Interoperable Semantic Application Design using Distributed Data Models

A. Kavidha [1], Dr. A. Saradha[2]

[1]Assistant Professor (senior)
Institute of Road and Transport Technology, Erode, Tamilnadu – 638316.
[2]Associate Professor
Institute of Road and Transport Technology, Erode, Tamilnadu – 638316.
kavitha.irtt@gmail.com

Abstract: The paper proposes a framework for building intelligent interoperable application 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. Semantic web technologies have the potential to build intelligent interoperable application using multiple heterogeneous data models as it uses ontology for knowledge representation. Semantic technology based application also provides the key benefits like improved data sharing, higher level of abstraction, best query response, independent maintenance of the model. Semantic agent in the proposed framework transforms the user queries semantically for possible integration between heterogeneous data models to drive intelligent inference. In order to reuse the distributed ontologies in collaborative knowledge models, they are transformed to one another to elevate the semantic inconsistency among concepts, attributes and individuals in the contributing ontologies. This paper also proposes a mapping algorithm for automatic mapping of contributing ontologies in various levels. The proposed framework and algorithm is tested for a case study for building employment exchange system with heterogeneous data models. In the proposed case study, the collaborating data models are mapped for structure, syntax and semantics in different granularity level to reconcile the conflicts and mismatches exists among them and the intelligent application is built using the framework to provide best query response.

[A. Kavidha, Dr. A. Saradh. **Interoperable Semantic Application Design using Distributed Data Models.** *Researcher* 2019;11(7):78-86]. ISSN 1553-9865 (print); ISSN 2163-8950 (online). http://www.sciencepub.net/researcher. 8. doi:10.7537/marsrsj110719.08.

Keywords: Interoperable Knowledgebase, distributed Knowledge models, ontology integration, semantic integration, collaboration of knowledge models, ontology mapping, semantic web application

1. Introduction

Semantic web provides data integration capabilities from the semantics of terminologies in the distributed heterogeneous data models. Ontologies are conceptual backbone of semantic web provides efficiency in data integration. Ontologies based knowledge representation support formal specification of a shared conceptualization of a domain in which the meaning of terms and relations are defined with different levels of formality. Ontologies enable integration of knowledge that can be reused by several applications across governance or business [Gardner S. P, 2005]. 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 mapping 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.
- Managing multiple knowledge bases for application design.
- Interoperating the knowledge bases for any semantic application design.

In the proposed approach, application is built using two different data models by integrating the relations among the two ontologies. The interoperability between the ontologies is achieved by the careful selection of behaviors and vocabularies. Concept in one ontology can be used as relation in another ontology. Sometimes semantically equivalent concepts of ontology mean the same but they use different terminologies. For example, the terms ‘gender’ and ‘sex’ mean the same but they are syntactically different. Data models with such semantic inconsistencies are resolved by the proposed mapping algorithm and are integrated to drive

intelligent inference. The domain of employment exchange is proposed as the candidate for building interoperable, co-operative and 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. Ontologies ensure an efficient retrieval of Web resources by enabling inferences based on domain knowledge as said in [Olivier Corby, Rose Dieng-Kuntz, and Fabien Gandon, 2001]. The employer and jobseeker behaviors allow them to function together by mapping the concepts and attributes of two ontologies. 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. 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. Collaboration among these data model is possible when mapping them for equivalence as per the mapping algorithm. The agent processes the semantic query posed by both employer and jobseeker and produces best response. Also, it is possible for the agent to give automatic response for the specified criteria. The application gives better response for the query involving vocabularies of different concepts.

2. Related Works

The importance of integration has been highlighted in the following papers. Earlier work on development of application using ontology integration is not explicitly mentioned in these papers. We find the some methodologies adopted for integration of ontologies.

- A key challenge is integrating the abundance of publicly available data sources of varying quality, inconsistent data formats, data models and terminologies, said by Susie Stephens**,** Oracle Alfredo Morales et al. (2006).
- Semantically enabled interoperability of the government services was emphasized many times as a key challenge, said by Furdiki K, et al., (2010).
- Approaches to implementing RDB to RDF was discussed by Dr. Mohammed T. Al-Sudairy and T. G. K Vasista (2011)
- It is cited that the earlier works on semantic data integration are done in fields such as semantic annotation of geodata. (Kieler, 2008).
- It is cited that 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. (Dr. Mohammed T. Al-Sudairy and T. G. K Vasista 2011).
- B. Orgun, 2006 said that if ontologies for multi-agent systems and the semantic web are to realize their full potential, it is important to fully automate the semantic translation among ontologies.
- Ontology mapping algorithm presented by Natalya F. Noy and Mark A. Musen, 2003 is a graph based method and supports only frame based ontology model. They claim that mapping algorithm “Anchor Prompt’ will not work if source ontologies are constructed differently.

