Webology, Volume 10, Number 2, December, 2013

Home	Table of Contents	Titles & Subject Index	Authors Index

Intelligent interoperable application for employment exchange system using ontology

Kavidha Ayechetty
Senior Assistant Professor, Department of Computer Science, Institute of Road and Transport Technology, Erode, Tamilnadu, India. E-mail: kavitha.irtt (at) gmail.com

Saradha Arumugam
Ph.D., Associate Professor, Department of Computer Science, Institute of Road and Transport Technology, Erode, Tamilnadu, India. E-mail: saradha.irtt (at) gmail.com

Received July 2, 2013; Accepted October 25, 2013

Abstract

Semantic web technologies have the potential to simplify heterogeneous data integration using explicit semantics. The paper proposes a framework for building intelligent interoperable application for employment exchange system by collaborating among distributed heterogeneous data models using semantic web technologies. The objective of the application development using semantic technologies is to provide a better inference for the query against dynamic collection of information in collaborating data models. The employment exchange system provides interface for the users to register their details thereby managing the knowledge base dynamically. Semantic server transforms the queries from the employer and jobseeker semantically for possible integration of the two heterogeneous data models to drive intelligent inference. The semantic agent reconcile the syntax and semantic conflicts exists among the contributing ontologies in different granularity levels and performs automatic integration of two source ontologies and gives better response to the user. The benefits of building interoperable application using semantic web are data sharing, reusing the knowledge, best query response, independent maintenance of the model, extending the application for extra features.

Keywords

Interoperable application; distributed application integration; ontology integration; semantic integration; collaborative application; ontology mapping

Introduction

The Semantic web provides data integration capabilities considering the semantics of terminologies in the data models. Semantic interoperability addresses the heterogeneity gap between ontologies. Ontologies are conceptual backbone of semantic web provides efficiency in data integration. Ontologies enable integration of knowledge that can be reused by several applications across governance or business (Gardner, 2005). Ontology is an engineering artifact provides formal specification of a shared conceptualization. It provides a common vocabulary of a domain in which the meaning of terms and relations between them are defined with different levels of formality. When different ontologies contain facts about the same resources, we can find new and interesting relationships between the resources in those ontologies. In the proposed work, ontology integration uses the following facts to bring out the benefits.

- Identification of common concepts and resources shared between ontologies.
- Expression of mappings between ontology concepts and attributes.
- Accessing distributed contents of ontologies.
- Mapping of individuals in different applications
- Users are allowed to query the integrated application.
- Ontologies are cooperative to each other when the semantic conflicts are resolved using mapping.

The domain of employment exchange is the candidate for building interoperable, co-operative, collaborative application using ontology approach. It involves two major data models namely employer and jobseeker. They collaborate to produce meaningful application. Employer Ontology and Jobseeker Ontology use different abstractions for the same vocabulary. Application is built using these two different data models by integrating the relations among them. The interoperability between the ontologies is achieved by the careful selection of behaviors and vocabularies. The employer and jobseeker behaviors allow them to function together by mapping the concepts and attributes of two ontologies. Concept in one ontology can be used as relation in another ontology. Sometimes semantically equivalent concepts in ontology mean the same but they use different terminologies. When such applications are integrated, right interpretations for the abstractions are assigned by the semantic agent. Applications are integrated using semantic web to drive intelligent inference. Semantic web based employment exchange system involves two main entities namely employers and jobseekers which share same conceptual domain and behave as producers and consumers. It applies semantic integration for deriving the best query-response. The data submitted by a jobseeker is considered by any number of companies and the job posted by a company is used by pool of jobseekers. The application gives better response for the query involving vocabularies of different concepts. Ontologies ensure an efficient retrieval of Web resources by enabling inferences based on domain knowledge (Corby et al., 2006).

Motivation behind building employment exchange system using ontology approach

The proposed ontology based employment exchange system has the following advantages:

- To replace the existing applications based on database technologies by ontology based approach to gain advantages of semantic web technology
- The application by ontology approach is intuitive and easy to build distributed applications.
- The application designed by semantic technology is reusable and extendable
- Effective use of the jobsites by the users and provide a wide collaboration among jobseekers and employers
- Employers refine their searches for candidates who exactly matches all their requirements
- To provide a job pool in which every qualified individual can pick his/her suitable job
- The knowledge base content is accurate and free of redundancy
- (Klyne , 2004) said Open access to information provides way to build new applications.
- Dynamic adoption of requirements.

Methodology

Semantic Web is an open-ended framework which combines and exploits information from a wide range of sources. The semantic web technologies enable proper integration of knowledge in ontology based applications. Shared understanding is necessary to overcome differences in terminology (Antoniou & Van Harmelen, 2004). For example, the term ‘course’ may be used with two different usages. It may refer ‘degree’ or ‘subject’. The differences in terminology between ontologies are solved by mapping logic. A semantic agent in the proposed system is using the logic to search for relation “owl:equivalentProperty” between terms. The terms related with ‘equivalentProperty’ are linked. Linking of terms derive semantic interoperability among cooperating ontologies. Also, mapping logic uses some of the object properties between terms to link them. Thus, semantic heterogeneity is solved by the use of ontologies automatically.

Ontology is an explicit specification of concept. We use ontologies to relate concepts from different information sources. In the proposed semantic agent based Employment Exchange system (EES), Jobseeker and Employer are the main entities in the domain of employment exchange system and they are the two source ontologies. We represent main information entities using different ontologies and the architecture simplifies the collaboration task by supporting addition and removal of sources. Collaboration between ontologies is difficult if they have minimal common vocabulary and it is overcome by inter-ontology mapping. The inter-ontology mapping identifies semantically related terms of different source ontologies. It considers different views on a domain such as different granularity of ontology concepts. Semantically equivalent terms appear as concept, attribute in two different ontologies. EES is an interoperating system defines concepts on the basis of common meaning with different terminologies. These concepts result in semantic conflicts whenever two contexts do not use same interpretation of information and requires mapping. EES’s semantic server includes two important processes such as dynamic data acquisition and query-response. The dynamic data acquisition system and query-response system is shown in Figure 1.a and Figure 1.b respectively. User on both models here is considered either as employer or jobseeker.

