Journal of Southeast University (English Edition) Vol. 25, No. 4, pp. 473 – 476 Dec. 2009 ISSN 1003—7985

Approach for knowledge sharing based on ontology context immigration

Ma Yinglong[1] Zhang Hongxian[1] Liu Shaohua[2]

([1] School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, China)
([2] School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: In order to better achieve knowledge sharing based on distributed ontologies, an approach based on ontology context immigration (OCI) is proposed. Compared with traditional approaches such as ontology integration and mapping, the proposed approach can reduce the implementation complexity. This approach can be mainly divided into three phases: ontology context determination for a given term, ontology semantic similarity computation between ontology terms, and ontology context immigration. As for a local semantic term based on distributed ontologies, an appropriate ontology context of the term is determined and extracted from a local ontology most associated with the term by using semantic similarity computation. Then, the ontology context is dynamically immigrated to the source ontology for enriching semantic information related to the term. A system called distributed knowledge sharing system (DKSS) is developed to illustrate this approach. The system adopts multi-agent technology for better communication and coordination between different ontology information sources. The experimental results show that it is efficient for distributed ontology knowledge sharing. The proposed approach does not require the support of a global ontology or the maintenance of complex ontology mapping relations, and thus it has better maintainability and scalability.

Key words: ontology; knowledge sharing; ontology context; multi-agent system

Knowledge sharing of distributed ontologies in open environments has attracted much attention in recent. As the core of the semantic web[1], ontologies have been adopted as the conceptual backbone of enterprise application requirements. There is a tendency that distributed ontologies will become more dynamic rather than static because of requirement changes. It is inevitable to lead to a large number of semantic heterogeneities between distributed ontologies. This will impede knowledge sharing of distributed ontologies. For efficient knowledge sharing of distributed ontologies, ontology integration and mapping methods are widely adopted[2-5]. The ontology integration method commonly depends on a global ontology that contains the reproduction of semantic terms of all distributed ontologies. This method is time-consuming and manually made. However, as the global ontology must be correspondingly updated and maintained in accord with the changes of local ontologies, it will lead to poor maintainability and scalability of the whole distributed knowledge system. In contrast, ontology mapping does not require the global ontology but needs to create semantic mapping relations between different semantic terms. This method thoroughly considers the dynamic characteristics of distributed ontologies, but it also has to maintain complex ontology mapping relations. Furthermore, the semantic query process based on ontology mapping will greatly increase the reasoning complexity of the distributed knowledge system.

This paper proposes a novel approach for distributed ontology knowledge sharing based on ontology context immigration (OCI). When there are semantic terms requiring semantic context in a local ontology, this approach first calculates the semantic similarities between them and the terms in other local ontologies, and creates ontology mapping relations between associated semantic terms. In order to achieve ontology interoperability between two local ontologies, the ontology context (OC) of a given term will be determined and extracted from a local ontology most associated with the term. Then, these OCs will be dynamically immigrated to the source ontology for enriching semantic information related to the term. A system called the distributed knowledge sharing system (DKSS) is developed to illustrate this approach.

1 Ontology Knowledge Representation

Ontology was introduced into computer science by the artificial intelligence community to describe data and knowledge representation models which are conceptually independent of specific applications. The ontology representations in some web ontology languages, such as RDF and OWL, can be translated into a set of triples of the form (T, R, T'), where T and T' represent the terms (e. g., classes, properties or individuals), and R represents a binary relation between terms. Ontology languages provide some entailment rules for semantic reasoning. In addition, some toolkits such as Jena[6] and query languages such as RDQL[7] can be used to program ontology applications and perform semantic queries based on these applications.

2 Ontology Context Immigration

2.1 OCI elements

In distributed environments, domain ontologies are commonly developed separately by different developers, and each of them describes a partial knowledge of the whole domain of interest. In order to achieve ontology knowledge sharing based on the whole distributed system, these ontologies must cooperate for a given information query. Specifically speaking, we need to eliminate the semantic heterogeneities between terms of ontologies as much as possible, and

Received 2009-06-30.

Biography: Ma Yinglong (1976—), male, doctor, associate professor, yinglongma@ gmail. com.