3. 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. For example, the term ‘course’ may be used with two different usages. It may refer ‘degree’ or ‘subject’ [Grigoris Antoniou and Frank van Harmelen, 2004]. The differences in terminology between ontologies are solved by mapping logic. The Semantic agent in the proposed system search for relations “owl: equivalent Property” and “strongly equivalent concept” between terms to make possible collaboration activities in the application. Mapping logic uses these object properties between terms to link them and linking of terms derive semantic interoperability among cooperating ontologies. Thus, semantic heterogeneity is solved by the use of ontologies automatically.

Main information entities are represented in a domain to build ontologies and in turn architecture simplifies the collaboration task by supporting addition and removal of sources. Collaboration between ontologies is difficult if they have minimal common vocabulary and overcome by inter-ontology mapping. Common vocabulary defined for different concepts across different ontologies avoids arbitrary mapping [Wache, H., Vögele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., Hübner, S,2001]. 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 and differences in syntactic representation. Semantically equivalent terms appear as concept or as attribute in different ontologies. The proposed interoperating system also defines concepts on the basis of common meaning with different terminologies. Semantic conflicts occur whenever two concepts with the same name do not use same interpretation of information and does require mapping or transformation. For example, the term

'course' in source ontologies refers in different contexts 'subject' and 'degree'. So, syntactical or lexical equivalence of terms in collaborating data models can't be mapped as equivalent automatically.

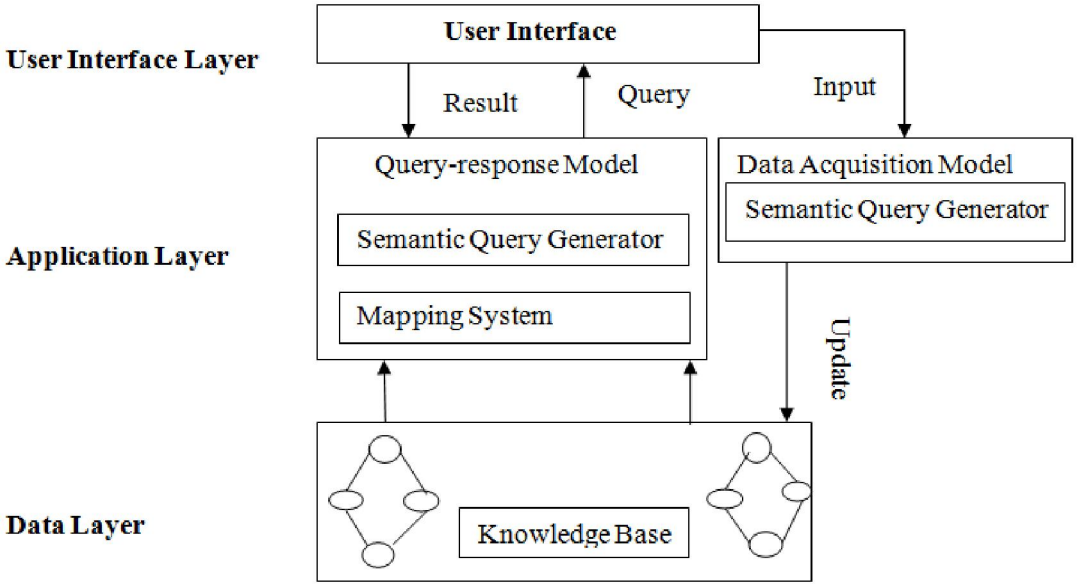

Figure 1. Framework for Collaborative Knowledge based application development Hence they are subjected to manual transformations.

Knowledge based approach for interoperable application development is designed and is shown in the Figure.1. Architecture shown in the proposed framework has user interface layer, application layer and data layer. The user interface of the framework supports the process of query in natural language and input the user information in web form. Data layer includes collaborative knowledge models represented using multiple ontologies. Application layer consists of Query-response model and Data Acquisition models. The paper proposes the case study on employment exchange system for building collaborative knowledge based approach in intelligent interoperable application. The following models are explained based on employment exchange system (EES).