Figure 1 a. Data Acquisition system

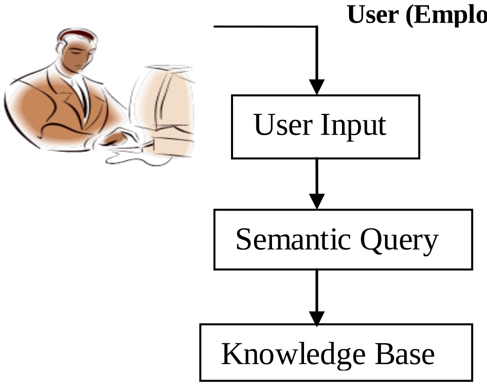
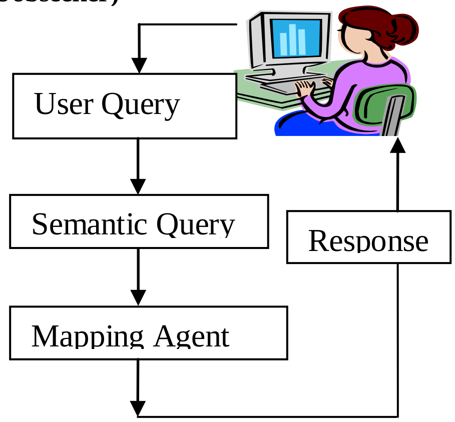


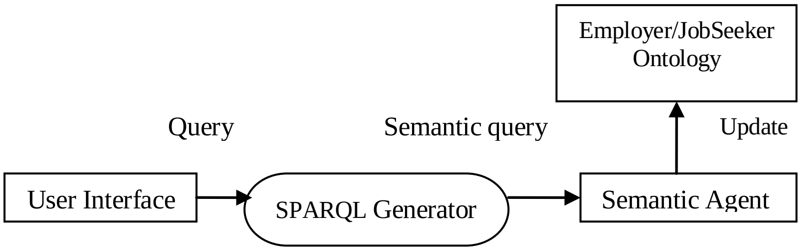
Figure 1 b. Query-Response system



Dynamic Data Acquisition System

The ontology based online employment system allows users (employers, jobseekers) to register their details. Dynamic data acquisition system receives user information during initial registration process based on user's category. The registration process creates two different pools of employers, jobseekers separately. The system is capable of updating ontologies when new users register in EES. Figure 2 shows process flow of dynamic data acquisition system for both employer and jobseeker. Two different sets of individuals are maintained along with their knowledge base in different locations.

Figure 2. Individual creation from the UI of Employment Exchange System (EES)



Employer registration includes acquisition of company and job posting details. Employer ontology tries to conceptualize the details of the organization details, job posting detail, details of branch posting the job. The jobseeker registration includes collection of details of the job seeker such as educational qualification, experience and job preference. The preference detail of the jobseeker covers (i) company they wish to apply for job (ii) location at which they may be willing to get job (iii) the post they are applying (iv) if there is any salary expectation and (v) if the person willing for part/full time. Based on the registration details, the agent creates new individuals in the appropriate data models of both employer and jobseeker.

Query-Response Model

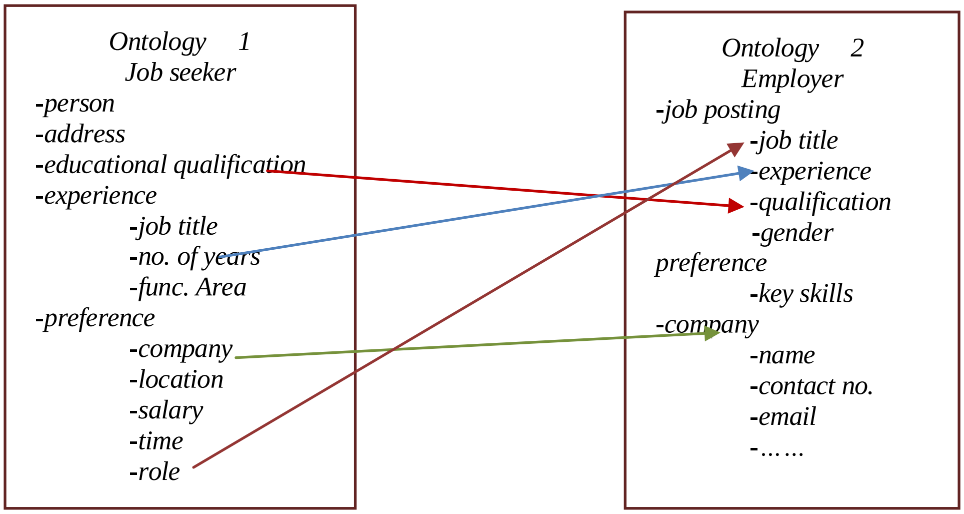
Like producer and consumer process, employee and jobseeker ontologies function together in building the EES application though they are designed and maintained separately. GUI based query formulation makes it easier and understandable for the user to post the query. The response includes the contact details and other prime particulars of the user. The query facility allows the users to query based on several search criteria. The search criteria maintained for users (employer, jobseeker) are the concepts such as company, location, role, salary and time. For example, to know whether any jobseeker has applied for particular company, the search criteria 'company' is used and by entering the name of the company, the employer gets details about jobseeker information like name and other contact information as the response. Jobseeker use the same search criteria 'company' to query whether any company has posted for job of their preference. Agent uses the employer data model to map with the jobseeker preference and mapped employer details are sent as response to the jobseeker.

Mapping Model

The system is able to access the distributed content of several ontologies by taking the advantage of mapping while they are integrated together for a query. In the case of single ontology, all the concepts needed across application are available in common ontology. In multiple ontology approach, each application operates with their ontology and the other ontologies are used when necessary by resolving semantic difference exists among them. Integrating ontologies involves building new ontologies by assembling, extending, specializing or adapting other existing ontologies (Orgun et al., 2008). In order to overcome the limitations in the existing system, we employ semantic web technology for interoperation between the data models of the system. When different ontologies contain facts about the same resources, we can find new and interesting relationships between other resources in those ontologies. Many of the existing information integration systems use more than one ontology to describe the information.