Foundation item: The National Natural Science Foundation of China (No. 60703036).

Citation: Ma Yinglong, Zhang Hongxian, Liu Shaohua. Approach for knowledge sharing based on ontology context immigration [J]. Journal of Southeast University (English Edition), 2009, 25(4): 473 – 476.

make the shared knowledge more relevant. In this case, we propose an approach called ontology context immigration. The ontological context of a given term is the set of elements of all possible superclasses (superproperties), subclasses (subproperties) and individuals most semantically related to the term, which can be generally regarded as a subontology of the ontology in which the term is defined.

Fig. 1 shows three domain ontologies (i. e., ontologies A, B and C) that are respectively maintained in three different agents. Ontology A imports semantic term B4 originally defined in ontology B. Ontology B imports term A4 originally defined in ontology A. Ontology C imports terms A3 and B2 that are defined in ontology A and ontology B, respectively. In ontology A, the imported term B4 has no more semantic information. Term B4 has more semantic information hidden in its super- or sub-relations in ontologies A and B. Therefore, based on ontology A, if an agent executes a semantic query containing term B4, then the agent probably needs to know more information about B4. Thus, it will request the OC of B4 from ontology B. According to the request, the agent of ontology B will extract the OC of B4 from ontology B, and further immigrate the information to the agent of ontology A.

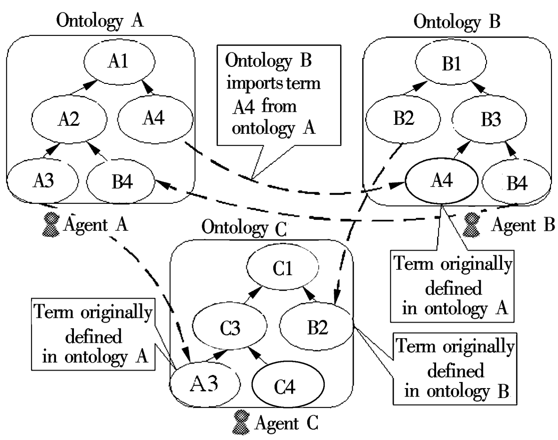

Fig. 1 Knowledge sharing of ontologies

In addition, each agent in the distributed system not only can request remote OC information but can also respond and provide its own ontology knowledge to other remote agents. Moreover, the interaction among agents in the whole distributed system does not depend on any middle entity (e. g., a mediation server). That is, each agent can request OC information from other agents. At the same time, it also can make responses to OC requests of other agents and immigrate the corresponding OC information to other agents. Considering such a system architecture, to add an agent into or remove it from the distributed system will not cause failures of the whole system. Therefore, this system has better capabilities of maintainability and scalability.

2.2 Ontology context determination

In order to determine the ontology context of a given term, some issues should be considered. If the term has been defined in some ontology, we directly extract and obtain the semantic context of the term from the ontology. If it is not explicitly defined in other ontologies, we need to compute the semantic similarities[8] between the given term and the terms in other ontologies. We will find the most semantically relevant term for the given term. Then we can establish the mapping relationship between them. The ontological context of the most relevant term will be extracted and further immigrated to the agent maintaining the given term.

2.2.1 Semantic similarity computation

As described above, when a semantic query refers to external terms and wants to know more semantic information about these terms, the immigration of the OC will be automatically performed. However, there probably exists a large number of semantic heterogeneities among distributed ontologies. For example, a semantic term "cat" may be defined as "pussycat" in another ontology, "coach" and "drillmaster" may be used in different domain ontologies for the same or similar meanings. In this situation, if a semantic query involves the external term "cat" but cannot find the exact matching term in other ontologies, it is necessary for the query agent to import the OC of "pussycat" so as that the system can also achieve ideal query answering. Because the semantic term of "cat" and "pussycat" has higher similarity, it is obvious that more satisfiable query results will be returned if the term "cat" is replaced with "pussycat", in addition to the ontology context of term. In order to calculate the similarity between semantic terms, we adopt two kinds of similarity methods: syntax similarity and semantic similarity. Syntax similarity is defined as the string matching degree of term names. This paper adopts the Levenshtein method[9] for syntax similarity calculations. Semantic similarity is defined as the semantic matching degree of different terms. Semantic similarity can be measured by semantic distance between different semantic terms. In this paper, we first use the WordNet to calculate the semantic distance between two adjacent nodes. The semantic similarity between any two nodes is defined as follows:

