Domain-Based Ontology Architecture

Abdolreza Hajmoosaei and Sameem Abdul Kareem

***Abstract*—Ontologies are used for enhancement of explicit semantic information of data sources. We can establish a consensus between communities and resolve semantic heterogeneity conflicts between data sources through ontology. The reason ontologies are becoming so popular is in large part due to what they promise: a shared and common understanding of some domain that can be communicated between people and application systems. The creation of the ontology is a major issue in use of ontology. In order to create an ontology we need to extract all detailed specifications from the domain conceptualization, modeling these specifications based on a data model and formalizing these based on a formal language. Most ontologies are related to a specific domain area (such as university, music, movie, and so on) and thhee use of a particular vocabulary related to the domain area. We call these ontologies as domain-based ontology. In this paper we introduce a model for representing of domain-based ontologies. Subsequently we build an University ontology as a domain-based ontology, based on our proposed model. Finally we propose domain-based ontology architecture as an efficient approach in ontology–based data integration.**

***Index Terms*—Domain-based ontology, ontology representation model, domain-based ontology architecture.**

I. Introduction

Ontology is becoming an effective means in intelligent information integration, information interoperability, information retrieval, electronic commerce and knowledge management [1]. The reason ontologies are becoming so popular is in large part due to what they promise: a shared and common understanding of some domain that can be communicated between people and application systems. Applying ontologies guarantees consistency in communities' understanding of statements made during a communication. By using ontologies, communities are able to communicate based on their defined ontologies. The complication of creating ontologies is due to the resolution of ambiguity and semantics heterogeneity in communication [2].

The major tasks in creating ontologies are extracting all detailed specifications from the domain conceptualization, modeling these specifications based on a data model and formalizing these based on a formal language. Building of ontologies is highly expensive in terms of time, complexity and quality evaluation. The complexity of the building and formalizing ontologies is a disadvantage in the use of ontology [3], [4]. Reducing of ontology creation cost is an area of interesting of ontology users. For this purpose, researchers have made efforts to produce tools for building and evaluating ontologies such as Protégé [5], Ontolingua [6], or Chimaera [7] as ontology-editing environments.

In this paper we propose a model for representing of ontologies and then we discuss the ontology development methodology based on our proposed representation model. Subsequently we create a domain-based ontology as a case study in the domain of the university. Finally we explain role of domain-based ontologies in semantic data integration.

II. Ontology Representation Model

Before the creation of an ontology we need to define a representation model for representing the ontology. The representation model should be general, expressive and compatible. Our proposed representation model is general and any ontology with any representation model can be transformed to this representation model [8].

Definition 1: T:=(C,A,R,V), each ontology element (term) is one of following entities:

C: concept or instance of one concept

A: attribute of one concept

R: relationship between concepts

V: value range of one relationship

For example student (concept), age (attribute), master student (instance of student, it is considered as sub-concept of student in our model), attend (relationship between student and class) and "<20" (value range of "max-credit-course" relationship) are some terms in the university ontology.

Definition 2: C:=(name, syn-set, A, key-A, key-R), each concept is defined with its name, set of its synonyms, attributes, its key attributes, and key relationships with other concepts. The key attributes and key relationships are subsets of concept attributes and relationships. The key attributes and key relationships are specific properties of one concept that characterize the concept.

Definition 3: A:=(name, syn-set), attribute is defined with a name and a set of synonyms.

Definition 4: R:=(name, syn-set, domain, range), each relationship is defined with a name, set of synonyms and its domain and range.

Definition 5: V:=(value), this feature is used for representation range of one relationship that is a value. One value Begins with one of these characters: "=", "<", ">" or "< >" and one string that show the value of its range.

Definition 6: O:=(G, G$'$), each ontology is represented by two graphs.

Definition 7: G:=(N,E), N=<C>, E=<is-a>, G is acyclic directed rooted graph that consists of nodes and edges. Each node is a concept (or instance of a concept). Each edge is *"is-a"* relation that shows sub-concept (subclass) relation between nodes. *G* is a hierarchy concept model of ontology.

Manuscript received October 13, 2012; revised December 19, 2012.