The mapping system available in the agent uses the mapping logic for the possible match between the preference of jobseekers and the job posting details of employer for information retrieval from both sides. The Job posting concept of the employer ontology represents Employer requirements. The preference concept of the jobseeker ontology represents Jobseeker requirements. The agent formulates the semantic query equivalent to the query submitted by the user using Graphical user interface. If the query posted by employers, then retrieval of information takes place from jobseeker data model. At the same time**,** when the jobseeker preference is matching with the employer's job posting concept in the employer ontology, the agent sends the details of employer posting the job to job seeker automatically. From the employer and job seeker ontology we bring about the interoperation by performing the initial step of identifying the common resources. The concept "job posting" in employer ontology and the concept "preference" in job seeker ontology are mapped for effective knowledge retrieval as shown in Figure 3. Various kinds of mapping carried out between two data models include concept to concept mapping, attribute to attribute mapping, concept to attribute mapping and attribute to concept mapping.

Figure 3. Ontology Mapping



Attribute to attribute mapping

An attribute of a concept in ontology is mapped with attribute of a concept in another ontology. When the attributes are semantically equal, they are integrated with the relation 'owl:equivalentProperty' indicating they refer to same kind of information. The attribute "role" in Job Seeker ontology is equivalent to "job title" attribute in Employer ontology and the attribute 'company' in jobseeker ontology is equivalent to 'name' in employer ontology. This kind of mapping takes place when the attributes with different terminology appear under different concepts of their ontologies and sharing same semantic value. The application considers the terminological differences and exploits the semantic similarity for integration.

Concept to attribute mapping

In Concept to attribute mapping, concept in one ontology semantically overlap with attribute of concept in another. The proposed Employment exchange system uses the concept 'educational qualification' in jobseeker ontology and attribute 'qualification' of concept 'Job Posting' in employer ontology to refer qualification of a person. Mapping concept of jobseeker ontology to attribute of employer ontology takes place to satisfy information need for the query involving information about 'qualification'. Mapping logic checks for 'equivalent' relation between concept and attribute of the source ontologies. It proceeds further to check for equality between value of attribute (undergraduate, postgraduate) of 'educational qualification' concept and value of attribute ‘qualification’ in employer ontology. If values are find equal, then the server retrieves information based on the query.

Concept to concept mapping

If concepts between different input sources contain same information, a concept to concept mapping takes place. Concepts ‘Search Criteria’ and ‘Address’ in both the ontologies are having same structural equivalence and uses same term. So they are related to one-another with ‘equivalent’ relation. The concept to concept mapping is possible when they have same structural relationship. Semantic mapping between concepts takes place when names of the concepts differ besides being structural equivalent.

Attribute to concept mapping

Mapping logic of the semantic server checks for ‘equivalent’ relation between the attribute ‘experience’ in employer ontology and any attributes of concepts in jobseeker ontology. The attribute ‘no.of years’ of the concept ‘Experience’ in jobseeker ontology is having equivalent relation with the attribute ‘experience’ in employer ontology as in Figure 3.Then the server compares the individual’s value of the attribute ‘experience’ with individual value of attribute the ‘no. of years’. If the values are equal, then the server retrieves the information based on the query.

Ontology Development

The ontology framework enables the construction of ontologies at the Data model layer. The inability of existing integration strategies to organize and apply the available knowledge to the range of real scientific, business and governance issues is impacting on not only productivity but also transparency of information in crucial safety and regulatory applications (Al-Sudairy & Vasista, 2011). The potential benefits of ontologies in general-purpose software technology have been widely discussed and ontologies have not achieved a major breakthrough yet (Vladan, 2002). Among the XML conforming languages, we use Web Ontology language (OWL) in this paper for simpler implementation and powerful inference of ontology. Ontology is an explicit description of a domain and it involves following steps:

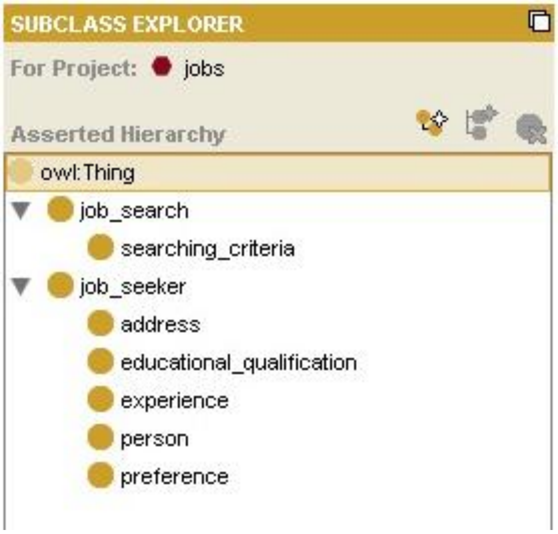
- Defining concepts
- Deriving the class hierarchy
- Defining data and object properties of each class
- Setting up domain and ranges for each property
- Providing constraints
- Adding individuals
- Checking consistency
- Testing the ontology with queries and responses

In the employment exchange system, we developed two ontologies namely 'Employer' and 'Job seeker'. User interface model allows users to dynamically update the ontologies.

Job Seeker Ontology

The job seeker's information need is to find the job which suits them well from the pool of employers. In the employment exchange system, the information appears in Job Seeker ontology as shown in Figure 4 and it is having common vocabulary with employer ontology. But, they appear in different granularity levels in these data models.

Figure 4. Job seeker Ontology



SUBCLASS EXPLORER

For Project: ● jobs

Asserted Hierarchy

- owl:Thing
- ▼ job_search
  - searching_criteria
- ▼ job_seeker
  - address
  - educational_qualification
  - experience
  - person
  - preference

It is important to impose restrictions for the classes. The Resource description framework statement in the ontology is a three valued statement (i.e.) subject, predicate and object. Here subject is viewed as domain, predicate is viewed as property and object is viewed as range. There are two types of properties namely data type property and object property for any concept. Domain and range specifications are assigned for properties (data, object) of all the concepts in the ontology. Data type property relates an individual with a set of values. Domain is an individual of any class and range is a set of values of a particular data type. Object property relates an individual with another individual i.e., here both domain and range of three valued statement are individuals. The data type properties for the concepts in the jobseeker ontology are as follows:

