Journal of Southeast University (English Edition) Vol. 22, No. 3, pp. 375 – 380 Sept. 2006 ISSN 1003—7985

Extend OWL: toward a clustering integration of collaborative ontologies and agents

Xiao Ruliang[1,2] Tang Shengqun[1] Li Ling[1] Fang Lina[1] Xu Yang[1]
Xu Youwei[1] Deng Xinguo[1]

([1] State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, China)
([2] Department of Information and Management, Hunan Finance and Economics College, Changsha 410205, China)

Abstract: In order to solve the problem of how to collaborate with foreign agents and ontologies, a restricted clustering integration approach is proposed. It differs from the traditional approaches in which web ontology language (OWL) is extended by adding some new collaborative interfaces (i. e., agent-link and ontology-link) to it instead of owl: import. Syntaxes of the interface for foreign ontologies and foreign agents, respectively, and a meta-method of clustering integrated collaboration are discussed. The approach focuses on taking advantage of OWL itself to solve the collaborative problems, and it is feasible to track the contexts of new-added knowledge concerning ontological collaboration.

Key words: extended OWL; ontology; agents; collaboration; clustering integration

A large knowledge engineering application usually requires the development of many different ontologies to process their respective businesses, and many agents being correlated with the application need to communicate with these ontologies in order to fulfill their tasks. During the last decade, multi-agent systems have rapidly been growing, and they are now used in industrial contexts. The knowledge representation field and ontological application are also gaining momentum with the semantic web supported by the W3C. An ontology, that can be used to describe more complicated semantic structure than that of general data, is usually used to construct the systematic skeleton in a knowledge-based system, and it is suitable for describing heterogeneous and distributed information resources. Both the agent, as an entity with faith, goal and behavioral functions, and the ontology are unattached formalisms. The second generation web ontology language standard OWL[1] has no satisfactory solution to how agents collaborate with an ontology acting as a knowledge base formalism. It only provides an interface structure "owl: import" that allows an ontology to import other entities, where there are no restrictions and filters by means of using a reference URI.

In fact, their communication with each other is a collaborative process, which should be restricted, and not open completely. Usually, the environment changes unceasingly and the semantics and the structural style contained in the knowledge must also change unceasingly. Therefore, the ontology must not only be updated in time along with the changes in domain knowledge, but it also must adjust its cooperation unceasingly with others to aid in its evaluation in order to express the semantics context. Hence, there exists a crucial issue on how to integrate the collaboration between an ontology and much unattached formalism, whether the inferences among these entities are determined or not. But the standard OWL does not have a better solution. An interface "owl: import" is obviously insufficient and it is to solve the problem for heterogeneous entities.

In this paper, we propose a restricted ontological collaborative method with agents and other ontologies. It differs from the traditional approaches, in which OWL is extended by adding some new collaborative interfaces (i. e., agent-link and ontology-link) instead of using "owl: import" and so on. In order to achieve this kind of collaboration in an effective way, the extensive application of the semantic web is necessary. Considering that OWL-Full goes beyond the scope of the abstract description logic (ADL) which is the basis of OWL, our work is restricted to the area of OWL-Lite or OWL-DL. Three aspects of the approach are discussed: syntax of the interface for agents and foreign ontologies, respectively, and the meta-method of clustering integration while an ontology is collaborating with foreign agents and ontologies.

1 Related Work

For the collaboration among ontologies, much

Received 2006-04-20.

Biographies: Xiao Ruliang (1966—), male, graduate, xiaoruliang@163.com; Tang Shengqun (1946—), male, professor, taozhou@whu.edu.cn.