The authors are with Faculty of Computer Science Taylors University, Selangor, Malaysia (e-mail: Abdolreza.Moosa@taylors.edu.my, sameem@um.edu.my)

DOI: 10.7763/IJMO.2013.V3.224

Each node has one father and may have none, one or more child nodes. If one node has two fathers, the model resolves this problem with repeating child node for each one of its fathers.

Definition 8: $G^{'}:=(N,E^{'})$, $N=<C,V>$, $E^{'}=<R>$, $G^{'}$ is cyclic graph that consists of nodes and edges. Each node is a concept (or instance of a concept) or one value. Each edge is a relationship between two nodes that show the relationship between concepts. G' is a concept relationship model of ontology.

In our uniform representation model, all elements (concepts, attributes, relationships and values) are string (chain of characters). Our representation model of ontology is very general, so that our proposed approach which uses this formalization will work with any ontology representation languages. This representation model represents the main exploitable information in ontology.

III. Ontology Creation Methodology

There are approaches for creation of ontologies that have been proposed by other researchers. Some of these approaches in ontology creation domain are [3], [4], [9], [10]. Our approach was motivated by some ideas of the above approaches. We refer the reader to [3] for a discussion on ontology creation approaches and problems related to them.

As mentioned in [10] there are some fundamental rules in ontology design that we should pay attention to:

There is no one correct way to model a domain, there are always viable alternatives.

Ontology development is necessarily an iterative process.

Concepts in the ontology should be close to objects (physical or logical) and relationships in your domain of interest.

We will certainly need to revise the initial ontology. This process of iterative design will likely continue through the entire lifecycle of the ontology. We suggest the following phases for creation of ontologies based on our proposed ontology representation model (shown in Fig. 1).

Phase 1. Determination of domain and Scope of ontology : We start the development of an ontology by clarifying its domain and scope. That is, answer several basic questions [10]:

- What is the domain that the ontology will cover?
- For what purpose are we going to use the ontology?
- For what types of questions should information in the ontology provide answers to?
- Who will use and maintain the ontology?

We should stay away from application domain requirements and try to be as general as possible. This is due to the fact that ontologies describe universal intension of terms for a community.

Phase 2. Concept (or instance) extraction: Based on the defined scope we extract relevant concepts from the information sources in the context.

Phase 3. Concept hierarchy creation: we arrange concepts and instances in a taxonomic (sub-concept, super-concept) hierarchy. If a concept A is a super-class of concept B, then every instance of B is also an instance of A In other words, the concept B represents a concept that is a "kind of" A. There is no single correct concept hierarchy for any given domain. The hierarchy depends on the possible uses of the ontology, the level of the detail that is necessary for the application, personal preferences, and sometimes requirements for compatibility with other models [10].

Phase 4. Attribute and relationship extraction: Based on the defined scope we extract relevant attributes for each concept and existing relationships between concepts from the information sources in the context.

Phase 5. Key-property determination: we determine the key properties for each concept chosen from its attributes and relationships. Key properties define role and main semantic of concept. For example "teaching" is key property for the "lecturer" concept that shows essence and semantic of "lecturer".

Phase 6. Synonym determination: we specify synonyms of each concept, attribute and relationship. Synonym of each term is other name that is used by other communities for representing the same term.

Phase 7. Modeling: we represent the extracted terms in an ontology representation model. We use an edit ontology tool for this purpose.

Phase 8. Implementation: Finally we implement and store the uniform representation model of ontology in a relational data base.

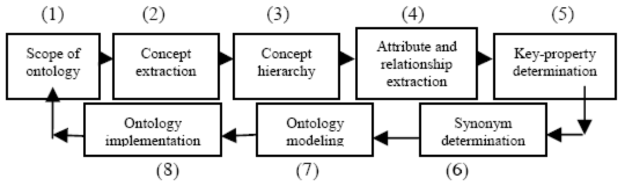

Fig. 1. Ontology creation steps.

A practical question often asked is, "Whose role is it to build ontology?" The person building ontology should have a good understanding of the vocabulary and the conceptualization of the domain. Such knowledge helps practically to ensure the accordance of ontologies with the community's conceptualization as a measure of quality for ontologies [2].

