in multiple languages and with several inconsistencies among them). The proposed tool supports the experts in defining relationships among the most recurrent concepts, reducing the time to SKOS production and allowing assisted production. The validity of the produced knowledge base has been assessed by using SPARQL query interface and a precision and recall model. The solution has been developed for Open Space Innovative Mind project, with the aim of creating a portal to allow industries at posing semantic queries to discover potential competences in a large institution such as the University of Florence, in which several distinct domains are associated with its own departments.

Keywords: SKOS; Semantic Web; Skills Management System, knowledge management, semantic queries, validation.

1. Introduction

In many research and application areas, information retrieval and acquisition of data from different web resources has become a common approach. In such contexts, the information needed usually reaches a quite high level of specialization. Therefore, especially non-expert users may encounter difficulties in finding relevant results. This is due to the fact that Internet search engines mainly rely on textual keywords matching; besides, users' approach is usually based on natural language queries. Because of these reasons, the necessity has arisen to develop information retrieval systems which are able to infer and maintain semantic relationships among text terms. In the Semantic Web era, representing knowledge in the form of ontologies, thesaurus, taxonomies and other type of semantic data has become increasingly mandatory. The semantic markup is widely available, both to enable sophisticated interoperability among agents and to support human web users in locating and making sense of information.

Among the models for content semantic classification and enrichment, the Simplified Knowledge Organization System, SKOS, is probably the most diffused [1]. It is a data model for sharing and representing Knowledge Organization Systems (KOS) such as thesauri, classification schemes, term lists, controlled vocabulary and taxonomies within the framework of the Semantic Web. The SKOS model is typically defined using the Resource Description Framework (RDF) [2], and it allows information to be machine-understandable and computable by automatic software agents. Thus, for content classification, the adoption of a SKOS is one of the first step to enter in the semantic world. The adoption of a knowledge representation structures for content involves many advantages:

- the semantic information is available in machine readable format and can benefit from the emergent technologies of the semantic web;
- the increasing spread of the semantic search engines like Swoogle [3], Watson [4, 5] and SIndice [6] helps to find information with a high degree of precision, especially when the solutions are integrated with Natural Language Processing (NLP) capabilities;
- the users are helped in tagging their content using a predefined set of terms (also called vocabulary or terms, which can be multilingual and supported by a thesaurus) belonging to the SKOS that should be validated and accepted by the community. The terms may be also provided in different languages, and their translations validated as well.

The SKOS can be a model at which one may tend after the collection of free tags from the users, the so called folksonomy. The free tags could be statistically analyzed to build a taxonomy and finally a SKOS adding related relationships. Tools for SKOS editing are quite diffuse, such as thManager [7] (a simple SKOS editor), SKOSEd [8], a Protégé [9] plug in for editing SKOS.

1.1. *Related Work and State of the Art*

In order to produce a SKOS or an ontology modeling concepts and their relationships, many techniques have been developed. As a matter of fact, the production of ontology learning frameworks has been recently focused on more specific knowledge sub-domains, starting from manually created or semi-automated seed ontologies [10, 11, 12]. Techniques based on terms co-occurrence (assuming that entities occurring together in a sentence are related, without typically capturing the semantics of the relationships [13]), synonyms or hierarchical similarity are often joined with statistical analysis to derive semantic similarities between entities [14].

However, despite the presence of emerging systems for developing and managing structured taxonomies and/or SKOS models for a given domain, the automated production and maintenance of domain knowledge bases is still a process often very expensive and time consuming. Most of the above mentioned solutions provide reasonable results only when the number of documents in a specific domain is very huge, in the order of several tens of thousands.

In the literature, there exist a few tools that support the process to pass from text to ontologies and/or SKOS, among them:

- PoolParty [15]: for creating and maintaining multilingual SKOS thesauri. PoolParty, providing text mining and linked data capabilities, helps to expand a thesaurus analyzing documents (e.g., web pages or PDF files) relevant to a given domain in order to clean candidate terms for the thesaurus;
- Text2Onto [16]: it provides a general architecture for discovering conceptual structures and engineering ontologies from free text. The main aim of Text2Onto is support developers in the ontology construction process by applying text mining techniques;
- Some other methods [17] for semi-automatic conversion from well-defined thesaurus like MESH [18] or NCI [19] thesaurus in SKOS format. A novel approach, called “Information Theory Principle for Concept Relationship” has been presented in [20] in order to redefine the SKOS “broader” and “related” relationships between two concepts or topics. For this purpose, content analysis is performed on the basis of probabilistic models, such as Latent Dirichelet Allocation (LDA) and Probabilistic Latent Semantic Analysis (pLSA).

In point of fact, when the domain is narrow and the number of source document is small a manual work is needed to create the seeding knowledge for indexing. In those cases, software products on the market still require an important phase of manual collection of information from the domain, and do not provide satisfactory mechanisms for the coordination of the production process performed by several groups. Thus, the modeling of domain-based SKOS often turns out to be a manual process and, in most case, it is time consuming and hard. It may involve a number of personnel that are not easy to be coordinated to work together to a unique SKOS.

These problems become very critical when a large knowledge modeling, comprised by several smaller domains, is needed. When information sources are multiple and big in size, locating the relevant data, capturing their semantics and providing an overall view of the available information becomes difficult [21]. This occurs, for example, when one has to SKOSify the activities/documents of a local govern, or of a large University, or of a commercial/industrial district, or of a large editor/press such as IEEE, ACM, Springer, etc. The already in place standard classifications for companies (e.g., European and national codes, for instance the Italian ATECO or ISTAT), or for institutions or researchers (e.g., in Italy SSD, and CUN areas) are not suitable to match documents and content with the definitions of their “*terms*” and “*key phrases*”. Most of those classifications have been frequently produced years ago, with the purpose of a final manual fitting, and not with the purpose of using them for automated classification, or for reasoning with semantic tools.

On the other hand, once the seeding knowledge is created, expansion query methods can be used, in order to obtain a generalization of the semantic indexing procedures of the domain in easier ways than in the case of more specialized knowledge datasets. However, since automated tools usually require a huge training corpus, making intensive use of natural language processing algorithms (e.g., Aqualog [22], Watson [4]), the results are not always satisfactory [10], especially when there is the necessity to infer a higher morphological or semantic level in information retrieval systems. Pseudo-relevance feedback (PRF) is a technique used to automatically expand search queries by finding semantically relevant expansion words in top ranked documents [23]. The main problems with PRF query expansion are the poor performances when expansion words are obtained from not relevant indexed documents. Another issue is that many PFR methods use external corpora to collect expansion keywords and key-phrases, for instance Wikipedia and WordNet [24]. In general contexts, where a high level of specialization of knowledge is not requested, the semantic indexing of such large datasets can be performed using several strategies in order to find morphological, syntactic and lexical relationships or patterns [25].

Moreover, in the case of a large knowledge domain based on several smaller domains the manual production of SKOS may leads to stress some knowledge areas/subareas, depending on the knowledge of the experts involved, without taking into account the effective distribution of data to be ingested and the amount of concepts to be indexed into the knowledge base. The users frequently discover them partially out of context (some of the nodes/concepts are rarely used), hence it is hard and often misleading to use them as a knowledge base for *automated classification of documents*. These are common problems of most of the manually produced classifications. As a matter of fact, the task of creating a SKOS requires a deep knowledge of the specific domain, and it implies:

- the precise understanding of the semantic model behind a SKOS, in order to avoid the production of terms which are not related each other by a specialization/generalization and/or relationships;
- the adoption of skilled personnel in both modeling knowledge and application domain or sub-domain;
- the domain analysis and the subsequent collection of terms in an organized form highlighting their relationships;
- a mechanism for coordination of activities in the various stages of the production task;
- the adoption of rules to avoid over-classification (over specialization in the SKOS hierarchy) and under-classification in some areas.

The previous description puts in evidence why the knowledge base production process is a time and resource consuming task and prone to errors, even if the target is the production of a SKOS. A solution could be starting from the data/content to be classified, and directly extracting the SKOS from these sources with an automated or semi-automated process (by taking into account natural language constructs to identify keywords, stop-words, concepts and relationships). In the context of the SKOS, the most important aspects are related to the identification of keywords and concepts.

This paper proposes a solution for assisting expert users in the development and management of a unified knowledge, modeled as a SKOS, by modeling a large knowledge comprised of a number of domains represented by a limited number of documents. The main idea is to realize a solution and tools to strongly accelerate the process of SKOS production, exploiting the real documents/content and web pages to be indexed, and involving the experts in creating relationships among the most recurrent concepts. The solution proposed has been developed in a wider project called Open Space Innovative Mind, OSIM, which has been founded by ECRF, Ente Cassa di Risparmio di Firenze. The OSIM project has as main objective the realization of a portal on which the industries and students can pose questions with the aim of identifying the competences in terms of researchers, groups, structures and/or courses in the large knowledge domain of the University of Florence, ranging from humanities to engineering, from medicine to agriculture.

The solution addressed in this paper allows accelerating the production of SKOS when the domain knowledge is wide but comprised of several smaller domains, where the amount of information to be processed for each domain is small in terms of documents, and cannot be generalized by using external sources without compromising the prevision of the query results. The indexing and query generalization, performed by using external knowledge, may also create spurious results due to the addition of concepts that, de facto, are not present in the sources to be indexed. These activities may decrease the performances and the information retrieval capabilities of the system, in terms of recall and precision, as later discussed in Section 6.

The rest of paper is organized as follow. In Section 2, an overview of the Open Space Innovative Mind system is provided; the context of OSIM domain, an overview of OSIM ontology in the context, and what kinds of semantic relations we are addressing, are discussed in detail. Section 3 reports the requirements for the tool for the assisted generation and management of SKOS. Section 4 shows the software architecture that implemented the solution in the global project framework, putting particular emphasis on multi language RDF-SKOS manager and related mining algorithms and ontological model. In Section 4.4, an overview of the semantic query support and query wizard is reported. In Section 5, the achieved experimental results and an overview of some user experiences. In Section 6, experimental validation of the system performance, in terms of recall and precision is presented and discussed. Finally, conclusions are drawn in section 7.