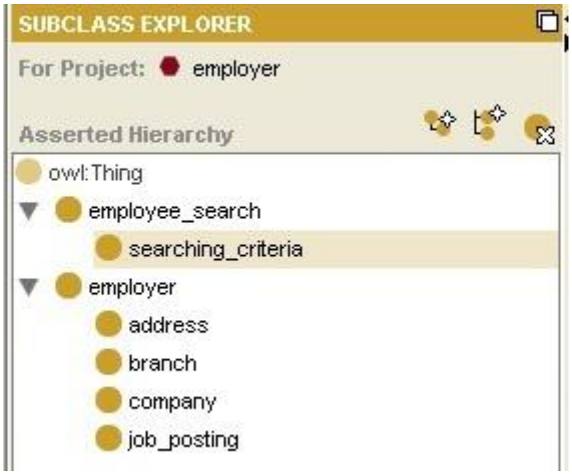
Educational qualification: basic level, ug, pg, key_skills
Address: city, door_no, street, location, state, pincode
Experience: Functional_area, job_title, no_of_years
Person: contact_no, gender, dob, email_id, community, name, nationality
Preference: location, time, salary, role, company

The object type property of Jobseeker ontology includes has_preferred to, is_location_of, is_preferred_by, had_preffered_to, has_experienced_With, is_located_at and has_qualified_with.

Employer Ontology

The employers work is to find the right jobseekers that suit the job requirement of available job in the company. During registration, employers provide company/organization details. After successful registration, employers input job posting details in the input form available in the user interface. Agent uses the search criteria concept in this ontology to search the jobseekers based on various criteria like company, location, role and time. The Employer ontology has class hierarchy as shown in Figure 5 and the data and object properties are described as follows:

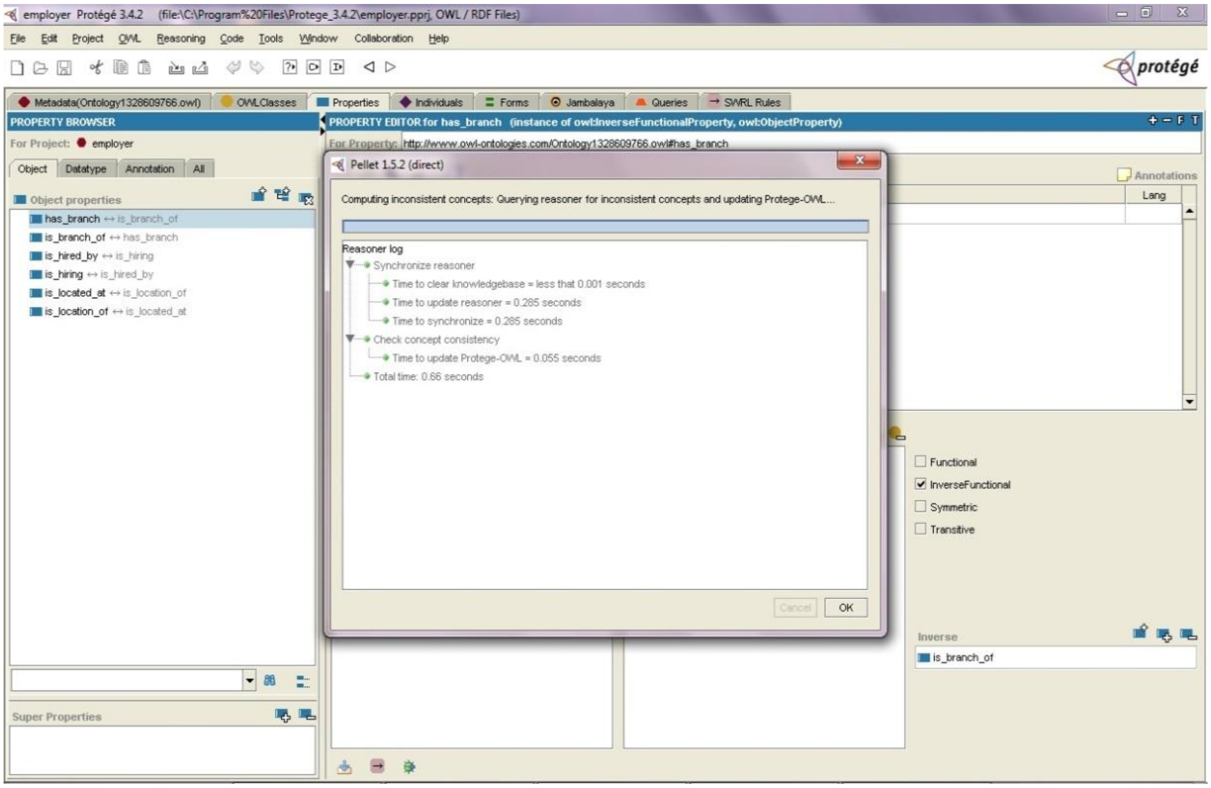
Figure 5. Employer Ontology



<u>Address</u>
Data type property: city, pin_code, state, street
Object property: is_location_of
<u>Branch</u>
Data type property: branch_contact_no, branch_name
Object property: is_branch_of
<u>Company</u>
Datatype property: e_mail, contact_no, name, no_of_employee, type, web_site, year_of_estabishment
Object property: has_branch, is_hiring, is_located_at
<u>Job_posting</u>
Datatype Property: experience, gender_preference, job_title, key_skills, qualification, referal_contact_no, referal_e_mail, refereal_name, salary, time and vacancy
Object property: is_hired_by

Necessary constraints are assigned for the resources in both the ontologies after building the class hierarchy and defining properties. The next important step is testing the ontology for consistency and the ontologies are consistent as in Figure 6. The Pallet reasoner plug-in of protégé is used for reasoning the ontology and to check for inconsistencies. Once the ontology is found to be consistent, then we can add individuals to each property. The ontology should be classified for the purpose of error detection. Classifying the ontology is must for querying purposes.