Case Study: Creating of University Ontology

We follow the above methodology for building an university ontology.

Phase 1: Domain and Scope clarification. Scope of our domain ontology is university.

Phase 2: Concept (or instance) extraction. For extraction of existing concepts in the university we need to investigate and study conceptualization of this domain in detail. We exploit some information sources for this purpose such as knowledge of people, who there are in university (such as staff, students), university websites, university specific vocabularies and other university ontologies (such as SHOE university ontology).

Phase 3: Concept hierarchy creation: We create concept hierarchy from extracted concepts in a taxonomic form (Fig. 2).

Phase 4: Attribute and relationship extraction: We determine attributes of each extracted concept and relationships between concepts.

Phase 5: Key-property determination: We exploit dictionaries to specify key properties of each concept. These key properties show main meaning and semantic of each concept which characterizes a concept from others. We can discover them in the definition of expressions of each concept in the dictionary.

Phase 6: synonym determination. We exploit dictionaries and universities websites to specify synonyms of each extracted term (concepts, attributes and relationships).

In the following section we present results of the above mentioned phases. The first we present created concept hierarchy in a taxonomy form. We present extraction attributes for each concept inside [brackets], synonyms inside {braces} and key properties inside (parentheses). Afterwards, we present relationship between concepts and the set of synonyms of attributes and relationships.

University *{University college} [home-page, name, country]*

College {Faculty, School, Academic-Department} [home-page, name]

Department *{Academic-Department} [home-page, name]*

Laboratory *{Lab} [home-page, name]*

Staff *{employee, worker, people, educational-employee} [home-page]*

Academic-staff *{Faculty, Academic-people, faculty-member, academician, academic} [home-page]*

Professor *{Prof} [URI] (key-relationship: teach)*

Assistant-Professor *{Assist-Prof} [name, Email, research-interest, home-page]*

Associate-Professor *{Assoc-Prof} [name, Email, research-interest, home-page]*

Full-Professor *{Professor, Prof} [name, Email, research-interest, home-page]*

Visiting-Professor *{Adjunct Professor} [name, Email, research-interest, home-page]*

Lecturer *(key-relationship: teach)*

Fulltime-Lecturer *{Lecturer, Senior-lecturer} [name, Email, research-interest, home-page]*

Part-time-Lecturer *[name, Email, research-interest, home-page]*

Visiting-Lecturer *{Adjunct Lecturer} [name, Email, research-interest, home-page]*

Gusting-Lecturer {Adjunct Lecturer} [name, Email, research-interest, home-page]

Post-Doctor {Postdoctoral, Post-doctorate} [name, Email, research-interest, home-page] (key-relationship: teach)

Tutor {instructor, trainer, educator} [name, Email, home-page] (key-relationship: teach)

Research-Assistant {RA, research-fellow}[name, Email, research-field, home-page](key-relationship: research-field)

Teaching-Assistant {TA} [name, Email, home-page]

Researcher {research-worker, scientist} [name, Email, Research field, home-page] (key-relationship: research-field)

Administrative-Staff {Non-Academic-Staff, management-staff}

University-Dean {Chair} [name, Email, home-page]

Deputy-Dean {dean- assistant}

Academic-Deputy-Dean {academic-dean-assistant, deputy-dean} [name, Email, home-page]

Development-Deputy-Dean {development-dean assistant, deputy-dean} [name, Email, home-page]

Head-Department [name, Email, home-page, Department-name]

Head-Library [name, Email, home-page]

Administrative-Assistant {administrator}

Assistant-Registrar [name, Email]

Clerical-Assistant [name, Email]

Official-Assistant [name, Email]

Student {current-student, prospective student}[home-page] (key-relationship: study-in)

Undergraduate-student

Diploma-Student [home-page]

Bachelor-Student {degree student}[home-page]

Postgraduate-Student *{Graduate-Student}*

Master-Student *[home-page]*

Doctorate-Student *{PhD-Student} [home-page]*

Postdoctoral-Student *{Post-doc-student} [Home-page]*

No-graduating-Student *[home-page]*

Exchanging-Student *[home-page]*

Research-group *{Research} [home-page, Research-field] (key-relationship: Research-field)*