2. Overview of Open Space Innovative Mind System

As previously stated, the main goal of the OSIM project is to realize a service to industries and students on which they can pose questions with the aim of identifying researchers, groups, structures and courses with the needed competences and knowledge among those of the University of Florence. The University of Florence includes about 50 different departments belonging to all the scientific sectors areas, and hosting more than 2200 permanent researchers (at which one should add phd students, and temporary research contracts), and more than 400 labs with their web pages. Each researcher typically teaches at 2-3 courses; thus, there are about 6000 course programs that may be considered competence descriptors as well. Moreover, the several research departments and researchers participate to research projects, for a total of about other 20.000 descriptors, etc. In such context, it is very hard to identify a manageable number of people that could be reasonably entitled in terms of skills to create a shared common SKOS among all domains. This is due to the fact that the whole knowledge model has to be extracted from a limited amount of information from each domain, ranging from health care to geometry and math, from engineering to agriculture, from mechanics to statistic and pharmaceutics, etc. Furthermore, the sources of this knowledge may change quite dynamically, the courses are updated every 6 months, the CV of people may frequently change, other publications and projects are constantly added or updated, etc. For example, for a department (that may be considered a domain) an average of 50 people and 250 small documents are obtained.

On the basis of the above description, the available information has to be ingested from a rage of different sources. This highly dynamic collection of sources may be automatically gathered through the use of software agents and crawling tasks. In this way, the tool and service are able to update the knowledge-base regularly, with marginal or not human intervention. The information gained can be used by a semantic search engine to answer user queries with a high degree of precision. For example, by using an assisted semantic query interface with natural language query engine such as Aqualog [22], Watson [4]. This scenario is shown in Figure 1.

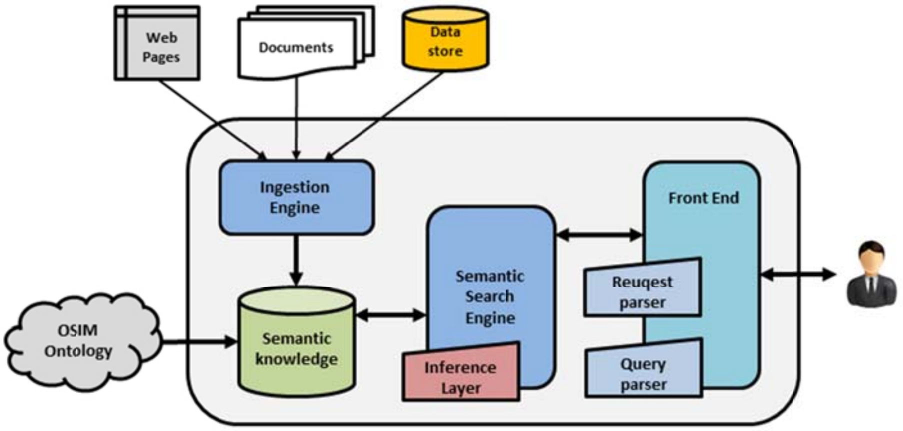


Fig. 1. Ingestion and querying semantic knowledge base

This general model is similar to those used in other mentioned state of the art solutions. It may typically include: a set of data ingestion processing tools with natural language processing support; a semantic database including the domain knowledge based on SKOS, with a certain general structure ontology such as FOAF (Friend of a Friend Ontology [26]); the semantic engine with inference capabilities; a front end engine in which semantic queries are posed and results are provided to the users.

2.1. *Domain Knowledge*

In this section, the structure of the knowledge based managed by OSIM is presented. Then, a very brief overview about the usage of ontology in our context is described, followed by a description of the ontology population process.

2.1.1. *Description of the domain knowledge*

As a first step we had taken into account the fact that, the large amount of data to be processed belong to governmental institutions such as the university and thus also on people. More specifically, the OSIM knowledge is composed by four self-supporting ontologies which are related by semantic relationships, as depicted in Figure 2.

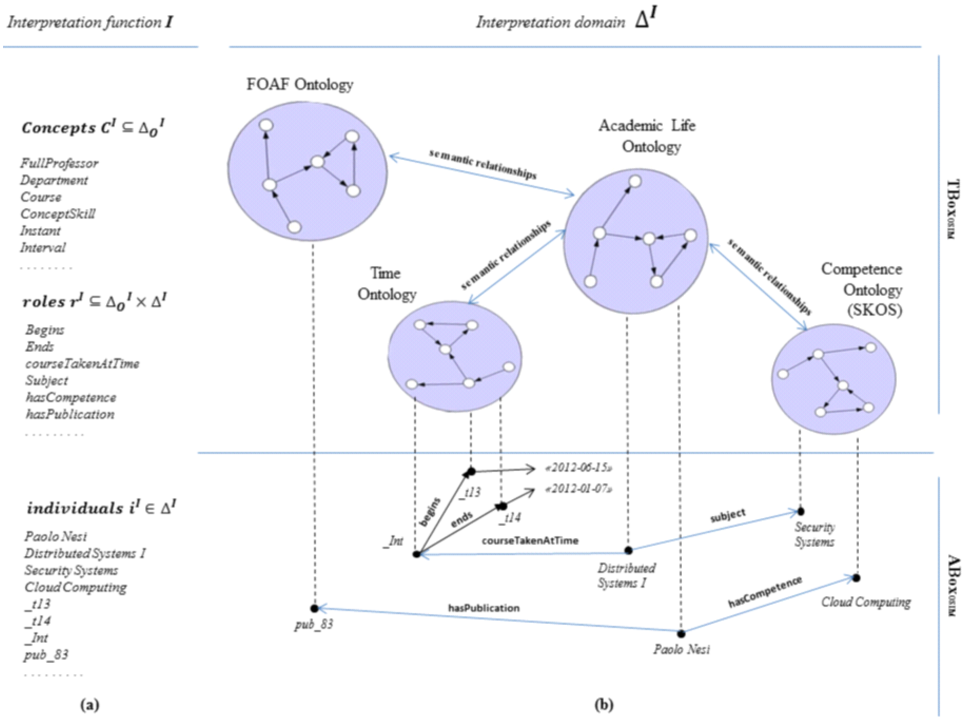


Fig. 2. OSIM knowledge base. (a) Interpretation function (b) Interpretation domain

In the following each of those ontologies is described and explained in the OSIM domain. **Friend of a Friend (FOAF) ontology** is used to model properties about Person and Organization class (in particular for: professor, Ph.D. students, students, researchers, contractors, their relationships, research classification as Disciplinary Scientific Sector, and many other): the name, the surname, the e-mail properties and the *knows* relationship (applicable to individual belonging to the Person class). The FOAF ontology is very useful for describing people, the semantic links between them and the things they create and do. The ontology comprises only the definition and no instances are declared for any defined class; the instances are populated by the ingestion process taking the information from external databases and web pages. Ontology and data are identified by a common namespace.

Academic Life Ontology has been developed specifically for the Italian University case structure and terminology, that defines elements for describing universities and the activities that occur at them (labs, departments, faculties, research centers, groups, projects, courses, curricula, matter, projects, integrated labs, etc.). In the OSIM context, Academy Life Ontology describes the University of Florence structure. The main concepts:

- **Organization** class describes physical structures of university like research centers, departments, faculties, courses and laboratories;
- **People and role** describe instances like full-professors, researchers, PhD students, research contractors, external researchers, related and derived from FOAF concepts;
- **Activity entities** that cover aspects like past projects, ongoing projects, academic publications, affiliations and teaching. To each person the specific publications are added as well, establishing in this way also relationships among the different authors; the specific affiliations and teaching are added, establishing relationships between people and instances of Organization concepts.

Competences SKOS is the SKOS ontology that describes the hierarchy of the technical skills of structures and people belonging to the given application domain: department, centers, etc. This part of the knowledge is the most dynamic and has to be extracted and populated by processing the several instances of textual informal descriptors of: people, projects, publications, documents, departments, centers, and so on.

Time ontology [27, 28] provides a vocabulary for expressing facts about topological relations among instants and intervals, together with information about both durations and date information. Each fact related to a FOAF concept or to an Academy Life concept, is referred by a temporal relationship. For example, we can state that an associated Professor became full Professor at specific time, a researcher has been involved in a project from/to, a Professor has taken a new course at specific academic year. Note that, in this way, we are able to infer scientific careers of persons and their temporal relationships.

2.1.2. *Ontology Representation in the context*

The Web Ontology Language (OWL) is designed for use by applications that need to process the content of information instead of just presenting information to humans.

OWL is a family of three ontology languages: OWL-Lite, OWL-DL, and OWL-Full. The first two languages can be considered syntactic variants of *SHIF(D)* and *SHOIN(D)* description logics (DL), respectively, whereas the third language was designed to provide full compatibility with RDF(S). We focus mainly on the first two variants of OWL because OWL-Full has a nonstandard semantics that makes the language un-decidable and therefore difficult to implement. OWL comes with several syntaxes, all of which are rather verbose. Hence, in this paper we use the standard DL syntax. For a full DL syntax description, please refer to [29]. The main building blocks of DL knowledge bases are concepts (or classes), representing sets of objects, roles (or properties), representing relationships between objects, and individuals representing specific objects. OWL ontologies consist of two parts: intentional and extensional. The former part, consisting of a *TBox*, contains knowledge about concepts (i.e., classes) and complex relations between them (i.e., roles). The latter part, consisting of an *ABox*, contains knowledge about entities (i.e., individuals) and how they relate to the classes and roles from the intentional part. Our knowledge base KB$_{\text{OSIM}}$ is just a TBox_{OSIM} plus an ABox_{OSIM} . Figure 2(a) shows a very small sketch of KB$_{\text{OSIM}}$. An interpretation $I = (\Delta^I, \cdot^I)$ is a tuple where $\Delta^I,$ the domain of discourse, is the union of two disjoint sets ${\Delta_O}^I,$ (the object domain) and ${\Delta_D}^I$ (the data domain), and I is the interpretation function that gives meaning to the entities defined in the OSIM ontology. I maps each OWL class C to a subset $C^I \subseteq {\Delta_O}^I$ each object property P_{obj} to a binary relation ${P^I}_{obj} \subseteq {\Delta_O}^I \times {\Delta_O}^I$, each datatype property ${P^I}_{data}$ to a binary relation ${P^I}_{data} \subseteq {\Delta_D}^I \times {\Delta_D}^I$, and r is the union of two disjoint sets ${P^I}_{obj}$ and ${P^I}_{data}$. The whole definition is in the OWL W3C Recommendation (http://www.w3.org/OWL/). In the following, according to the syntax defined in [31], an example describing a very small TBox_{OSIM} fragment is provided:

(1) $\mathit{Faculty} \sqsubseteq \mathit{Organization}$

(2) $\mathit{Professor} \sqsubseteq \mathit{Researcher}$

(3) $\mathit{TemporalEntity} \sqsubseteq \mathit{Interval} \sqcup \mathit{Instant}$

(4) $\mathit{Professor} \sqsubseteq \exists \mathit{hasSDS.ScientificDisciplinarySector}$

(5) $\mathit{Course} \sqsubseteq \exists \mathit{courseTakenAtTime.Interval}$

(6) $\mathit{Person} \sqsubseteq \exists \mathit{hasCompetence.ConceptSkill}$

(7) $\mathit{Department} \sqsubseteq \exists \mathit{subject.ConceptSkill}$

where: (1), (2), and (3) are simple classes; (3) is a class specified in terms of other ones, in particular is defined as the intersection of the classes *Instant* and *Interval*. (4) and (5) are examples of object properties: (4) binds the class *Professor* (defined in terms of the FOAF ontology) to the class *Scientific Disciplinary Sector*, and (5) over the *Interval* class, meaning when a *Course* has been given. Finally, (6) and (7) define object properties binding the hierarchy of the technical skills to people and structures, respectively.

2.1.3. *Ontology Population Process*

The components related to the **Academic Life Ontology** and to the **FOAF** are initialized and directly populated by gathering information from the central database of the University and of other institutions. Among them, the central CINECA servers [30]. This operation is performed with a set of crawling tasks realized by using SOAP Client implemented in JAVA making use of JAX-WS [31].

On the basis of the described context, the most critical aspect is the modeling and population of the above mentioned **Competence SKOS** for the whole university area.

Individuals populating above ontologies constitute the $ABox_{OSIM}$ which is interlinked with the intentional knowledge. A very small example of $ABox_{OSIM}$ related to the previous $TBox_{OSIM}$, as depicted in Figure 2(b), is reported in the following:

$ABox_{OSIM}$ = {

Paolo Nesi: FullProfessor,
Distributed Systems I: Course,
Security Systems: ConceptSkill,
Cloud Computing: ConceptSkill,
. . . .
_Int:Temporal Entity,
_t13:Instant,
_t14:Instant,
(_t13,2012-01-07),
(_t14,2012-06-15),
(_Int, _t13):begins,
(_Int, _t14):ends,
(Distribuited Systems I, _Int):courseTakenAtTime,
(Paolo Nesi, Cloud Computing):hasCompetence,
(Distributed Systems I, Security Systems):subject
. . . .

}

Concerning to the Competence SKOS, typically the solution proposed is to manually produce a coarse classification. On the other hand, what it is really needed is to arrive at a SKOS populated by instances directly related to the real sources of descriptors to allow the automated classification and reasoning, bringing the final users performing the query to the effective sources of information and thus to people and structures of the Academy Life Ontology. The adoption of a direct Natural Language Processing solution for indexing all elements resulted not viable since the amount of not identified terms is huge in technical documents. This vanishes any attempting to automatically processing the documents, courses, CV and publications with a simple NLP to ontology parser. An example is the direct usage of Stanford Parser [32] and /or treetagger [33] parsers which adopt GATE [34] as processing tool and which also needs an ontology and gazetteers to correctly parse the documents.

For these reasons we started with the idea of producing a solution for assisting expert users in the assisted development and management of a seeding knowledge based on **Competence SKOS**, the **Collaborative SKOS Accelerator and Manager,**

CoSKOSAM. With the aim of accelerating the supervised process of SKOS production and population. In the next section the identified requirements are presented.

3. Requirements of CoSKOSAM

The CoSKOSAM is a SKOS production tool and manager. Its requirements put in evidence that the aim was to create a collaborative environment in which several experts can contribute to the production and the management of the same SKOS with multiple domains. The system may help them in identifying the keywords and concepts which are located in the real documents and sources to be finally automatically classified/indexed, without losing the relationships from concepts to sources, from user and structures.

The main functional requirements for the CoSKOSAM have been outlined to provide the capability of:

- ingesting and analyzing content from different sources (web pages, cv, documents, publications, etc.) to extract keywords and concepts and keep them linked with the original context/source, putting this information into a knowledge base and store it in suitable semantic structures;
- updating the crawling and ingestion of the content and thus the update of the semantic structures related to the identified keywords and concepts;
- helping the area editors with suitable tools that allow them to identify the most relevant keywords and concepts, cleaning the information from stop-words and not useful information;
- managing multi-language content , in order to map the concepts into a multilingual knowledge base, also exploiting translation services and utilities;
- creating and editing a multi-language SKOS about the identified concepts/ competences/skills of personnel and research centers of the academic structure. Typically the professors provides their CV in English and their courses in Italian. Therefore, the translation allows to match the concepts and keywords;
- supporting collaborative crawling, management and editing of the SKOS structures, synchronization of write operations on different logical sections of the knowledge base, ensuring protected access to the system to prevent unauthorized tampering underlying knowledge production;
- allowing the incremental and distributed production of knowledge, giving access to single domain SKOS production and management to separate users;
- supporting the integration of produced SKOS in terms of related terms and synonyms, aggregating multiple instance of them.

The benefits of creating a **Competence SKOS** via a Web-based collaborative environment and management are many. The CoSKOSAM skilled staff, composed by directors of departments, or sector referent people, is enabled to build and manage the knowledge base by incrementally and collaboratively confronting themselves with their colleagues, toward the common goal of realizing a high quality product. Moreover, the administration mechanisms of CoSKOSAM prevent the access by unauthorized personnel and coordinates the work keeping the knowledge base protected.

The benefits of the proposed system are manifold: assisted support at all stages of the production process; support for generating a multilingual SKOS taxonomy; reducing the development time; reducing the amount of personnel involved; ability to work collaboratively and coordinately; safe access to the system and prevention from tampering by unauthorized personnel.

4. CoSKOSAM Architecture

The CoSKOSAM architecture is shown in Figure 3. It has been implemented in JAVA as a web application, using a client-server architecture.

The CoSKOSAM architecture consists of some primary building blocks:

- **SKOS Manager** provides services for creating and maintaining a SKOS of keywords and concepts formalized according to a SKOS vocabulary. It provides a Web based user interface through which users are able to access all the services offered by the underlying layers;

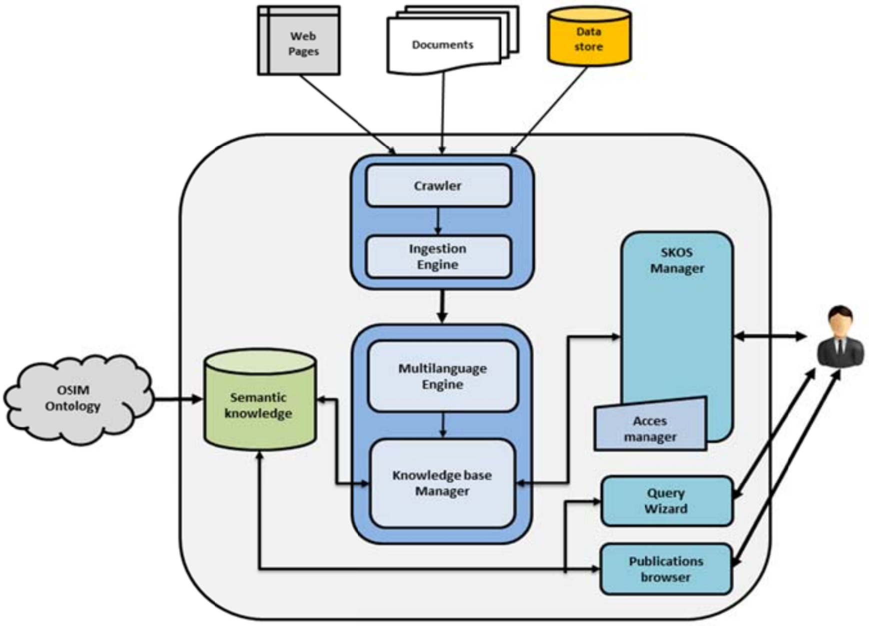


Fig.3 Architecture of Knowledge Base Management System.

- **Access Manager** offers services for the security login and grant check. It helps to synchronize and coordinate the access to the various sub-sections of the SKOS among different users. The privileges pyramid provisioned by system can be grouped in the following three main categories:

  - *Administrators users*: users who have the highest privileges on semantic store. They can edit, read, write, execute backup and crawling tasks;
  - *Writer users*: users who can edit, read and write the working knowledge base but cannot execute backup and crawling activities;
  - *Reader users*: guest users. They can only read the knowledge base but cannot change it in any way;
- **Crawler and Ingestion Engine** offers the ability for crawling and ingesting sources to feed the knowledge base. The crawling task performs a breadth-first search on graph of the web structure of University and ingests some kind of domain-information like personnel, courses, staff curriculum, advertising, research centers, etc. For each skills ingested from the personnel web pages, the crawler keeps track of the occurrences of the related label. Specifically, it makes use of the GATE NLP Platform [34] to implement the crawler architecture;
- **Multilanguage Engine** provides services for the management of multilingual SKOS taxonomy. The information ingested by crawling task is automatically translated by Multilanguage Engine in the languages handled by module. The sources are frequently produced in only one language while the keywords and concepts have to be declined in multiple language (e.g., in Italian and English at least). This is due to the fact that, descriptors about courses are typically in Italian, while CV or researchers, projects and papers are in English;
- **Knowledge Base Manager** provides the software API interface for manipulating the domain ontology. The knowledge base has been built by using the API provided by the Sesame framework [35].
- **Query Wizard** is an assisted search engine, equipped with fuzzy technology, which allows the user to make queries within the whole knowledge base. The user can insert general text queries, or restrict the search to specific fields, such as departments and courses. There is also the possibility to browse the single person for competence-related searches.
- **Publications Browser** is another search engine specific for browsing the publications produced by the whole research personnel of the University of Florence. It allows searches based on people name, publication type (journal article, workshop or conference proceedings, book chapters, etc.), research topic areas and year of publication. Furthermore, the inference built in the knowledge base allows keeping traces of collaborations among researchers; therefore it is possible to make cross-searches among authors and common publications. For each person there is a personal page where there are also collect all the information about type, number and people who assisted him/her in his/her publications and, finally, in which journals they were published. Is also possible view an histogram of the publications for each year.

