Abstract

The scholarly activities query, retrieval, maintenance and management of institution repository data is often difficult, because repositories are loaded with documents in a myriad of formats generated by an Institution. The current data retrieval techniques that are used in institutional repositories make use of indexing and keywords in order to retrieve information which results in reduced accuracy in matching user requests.

This paper proposes a system called SEMKNOW\(^1\), which will enable machine involvement to enhance the efficiency of information seeking process through flexible query option and well conceptually organized structure of documents inside the repository for researchers.

Adding semantic technologies to the educational repository of the institution will add support for the functions carried out by the various roles involved in knowledge development and dissemination in academic institutions.

Keywords: Ontology, Semantic web (web 3.0), Institution repository, Knowledge management system.

Introduction

One of the main objectives of universities is the dissemination and creation of knowledge. It is important for an academic institution to perpetuate and evolve knowledge. To this end they make use of digital repositories in order to easily access, maintain and distribute materials that are developed in the course of conducting research. The nature of distribution of these materials is more user-centric and more focus is placed on creating relevant folksonomies in order to enable easier retrieval and access for the user, however, the retrieval process would be more effective if the tags were machine understandable and so the system would be actively involved in aiding the user to find relevant documents.

The open source SEMKNOW project aims to provide intelligent system that agreements within-built interfaces for the representation, integration, management and querying of knowledge through semantic web. In this paper, I describe an initial web-based application that adopts an ontology centric model to perform semantic query answering over Publications (Conference papers, Theses, E-Journals, E-Books, Tech Reports, Manuals, Unpublished articles, Research and Development Projects). Ontologies in this application are described using OWL\(^2\).

This paper is organized as follows. The next section presents the problem statement and motivation for my work. It is followed by a section describing the SEMKNOW architecture and implementation. Next I describe the SEMKNOW systems domain ontology, application ontology and how they both interoperate with each other. This is followed by a section that describes how this are represented in RDF\(^3\). Then I go on to describe the prototype built and finally I present the conclusions of my work.

Motivation

In the knowledge-intensive view of Education, it requires knowledge-bases and knowledge systems built to effectively store and retrieve the generated

\(^1\) SEMKNOW – Semantic web Knowledge Management System.

\(^2\) OWL – Web Ontology Language

\(^3\) RDF - Resource Description Framework
knowledge in the right context to advance learning. Over recent years Universities and higher education institutions are pioneering knowledge management into their organization and there is quite tremendous growth in publications by Indian researchers, contributing to population of web with research literature.

A free platform for knowledge sharing not only among academia, but also for students and other researchers around the globe is seen as the result of digital libraries, Institutional repositories emergence. Nevertheless, the complex collaboration processes that take place in the teaching-learning process still poses major challenges to research mainly in Knowledge Management itself and Artificial Intelligence as a supporting field due to the informality that in most cases surrounds learning.

Today technologies, initiatives and strategies such as the OWL, RDF, and DCMI’sub sist or are being developed which allow identifying and describing knowledge and information resources. It is therefore important that knowledge management technologies and strategies be researched, developed and applied in education to realise the future workforce for the new economic model and simultaneously enrich their learning environment.

It is for that reason so essential to develop ontologies and tools that allow managing different types of documents that build research literature and knowledge needs to be stored and retrieved in right context. This stated necessity generates a justifiable motivation for the development of tools to assist students and professors in storing and retrieving information documents that result from their teaching-learning process and thus generate knowledge-centric collaboration beyond the confines of classroom.

System Architecture

SEMKNOW was implemented using the Java programming language and interoperates with other open source and Semantic Web technologies, including the Sesame\(^5\) Java API, Pellet\(^6\)and Protégé\(^7\). The system architecture shown in Fig.1 is composed of several components including an application aggregator, a web user interface, an ontology repository, an ontology index, a reasoner that exhibit inferred relations, properties, classes and a SesRQL\(^8\) query engine for querying a triple store.

![SEMKNOW Architecture](image)

**SEMKNOW Aggregator:** The SEMKNOW aggregator is the nucleus element of the system which integrates the decoupled relations among systems modules, operates the system behavior, and manages ontologies used by the application. The aggregator loads requested ontology as per the context and the retrieved ontology. The aggregator enables communication between components. The manipulation of OWL documents and data structures by multiple SEMKNOW components is facilitated by the Sesame Java API.

**Ontology Repository:** Storage and retrieval of URI identified ontologies is supported by ontology repository. This implementation in SEMKNOW system uses a sophisticated MySQL database storage.

**Ontology Indexer:** The ontology indexer though behaves like lookup table; it enables fast and easy access to entities in ontology. Indexed entities include class, property, their types and individual names. Thus, this component supports partial entity name matches, and retrieval of specific types of entities. The current implementation leverages the speed and power of the Lucene Text Indexer; therefore, it reduces the need for memory data structures for caching entity metadata.

**SeRQL Query Engine:** The SeRQL Query Engine interface provides the ability to query a triple store repository using SeRQL and to obtain result set through its API. Implementations for sesame serve as wrappers for the vendor-specific APIs, hiding them from the components that use the services.

**Graphical User Interface:** The SEMKNOW systems graphical user interface is built on Java Server Pages (JSP). The user interface has been tested and optimally with IE9, Mozilla Firefox, Google chrome, Safari 5.

**Semantic Reasoner:** A semantic reasoner, reasoning engine, rules engine, or simply a reasoner, is a piece of software able to infer logical consequences from a

---

\(^4\) DCMI –Dublin Core Metadata Initiative


\(^6\) Pellet - [http://clarkparsia.com/pellet/](http://clarkparsia.com/pellet/)

\(^7\) Protégé - [http://protege.stanford.edu/](http://protege.stanford.edu/)

\(^8\) SeRQL –Sesame RDF Query Language
set of asserted axioms. Reasoner use first-order predicate logic to perform reasoning. The semantic reasoner implemented in system is Pellet.