Project *[Project-Title, home-page] (key-relationship: Project-Title)*

Center *{unit} [Name, home-page]*

Institute *{academic centre}[Name, home-page]*

Program *{Degree-Program, Academic-program, and Educational-Degree} [home-page] (key-relationship: list-of-course)*

Undergraduate-Program *[web-page]*

Diploma-Program *[web-page, Field]*

Bachelor-Program *[web-page, Field]*

Postgraduate-Program *{Graduate-Program} [web-page]*

Master-Program *{Master-of-science} [web-page, Field]*

Doctorate-Program *{PhD, Doctor-of-philosophy} [web-page, Field]*

Postdoctoral-Program *{postdoctoral-position, postdoctoral –fellowship}[web-page, Research-field]*

No-graduating-Program *[web-page, Field]*

Student-Exchanging-program *[web-page, Field]*

Admission *{academic-admission}[home-page]*

Undergraduate-Admission *{degree-admission}[web-page, Field, Requirement, Admission-date] (key-attribute: requirement)*

Postgraduate-Admission *[web-page, Field, Requirement, Admission-date] (key-attribute: requirement)*

Tuition-Fee *{admission-fee, fee, charge, fee-structure} [home-page]*

Diploma-Fee {*diploma- charge} [web-page, amount] (key-attribute: amount)*

Bachelor-Fee {*bachelor- charge} [web-page, amount] (key-attribute: amount)*

Master-Fee {master- *charge} [web-page, amount] (key-attribute: amount)*

Doctorate-Fee *{PhD- charge} [web-page, amount] (key-attribute: amount)*

Financial-Aid *{financial-assistance}[home-page]*

Undergraduate-Financial-Aid *[web-page]*

Scholarship *[web-page, amount] (key-attribute: amount)*

Fellowship *[web-page, amount] (key-attribute: amount)*

Loan *[web-page, amount] (key-attribute: amount)*

Postgraduate-Financial-Aid *{Graduate-Financial-Aid}*

[web-page]