4.1. *SKOS Manager*

The SKOS Manager provides services for creating, managing, and maintaining a multilingual SKOS model as comprised of concepts compounded from keywords extracted from several kinds of domains and sources by the Ingestion Engine. The approach allows the organization of concepts into concept schemes where it is possible to indicate semantic relationships between terms. The SKOS Manager enables the complete development of a SKOS by providing a range of services, for the incremental, collaborative and multilingual development of the common knowledge via WEB.

Each information about Competence SKOS ingested by previous crawling processes is stored in the knowledge base as an instance of the SKOS class: *Concept*. A user of the SKOS Manager, which is expert in a given sub-domain such as a single department, has the chance to change the knowledge base by adding new concepts and inserting semantic relationships among already existing concepts.

The relations among concepts over the SKOS vocabulary allow adding semantic information to the knowledge base.

The allowed relations are:

- **skos:conceptSchema** relation provides the ability to express the origin of a concept in a concept scheme;
- **skos:hasTopConcept** relation provides the ability to express the mayor topics that are wrapped up into a concept scheme;
- **skos:broader** relation must be used to express the fact that a concept is in some way less general than another. It implies that the concepts involved may reasonably be arranged into a hierarchy, without being too strict about the exact meaning of the hierarchical relationship;
- **skos:narrower** is the inverse relation of skos:broader;
- **skos:related** relation provides the ability to create associative links between concepts. The property carries weak semantics and it expresses the fact that two concepts are in some way related, and that the relationship should not be used to create a hierarchy but for create links between branches of a hierarchy of concepts;
- **skos:prefLabel** is the preferred label associate to concept in a given language. A label is any word, phrase or symbol that can be used to refer to the concept by people. A concept may have only one preferred label for language. When a user adds a label to SKOS the system provides to automatically translate the text-label in the right language by exploiting an external service (in any case the translation may be corrected by the user).

A graphic explanation of generated SKOS vocabulary is shown in Figure 4. The multilingual features provided are available through the use of a translation service. The user has the option of choose the working language and, whenever he/she decides to add a new concept and thus an associated label, the framework translates automatically the value in the language where the label is missing, ensuring a consistent state. Multilingual and multicultural issues are dealt to assure a wider and more effective exploitation of data beside the background of the operator and their location. Furthermore a multilingual approach helps to improve *precision and recall* of popular search engine, which are very good at retrieving the accurate information.

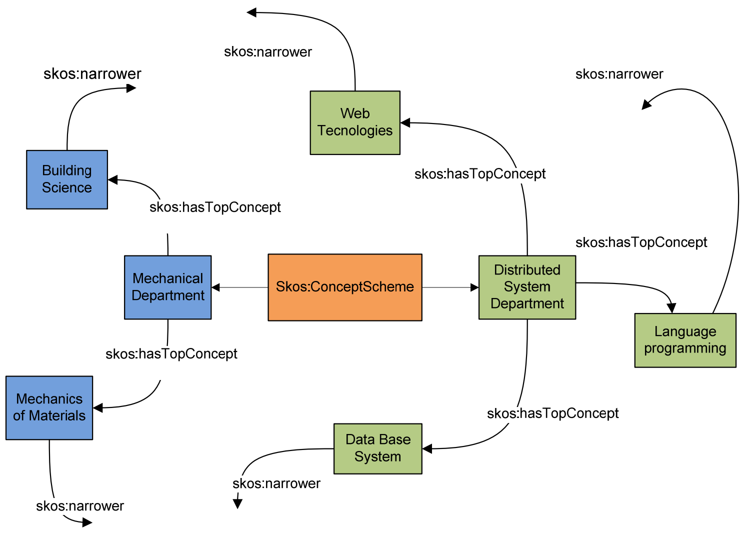


Fig. 4. Generated SKOS Knowledge Base (a fragment)

Other features provided by the system are:

- searching by label in the working taxonomy and discovery branches that contain the target label;
- view the semantic information about personnel or structures related to each *skos:Concept*; that is the link to the original sources;
- view the frequency with which the concepts occur in source web documents;
- filtering the concepts by frequencies. In this way the user has the opportunity to discard some labels that may be considered statistically less remarkable or highly specializing;
- ability to edit concepts in multilingual mode taking advantage of automatic translations. An user can also manually re-translate a label for achieving an higher quality result;
- visual system log for provide additional information about working knowledge base;
- more than one graphical view for displaying the SKOS concepts:
  - *only label*: displays the label in the current language
  - *label with frequencies*: displays next the label, the frequencies of the current concept;
  - *label with language*: displays next to the label, the dual language to the current one. In this way the user has the ability to edit and fix any inaccuracies in the automatic translator.

4.2. *Crawling Publications Metadata from CINECA*

CINECA is an interuniversity consortium which includes 50 Italian Universities and the National Research Institute (CNR). It offers advanced telematics and informatics services supporting research activities and products. Besides, the CINECA database collects the researchers' publications metadata, structured according to the following fields: Authors; Title; Research subject / topic; Publication type; Editorial details (volume and issue numbers, press, number of pages); Year of publication; Publication URL; Abstract.

A crawler, similar to the one used for the keywords extraction in the SKOS manager, is used to extract these information regarding the publications produced by University of Florence research personnel. The metadata extracted by the crawler are collected into a MySQL database and, subsequently, an automated procedure store them in the RDF Semantic Knowledge Base, inferring additional knowledge such us authors collaboration and common publications.

Researchers registered in the CINECA database have a unique CINECA identifier, which makes the knowledge inference process unambiguous. The proposed system, however, is also able to collect relations among not registered user, by processing and matching string authors instead of numerical ids.

The Publication Browser allows the user to make basic searches by authors' name, subject or topic, year of publication. In addition it is possible to make more advanced searches such as publications of a single author within a single subject and, as already mentioned, seek for authors' collaboration and common publications.

4.3. *Multilanguage Competences Extraction with GATE*

The *Competence Extraction* process consists of an automated keywords extraction (carried out by a web crawler) joined with a supervised keywords selection, which represents the preliminary phase for the second and last competence extraction by the crawler. Figure 5 shows how each phase is realized.

These phases are summarized into the following steps.

Basic keyword crawling phase. First of all, the web pages in which people and departments, courses, faculties, etc., information is present (personnel information, biographies, interests, curricula, courses, additional publications, etc.), are collected. This operation is fulfilled by the Annie Gate plugin. Every single page is then split in different sentences stored in different files; the language of each sentence is identified by the GATE language detector, so that all the sentences of the same language can be assembled together within a file. At the end of this process, sentences are collected into distinct English and Italian repositories. This allowed us to treat them in different manners.

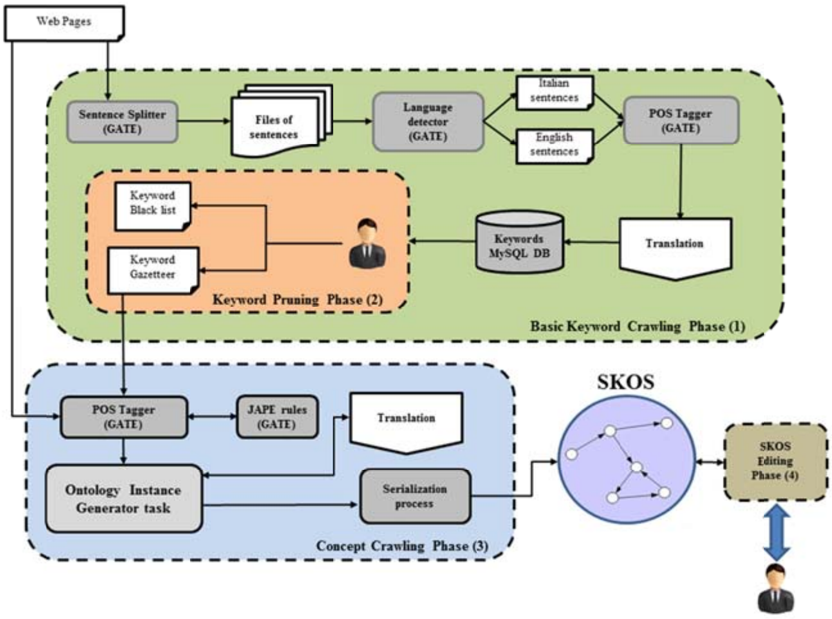
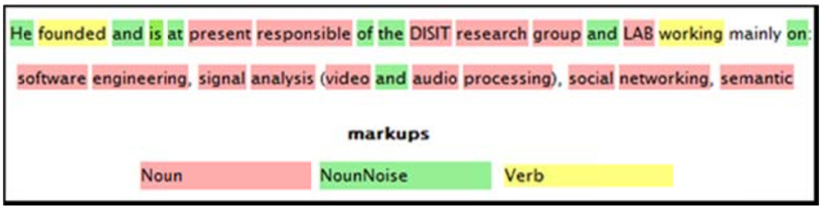


Fig. 5. Competence crawling process.

Specific tree-tagger algorithms (tools for annotating text with part-of-speech and lemma information) are executed on the respective text to extract the words identified with *Noun* tag of the sentences (only single words, in the following called keywords). Figure 6 shows an example of how tree-tagger works. Words tagged as “noun” are keywords candidate to be used in the successive phase.