Figure 6. Consistency Checking



employer Protégé 3.4.2 (file:\C:\Program%20Files\Protege_3.4.2\employer.pprj, OWL / RDF Files)
File Edit Project OWL Reasoning Code Tools Window Collaboration Help
protégé
Metadata(Ontology1328609766.owl) OWLClasses Properties Individuals Forms Jambalaya Queries SWRL Rules
PROPERTY BROWSER
For Project: employer
Object Datatype Annotation All
Object properties
has_branch ↔ is_branch_of
is_branch_of ↔ has_branch
is_hired_by ↔ is_hiring
is_hiring ↔ is_hired_by
is_located_at ↔ is_location_of
is_location_of ↔ is_located_at
Super Properties
PROPERTY EDITOR for has_branch (instance of owl:InverseFunctionalProperty, owl:ObjectProperty)
For Property: http://www.owl-ontologies.com/Ontology1328609766.owl#has_branch
Annotations
Lang
☐ Functional
☑ InverseFunctional
☐ Symmetric
☐ Transitive
Inverse
is_branch_of
Pellet 1.5.2 (direct)
Cancel
OK

Computing inconsistent concepts: Querying reasoner for inconsistent concepts and updating Protege-OWL...

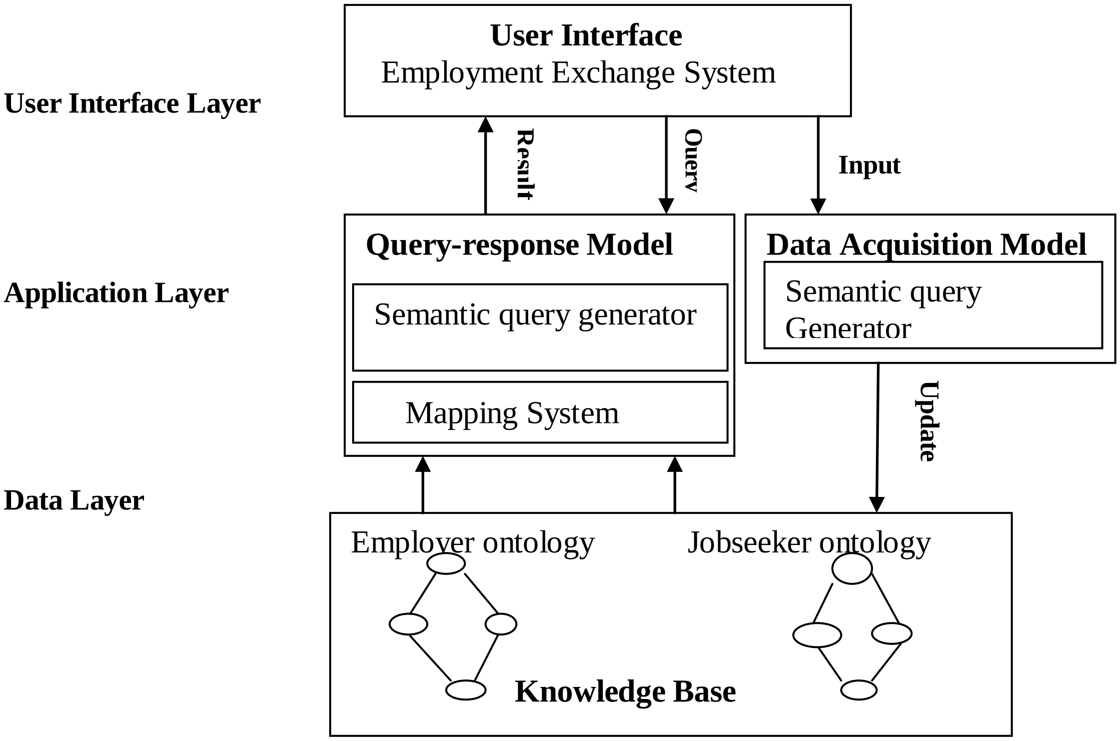
```
Reasoner log
▼ Synchronize reasoner
    Time to clear knowledgebase = less that 0.001 seconds
    Time to update reasoner = 0.285 seconds
    Time to synchronize = 0.285 seconds
▼ Check concept consistency
    Time to update Protege-OWL = 0.055 seconds
  Total time: 0.66 seconds
```

Results and Discussion

Employment Exchange System use semantic technologies in the development of framework. **S**PARQL **P**rotocol **a**nd **R**DF **Q**uery **L**anguage *(*SPARQL*)* is an RDF query language, able to retrieve and manipulate data stored in RDF format. Agent program uses SPARQL to query ontologies. Jena is java based semantic framework which provides an API to read/write data from/to RDF graphs. Semantic agent is built using Jena framework model that supports SPARQL and allow reasoning support for the underlying OWL data model. Abstracting domain knowledge to an ontology layer avoids extensive reliance on custom procedural programming and the need to rewrite legacy code whenever a model, schema, or policy changes (Stephens et al., 2006).

The framework for Employment Exchange System has user interface layer, application layer and data layer as in Figure 7. Data layer includes Employer ontology and job seeker ontology. Ontologies have concepts and attributes with terminological differences but with semantic consensus. Similar terms appear in different granularity levels of the ontologies. When ontologies integrate for semantic similarity, collaboration between two heterogeneous data models is achieved. Developing ontologies in this manner proves the benefit of building the interoperable application.

Figure 7. Framework for Employment Exchange System



User Interface Layer

The user interface of the framework allows to perform the process of search and to input the user information in user understandable form. The users are in two categories namely job seekers and employers. The role of each user is described below:

(1) Job seeker

The job seekers can login into the system by providing their e-mail ID and password. Users input this information during registration. Upon successful login, the candidate can do following activities. They can record their experience, preferences and other personal information. They can search for jobs posted by the employers. Jobseekers use any search criteria like company, location, role and salary to know the employer information. Special object properties relate the jobseeker individuals with the employer individuals. Object properties like is_preferred_by and has_preferred_to relate jobseekers individuals with employer individuals.

(2)Employer

The employers can login to the system by providing their e-mail id and password. Two major functions carried out by the employers are i) posting the job details. ii) Search from the pool of jobseekers according to search criteria like name of the company, location, role of the job and time. They can draw necessary list of jobseekers from the system at employer's requirement.

Job Posting: The employer can post the available vacancies in the Web application. They submit the necessary details like name of the post, number of vacancies, qualification required, compensation offered, last date for application etc. This will be made available to all registered candidates who logs on to the system.

Search for Candidates: The employer search for jobseekers whose information fulfills employer's requirement. Employers use any one of the search criteria for selecting the jobseeker from the pool of jobseekers. In addition, the employers get the information automatically as soon as the jobseeker details matches their job posting details.