work has been done recently. It mainly includes three classes. First, the mutual operation is carried out by the mapping method between the distributed ontologies. Because each ontology is an independent formalism, the mapping relation is bridged between two unattached formalisms[2-3], between the two individuals, and between the concepts that belong to two different formalisms:

$$C_i \underset{\rightarrow}{\sqsubseteq} C_j, \quad C_i \underset{\rightarrow}{\sqsupseteq} C_j, \quad a_i \rightarrow b_j$$

where C_i, C_j are classes; a_i, b_j are individuals in the ontologies O_i, O_j, respectively. This kind of mapping supports the interoperation between two ontologies which is a solution from OWL-DDL based on OWL. Secondly, using function-restricted agents forms a mapping between two ontologies[4]. In general, this solution defines some agents such as the mapping agent (MA), the interface agent (IA), the similarity agent (SA), the ontology agent (OA), the query agent (QA). The OA answers for the relative tasks with the ontology, which is driven by different resources, can operate directly over the structure of an ontology and its mapping files, then the IA makes the certain application domain, after the relative information provided by OA and SA is obtained, MA begins to work, finally SA captures the understandable knowledge by querying. Thirdly, the method of ε-connections[5] builds a linked relationship among ontological individuals over corresponding no-intersectional domains, in which the ADL frame is used to integrate different formalisms. It suggests that the ontology should be the vertex in the direct graph, the link should be the direct edge, and they compose the complex network graph.

There exist some shortcomings that cannot be ignored in the mainstreams above. The expressive ability in the third class is much stronger than that in the first one based on OWL-DDL, but it is easy to lead undecidable state. And the greater costs of communication in the second one make it not worth employing extensively. And because the technology contained in the way of ε-connections causes complicated connections, it is difficult to build complicated multi-field practical modeling.

For the ontological collaboration with agents, much research has also been done recently. They are mainly divided into two classes. The first, the interoperation standard of intelligent physical agents recommended by the FIPA organization[6], which uses a so-called OKBC (open knowledge base connectivity) as a complete open and no restricted knowledge base, supports the heterogeneous agents to communicate with each other, and achieves the ontological cooperation with agents in the sight of its semantic or functional sense based on ACL (agent communication language by FIPA). Zhang et al.[7] presented a method that holds the two characters above to construct ontology using FIPA ACL. As for the second, people usually implement this kind of collaboration with DAML + OIL[8] technology. DAML (Darpa agent markup language)[9] is based on the RDF; OIL (ontology interface layer) is called ontology interface language. They unify the frame system based on RDF(s) and many methods based on description logics to support the collaboration. There is also a combination of the two methods above to transform the ontology in order to carry out the cooperation[10].

The two methods above need to construct a shared intelligent interface between two entities. Because of heterogeneous formalisms, there is an obvious shortcoming. On the one hand, the ontology is so completely shared from the point of the agent that all operations on it will not be restricted. On the other hand, ontology utilizes only the interface "owl: import" to import foreign information from agents without filtering and restriction, so the expression becomes undecidable easily. In addition, there also exist great costs of communication. Therefore, it is difficult for all the above methods to be applied broadly.

We will take advantage of ε-connections, OWL-DDL to resolve this difficult problem by adding a collaborative interface between an ontology and agents, between an ontology and foreign ontologies, i. e., agent-link and ontology-link. All our work will be confined inside the OWL-DL domain. Recently, OWL-DL has become the basis of ontological development. In order to implement this kind of collaboration in an effective way, it is necessary for the extensive application of the semantic web to extend the standard from the OWL itself.

2 Restricted Method of Collaboration

On the semantic web, a large-scale domainal application of ontologies usually needs to communicate with many relative foreign entities (ontologies, agents). In the course of ontological (also called upper-ontology) collaboration, in order to reach the sharing and reusing of knowledge when entities express their domain knowledge, keeping their semantics identical is required for these entities. This kind of collaboration should follow three basic semantic principles:

① Mutual semantic relationship. If there is a mutual semantic relationship between two entities, this kind of relationship must be the semantic intersection of the two domains. Otherwise, a clash must appear. Then, it is a primary issue to eliminate the clash.

② Semantic completeness. If the semantic element from a foreign entity accords with an application demand for upper-ontology, it must be complete.

③ Semantic structure consistency. The semantic structure of upper-ontology and foreign entities should set up an abstract data structure which can be matched automatically.

On the basis of the above principles, we can construct an ontological collaboration model with foreign entities (see Fig. 1). It can be divided into five main components:

① Registration All the foreign entities cooperating with this upper-ontology should be registered in the ontology, *EnumeratedAgentClass* and *EnumeratedOtologyClass* need to be defined in the extension of OWL.