**Implementation**

SEMKNOW, a Semantic web based system is developed with several new practices and technologies as of today. Particular challenges and issues are focused and solved in this system when compared to traditional semantic systems or existing knowledge based semantic systems.

The systems ontology is developed in particular using W3C approved standards through DC (http://purl.org/dc/elements/1.1) and BIBO (http://purl.org/ontology/bibo/) entries. And Imports of these namespaces makes SEMKNOW ontology standardized.

And the implementation of SEMKNOW system is carried out considering the Indian universities and educational institutions running the digital library repositories on relational databases. The shift from relational databases to semantic knowledge bases with triple storage is made easy R20 mapping of tables and their columns with entities of ontology. Alignment API is implemented in the system for this purpose by maintaining threshold value of 0.8 for matching from relational database to ontology. Fig. 2 shows the R20 mapping between SEMKNOW ontology and relational database.

**Fig. 2: Mapping between Relational Database and Ontology.**

In figure 2, Left hand-side is the tables and its columns from relational database. And right hand-side is the ontology with its classes and properties

The challenge in this alignment is API starts troubling as number of records increases. This is dealt with R20 mapping that is efficient even for millions of records.

**Fig. 3: R20 mapping.**

```xml
<xml version="1.0" encoding="UTF-8">
<r20>
  <ns:desc name="sendocs">
    <ns:table name="journal">
      <ns:desc name="idJournal"/>
      <ns:desc name="Year"/>
      <ns:desc name="Category"/>
      ...
      <ns:desc name="Keywords"/>
      <ns:desc name="Abstract"/>
    </ns:table>
  </ns:desc>

  <ns:desc name="sendocs">
    <ns:table name="journal">
      <ns:desc name="idJournal"/>
      <ns:desc name="Year"/>
      <ns:desc name="Category"/>
      ...
      <ns:desc name="Keywords"/>
      <ns:desc name="Abstract"/>
    </ns:table>
  </ns:desc>
</r20>
```

The system though being developed through JSP\(^9\) technology for front-end, the standards are validated in security and maintenance through EJB\(^10\) technology.

**SEMKNOW Ontology**

In process of learning by students and researchers various sources of knowledge are being generated in form of publications, articles. Today all this sources of information is available in the internet or in data bases in digital format, which enables their search and use. But search returns more efficient results only if it is machine understandable. This is achieved through ontologies, and so this system is built with specifically designed SEMKNOW application ontology to store and retrieve all the literature documents from various heterogeneous databases.

\(^9\) JSP –Java Server Pages Technology

\(^10\) EJB –Enterprise JavaBeans Technology
In this figure 4, the hierarchy is designed through \(<\text{rdfs:subClassOf}>\) and properties of instances (ISBN, Author...) for each Class (Book, Journal Article...) are defined through \(<\text{rdf:datatype}>\) in RDF.

It’s obvious that there is a tight-bond relation between document and its topic. To facilitate complete effective search the different types of documents organized in application ontology are related with a suitable domain- or subject-based ontology. So for this purpose SEMKNOW domain ontology has been designed using the ACM digital library hierarchy.

In the figure 6, the left hand side represents the application ontology and right side is the domain ontology. The instance B1 of class Book has other properties like hasAuthor, hasISBN, hasPublisher..., besides shown isOfSubject property.

The two ontology’s are related with \(<\text{owl:ObjectProperty}>\ rdf:about="&;hasDocs"> in domain ontology and \(<\text{owl:ObjectProperty}>\ rdf:about="&;isOfSubject"> in application ontology. These relations are established through Ontology mapping enabling the navigation from TypeOfDocument to SubjectOfDocument and vice-versa.

**Prototype**

The SEMKNOW systems prototype developed in Java shows that users perceived the system as useful in assisting research work and the retrieval of information is of adequate quality. The system is built with basic services justifying the objective of the system like browsing and semantic querying. Besides few additional modules are included namely -

- Administrator module: Metadata of new documents can be added and updated to Institutional repository. The system provides user interface for admin to populate triple store without having prior knowledge about adding RDF statements to .owl files every time a new record to be updated, unlike traditional semantic web based systems. And handles user activities.
This shows the representation of SEMKNOW triple store using Sesame MySQL RDF store. Adding new records in deposit module of system dynamically updates repository through Sesame Java API.

- Deposit Module: A new document can be added to repository by user but updated to repository only on admins approval.
- Discussion Forum: Knowledge Management system is justified on enabling knowledge collaboration and sharing, that is achieved through this module.

Conclusions and Future Work

In this paper, I described the system architecture and general functionality of SEMKNOW, an easy to use semantic web-based application for the scholars, researchers, and students to perform scholarly activities. Current work includes semantic filters enabling content browsing, querying, and knowledgebase development through deposit module in system, system internationalisation with multi-language support depending on geographical location and interactive Knowledge collaboration through discussion forum and blogs. The ontology engineering greatly helped in enabling the identification of the more significant concepts and relationships used more often in a knowledge domain, thus becoming a prevailing knowledge representation metadata for knowledge repositories. The main contribution of this system has been in developing a Human-Machine System that supports flexible interactions between their components and finally offering the expected information to the end users of SEMKNOW system.

The following stage in the development of this system will be in shifting RDF triple store to Jena TDB that supports uploading, querying millions of triples in petite time compared to current existing triple stores. And also to implement a hybrid existing triple stores.
inferences into the system to overcome expense of time and space when dealing with large ontologies.

References