Application Layer

In contrast with database technology, in the ontology approach missing information is treated as unknown. Schemas in database technology behave as constraints on structure of data whereas ontology axioms are used as inference rules. The application logic layer consists of two main processes namely data acquisition process and query-response process. Agent updates the data models dynamically whenever user registration takes place. The users (employers, jobseekers) register their details to join in the pool of information model. Following actions are carried out during search process.

1. The user submits the query in user understandable form in the user interface provided in the framework.
2. Query submitted by the user is transformed to SPARQL Query by the Semantic Query Generator before being processed by the mapping system in the Semantic Agent.
3. After analyzing the Query, matching between two ontologies is performed as required by the SPARQL query and response is produced in the RDF format.
4. Response produced by the agent is then converted in to user understandable format.

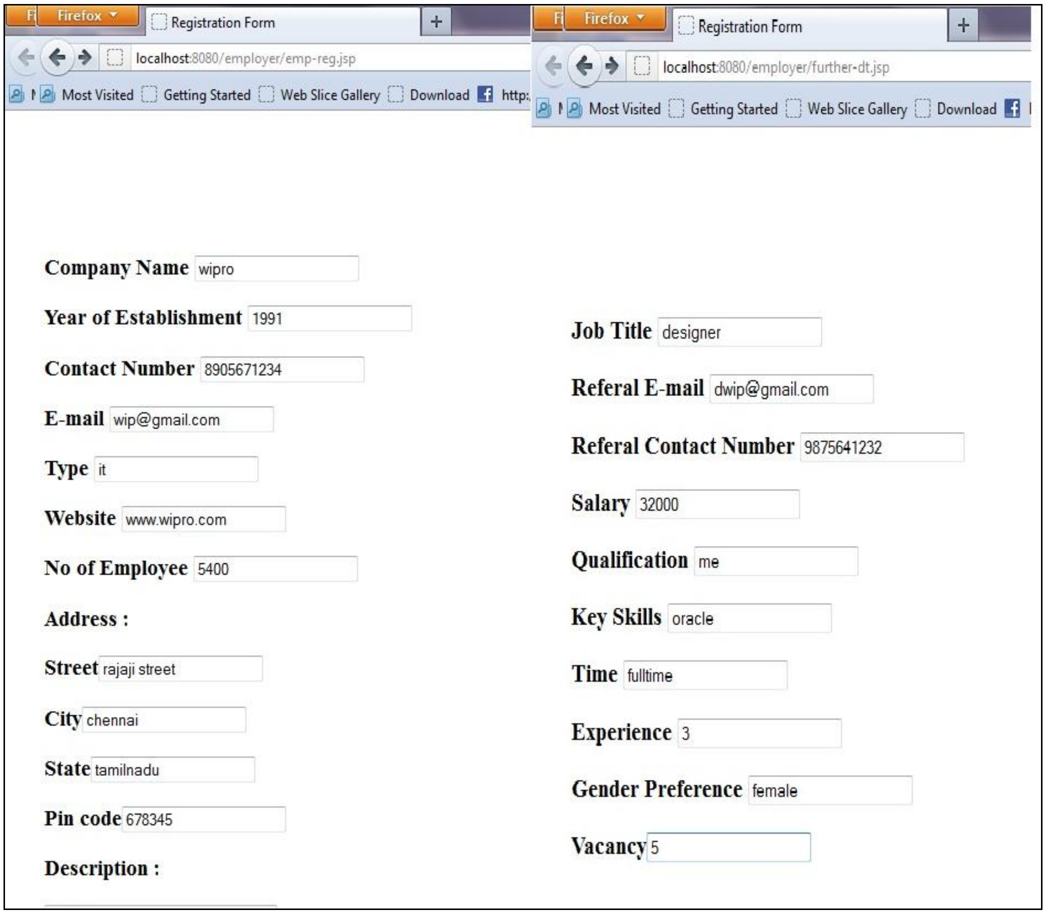
Search / Input data

Searching for data and inputting data are the options available in the User Interface of the framework for the end users (jobseekers, employers). Semantic Query generator converts user input available in user understandable form to machine understandable format. Figure 8 shows the employer registration form and employer job posting form in the user interface of data acquisition model. Figure 9 shows the job seeker registration form used for capturing personal, experience details of a job seeker along with their preferences during data acquisition phase. The data acquisition phase allows the users to input data dynamically as shown in figures 8 and 9. The Search or Query option and relevant response from the Query-response model are shown in Figure 10 and 11. Figure 10 shows the response about job seeker information for the employer's query. Employer query uses company 'tcs' as the searching option. Job seeker information available in the response includes details of mobile number, name of the person, e-mail id, experience and his job title preference. Jobseeker's query and response is shown in Figure 11. The query by Jobseekers uses search criteria company with the value 'tcs' and they get the response about the 'tcs' company information like website address, referral contact number and e-mail id. In the case of Input option, the agent translates the user input query to triple format (SPARQL) and updates the respective ontology dynamically without the need for any change in the framework.

Semantic Agent

It is important to fully automate the semantic mapping among ontologies for the semantic web to realize its full potential. Semantic Agent is built using Jena framework model. Semantic Query Generator in the semantic agent translates the query posted by the end users into equivalent SPARQL query. Query formation also takes place when the jobseeker submits the details (input). Then the agent processes the machine understandable query. Information retrieval takes place from the data models based on the mapping constraints. Mapping constraints finds for semantic overlap between particular individual's attributes/concept in one ontology and concepts/attributes of the individual in the cooperating ontology. Information from the data models are in triple form. Hence, agent formulates the response from triple format to user understandable form. Response includes individual's information which satisfies the mapping constraints.