② Domain partition The ontology should provide unified data structure to every kind of foreign entity, so it is easy to realize division of field, and every class has a unified *clusterID* as a chain beginning of agent-link and ontology-link for all registered entities.

③ Classification The upper-ontology should classify foreign entities by utilizing *AgentClass* and *OntologyClass*. All entities carrying on mutual operations over the same semantic data structure belong to the same clustering (or class).

④ Modality Foreign entities' active state should be written down, and their modality sign should express that the ontology can arouse the cooperation under a ready state, and offer a kind of mechanism for triggering a foreign entity.

⑤ Restriction One integrated chain, not only defines the subdomain to offer an operable field to the other side, but also defines its own range. In addition, a foreign entity should be also identified. This kind of restriction tracks the context of the newly increased knowledge of the ontology.

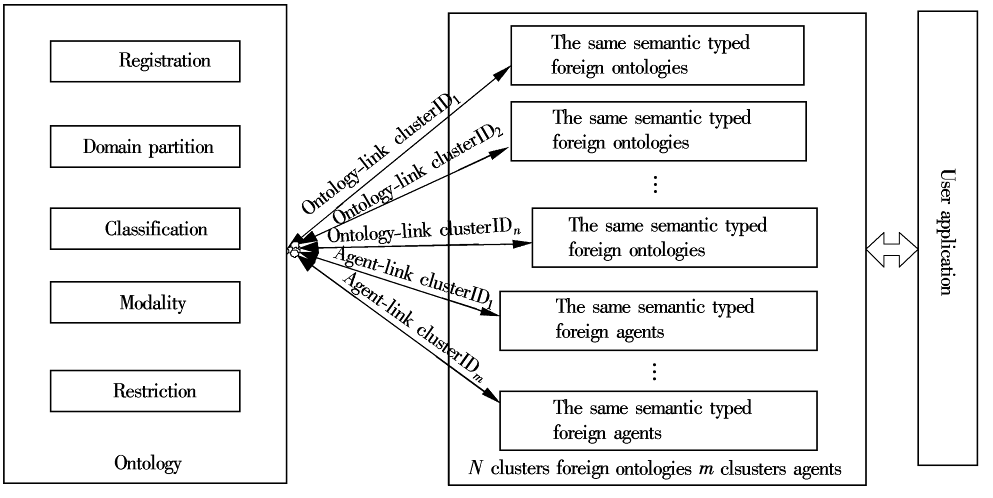

Fig. 1 A restricted ontological collaborative method with foreign entities

In the next section, we will extend the standard OWL on the basis of the above model, and add foreign ontological or agent collaborative interface into OWL. Hence, according to the relative properties of subdomains, we can construct an ontological layered integrative structure for collaboration.

3 Extend OWL: Integrating Collaboration

OWL, which acts as the second generation web ontology language recommended by W3C, is based on description logic. Considering OWL-Full goes beyond the scope of the ADL, our work is restricted to the area of OWL-Lite or OWL-DL. OWL only reserves one interface "owl: import". An ontology can be channeled into a foreign ontology or other entities. This import by using a URI reference is a kind of "ways of affirming totally" that lacks some restraints to fulfill the filter.

With the changing environment, ontologies develop dynamically, therefore, not only do their knowledge and semantic structures update constantly, but also the collaboration between the ontology and other entities must be adjusted constantly. Considering the factor of security, an ontology, as a knowledge base, should not offer a total and open operation mode for all foreign entities. So it is necessary to add an interface definition inside the OWL frame to achieve a restricted ontological collaboration with foreign entities.