3.1 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 the web form. 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.

3.2 Query-Response Model

Like producer and consumer process, employer 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 by entering the name of the company, the employer gets details about jobseeker information like jobseeker's 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. 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.

3.2.1 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 [B. Orgun, M. Dras, A. Nayak and G. James,2006]. 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 agent automatically identifies the similar concepts and its similar attributes and relates them as equivalent based on the following algorithm.

3.2.2 Mapping Algorithm

1. Check concepts in ontologies for their lexical and semantic similarity maintaining their structural equivalence (same super and subclass)

2. Attributes of equivalent concepts are compared for term similarity as well as for semantic similarity. Further, they are checked if they are of same data type and the restrictions of values they represent are same. Sometimes there may not be a single equal attribute even if the concepts are equivalent.

3. Object properties of the equivalent concepts are compared for syntactic and semantic similarity if they have equivalent domain and range. The domains of the property in both ontologies are found to be equivalent if they are already classified as equivalent in step 1. Ranges of equivalent concepts are compared for similarity of terms as well as the cardinality of the values represented by the terms.

4. Concepts in source ontologies are **strongly equivalent** if they satisfy all the above criteria and hence they are categorized as strongly equivalent automatically.

5. The concepts which are not strongly equivalent are identified and they are aligned (or) mapped manually.

The syntactic similarity is found out using Dice similarity method and Word Net API edu.mit.jwi_2.2.3 is used for semantic similarity in the contributing ontologies. The above mapping logic is used for finding equivalent concepts automatically. Manual alignment among concepts and attributes takes place if they are not selected as equivalent automatically. The mapping system available in the agent uses the manual mapping 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 is 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. Mapping used between them is shown in Figure 2 below. The concept 'preference' in both ontology is syntactically equal but the proposed mapping algorithm does not identify them as strongly equivalent as they do not have equivalent attributes. Hence manual alignment or mapping takes place in various categories. Various kinds of manual mapping carried out between two data models include attribute to attribute mapping, concept to attribute mapping and attribute to concept mapping.

3.2.3 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 and syntactically equal, they are integrated with the relation 'owl: equivalent Property' 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 and identified them as

equivalent manually. The application considers the terminological differences and exploits the 'equivalent property' for attribute to attribute integration.

3.2.4 Concept to Attribute Mapping

In Concept to attribute mapping, concept in one ontology semantically and/or syntactically 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'. Equivalent relation between concept and attribute of the source ontologies is actually between any attributes of concept having semantically equaled with attribute. Application 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. It is equivalent to the conversion of primary data type to user defined data type in programming paradigm.

3.2.5 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 as in Figure 2. 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.

3.2.6 Concept to Concept Mapping

If concepts between different input sources contain same information, a concept to concept mapping takes place. Concept 'Search Criteria' 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.

4. Results and Discussion

Employment Exchange System (EES) use semantic technologies in building collaborative application development and allow reasoning support for the underlying OWL data model. EES uses two distributed interoperable knowledge models namely Employer and Jobseeker. The concepts of both data models may share some common vocabularies which are identified as strongly equivalent by the proposed mapping algorithm.

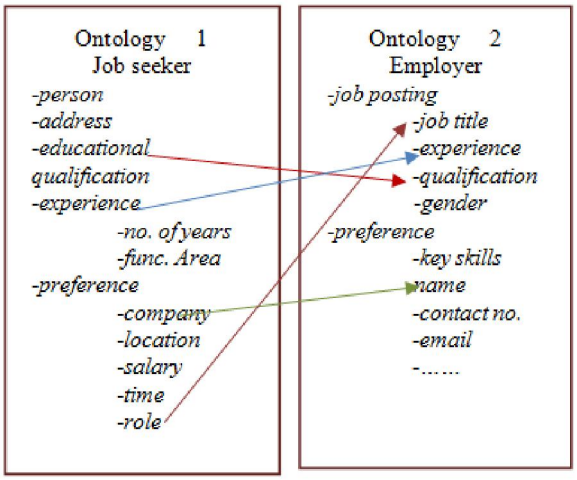

Figure 2. Ontology Mapping

The data models are distributed and they are reused without making any alteration in the application development. ***S**PARQL **P**rotocol **a**nd **R**DF **Q**uery **L**anguage (SPRQL)* 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.