Figure 8 Employer registration Form



Company Name wipro

Year of Establishment 1991

Contact Number 8905671234

E-mail wip@gmail.com

Type it

Website www.wipro.com

No of Employee 5400

Address :

Street rajaji street

City chennai

State tamilnadu

Pin code 678345

Description :

Job Title designer

Referal E-mail dwip@gmail.com

Referal Contact Number 9875641232

Salary 32000

Qualification me

Key Skills oracle

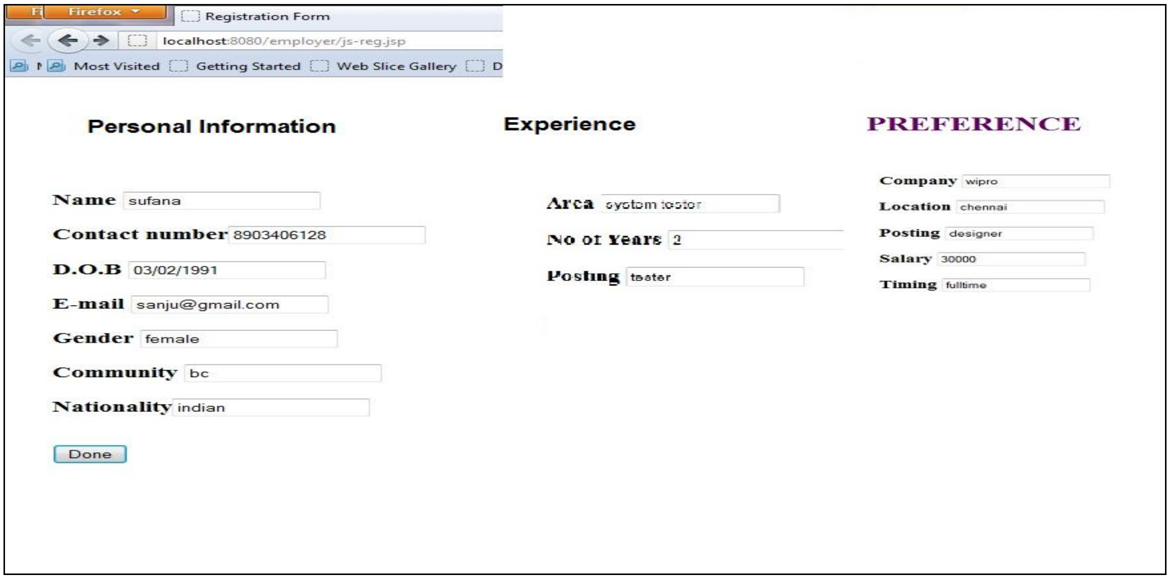
Time fulltime

Experience 3

Gender Preference female

Vacancy 5

Figure 9 Jobseeker registration Form



Personal Information

Name sufana

Contact number 8903406128

D.O.B 03/02/1991

E-mail sanju@gmail.com

Gender female

Community bc

Nationality indian

Experience

Area system tester

No of Years 2

Posting tester

PREFERENCE

Company wipro

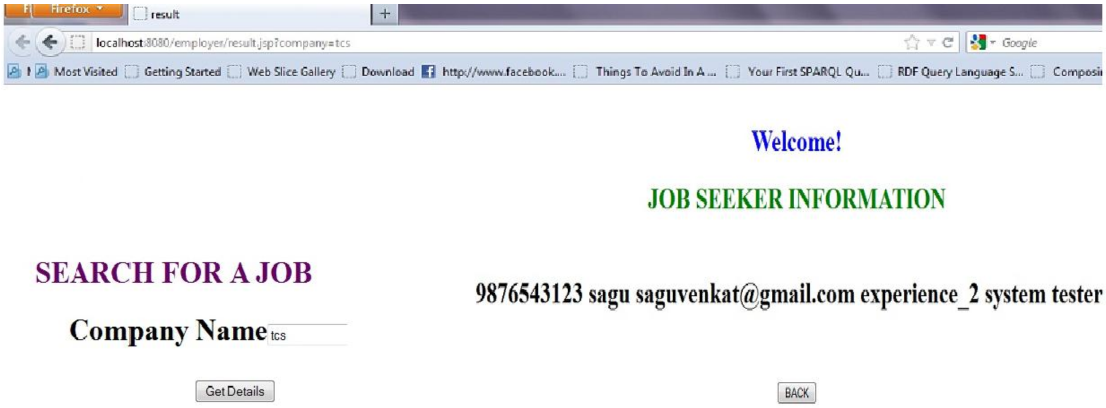
Location chennai

Posting designer

Salary 30000

Timing fulltime

Figure 10 Query by Employer with criteria-company='tcs' and response jobseeker details



SEARCH FOR A JOB

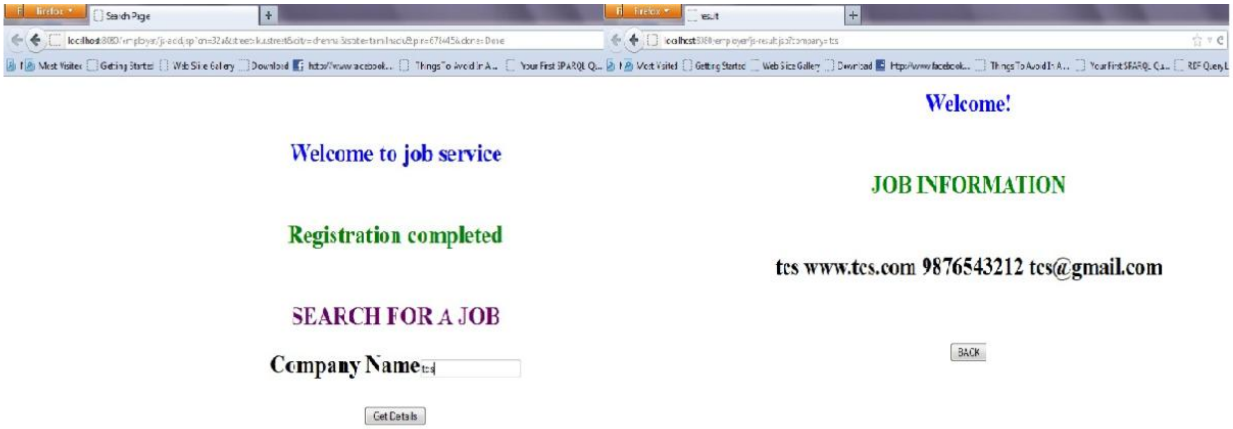
Company Name tcs

Welcome!

JOB SEEKER INFORMATION

9876543123 sagu saguvenkat@gmail.com experience_2 system tester

Figure 11. Query by Job Seeker with Company Name='tcs' and response company details



Welcome to job service

Registration completed

SEARCH FOR A JOB

Company Name tcs

Welcome!