Scholarship *[web-page, amount] (key-attribute: amount)*

Fellowship *[web-page, amount] (key-attribute: amount)*

Loan *[web-page, amount] (key-attribute: amount)*

Teaching-Assistantship *[web-page, amount] (key-attribute: amount)*

Research-Assistantship *[web-page, amount] (key-attribute: amount)*

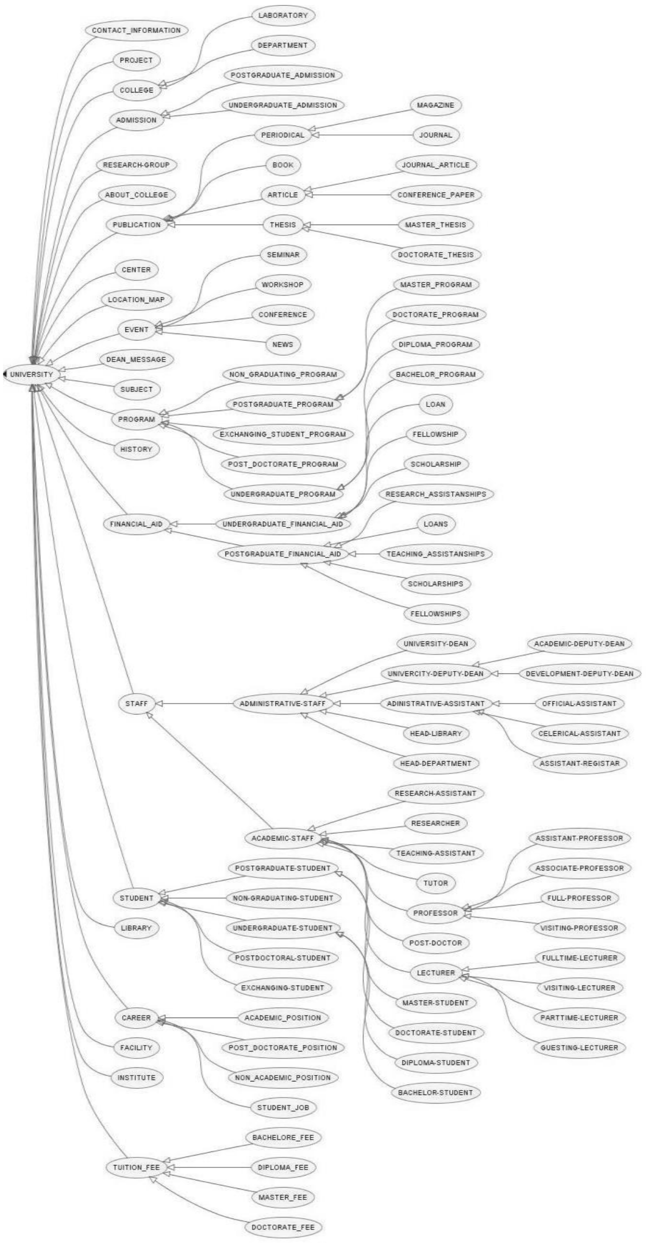

Fig. 2. Concept hierarchy diagram.

About-university {Overview, about-us, about-college} [home-page]

Dean-Message {Chair-message, mission} [home-page History [home-page]

Map-Location {Place,get-there, how-get} [home-page]

Contact-Info {contact, phone-directory}[home-page, address, Phone, fax, Email] (key-attribute: address, phone)

Publication {research-publication}[home-page] Article {paper}

Journal-Article {Journal-publication, Journal-paper} [web-page, Title, publish-date, Author] (key-attribute: author)

Conference-Paper {conference-proceeding} [web-page, Title, publish-date, Author] (key-attribute: author)

Book [web-page, Title, publish-date, Author] (key-attribute: author)

Periodical {journal}

Journal [web-page, Title, publish-date, Author] (key-attribute: author)

Magazine [web-page, Title, publish-date, Author] (key-attribute: author)

Thesis {dissertation}

Doctoral-Thesis {PhD-thesis} [web-page, Title, publish-date, Author] (key-attribute: author)

Masters-Thesis [web-page, Title, publish-date, Author] (key-attribute: author)

Event {news, event-news}[home-page]

News [web-page, Description, Event-date]

Conference [web-page, Description, Event-date]

Seminar [web-page, Description, Event-date]

Workshop [web-page, Description, Event-date]

Career {employment, recruitment, job-opportunity, vocation} [home-page] (key-attribute: position)

Academic-Position {Educational-position, lecturer-position}} [web-page, Position, requirement, deadline]

Postdoctoral-Position [web-page, Position, requirement, deadline] (key-attribute: Position)

Non-Academic-Position {staff-position} [web-page, Position, requirement, deadline] (key-attribute: Position)

Student-job-Position [web-page, Position, requirement, deadline] (key-attribute: Position)

Facility {service}[home-page]

Library [home-page]

Subject {course} [web-page, title]

Concept relationships:

Advisor (Academic-staff, Student)

Org-Publication (center, Publication)

Org-Publication (Institute, Publication)

Research-Publication (Research, Publication)

Project-Publication (Project, Publication)

Head (Center, Staff)

Head (Institute, Staff)

Head (Department, Staff)

Work (Staff, Department)

Teach (Academic-staff, Subject)

Member (Center, Staff)

Member (Institute, Staff)

Member (Department, Staff)

Member (Research, Staff)

Member (Project, Staff)

Offers (Department, Program)

List-of-course (program, Subject)

Study-in (student, program)

The synonym sets of attributes:

Webpage: {URI, Web-address, URL, homepage}

Name: {Full-name, first-name, last-name, surname, given-name}

Email: {electronic-mail, mail}

Phone: {telephone, hand-phone, phone-number, Contact-number}

Research-interest: {research-field, research-area}
Homepage: {personal-webpage, personal-homepage, personal-page, URI, URL}
Title: {subject}
Field: {area, major, program}
Research-field: {Research-area}
Requirement: {qualification}
Admission-date: {date, deadline, start-date, end-date}
Amount: {fee}
Address: {Permanent-address, Postal-address, Add.}
Publish-date: {Pub-date, date, year}
Author: {writer, Published-by, Publisher}
Position: {post, job, Job-title}
Event-date: {start-date, date, deadline,important-date}
Description: {Description}
Project-Title: {title, subject}
Language: {}