$$\mathrm{SemSim}(c_i,\ c_j) = \sum_{e \in p(c_i,\ c_j)} w_e + \frac{D_{c_i} + D_{c_j}}{D_{c_i} + D_{c_j} + 2D_{\mathrm{LCA}}}$$

where $\sum_{e \in p(c_i,\ c_j)} w_e$ represents the sum of weighed distances in the shortest path between nodes c_i and c_j, and $p(c_i,\ c_j)$ refers the shortest path between nodes c_i and c_j. D_{c_i} and D_{c_j} represent the weighed distance of c_i and c_j to their shortest common ancestor, respectively. D_{LCA} represents the weighed distance from this shortest common ancestor to the root. Relying on semantic distance, we define the semantic similarity. The ultimate similarity is as follows:

$$\mathrm{Sim}(c_i,\ c_j) = k\mathrm{SynSim}(c_i,\ c_j) + (1-k)\,\mathrm{SemSim}(c_i,\ c_j)$$

where $\mathrm{SynSim}(c_i,\ c_j)$ is the syntax similarity calculated by the Levenshtein method. k is a coefficient in [0, 1], which can be used to appropriately adjust the weight of syntax similarity and semantic similarity. This paper takes $k = 1/3$ for similarity computation.

2.2.2 OC immigration

By similarity computation, the system can figure out which is the most similar term in distributed ontologies. The following task is to determine the OC of the most similar term. We take Fig. 2 as an example to illustrate our approach.

In Fig. 2, ontology B imports term A4 that is originally defined in ontology A(or A4 in ontology B is an external term and is most similar to the term A4 defined in ontology B by similarity computation). At this time, the super terms of A4(i. e., A2, A6 and A1), the sub terms of A4(i. e., A5 and A7), and the relationships among them, as well as the total semantic information about A4, will be used to determine the semantic context of A4. Specifically speaking, if a local semantic query based on ontology B involves a term A4 that is directly imported from other ontologies or an external term obtained by similarity computation, then the ontology context of the term will be dynamically extracted from ontology A and immigrated to agent B. The enriched ontology B will help users find more relevant semantic results.

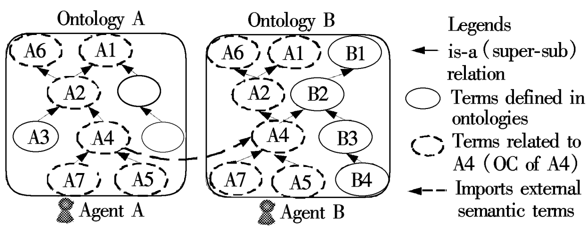

Fig. 2 Ontology context

2. 3 System architecture

The prototype of the DKSS based on ontology immigration is developed by using multi-agent technology. Fig. 3 shows a skeleton of the system architecture. The system currently consists of three agents. Of course, it can be extended and probably contains more ontology agents. In the current DKSS system, each agent contains three key components: LocalSemanticQuery, SimilarityComputation and OCExtraction. Component LocalSemanticQuery executes local semantic query and reasoning. When the query involves external semantic terms and needs to immigrate the OC of those terms, component SimilarityComputation will be invoked to figure out which semantic terms in the distributed system are the most similar ones. The subcomponent RequestModule will be invoked to request the OC of those terms from remote agents. The subcomponent ResponseModule responds to the remote request. Component OCExtraction extracts certain OC depending on the requests. In the end, the subcomponent ResponseModule will sends these OCs to the remote request agents. For example, a semantic query based on agent B may refer to some external terms that are most similar to those terms defined in agent A or agent C. At this time, SimilarityComputation of agent B will be invoked to compute the most similar terms in agent A and agent C, and then RequestModule will be executed to request OCs of these similar terms. If agent B and agent C contain the request information, their OCExtraction components will dynamically extract the OCs from local ontologies and send them to agent B by RequestModule. Agent A and agent C are similar.