JOB INFORMATION

tcs www.tcs.com 9876543212 tcs@gmail.com

Ontology Mapping Metrics

Mapping process considers equivalent correspondence between different entities of the contributing ontologies. Different entities of ontology include concepts and attributes. Mapping carried out in employment exchange system finds out equivalent correspondence among the entities of Employer and Jobseeker Ontology. Interoperability of heterogeneous data sources is possible only when essential mapping are done. In Employment exchange system, Job Posting concept in Employer ontology and Preference concept in Jobseeker ontology are treated essential mapping for the application and since some of the entities are not mapped automatically by the mapping logic, manually mapping is done. Metrics used for ontology mapping are Precision, Recall and f-Measure. Table 1 below shows metrics applied for the mapping logic used in employment exchange system.

Definition : (Precision and recall). Given a reference mapping R, the precision of some mapping A is given by $P(A, R) = |R \cap A| / |A|$ and recall is given by $R(A,R) = |R \cap A| / |R|$

Definition: F-Measure is given by (2*P*R) / (P+R).

Concept to Concept Category:

M1 <address (Employer ontology) = address (Jobseeker ontology)>
M2 <search-criteria (Employer ontology) = search-criteria (Jobseeker ontology)>
Above reference mappings M1 and M2 are found automatically. Hence, Precision=1 and Recall= 1

Concept to Attribute Category:

M1 <company (Employer ontology) = company (Preference concept in Jobseeker)>

M2 <educational-qualification (Jobseeker ontology) = qualification (Job posting in Employer ontology)>
M3 <experience (Jobseeker ontology) = experience (Job posting in Employer ontology)>
are reference mapping and they are mapped, hence, Precision=1 and Recall=1.

Attribute to Attribute Category:

There are 7 reference mapping. Out of which 6 are found by mapping logic. Hence, precision=1, recall=.85.Along with standard precision and recall, f-measure is calculated and shown in Table 1.

Table 1. Ontology Mapping Metrics for employment exchange system

	Concept to Concept Mapping	Concept to Attribute Mapping	Attribute to Attribute Mapping	Average of all mapping
Precision	1	1	1	1
Recall	1	1	.85	.95
F-Measure	1	1	.91	.97

Mapping requirement vary according to the application and always contributing distributed ontologies may not possess the compactness or completeness expected in the ontology mapping and is always not possible to find out all mapping automatically by the logic. It also depends on the nature of vocabularies used in ontology design. In most of the research papers, mapping or alignment discussed only for concept to concept. It is significant that the other category of mapping discussed in this paper also play role in ontology based application like employment exchange system. The need for ontology mapping of library ontology was discussed in (Shenoy et al., 2011). Mapping for hotel domain and Query re-writing was discussed in (Juarna et al., 2012). Prompt algorithm which provides semi-automatic merging and alignment of ontologies is given in (Noy & Musen, 2003). The above references highlight only ontology matching techniques and the previous literature does not mention about the application using ontology mapping. Requirement of ontology mapping for interoperability can be understood only by building real applications.

Conclusion

The proposed semantic application framework used for employment exchange system proves that the semantic web application can be built using semantic web technologies to integrate data from multiple distributed data sources. Conflicts and mismatches in the knowledge base of employment exchange application are reconciled by semantic web based application development. Application allows the job seeker to access one single interface to avail all the information available in the market and quality driven query answering. The key benefits of building interoperable application using ontology mapping improves data sharing between jobseeker and employer knowledgebase. The system provides higher level of abstraction because the end users do not know any details about underlying data or its representation. Semantic mapping between the data models facilitates the best query response. As opposed to the distributed database applications, ontology based application promises rich consistency and at the same time gaining advantage of distributed application development. Independent maintenance of the model decreases the interdependencies among layers of the application framework. Addition of new concepts in the ontologies does not cause any reengineering of the application design. Ontology metrics precision, recall and f-measure drawn from mapping logic enhance the correctness, completeness and definiteness of mapping in the application design.

Future Enhancement

- The query processing system will be designed in such a way that it can generate the SPARQL query for any natural language query.
- Ontology store can be built in future to sustain the increasing number of users
- Enhancement of functionalities by building richer knowledgebase
- Same Framework can be applied to any multi model interoperable applications
- Application can be extended to incorporate some more functionalities

References

Al-Sudairy, M.T., & Vasista, T. G. K. (2011). Semantic data integration approaches for e-governance. *International Journal of Web & Semantic Technology*, 2(1), 1-12.

Antoniou, G., & van Harmelen, F. (2004). *A Semantic Web Primer.* (2nd ed.). Cambridge, MA: MIT press.

Corby, O., Dieng-Kuntz, R., Faron-Zucker, C., & Gandon, F. (2006). Searching the semantic Web: Approximate query processing based on ontologies. *IEEE Intelligent Systems & their Applications*, *21*(1), 20-27.

Gardner, S. (2005). Ontologies and semantic data integration. *Drug Discovery Today*, *10*(14), 1001-1007.

Juarna, A., Refianti, R., Mutiara, A.B., & Pratiwi, D.I. (2012). Ontology mapping for interoperability representations in hotel domain. *Journal of Theoretical and Applied Information Technology*, 35(2), 135-140.

Klyne, G. (2004). Semantic web applications. Retrieved January 10, 2013, from http://www.ninebynine.net/Papers/SemanticWebApplications.pdf

Noy, N. F., & Musen, M. A. (2003). The PROMPT suite: interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, 59(6), 983-1024.

Orgun, B., Dras, M., Nayak, A. and James, G. (2008), Approaches for semantic interoperability between domain ontologies. *Expert Systems*, 25(3), 179-196.

Shenoy, M.K., Shet, K.C., & Acharya, D.U. (2011). Interoperability using ontology mapping. *Journal of Computer Science and Engineering*, 5(1), 2-4.

Stephens, S., Morales, A., & Quinlan**,** M**.** (2006). Applying semantic web technologies to drug safety determination. *Intelligent Systems*, *IEEE*, 21(1), 82-88.

Vladan, D. (2002). Understanding ontological engineering. *Communications of the ACM*, 45(4), 136-144.

Bibliographic information of this paper for citing:

Ayechetty, Kavidha, & Arumugam, Saradha (2013). "Intelligent interoperable application for employment exchange system using ontology." *Webology*, 10(2), Article 113. Available at: http://www.webology.org/2013/v10n2/a113.pdf