The synonym sets of relationships:

Advisor: {supervisor}
Org-Publication: {organization-publication}
Research-Publication: {}
Project-Publication: {}
Head: {manager, chair}
Work: {employ}
Teach: {educate}
Member: {work, team}
Offers: {program}
List-of-course: {list-of-subject, course-material, curricula}

Phase 7: Modeling: In this step we design a concept hierarchy and relation concept graphs by one ontology-editing tool. In this phase, we use Protégé-2000 for our example. Protégé-2000 was developed by Mark Musen's group at Stanford Medical Informatics (shown in Fig. 2).

Phase 8: Implementation. Finally we implement and store a uniform representation models (hierarchy concept and relation concept graphs) of ontology in SQL/SERVER DBMS. We illustrate the required tables for storage of ontologies in the uniform representation model form as follows: (In all tables first columns are primary key)

IV. Domain-Based Ontology Architecture

In all ontology–based integration approaches, the ontologies are used for the explicit description of the semantics of information sources. But there are different ways of how to employ the ontologies. In general, four different directions can be identified: single ontology approaches, multiple ontologies Approaches, Top-level ontology approaches and Shared vocabulary approaches [11].

A single ontology approach uses one global ontology providing a shared vocabulary for the specification of the semantics (shown in Fig. 3). All information sources are related to the global ontology. A prominent approach of this kind of ontology integration is SIMS [12]. The single ontology approach can be applied to integration problems where all information sources to be integrated provide nearly the same view on a domain. If one information source has a different view on a domain, finding the minimal ontology commitment becomes a difficult task [13]. These disadvantages led to the development of the multiple ontology approach [11].

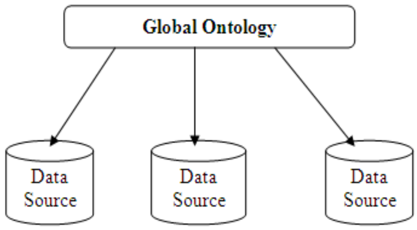

Fig. 3. Single ontology architecture from [11]

In a multiple ontology approach, each information source is described by its own ontology (shown in Fig. 4). For example, in OBSERVER [14] the semantics of an information source is described by a separate ontology. At a first glance, the advantage of a multiple ontology approach seems to be that no common and minimal ontology commitment about the global ontology is needed [13]. This ontology architecture can simplify the change, i.e. modifications in one information source or the addition and removal of sources. But in reality the lack of a common vocabulary makes it difficult to compare different source ontologies.

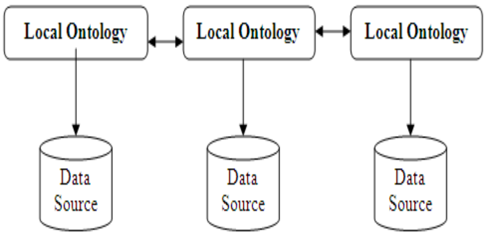

Fig. 4. Multiple ontology architecture from [11]

To overcome the drawbacks of the single and multiple ontology approach, top-level ontology approaches were developed (shown in Fig. 5). Similar to the multiple ontology approach, the semantics of each source is described by its own separate ontology. In order to perform a suitable matching between ontologies we need a basic agreement between application ontologies. Such agreement is made by means of higher-level ontologies. An ontology for a large number of communities cannot be complete or highly specialized. It is difficult to reach a consensus within the community or between communities over detailed definitions in the ontology. A community can adopt a higher-level ontology and specialize it by adding its own definitions to it. As a result, a specialized ontology cannot remove any constraints or terms of a higher-level ontology without the agreement of the communities already committed to that ontology. The advantage of a top-level ontology approach is that new sources can easily be added without the need to modify the mappings or the top-level ontology. It also supports the acquisition and evolution of ontologies. The use of a top-level ontology makes the source ontologies comparable and avoids the disadvantages of multiple ontology approaches. The drawback of a top-level ontology approach, however, is that, existing ontologies cannot be reused easily, but has to be re-developed from scratch, because all source ontologies have to refer to the top-level ontology [11].