He founded and is at present responsible of the DISIT research group and LAB working mainly on: software engineering, signal analysis (video and audio processing), social networking, semantic

markups

Noun NounNoise Verb

Fig. 6. Example of pos-tagged sentence, nounNoise are the so called stop words.

Each keyword is stored in a temporary data structure, and its number of occurrences found in the parsed pages, is saved. Then, a translation is performed in order to complete the couples Italian-English and vice versa. Finally, keywords, translations, and related frequencies, are stored in a database.

Keyword Selection phase. In this phase, the user analyzes the extracted keywords, eventually choosing which are important and significant, and discarding the ones which are not relevant, too generic or out of context. To this end, the CoSKOSAM tool supports the expert users with a set of tools and information such as keywords frequency (occurrence), language, status, translations, etc. (see Section 5). The relevant keywords are stored in a special GATE list, called *gazetteer,* which is used in the successive phase for concept crawling. The discarded keywords can be put in a *blacklist* instead.

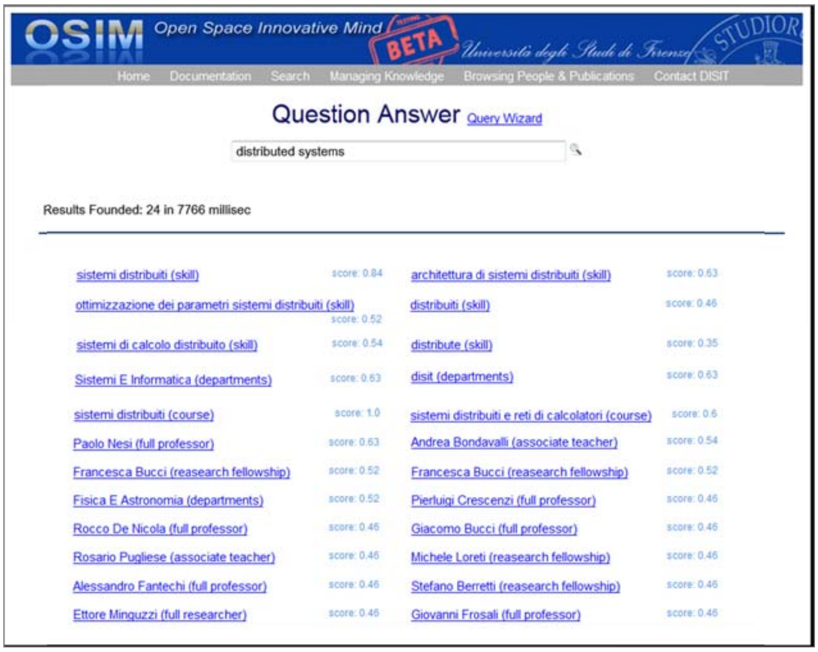
Concept crawling phase. In this step, all the web pages, containing the information described in the first phase, are re-processed taking into account the keywords highlighted in the first phase. For each page, GATE capabilities are used to find the keywords (stored in the GATE gazetteer) in the texts. GATE exploits JAPE (Java Annotation Patterns Engine) rules [36] to identify concepts over pos-tagged texts. JAPE provides finite state transduction over annotations based on regular expressions, and allows recognizing regular expressions in annotations on documents. A JAPE grammar consists of a set of phases, each of which consists of a set of pattern/action rules. The phases run sequentially and constitute a cascade of finite state transducers over annotations. The left-hand-side (LHS) of the rules consists of an annotation pattern description. The right-hand-side (RHS) consists of annotation manipulation statements. Annotations matched on the LHS of a rule may be referred to on the RHS by means of labels that are attached to pattern elements. We defined four rules for handling concepts. For example, the following rule permits to aggregate of keywords in the texts to create more specific concepts:

```
Rule: FindComposedENCompetence
Priority: 20

(
   ({Lookup.majorType == "AteneoCompetence", Lookup.minorType == "keywordAteneo", Lookup.mayorType != "stopword })
      (
         ({Token.category == CC} | {Token.category == DT} | {Token.category == IN} |{Token.category == JJ})*

            ({Lookup.majorType == "AteneoCompetence",Lookup.minorType == "keywordAteneo", Lookup.mayorType != "stopword”})
      )*
) :tmpCompetency
 →
:tmpCompetency.UnifiCompetence= {kind = "competence" rule = "FindComposedENCompetence" lang = "en"}
```

The LHS specifies a pattern to be matched to the annotated GATE document. *CC*, *DT*, *IN*, *JJ* are special categories of stop words (referred by NounNoise markup in Figure 6). CC, DT, IN, JJ are special categories of stop words. CC stands for Coordinating Conjunction, for example ‘and’, ‘but’, ‘nor’, ‘or’, ‘yet’, plus, minus, less. DT are determiners, IN are prepositions or subordinating conjunctions and JJ are adjectives. Referring to Figure 6, the concept “*video and audio processing*”, for example, is handled by the above rule. Then the translations are performed to Italian to English concepts and vice versa. In this phase, the concepts are associated with persons and courses. The task “Ontology Instance Creator” in Figure 5, creates ontology instances of persons, courses and the semantic relationships with their related concepts. Finally, also the links of concepts with the source page, are serialized and then stored and indexed in the RDF store. This phase produces a list of concepts of interest that can be managed by the user. Hereinafter these concepts are called competences and have associated with their corresponding frequency.

SKOS Management phase. The final phase is performed by the user, in order to infer semantic links between the concepts of interest. By this way, the SKOS ontology is created. The inferred ontology can be modeled as a tree, whose nodes are represented by the extracted competences. The competences are classified following criteria of increasing specificity. Therefore, the root of the tree should denote the more general concept, while leaf nodes are the most specific ones. The SKOS Editor allows even to copy a competence node, and to paste it under different semantic paths, following two strategies:

- *Paste as Reference*; this operation is equivalent to merge all the copied competences with the same label, so that all the copies are referred to the same object. In this way, the actions made to the object (rename, deletion etc.) are assigned to all the copies.
- *Paste as Related*; by this operation, the copies are distinct from the original object, maintaining a weaker semantic relation.


OSIM
Open Space Innovative Mind
BETA
Università degli Studi di Firenze
STUDIOR
Home
Documentation
Search
Managing Knowledge
Browsing People & Publications
Contact DISIT
Query Wizard

Question Answer

distributed systems

Results Founded: 24 in 7766 millisec

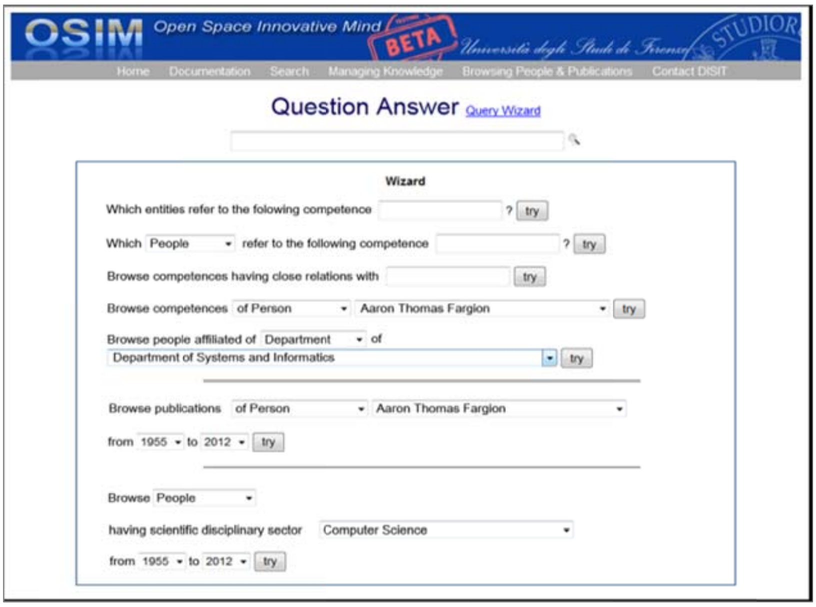
sistemi distribuiti (skill)	score: 0.84	architettura di sistemi distribuiti (skill)	score: 0.63
ottimizzazione dei parametri sistemi distribuiti (skill)	score: 0.52	distribuiti (skill)	score: 0.46
sistemi di calcolo distribuito (skill)	score: 0.54	distribute (skill)	score: 0.35
Sistemi E Informatica (departments)	score: 0.63	disit (departments)	score: 0.63
sistemi distribuiti (course)	score: 1.0	sistemi distribuiti e reti di calcolatori (course)	score: 0.6
Paolo Nesi (full professor)	score: 0.63	Andrea Bondavalli (associate teacher)	score: 0.54
Francesca Bucci (reasearch fellowship)	score: 0.52	Francesca Bucci (reasearch fellowship)	score: 0.52
Fisica E Astronomia (departments)	score: 0.52	Pierluigi Crescenzi (full professor)	score: 0.46
Rocco De Nicola (full professor)	score: 0.46	Giacomo Bucci (full professor)	score: 0.46
Rosario Pugliese (associate teacher)	score: 0.46	Michele Loreti (reasearch fellowship)	score: 0.46
Alessandro Fantechi (full professor)	score: 0.46	Stefano Berretti (reasearch fellowship)	score: 0.46
Ettore Minguzzi (full researcher)	score: 0.46	Giovanni Frosali (full professor)	score: 0.46

Fig. 7. OSIM full text search results.

4.4 OSIM Semantic Full Text Query and Query Wizard

The OSIM Knowledge Base created as described in the previous sections (by Crawler, Ingestion Manager and Knowledge Base Manager) can be queried by using a full text as on top of the interface presented in Figure 7. Each time a user makes a search, the correspondent query is processed by a Query Engine that transform it into a Well-Formed SPARQL query with which is possible to interrogate directly the RDF Knowledge Base to achieve results. The RDF database has been realized by using Sesame [35] database with OWLIM inferential engine [37]. In addition, the Knowledge Base concepts are also indexed by using traditional text based indexing tools Apache Lucene [38]. Therefore, the Query Engine exploits both the semantic indexing and the text similarity index. The results found are shown to the user after being sorted by a score calculated with an appropriate ranking algorithm. The algorithm employs the Lucene Similarity scoring function [39] to calculate the similarity score, which is additionally weighted by the numerical occurrence of the searched record in the indexed Knowledge Base. The details about this part of the architecture are not reported in this paper for the lack of space.