3.1 Ontological interface for foreign ontologies

We will define the ontological interface in the OWL frame so as to make an ontology able to collaborate with a foreign ontology (called the former upper-ontology). Because, for every ontology, its application contexts, its collaborative domains and semantics may be different from others, the upper-ontology needs to be able to track context as well as provide restricted shared knowledge in order to help dynamic evolvement. When we define the interface, we should not only consider that some meta data structures should be reserved, but also consider that some restrained conditions that are attached to the interface. By means of the expanded expression method of BNF (EBNF) (brace { } shows that item can appear arbitrary times; square bracket [ ] represents that this item can appear less than or equal one time, ended sign needs to add quotation marks of adding, and no ended sign needs not to add quotation marks). First, we add axioms into OWL as follows:

Axiom: : = "Cluster(" clusterID [deprecate] {annotation}{"domain("Description")"}")"

Axiom: : = "Ontology-Link (ontology-linkID [deprecate] {annotation} ["inverseOf (" ontology-linkID")"] {"domain (" Description ")"} {"range (" Description")"}{"datastruct("Description")"}{"ForeignOntology (" {ontologyClassID} | {ontologyID} ")"}

where "domain" acts as a description of a category of the ontological cluster. After adding the first axiom, a *clusterID* can be produced for the ontology. The second axiom can create many links inside the ontology for foreign ontologies, its domain, its range and linked ontological group. The ontologies group may be a set of many ontological individuals, or abstract classes set over an ontological group. So a link between an ontology and an ontological group constructs a clustering integration restricted by an upper-ontology. Next, we will define the restricted component for *ontology-linkID* and *clusterID*.

Restriction: : = "Restriction (" clusterID | ontology-LinkID {clusterRestrictionComponent | ontology-linkRestrictionComponent}")"

Cluster-linkRestrictionComponent: : = "allValueFrom("Description")" | "someValuesFrom("Description")"

Ontology-LinkID: : = "Ontology-LinkID (" {OntologyClass} | {EnumeratedOntologyClass} | [ontologyID] {directive}")"

Directive: : = "annotation (" ontologyPropertyID ontologyID ")" | "annotation (" annotationPropertyID URIreference ")" | "annotation (" annotationPropertyID dataLiteral ")" | "annotation("annotationPropertyID individual ")" | axiom | facts

Value: : = "value(" Ontology-LinkID | individualID ")"

We can obtain the corresponding right for every link by *cluster-linkRestrictionComponent*, and prepare weight value for parting application field, so as to customize syntax cluster from the point of semantic. In succession, we will resolve the problem of ontological classification. *OntologyClass* is an abstract from ontological group or ontological classes. Therefore, all the individuals of every class under *ontologyClassID* can form an enumerable set. Now, we define it as follows:

axiom: : = "OntologyClass (" ontologyClassID [deprecated] modality {annotation} {"domain (" Description")"}"datastruct("Description"))"

Modality: : = "active" | "inactive"

ontologyPropertyID: : = URIreference

OntologyClass helps to construct enumerable sets for some foreign ontologies which want to communicate the ontology. In the meanwhile, the ontological state "active" signs itself a kind of enable operation. Furthermore, we can define an enumerable foreign ontological class or individual set as follows:

axiom: : = "EnumeratedOntologyClass(" ontologyClassID [deprecated] {annotation} {individualID} ")"

These class or set under the management of an ontology link to cluster. Certainly, they are required to have no intersection between two clusters. We will define another axiom as follows:

axiom: : = "disjoinOntologyClass (" {description} ")" | "EquivalentOntologyClass (" {description}")" | "SubClassoOntologyClass (" {description} ")"

Hence, it is easy to partition the whole application field into subdomains. Thereinto, description can be defined as follows:

Description: : = "ontologyClassID" | : "restriction" | "unionOf(" {description} ")" | "intersectionOf(" {description} ")" | "complementOf(" description ")" | {ontologyID} ")"

Restriction: : = "Restriction (" Ontology-Link {ontologyRestrictionComponent}")"

ontologyRestrictionComponent: : = "allValueFrom("Description")" | "someValuesFrom("Description")"

Thus, these extensions of OWL resolve the hard problem on how to integrate an ontological collaboration with foreign ontologies from the point of syntax.