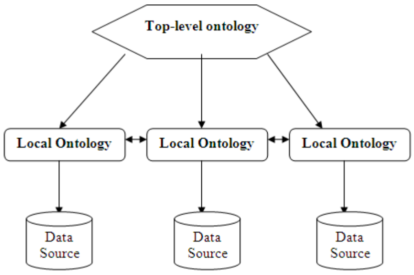

Fig. 5. Top-level ontology architecture

To overcome the drawbacks of the above ontology approaches, a shared vocabulary approach was developed (shown in Fig. 6). Similar to the multiple ontology approach the semantics of each source is described by its own ontology. However, in order to make the source ontologies comparable to each other they are built upon one global shared vocabulary [15], [16]. The shared vocabulary contains basic terms (the primitives) of a domain. In order to build complex terms of the source ontologies the primitive ones are combined by some operators. Since each term of the source ontology is based on the primitives, the terms become easier compared to the multiple ontology approaches [11], [16].

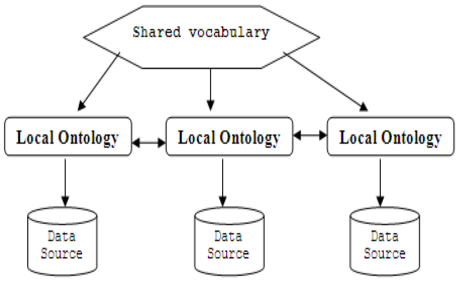

Fig. 6. Shared vocabulary architecture from [11]

TABLE I: BENEFITS AND DRAWBACKS OF THE DIFFERENT ONTOLOGY-BASED INTEGRATION APPROACHES FROM [11]

	Single ontology architecture	**Multiple ontologies architecture**	**Top-level ontology approach**	**Shared vocabulary approach**
Implementation effort	Straight forward	Costly	Reasonable	Reasonable
Semantic heterogeneity	Similar view of domain	Support heterogeneous views	Support heterogeneous views	Support heterogeneous view
Add/remove sources	Need for some adoption in global ontology	Providing a new source ontology, relating to other ontologies	Providing a new source ontology	Providing a new source ontology
Comparing of multiple ontologies	----------	Difficult, because of the lack of a common vocabulary	Simple, because ontologies inherit from a top-ontology	Easier, because ontologism use a common ontology

The advantage of a shared vocabulary approach is that new sources can easily be added without the need to modify the mappings or the shared vocabulary. It also supports the acquisition and evolution of ontologies. The use of a shared vocabulary makes the source ontologies comparable and avoids the disadvantages of the multiple ontology approach. The drawback of a shared vocabulary approach however, is that existing ontologies cannot be reused easily, and has to be re-developed from scratch, because all source ontologies have to refer to the shared vocabulary [11]. In the shared vocabulary approach the interesting point is how the local ontologies are described, i.e. how the terms of the source ontology are described by the primitives of the shared vocabulary.

Table I summarizes the benefits and drawbacks of the different ontology mapping approaches. It's impossible to use one global ontology, shared vocabulary or common top-level ontology for a large and dynamic environment like web because many data sources are involved and the number of involved data sources change.

Users are free to use their own terms and vocabulary and the schemata are subject to frequent changes. Therefore the multiple ontologies architecture is the most appropriate approach for ontology-based Integration of data sources in a large and dynamic environment. Each source ontology could be developed without considering other sources or their ontologies — no common ontology with the agreement of all sources is required. However, in reality the lack of a common vocabulary makes it extremely difficult to compare different source ontologies.

We introduce domain-based ontology approach to overcome the drawbacks of the above approaches (shown in Fig. 7). Similar to the multiple ontology approach the semantics of each source is described by its own ontology. However, in order to make the source ontologies comparable to each other they are mapped to domain-based ontology. Domain-based ontology is used as a global ontology. The domain ontology covers the semantic definition of terms which are required for user queries in a particular application domain. The domain-based ontology is built and modelled base on system uniform representation model and local ontologies are required to transform to system uniform representation model before mapping process. The advantage of a domain-based ontology approach is that new sources can easily be added without the need to modify the mappings or the domain ontology. The use of a domain ontology makes the source ontologies comparable and avoids the disadvantages of the multiple ontology approach. The drawback of a domain-based ontology approach however, is that local ontologies has to be in same domain because all source ontologies have to be mapped to the domain ontology.