Figure 7 shows an example of full text query for handling all entities related to "distributed systems" competence. The results are labeled as: (skill) that is competences, (departments), (course), (lab), (center), people (as: full professor, associate teacher, full researcher, etc.) which are in relation with the query. Results are sorted by the score. Please note that person, course, department, competence, have a personal result's page that collect all the related information; for example, each person have a personal result's page where is possible to find information about faculty, courses, affiliations to departments, competences and link to the corresponding publications page on the Publication Browser; while, on the result's page relative to a competence is possible to find all its broader, narrower and related concepts, the list of people that have this competence and the list of people that have more specific competence.


OSIM
Open Space Innovative Mind
BETA
Università degli Studi di Firenze
STUDIOR
Home
Documentation
Search
Managing Knowledge
Browsing People & Publications
Contact DISIT
Query Wizard
try

Question Answer

Wizard

Which entities refer to the folowing competence ?

Which People refer to the following competence ?

Browse competences having close relations with

Browse competences of Person Aaron Thomas Fargion

Browse people affiliated of Department of
Department of Systems and Informatics

Browse publications of Person Aaron Thomas Fargion
from 1955 to 2012

Browse People
having scientific disciplinary sector Computer Science
from 1955 to 2012

Fig. 8. OSIM Query Wizard.

The OSIM Query Wizard is an Assisted Search Engine and represents the guided model to navigate the Knowledge Base. With the Query Wizard the user can use the search form to restrict his search, to obtain results with specific pruned information (see Figure 8), for example requesting:

- Entities or people which have experience in a given competence.
- list of competences of a given person.
- departments which have experience in a given competence.
- list of competences of a given department.
- courses related to a given competence.

When generalized queries are posed, such as the request of providing the entities with a given competence, the system provides labeled results ordered by relevance.

5. A view into the CoSKOSAM tool and service

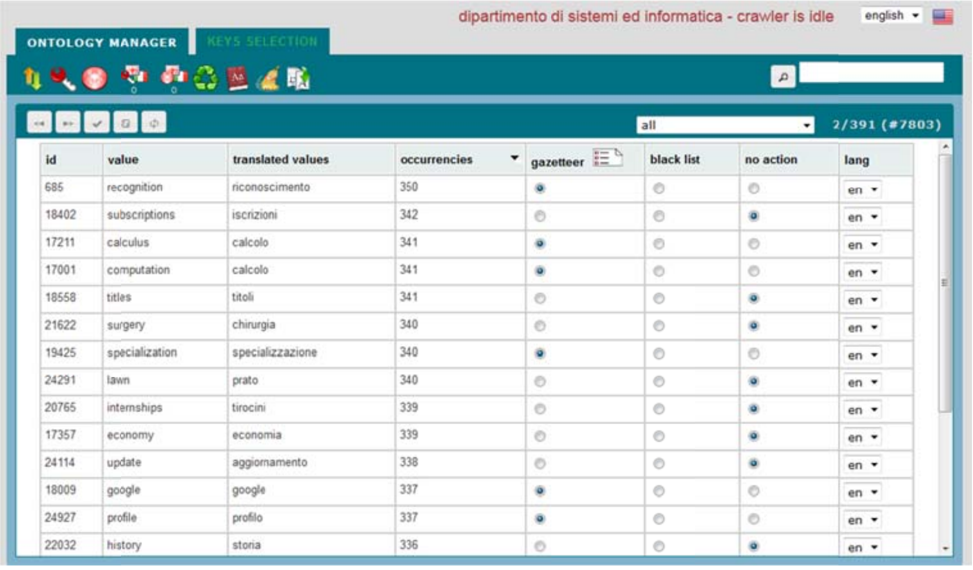
In this section, the achieved results and some user experiences are presented. The proposed tool has been used to develop the full knowledge base for indexing the knowledge of the whole structures of the University of Florence. It consists of 49 departments, about distinct 250000 keywords, coming from about 13000 documents (as CV, courses, etc.), and 2344 people that have courses and CV, while the total amount of researchers is much larger. Moreover, the publications collected from CINECA area are about 80000, with about 30000 authors including professors, PhD students, visiting professor, temporary researchers, contractors, etc.

The knowledge base building is performed keeping separate the domains in the ingestion. However, the concepts are integrated in the indexing process, and then merged together into a unique SKOS by joining concepts and adding *skos:related* relationships.

The typical department has about 60 permanent researchers and other 60-80 temporary researchers as contractor, PhD Students, etc. The permanent researchers (professors and researchers) have about 160 courses, 300 documents. In addition, each department may have distinct sectors. For example for the Department of Systems and Informatics (DSI) we had: sectors/research-areas such as math, operating research, computer science, computer engineering, automated control. The people and activities of those areas have different competences and skills, and they are difficult to be represented by a unique person. The solution proposed allows to work at level of (i) department, analyzing and processing the sources related to the whole department, or to work at level of (ii) scientific sector. In the latter case, the tool has to be feed with the list of people belonging to a given sector.

The aim of the activity for each department has been: (i) the ingestion of the whole sources related to the department; and (ii) the building of a multilingual Competence SKOS about department and its personnel. In the case of DSI it included 62 researchers (28 full professors, 13 associate professors, 21 researchers), and about 160 courses and programs, 335 documents, related publications and web pages, for more than 2000 publications.

To this end, the CoSKOSAM started crawling and ingesting all data, producing the list of basic concepts as presented in Figure 9 (in which the SKOS Manager is shown). The extracted basic concepts/keywords have been browsed by the reference domain expert to identify those that have been erroneously classified to the wrong language (the tool provides support for automatically correcting those situations and learning about them to avoid replicating the same misclassifications in the future).



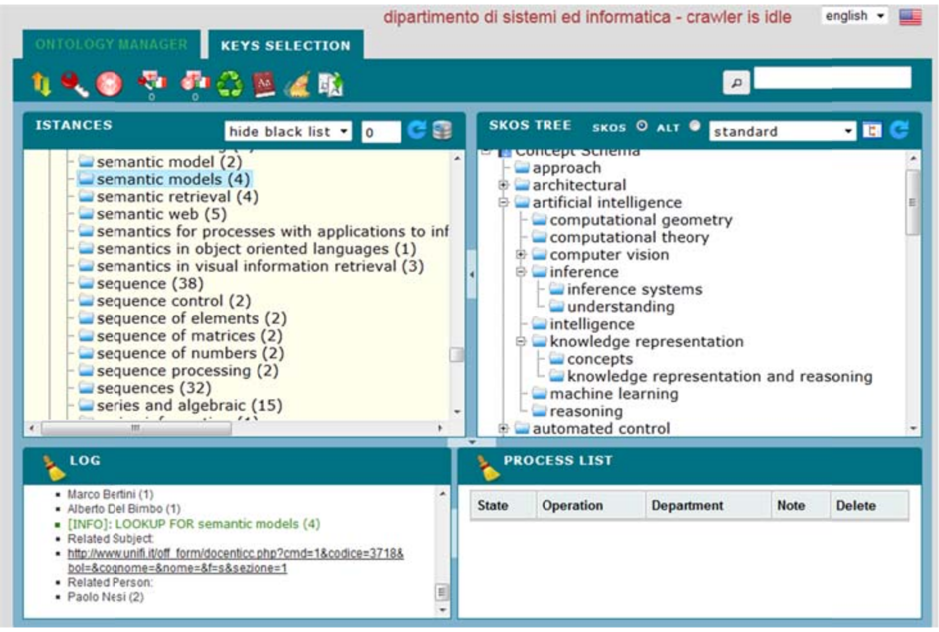
id	value	translated values	occurrencies	gazetteer	black list	no action	lang
685	recognition	riconoscimento	350	●	○	○	en
18402	subscriptions	iscrizioni	342	○	○	●	en
17211	calculus	calcolo	341	●	○	○	en
17001	computation	calcolo	341	●	○	○	en
18558	titles	titoli	341	○	○	●	en
21622	surgery	chirurgia	340	○	○	●	en
19425	specialization	specializzazione	340	●	○	○	en
24291	lawn	prato	340	○	○	●	en
20765	internships	tirocini	339	○	○	●	en
17357	economy	economia	339	○	○	●	en
24114	update	aggiornamento	338	○	○	●	en
18009	google	google	337	●	○	○	en
24927	profile	profilo	337	●	○	○	en
22032	history	storia	336	○	○	●	en

Fig. 9. SKOS Manager Web user interface – setting up crawling task stage for basic concepts

The list of them can be filtered to present only those that have a frequency greater than a given value. This allows the expert to focus its time on the most relevant one. At the same time, the expert can put in the black list some of them, for example if they are too generic. At the end of this process, the selected keywords are put in the gazetteer (see phase 2 in Figure 5). In the case of DSI, the process extracted have been about 1733 distinct terms aligned by the system in both languages, revised by the expert in about 1 hour of work over a total of 7800 distinct keywords, thus removing the general keywords that are not specific of the area.

After this phase, the basic concepts and keywords can be used to create the Competence SKOS, passing to a different interface of the CoSKOSAM, as shown in Figure 10 (the so called phase 3 depicted in Figure 5). On the left side, the list of basic concepts with their frequency (alphabetically ordered), while on the right side, the produced Competence SKOS are reported. The user of the SKOS Manager may look up at the original sources referred by both basic concepts or SKOS concepts; this activity is reported in the log box below. The SKOS tree, on the right side, can be manually created by adding new concepts or by using the drag and drop paradigm by exploiting and arranging the basic concepts on the left side.

The produced Competence SKOS for the DSI consists of different classes and has required 1 day of work for its completion, which is a reasonable time span with respect to the time needed for the manual production of the SKOS domain. Typically, the manual work takes several days of work to produce a SKOS that is not aligned with the effective concepts that one can find in the real document accessible on web. And thus the usage of the manual SKOS for indexing results to be strongly imprecise.