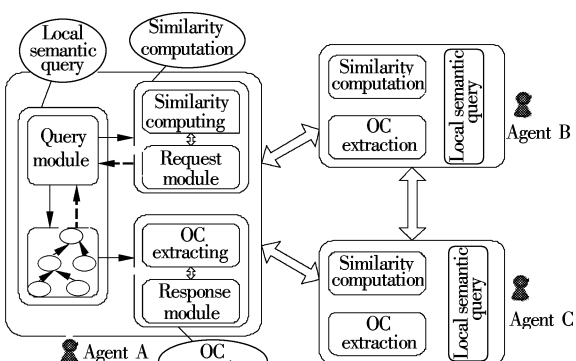

Fig. 3 Architecture of DKSS

3 Experiments and Evaluation

3. 1 Datasets and experimental index

Three domain ontologies such as Faulty, Department and Student are developed by using the Protégé tool[10] and further used as our experimental datasets. The Faculty ontology consists of 59 classes and many properties and individuals. It includes contents about the staff at a university such as title, departments that they work for, research areas, students and courses, etc. Ontologies Department and Student consist of more than 120 terms, and describes the semantic knowledge about departments and students, respectively.

In this paper, we apply three indices to our experimental evaluations. Precision, Recall and ResponseTime are defined as follows:

1) Precision

$$\text{Precision} = \frac{|\text{RelevantInformation} \cap \text{RetrievedInformation}|}{|\{\text{RetrievedInformation}\}|} \times 100\%$$

2) Recall

$$\text{Recall} = \frac{|\text{RetrievedInformation}|}{|\bigcup_{s_i \in S} \text{information}_{s_i}|} \times 100\%$$

where RetrievedInformation refers to the set of total information retrieved. RelevantInformation refers to the set of all relevant information existing in all given sources. RelevantInformation $\cap$ RetrievedInformation represents the set of relevant information retrieved. $\bigcup_{s_i \in S} \text{information}_{s_i}$ represents the summation of information contained in all datasets.

3) ResponseTime

The index ResponseTime is defined as the time interval from the beginning of semantic queries to the return of query results. Its time unit is in milliseconds. In fact, the ResponseTime in this paper can be regarded as the sum of network communication time, OC extraction and loading time, and semantic query transformation and semantic reasoning time.

3. 2 Data analysis and settings

The comparisons of Precision, Recall and ResponseTime between the OCI model and the non-OCI model are shown in Fig. 4 and Fig. 5. Where F, S and D represent datasets Faculty, Student and Department, respectively; FS, FD, SD and FSD represent the possible combinations of these three datasets.

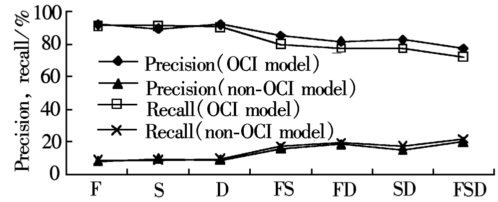

Fig. 4 Comparisons of precision and recall

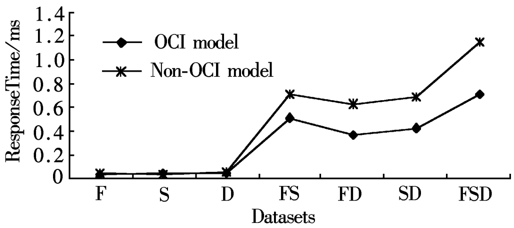

Fig. 5 Comparisons of ResponseTime

Fig. 4 shows the comparisons of precision and recall adopting the OCI model and the non-OCI model along with the increased volume of datasets. As for the precision, we find that precision values have a continual descent based on the two models. However, we observe that the precision values based on the non-OCI model has a smaller drop compared with that based on the OCI model. Furthermore, the precision values of the former are higher than those of the latter. As for the recall, we find that recall values have a continual ascent based on the two models. However, we observe that the recall value based on the OCI model has a smaller drop compared with that based on the non-OCI model. Furthermore, the recall values of the former are lower than those of the latter. The reasons are probably because immigration of ontology contexts causes more restrictive query conditions. In this situation, the retrieved results are relevantly reduced, which makes the query results more semantically relevant to a certain extent. In Fig. 5, we can observe an obvious increase in response time value. But relevantly speaking, the method adopting the OCI model is significantly lower than that adopting the non-OCI model with the increased volume of datasets. The reason is probably because the semantic queries based on the OCI model just need to load the OC information rather than the whole ontology information. This leads to a shorter response time for semantic queries.