3.2 Ontological interface for agents

First, we add an axiom into OWL for agents as follows:

Axiom: : = "Agent-Link (" agent-linkID ["deprecate"] {"annotation"} {"domain (" Description ")"} {"datastruct("Description")"}{"range ("Description")"} {"ForeignAgent ("{agentClassID} {agentID}")"}")"

Here, "domain" is the field of definition of this ontological interface, and it limits the field where this interface can be employed. The range described is the field where foreign agent can operate. The purpose of defining datastruct is to provide a semantic data structure for a foreign agent, and the datastruct should be consistent with range, and every entity agent should have its own and only one identification agentID. Certainly, an entity agent should also observe corresponding development standards. At the same time, *agentClassID* expresses the affiliated type of every chain which includes corresponding agentIDs inside. An agent-linkID, here regarded as a bridge between the ontology and one or more agents, has dispelled the wide gap.

Axiom: : = "AgentClass("agentClassID" ["deprecated"] modality{"annotation"}{"domain("Description") }datastruct("Description"))"

Modality: : = "active" | "inactive"

agentPropertyID: : = URIreference

"AgentClass" can help all agents which want to operate with the ontology to register, and it also makes the same semantic datastruct agents to be a cluster. At the same time, agent's active state should be written down, and the modality sign expresses that the ontology can arouse the cooperation under ready state and offers a kind of mechanism of triggering agent.

Next, in the extension of OWL, an enumerable agent class can be defined further.

Axiom: : = "EnumeratedAgentClass(" agentClassID["deprecated"] {annotation} {individualID}")"

When an ontology collaborates with agents, *EnumeratedAgentClass* defined above can form many clusters managed by the ontology. As to these defined clusters, they cannot cross with each other, that is to say, there is not any common individual agent in any two clusters. For OWL DL, an ontology should be allowed to manage these clustered agents by means of the layered way. We add further the following definitions.

Axiom: : = "disjointAgentClass ("{description}")" | "EquivalentAgentClass ("{description}")" | "SubClassofAgentClass("{description}")"

This kind of extension helps to achieve layered management in a larger application field. The description of the axiom above can be defined as follows.

Description: : = "agentClassID" | "restriction" | "unionOf(" {description}")" | "intersectionOf (" {description} ")" | "complementOf(" description ")" | {agentID} ")"

Restriction: : = "Restriction("Agent-Link {agentRestrictionComponent}")"

agentRestrictionComponent: : = "allValueFrom(" Description")" | "someValuesFrom("Description")"

From the point of view of a semantic component, the extension of OWL, which is constructed in this section, has solved the restricted collaborative problem between the ontology and the agents.

3.3 Meta method of clustering integration

In a larger-scale application field, it is necessary that an ontology collaborate with foreign ontologies and agents. The purpose of building an ontology only serves to share and reuse knowledge, but the ontology should not be open without restriction. According to our restricted ontological collaboration method with agents and foreign ontologies in the third section, after corresponding registration first, a foreign entity can begin to interoperate with the shared ontology (upper-ontology). If this kind of method is combined with our extension of OWL, we will obtain a meta method which integrates an ontological collaboration with agents and foreign ontologies to realize the clustered and layered management. It appears as a four-layer tree structure, see Fig. 2.

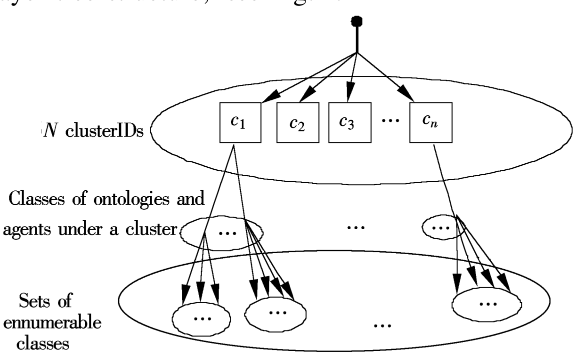

Fig. 2 Ontological layered clustering integration of collaboration with foreign entities

The first layer of the meta method is the set of cluster head *ClusterID*, every set of agents' classes or foreign ontological classes below the *ClusterID* form the second layer, the third lists sets of enumerable classes under every class, and the bottom is just each individual entities. According to the characteristic of the meta collaborative method, it is a layered and clustered integrative system.

The environment changes constantly. Information of the field itself, and the configuration and semantics of knowledge that are contained in agents and foreign ontologies are also changing constantly. The interface that is added above is completely limited to the scope of OWL-DL and, therefore, this kind of meta method that proposed in this paper, has solved how to integrate dynamic collaboration under the frame of OWL. Additionally, a very important characteristic is that the search algorithm concerning the tree may be used when we track the context of new-added knowledge.