- semantic model (2)
- semantic models (4)
- semantic retrieval (4)
- semantic web (5)
- semantics for processes with applications to inf
- semantics in object oriented languages (1)
- semantics in visual information retrieval (3)
- sequence (38)
- sequence control (2)
- sequence of elements (2)
- sequence of matrices (2)
- sequence of numbers (2)
- sequence processing (2)
- sequences (32)
- series and algebraic (15)

```
Concept Schema
  approach
  architectural
  artificial intelligence
    computational geometry
    computational theory
    computer vision
    inference
      inference systems
      understanding
    intelligence
    knowledge representation
      concepts
      knowledge representation and reasoning
    machine learning
    reasoning
  automated control
```

- Marco Bertini (1)
- Alberto Del Bimbo (1)
- [INFO]: LOOKUP FOR semantic models (4)
- Related Subject
- http://www.unifi.it/off_form/docenticc.php?cmd=1&codice=3718&bol=&cognome=&nome=&f=s&sezione=1
- Related Person:
- Paolo Nesi (2)

State	Operation	Department	Note	Delete

Fig. 10. SKOS Manager Web user interface – setting up Competences SKOS from the basic concepts.

More specifically, the generated knowledge base contains information about:

- domain ontology formalized according to the semantic web technology;
- competence SKOS comprised of many SKOS concepts: SKOS multilingual vocabulary obtained represents the hierarchy of DSI skills:
- the concept related to personnel skills are ingested from web portal of department and, for each person involved is analyzed: personal home page; curriculum vitae; courses, web pages related to teaching activities, The pages contain information about academic courses, educational material; publications;
- information about the number of occurrences for each Competence identified and classified;
- information on structures and people linked by semantic relations;
- information about academic courses taught by teaching staff.

At the end of the Competence SKOS production for DSI, the remaining basic concepts not allocated were 2056 and the produced SKOS was based on 888 (for each language) concepts organized in 27 levels.

The Competence SKOS produced has been validated by experts of the sectors by using the look up facility and, in short, by observing the sources identified and connected to a large sample of broader concepts. Another form of validation has been the production of simple SPARQL queries with and without the usage of the inferential engine. The inferential engine, in this case, exploited both the hierarchy of concepts and the related SKOS relationships, as described in the next section.

Presently, the version accessible from the service link http://openmind.axmedis.org contains only a part of all the available data and in particular: 80000 publications, 30000 authors, reconstructed from the registrations performed by about 4000 people of the University of Florence on the CINECA database of research products. 8 departments have been fully indexed for a total of about 9000 keywords, see details on the informative web page at the mentioned link. This highlights the critical aspects of creating a domain knowledge index when the number of source documents and information is not huge.

6. Assessment and Validation

The typical approach for the assessment of information retrieval, IR, systems is based on the notion of relevant and non-relevant documents. Relevance is defined relatively to an information requisite, and not with respect to a set of terms present in the single query result. For this reason, the assessment of relevance is usually carried on by human domain experts judgments; standardly, it is a binary decision (either relevant of non-relevant) made for each document in the set of documents retrieved by the information retrieval system. A document is considered relevant if it addresses the requested information need (by the query intention), not because it just contains all the keywords in a submitted query [40]. This distinction is not easy to apply in practice, actually this issue can be considered at the basis of the difference between keyword based IR and semantic IR. Considering document relevance as a criterion, Precision and Recall have become the most popular metrics for IR evaluation [41], regularly applied in word based IR. Actually, this approach is used in many assessment frameworks, for instance the widely adopted TREC (Text REtrieval Conference) standard [42], which is a series of experimental evaluation efforts conducted annually since 1991 by the U.S. National Institute of Standards and Technology (NIST). Precision is defined as the fraction of retrieved documents that are considered relevant with respect to the number of *retrieved items*:

$$Precision = \frac{\#(\text{relevant items retrieved})}{\#(\textit{retrieved}\,\text{items})}.$$

Recall is defined as the fraction of relevant documents that are retrieved with respect to the total number of the *relevant items*:

$$Recall = \frac{\#(\text{relevant items retrieved})}{\#(\textit{relevant}\,\text{items})}.$$

The context of the presented validation is the following. The same general conditions of the TREC evaluation framework have been adopted: a group of experts, taken from four departments within the University of Florence (Computer Science, Electronic and Telecommunications, Economics and Business Science), was asked to assess the relevance of the University of Florence web site resources, with respect to a given test set of queries, each one of them formally expressing an information need. The test set was divided into four subsets of queries expressing specific topics within the domain knowledge related to the four departments listed above. Information needs set is made up of key phrases composed by a minimum of one to a maximum of three keywords.

The OSIM full text engine facility have been used over the ontology created by annotating and indexing competences, personnel, courses and structures on the whole text corpus parsed from the University of Florence web site, as illustrated in the previous sections.

The produced results are sorted by relevance/confidence and labeled, as described in Section 4.4. The correct labeling and the confidence have been taken into account to perform the assessment. In such a ranked retrieval context, a retrieved set is given by the top n retrieved documents, where n is a fixed cutoff. For our tests, we used $n = 20$. For each set, precision and recall values has been plotted to provide a precision-recall curve [40]. We used the 11-points interpolated average precision-recall curve, where precision is measured at the 11 recall levels of $k \cdot 0.1$, where k is integer and $0 \leq k \leq 10$, using the software `trec_eval`, specifically designed for TREC evaluations [43]. The results are shown in Figure 11 for the keyword based indexing service adopted by the University of Florence (called Marsilius) and the OSIM solution

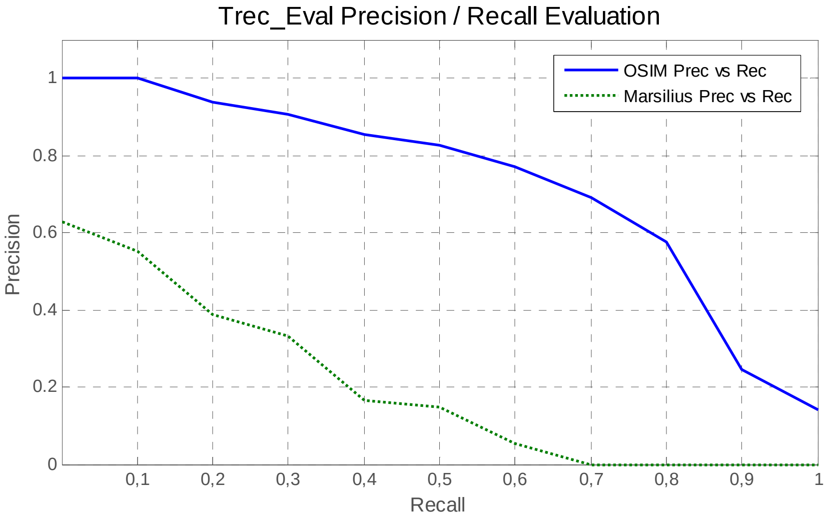


Fig. 11. Averaged 11-points Precision-Recall curve for OSIM Query Wizard and Marsilius search engines.

In order to perform a comparison on the basis of the same processed data, the estimation of the Precision-Recall curve has been made on the results obtained using the

Marsilius search engine. The Marsilius service is based on a Google Search Appliance device, that was configured for accessing to the same data ingested by OSIM. This comparison has not been made with the intent of giving an absolute estimation of the information retrieval performances of the two systems. Actually, they uses completely different search approaches upon the same text corpus. Therefore, the aim has been showing the different capabilities in extracting semantically correct results when the user expect them according to the concept of semantically relevant with respect to the user intentions. In this case, an evident problem exists in comparing a regular text matching based approach against a semantic approach. On the other hand, users are becoming every day more exigent in requesting semantically correct/labeled results, and thus are every day more reluctant in navigating in the indiscriminate results with the aim of understating from them which are the most competent experts, labs, departments, etc.

Another common evaluation metric used in the TREC environment is the Mean Average Precision (MAP), which provides a single-figure measure of quality across recall levels. For a single information need, the Average Precision is the average of the precision value obtained for the set of top k documents existing after each relevant document is retrieved, and this value is then averaged over information needs [40]. The `trec_eval` computes also this metric. MAP calculated for OSIM Query Wizard engine is 71.68%, while the one calculated for Marsilius search engine is 16.93%.

The comparison with other IR systems present in literature is not an easy task for several reasons: ontology learning tools usually depend on the specific domain they are designed for. Moreover, as stated before, the assessment of relevance of retrieved documents strongly relies on subjective judgments of experts or real users involved. We can cite some results found in recent literature that also reinforce these difficulties: Text2Onto obtained an average Precision/Recall of 31.71% / 25.16% for concept identification in the evaluation of the ontologization task of a corpus from SCORM manuals, performed in [44]. In [45], Text2Onto achieved a Precision/Recall of 6% / 35% extracting a concept ontology over a total of 80 web-sites relevant for 10 information goals, while the proposed tool showed a Precision/Recall of 78% / 87% (on the same test domain). Such percentages are to be interpreted with care, and cannot be directly compared with the results presented in this paper. Actually, these reports clearly indicate that the performances of a single IR system / learning ontology tool can sensibly vary, depending on the different text corpora analyzed. On the contrary, the comparison presented from OSIM and Marsilus are grounded on the same data and performed by the same experts.

7. Conclusions

This paper proposed CoSKOSAM, a web based solution for accelerating the production and management of knowledge base of a large entity comprised of several departments with different domain competences. The proposed solution automatizes the production of the hierarchical structure of the competences, as well as the definition of the semantic relationships among them using the SKOS vocabulary, providing a developing method to coordinated independent activities on separate domains that are automatically merged into a unified knowledge base. The methodology greatly reduces the time spent in the development process, aiding the users in all stages of the production process. Furthermore, the ontology is produced according to the OWL/RDF/SKOS rules and can benefit from emerging technologies and innovations offered by the semantic web. The generated ontology can be used as information domain by a demand and supply system about academic skills. It is currently in connection with a semantic database which can be interrogated by performing SPARQL queries allowing:

- semantic search to retrieve ranked information. For computing ranking it is possible to make use of term frequency as a factor weighting within the ranking algorithm;
- semantic indexing for search engine optimization and fuzzy queries;
- exploiting inferential engine to increase the system intelligence;
- improving the engine for providing results to the users and permitting them to navigate in the mesh of relationships among FOAF entities and results.