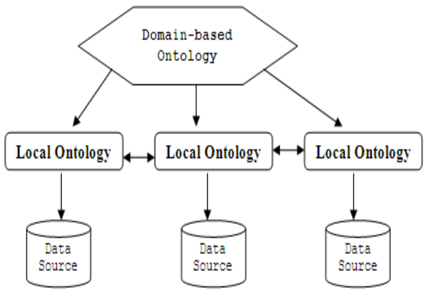

Fig. 7. Domain-based ontology architecture

V. CONCLUSION

In this paper, we recommended one development methodology based on our proposed representation model for the creation of ontologies. Subsequently we built a University ontology based on proposed methodology and representation model. Finally we introduced domain-based ontology architecture as efficient ontology architecture in data integration system.

REFERENCES

[1] D. Fensel, "Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce," *Springer-Verlag*, 2001.

[2] F. Hakimpour and A. Geppert, "Resolving semantic heterogeneity in schema integration: An ontology base approach," in *C. Welty and B. Smith,* editors, *Formal Ontology in Information Systems: Collected Papers from the Second Int'l Conf, FOIS'01*, pp. 297–308. ACM Press, 2001.

[3] F. Lopez, "Overview of methodologies for building ontologies," in *Proc. the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods: Lessons Learned and Future Trends. CEUR Publication*, 1999.

[4] M. Uschold, "Building ontologies: Towards a unified methodology," in *Proc. 16th Annual Conf. of the British Computer Society Specialist Group on Expert Systems*, Cambridge, UK, 1996.

[5] Protégé. (2000). *The Protege Project*. [Online]. Available: http://protege.stanford.edu

[6] A. Farquhar. (1997). *Ontolingua tutorial*. [Online]. Available: http://ksl-web.stanford.edu/people/axf/tutorial.pdf

[7] Chimaera. (2000). *Chimaera Ontology Environment*. [Online]. Available: http://www.ksl.stanford.edu/software/chimaera

[8] A. Hajmoosaei and S. Abdul-Kareem, "An ontology-based approach for resolving semantic schema conflicts in the extraction and integration of query-based information from heterogeneous web data sources," To be published in the Australian Computer Society as vol. 85, ISBN 1-920-68266-X, in *the CRPIT Series*. Australia.

[9] M. Uschold and M. King, "Towards a methodology for building ontologies," in *Workshop on Basic Ontological Issues in Knowledge Sharing*, held in conduction with IJCAI-95, 1995.

[10] F. N. Noy and L. D. McGuinness, *Ontology development 101: A guide to creating your first ontology*, 2000.

[11] H. Wache, T. V ̈ogele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hubner, "Ontology-Based Integration of Information — A Survey of Existing Approaches Intelligent Systems Group," Center for Computing Technologies, University of Bremen, P.O.B. 33 04 40, D-28334 Bremen, Germany.

[12] Y. Arens, Y. Ciiee, and A. Knoblock, "SIMS: Integrating Heterogeneous, Distributed Information Sources," in *Proc. the 1997 ONDCP/CTAC International Technology Symposium*, Office of National Drug Control Policy, Executive Office of the President. Chicago, IL. August 18-21, 1997.

[13] T. Gruber, *Toward principles for the design of ontologies used for knowledge sharing*, 1995.

[14] E. Mena, A. Illarramendi, V. Kashyap, and A. Sheth, "OBSERVER: An Approach for Query Processing in Global Information Systems based on operation across Pre-existing Ontologies," *International Journal on Distributed and Parallel Databases*, vol. 8, no. 2, pp. 223-271, 2000.

[15] C. H. Goh, "Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Sources," Phd, MIT, 1997.

[16] H. Stuckenschmidt, H. Wache, T. Vogele, and U. Visser., "Enabling technologies for interoperability," in Ubbo Visser and Hardy Pundt, editors, *Workshop on the 14th International Symposium of Computer Science for Environmental Protection*, pp. 35–46, Bonn, Germany, 2000, TZI, University of Bremen.