4 Conclusion

An ontological collaboration with foreign ontologies and foreign agents is a dynamic process. In this paper, we propose a meta method of clustering integration while an ontology is collaborating with foreign agents and ontologies. It differs from the traditional approaches in the direction: OWL is extended by adding some new collaborative interfaces (i. e., agent-link and ontology-link) instead of "owl: import" that allows an ontology to import another entities, where there is no restriction and filter. Our work focuses on taking advantage of OWL itself to solve the collaborative problem, and this method is feasible for tracking contexts of new-added knowledge about ontological collaboration. This paper is only a part of our current work, and there are many difficult problems regarding interoperation to be studied further.

References

[1] Patel-Schneider Peter F, Hayes Patrick, Horrocks Ian, et al. W3C recommendation, OWL web ontology language semantics and abstract syntax [EB/OL]. (2004-02-10) [2006-04-01]. http://www.w3.org/TR/2004.

[2] Cuenca Grau B, Parsia B, Sirin E. Working with multiple ontologies on the semantic web [A]. In: McIlraith Sheila A, Plexousakis Dimitris, van Harmelen Frank, eds. *The Third International Semantic Web Conference, LNCS* [C]. Hiroshima, Japan, 2004, **3298**: 620－634.

[3] Baader F, Lutz C, Sturm H, et al. Fusions of description logics and abstract description systems [J]. *Journal of Artificial Intelligence Research,* 2002, **16**: 1－58.

[4] Li Li, Yang Yun, Wu Baolin. Agent-based ontology mapping towards ontology interoperability [A]. In: Zhang S, Jarvis R, eds. *The 18th Australian Joint Conference on Artificial Intelligence, LNAI* [C]. Berlin: Springer, 2005, **3809**: 843－846.

[5] Kutz O, Lutz C, Wolter F, et al. E-connections of abstract description systems [J]. *Artificial Intelligence*, 2004, **156**(1): 1－73.

[6] FIPA. Foundation for intelligent physical agents [EB/OL]. (2005-06-08) [2006-04-01]. http://www.fipa.org.

[7] Zhang Wenyu, Lin Lanfen, Qiu Jiong, et al. An ontology-based functional modeling approach for multi-agent distributed design on the semantic web[A]. In: Shen W, ed. *Proceedings of the 9th International Conference on Computer Supported Cooperative Work in Design, LNCS* [C]. Coventry, UK, 2006, **3865**: 334－343.

[8] van Harmelen Frank, Patel-Schneider Peter F, Horrocks Ian, et al. Reference description of the DAML + OIL ontology markup language [EB/OL]. (2001-04) [2006-04-01]. http://www.daml.org.

[9] DAML—Darpa agent markup language[EB/OL]. (2000) [2006-04-01]. http://www.daml.org.

[10] Obitko Marek, Marik Vladimir. Ontologies for multi-agent systems in manufacturing domain [A]. In: *Proceedings of the 13th International Workshop on Database and Expert Systems Applications* [C]. IEEE Computer Society, 2002. 597－602.

扩展 OWL:聚合集成协作的本体和智能体

肖如良[1,2]　唐胜群[1]　李　玲[1]　房丽娜[1]　徐　扬[1]　徐有威[1]　邓新国[1]

([1]武汉大学软件工程国家重点实验室,武汉 430072)
([2]湖南财经高等专科学校信息管理系,长沙 410205)

摘要: 为了解决本体之间、本体与外部智能体之间协作的困难问题,提出了一个本体聚合集成协作的方法,与当前方法不同之处在于不使用 owl:import 而采取添加新的协作接口(agent-link 和 ontology-link);讨论了为外部智能体及本体所做的语法接口,叙述了本体与外部智能体和本体协作的元方法.该方法致力于从 OWL 自身来解决这个协作问题,而且容易跟踪本体协作过程中新增知识的上下文.

关键词: OWL 扩展;本体;智能体;协作;聚合集成

中图分类号: TP311