The solution and proposed tool have been used for the ingestion and analysis of the university knowledge and life, including afferent organization and technical skills. In this context, the solution has been assessed to evaluate the precision and recall with respect to the user expectation. The obtained results are quite interesting with respect to those produced by other solutions. To this end, the results have been compared with respect to a keyword based solution that is working on the same data set and information. The results have been strongly better for the proposed OSIM solution, encouraging us to strongly intensify the research and complete the indexing for the whole structures.

Acknowledgements

The authors would like to thank all contributors to the Open Space Innovative Mind project such as Dr. Stefano Martina and in particular Prof. Mauro Lombardi and Prof. Paolo Rissone.

References

[1] SKOS: Simple Knowledge Organization for the Web. [Online]. Available: http://www.w3.org/2004/02/skos/

[2] RDF – Vocabulary Description Lanaguage 1.0: RDF Schema. [Online]. Available: http://www.w3.org/TR/rdf-schema/

[3] D. Li, T. Finin, A. Joshi, R. Pan, R. S. Cost, Y. Peng, P. Reddivari, V. C Doshi, and J. Sachs (2004). Swoogle: A Search and Metadata Engine for the Semantic Web. *Proc. of the 13th ACM Conf. on Information and Knowledge Management (CIKM),* ACM, pp. 652–659.

[4] M. d'Aquin and E. Motta, Watson, more than a Semantic Web search engine, *Journal of Semantic Web*, Vol. 2(1), pp. 55-63, 2011.

[5] M. d'Aquin, Building SemanticWeb Based Applications with Watson, in *WWW2008 - The 17th Int. World Wide Web Conference - Developers' Track*, 2008.

[6] G. Tummarello, E. Oren, and R. Delbru. Sindice.com: Weaving the open linked data, in *Proc. of the 6th Int. Semantic Web Conference and 2nd Asian Semantic Web Conference*

(ISWC/ASWC2007), Busan, South Korea, volume 4825 of LNCS, pages 547–560, Berlin, Heidelberg, November 2007. Springer Verlag.

[7] ThManager – an Open Source Tool for creating and visualizing RDF SKOS vocabularies

[8] SKOSEd – Thesaurus editor for the Semantic Web. [Online]. Available: http://code.google.com/p/skoseditor/.

[9] Protégé, Protégé enviroment. [Online]. Available: http://protege.stanford.edu.

[10] C. Tao and D. W. Embley. Seed-based generation of personalized bio-ontologies for information extraction, in Proc. of the 2007 conference on Advances in conceptual modeling: foundations and applications, pp. 74-84, 2007.

[11] D. Eynard, M. Matteucci and F. Marfia. A modular framework to learn seed ontologies from text, in *Semi-Automatic Ontology Development: Processes and Resources*, IGI Global, DOI: 10.4018/978-1-4666-0188-8, pp. 22-47, 2012.

[12] C. Lange, P. Ion, A. Dimou, C. Bratsas, W. Sperber, M. Kohlhase, and I. Antoniou. Reimplementing the Mathematical Subject Classification (MSC) as a linked Open dataset structure of the MSC / SKOS concept scheme. *Conferences on Intelligent Computer Mathematics (CICM)*, Bremen, Germany, 2012.

[13] A. Coulet, N.H. Shah, Y. Garten, M. Musen, R. B. Altman. Using text to build semantic networks for pharmacogenomics, *Journal of Biomedical Informatics*, 2010, Vol. 43(6), pp. 1009-1019, 2010.

[14] T. Cohen and D. Widdows. Empirical distributional semantics: methods and biomedical applications. *Journal of Biomedical Informatics*, Vol. 42(2), pp. 390–405, 2009.

[15] Pool Party - SKOS Thesaurus Management. [Online]. Available: http://poolparty.punkt.at/

[16] P. Cimiano and J. Völker. Text2Onto: a framework for ontology learning and data-driven change discovery, in *Proc. of the 10th Int. Conf. on Natural Language Processing and Information Systems, NLDB*, pp. 227-238, 2005.

[17] M. van Assem, V. Malais´e, A. Miles, and G.Schreiber. A method to convert thesauri to SKOS, in *Proc. of the 3rd European SemanticWeb Conference (ESWC 2006)*, p. 95–109, Budva, Montenegro, June 11-14 2006.

[18] L.F. Soualmia, C. Goldbreich, and S.J. Darmoni. Representing the mesh in owl: Towards a semi-automatic migration, in *Proc. of the 1st Int'l Workshop on Formal Biomedical Knowledge Representation (KR-MED 2004)*, p. 81–87, Whistler, Canada, 2004.

[19] J. Goldbeck, G. Fragoso, F. Hartel, J. Hendler, B. Parsia, and J. Oberthaler. The National Cancer Institute's Thesaurus and Ontology, *Journal of Web Semantics*, 1(1), Dec 2003.

[20] W. Wang, P. Barnaghi and A. Bargiela. Learning SKOS Relations for Terminological Ontologies from Text, in *Ontology Learning and Knowledge Discovery Using the Web: Challenges and Recent Advances*. IGI Global, 129 - 152. ISBN 1609606256, 2011.

[21] M. Palmonari. AERIA: Extending SKOS for the practical, yet well-founded, representation and integration of Web schemas in the large, Journal of Emerging Technologies in Web Intelligence, Vol. 3(3), 2011.

[22] V. Lopez, E. Motta, E., V. Uren, and M. Pasin. AquaLog: An ontology-driven Question Answering System for Semantic intranets, *Journal of Web Semantics*, Vol. 5(2), p.72-105, 2007.

[23] M. K. Chinnakotla, K. Raman and P. Bhattacharyya. Multilingual PRF: English lends a helping hand, in *Proc. of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, UniMail, Geneva, Switzerland, July 19-23, pp. 659-666, 2010.

[24] F. Zhao, F. Fang, F. Yan, H. Jin and Q. Zhang. Expanding Approach to Information Retrieval Using Semantic Similarity Analysis Based on Wordnet And Wikipedia, *International Journal of Software Engineering and Knowledge Engineering* (JSEKE), Vol. 22(2), pp. 305-322, 2011.

[25] M. Ruiz-Casado, E. Alfonseca and P. Castells. Automatising the learning of lexical patterns: An application to the enrichment of Wordnet by extracting semantic relationships from Wikipedia, *Data & Knowledge Engineering*, Vol. 61(3), pp. 484-499, 2007.

[26] The Friend of a Friend (FOAF) Project. [Online]. Available: http://www.foaf-project.org/.

[27] J. R. Hobbs and F. Pan. An Ontology of Time for the Semantic Web. *ACM Transactions on Asian Language Processing (TALIP): Special issue on Temporal Information Processing*, Vol. 3(1), pp. 66-85, March 2004.

[28] F. Pan and J. R. Hobbs. Temporal Aggregates in OWL-Time. In *Proc. of the 18th International Florida Artificial Intelligence Research Society Conference (FLAIRS)*, Clearwater Beach, Florida, pp. 560-565, AAAI Press, 2005.

[29] F. Baader, I. Horrocks, and U. Sattler. Description Logics as Ontology Languages for the Semantic Web. In Dieter Hutter and Werner Stephan, editors, Mechanizing Mathematical Reasoning. Essays in Honor of Jrg Siekmann on the Occasion of His 60th Birthday, number 2605 in *Lecture Notes in Artificial Intelligence*, pp. 228-248, Springer, 2005.

[30] CINECA, Centro di supercalcolo, Consorzio di Università. [Online]. Available: http://www.cineca.it/page/chi-siamo

[31] JAX-WS. [Online]. Available: http://jax-ws.java.net/.

[32] The Stanford Parser: A statistical parser. [Online]. Available: http://nlp.stanford.edu/software/lex-parser.shtml

[33] TreeTagger - a language independent part-of-speech tagger. [Online]. Available: http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html.

[34] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A Framework and Graphical Development Enviroment for Robust NLP Tools and Applications, in *Proc. of the 40th Anniversary Meeting of the Association for Computational Linguistics* (ACL '02), Philadelphia (2002).

[35] A. Kampman, F. van Harmelem, and J. Broekstra. Sesame: A generic architecure for storing and querying rdf and rdf schema, in *Proc. of Int. Conference on Semantic Web Conference, ISWC 2002,* October 7-10, Sardinia, Italy (2002).

[36] H. Cunningham and D. Maynard and V. Tablan. JAPE: a Java Annotation Patterns Engine (Second Edition). *Technical report CS--00--10*, University of Sheffield, Department of Computer Science, 2000.

[37] OWLIM - semantic repositories and RDF database management systems. [Online]. Available: http://www.ontotext.com/owlim

[38] The Apache Lucene Project. [Online]. Available: http://lucene.apache.org/core/

[39] Lucene Similarity Scoring Function. [Online]. Available: http://lucene.apache.org/core/old_versioned_docs/versions/3_0_2/api/core/org/apache/lucene/search/Similarity.html

[40] C. D. Manning, P. Raghavan and H. Schütze. Introduction to Information Retrieval, *Cambridge University Press*, ISBN: 0521865719, 2008.

[41] T. Saracevic. Evaluation of evaluation in Information Retrieval, *Proc. of the 18th Int. ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 138-146, New York, 1995.

[42] Text Retrieval Conference. [Online]. Available: http://trec.nist.gov/

[43] Trec_eval. TREC IR systems evaluation software. [Online]. Available: http://trec.nist.gov/trec_eval/

[44] A. Zouaq, D. Gasevic and M. Hatala. Towards open ontology learning and filtering, *Journal on Information Systems*, Vol. 36(7), pp. 1064-1081, 2011.

[45] H. Xiao, B. Upadhyaya, F. Khomh, Y. Zou, J. Ng and A. Lau. An Automatic Approach for extracting process knowledge from the Web, in *Proc. of the IEEE Int. Conference on Web Services, ICWS,* pp. 315-322, Washington DC, USA, 2011.