4.1 User Interface Layer

The user interface of the framework in Figure-1 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:

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.

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.

i) **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.

ii) **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.

4.2 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.

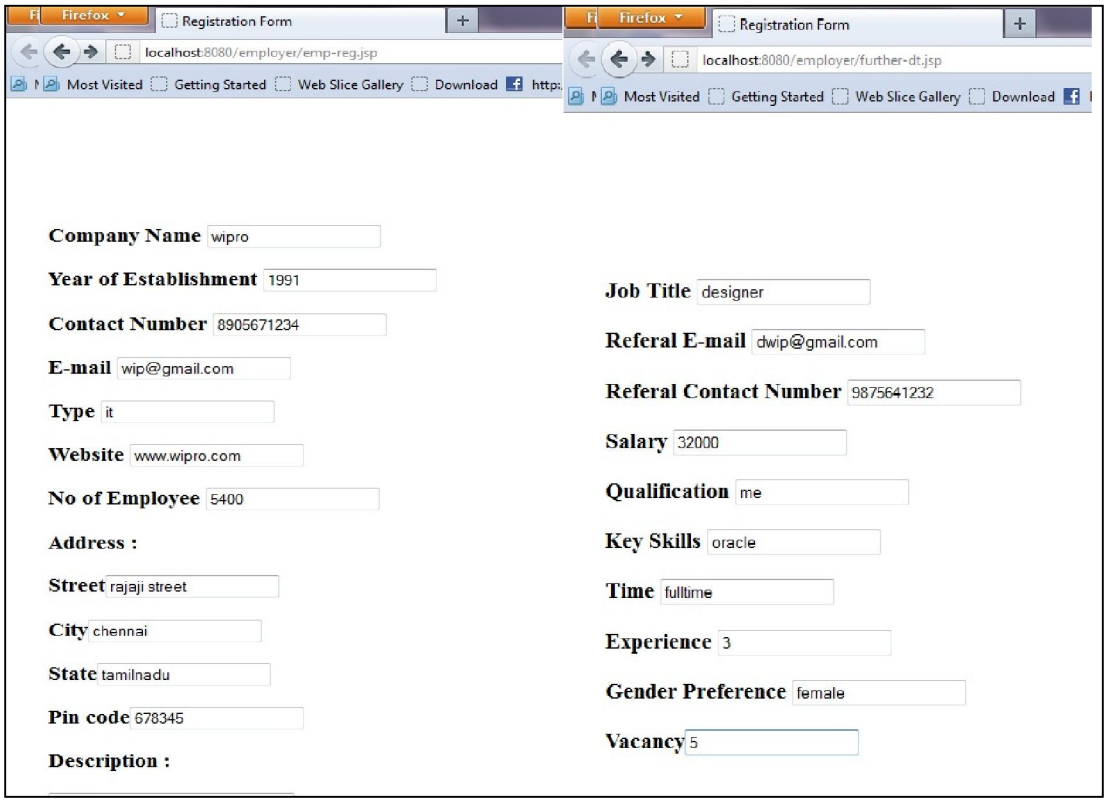

Figure 3. Employer registration

4.2.1 Search / Input data

Searching for data and inputting data are the options available in the User Interface of the framework for the end users (jobseeker, employers). Semantic Query generator converts user input available in user understandable form to machine understandable format (SPARQL). Figure 3 below shows the employer registration form and employer job posting form. Figure 4 below shows the job seeker registration form for capturing personal and

experience details of a job seeker along with their preferences. The Figure 5 shows the response about job seeker information for the employer's query using search criteria company name with the value "tcs". Job seeker information in the response includes details of mobile number, name of the person, e-mail id, experience and his job title preference and the response is given to the employers. Figure 6 shows the jobseeker's query with search criteria company with the value 'tcs' and 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.

4.2.2 Semantic Agent

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.