4 Conclusion

In this paper, we propose a novel approach based on the OCI for knowledge sharing of distributed ontologies. A semantic context of a given term can be regarded as a subontology of the ontology in which the term is defined. By specifying an appropriate ontology context for the given term, we can eliminate semantic heterogeneity between relevant semantic terms of distributed ontologies. The preliminary experimental results show that this approach is effective and efficient for knowledge sharing of distributed ontologies. This approach does not require the support of a global ontology or the maintenance of complex ontology mapping relations, and thus has better maintainability and scalability.

References

[1] Berners-Lee T, Hendler J, Lassila O. The semantic web [J]. *Scientific American*, 2001, **284**(5): 34－43.
[2] Noy N F, Musen M A. The PROMPT suite: interactive tools for ontology merging and mapping [J]. *International Journal of Human-Computer Studies*, 2003, **59**(6): 983－1024.
[3] Tempich C, Pinto H S, Sure Y, et al. An argumentation ontology for distributed, loosely-controlled and evolving engineering processes of ontologies(DILIGENT) [C]//*Proceedings of ESWC*. Heraklion: ESWC Press, 2005: 241－256.
[4] Kozaki K, Kitamura Y, Mizoguchi R. *Hozo*: *an environment for building/using ontologies based on a fundamental consideration of "role" and "relationship"* [M]. Berlin: Springer, 2002: 155－163.
[5] Tang Jie, Li Juanzi, Liang Bangyong, et al. Using Bayesian decision for ontology mapping [J]. *Journal of Web Semantics*, 2006, **4**(4): 243－262.
[6] HP Lab. HP labs semantic web toolkit software and documentation [EB/OL]. (2008-06-12) [2009-04-01]. http: //jena. sourceforge. net/.
[7] Seaborne Andy. RDQL—a query language for RDF [EB/OL]. (2004-02-10) [2009-04-01]. http: //www. w3. org/tr/owl-ref/.
[8] Cross V. Fuzzy semantic distance measures between ontological concepts [C]//*Annu Conf North Am Fuzzy Inf Process Soc NAFIPS*. Banff, Alta, Canada, 2004, **2**: 635－640.
[9] Levenshtein V I. Binary codes capable of correcting deletions, insertions, and reversals [J]. *Soviet Physics Doklady*, 1966, **10**(8): 707－710.
[10] The Protégé Project. Protégé: an environment for knowledge-based systems development [EB/OL]. (2009-06-18) [2009-06-24]. http: //protege. stanford. edu/.

基于本体环境迁移的信息共享方法

马应龙[1]　张红宪[1]　刘韶华[2]

([1] 华北电力大学控制与计算机工程学院，北京 102206)
([2] 北京邮电大学电子工程学院，北京 100876)

摘要：为了更好地实现基于分布式本体的知识共享，提出了一个基于本体环境迁移(OCI)的方法．该方法同传统的本体集成和映射方法相比，能减少实现复杂度．该方法可以分成 3 个阶段：给定术语的语义环境确定，本体术语之间的语义相似度计算以及本体环境迁移．针对分布式本体的一个本地术语，其本体环境可以通过使用语义相似度计算从与该术语最相关的本地本体中确定和抽取．然后，该环境将被动态地迁移到源本体以获取更丰富的语义信息．采用多智能体技术，开发了一个分布式知识共享系统(DKSS)以演示该方法的使用．实验结果显示，该方法对分布式本体知识共享是有效的，不需要维护全局本体或复杂的本体映射，因此具有更好的可维护性和可伸缩性．

关键词：本体；知识共享；本体环境；多智能体系统
中图分类号：TP313