5. Conclusion

The key benefits of building a framework for semantic web application using interoperable knowledge bases improves data sharing and knowledge management of several distributed heterogeneous knowledge bases. The proposed system provides higher level of abstraction because the end users do not know any details about underlying data or its representation. Mapping logic uses three levels of mapping structural, syntax and semantics. Semantic mapping between the data models facilitates the collaborative and intelligent query response. Earlier works on ontology alignment or mapping considered only similarity measure and is not semantically justified. Conflicts and mismatches in the knowledge base of employment exchange application are reconciled by our approach in building collaborative semantic web based application development. Mapping algorithm considers syntax, semantic and structural equality for strong equivalent of concepts among contributing ontologies. The Case study also proves the possibility of mapping between concepts and attributes of other concepts. This kind of mapping is essential if a term is represented as concept by one ontology developer and as attribute by the other ontology developers. The proposed methodology can be adopted for developing semantic application using heterogeneous distributed ontologies.

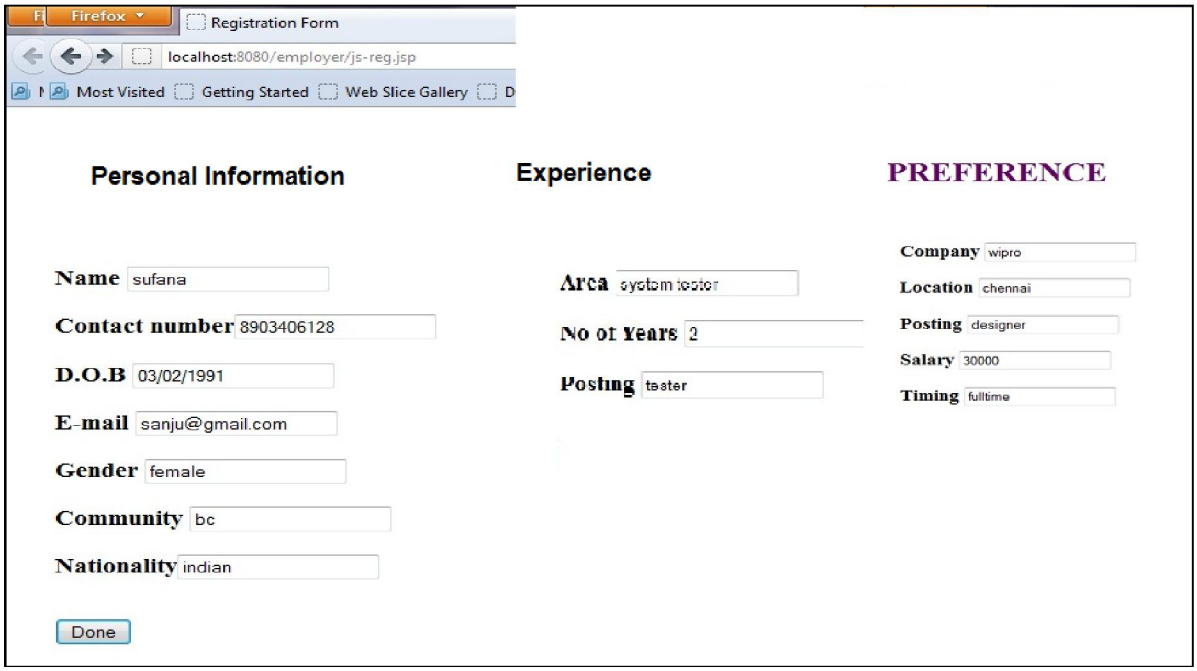

Figure 4 Jobseeker Registration

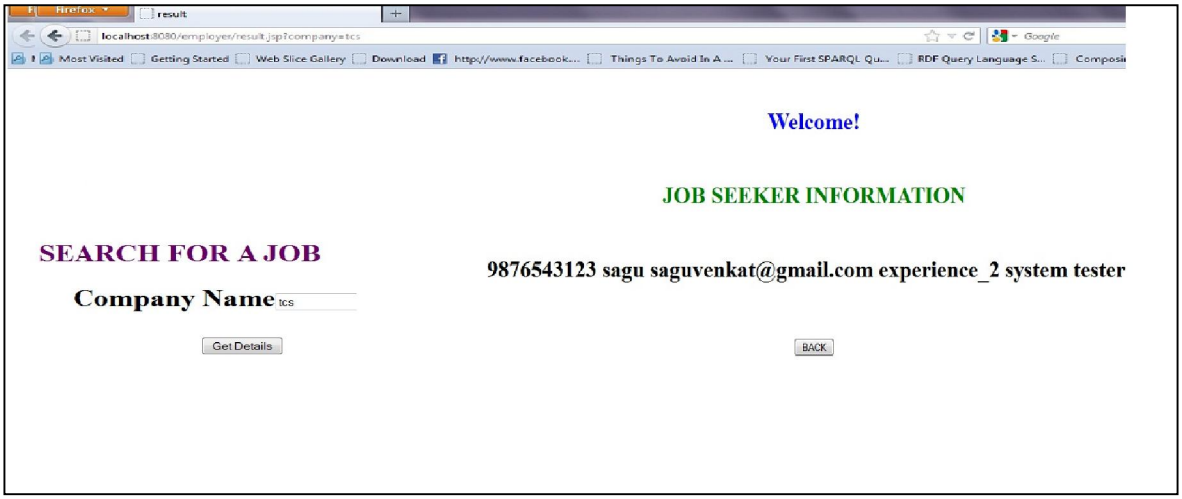

Figure 5 Query by employer using criteria company='tcs', response job seeker information

Figure 6 Query by Jobseeker with Company Name='tcs' and response company information

Corresponding Author:
A.kavidha
Senior Assistant Professor
Dept. of Computer Science & Engg.
Institute of Road and Transport Technology
Erode, Tamilnadu, India
638316

References

1. Gardner S. P. (2005). Ontologies and Semantic Data Integration. Drug Discovery Today, Vol.10, Issue 14, p1001-1007.
2. Olivier Corby, Rose Dieng-Kuntz, and Fabien Gandon, INRIA Catherine Faron-Zucker. (2001). Approximate Query Processing Based on ontologies. IEEE intelligent systems.
3. Susie Stephens, Oracle Alfredo Morales and Matthew Quinlan, Cerebra (2006). Applying Semantic Web Technologies to Drug Safety Determination. IEEE intelligent systems.
4. Furdiki K, Klischewski R, Paralic M, Sabol T, Skokan M (2010). E-Government Service Integration and Provision Using Semantic Technologies.
5. Dr. Mohammed T. Al-Sudairy1 and T. G. K Vasista (2011). Semantic data integration approaches for e-governance. Journal of Web & Semantic Technology Vol.2, No.1.
6. Kieler (2008). Semantic Data Integration across Different Scales: Automatic Learning of Generalization Rule. The International Archives of the Photogrammetry, Remote Sensing and

Spatial Information Sciences. Vol. XXXVII. Part B2. Beijing.
7. Dr. Mohammed T. Al-Sudairy and T. G. K Vasista (2011). Semantic data integration approaches for e-governance. International Journal of Web & Semantic Technology (IJWesT) Vol.2.
8. B. Orgun, M. Dras, A. Nayak and G. James. (AOW,2006). Approaches for Semantic Interoperability between Domain Ontologies. The Australasian Ontology Workshop), Hobart, Australia.
9. Wache, H., Vögele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., Hübner, S. (2001). Ontology-Based Integration of Information-A Survey of Existing Approaches. In Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01).
10. B. Orgun, M. Dras, A. Nayak and G. James. (2006). Approaches for Semantic Interoperability between Domain Ontologies. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 72.
11. Natalya F. Noy and Mark A. Musen (December 2003). The PROMPT suite: interactive tools for ontology merging and mapping. International Journal of Human-Computer Studies, Volume 59 Issue 6, Pages 983-1024.
12. Vladan Devedzi´c (2002). Understanding ontological engineering. Communications of the ACM, 45(4):136–144.
13. Amjad Farooq and M. Junaid Arshad. A Process Model for Developing Semantic Web Systems. New York Science Journal 2010;3(9):34-39]. ISSN: 1554-0200.
14. Grigoris Antoniou and Frank van Harmelen. (2004). A Semantic Web Primer. ISBN 0-262-01210-3, MIT press.
15. Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web. Scientific America, 284(5):34.43, 2001.
16. Golbeck, J., M. Grove, B. Parsia, A. Kalyanpur, J. Hendler, “New Tools for the Semantic Web”, Proceedings of EKAW 2002, LNCS 2473. Springer (2002) 392–400.
17. Tayybah Kiren, Muhammad Shoaib, Muhammad Tariq Pervez, Sonia Majid, Qazi Mudassar Illyas. AN Approach To Partially Import The Ontologies On Semantic Web Based Upon User Choice. Journal of American Science 2010;6(11):571-581.

7/20/2019