Journal of Southeast University (English Edition) Vol. 22, No. 3, pp. 338 – 342 Sept. 2006 ISSN 1003—7985

Context query and association discovery for collaborative environment

Wang Guiling Jiang Jinlei Shi Meilin

(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

Abstract: A context memory model and an approach for context query and association discovery are proposed. The context query is based on a resource description framework (RDF) dataset and SPARQL language. To discover collaborative associations, an approach of transforming RDF named graphs into "context graph" is proposed. First, the definitions of the importance of the nodes and the weight assignment for the "context graph" are given. Secondly, the implementation of a spread activation algorithm based on "context graph" is proposed. An infrastructure is also built up in the collaborative context space (CCS) system to support context memory and knowledge discovery in a collaborative environment.

Key words: ontology; context memory; semantic web; semantic association; spread activation algorithm

Context computing can enable computing environments to provide enhanced context information services. In the area of CSCW, context can be defined as any information that can be used to characterize the situation of entities in a collaboration space[1]. The information space can be constructed where the context data is sensed, acquired, stored, retrieved and inferred in order to present useful information according to user needs.

Until recently, most context-aware applications only allow storage of a context entity along with its attributes and queries on various attributes of an entity. However, sometimes more complex query and retrieval techniques are required, especially in the area of CSCW. For example, in a collaborative environment, a user may need not only to track the word documents modified during October 2005 by someone else but also to identify the potential collaborators on the next task.

The goal of this paper is to present how to implement complex context query and collaborative association discovery based on semantic web technology[2].

1 Context Query

Context memory or organizational memory can be seen as the complete knowledge of a collaborative group collected over the time of its existence. In our previous work[1], we classified the contextual information into eight categories (i. e., person context, task context, interactive context, artifact context, tool context, collaboration control context, environment context and historical context) and defined an ontology for contextual collaborative applications (OCCA) by OWL[3] maintained by Protégé 3.0[4]. OCCA makes a good basis for the specification and query of context memory. Continuing the work, we define context memories as RDF datasets composed of person context memory, task context memory and artifact memory etc. Each context memory CM_i is formally defined as an RDF dataset.

Definition 1 Context memory

$$CM_i = \{Cxt, (\langle u_1 \rangle, Cxt_1), (\langle u_2 \rangle, Cxt_2), \ldots, (\langle u_n \rangle, Cxt_n)\}$$

where $\langle u_i \rangle$ is a URI and is distinct from each other; Cxt is the aggregate graph; $(\langle u_i \rangle, Cxt_i)$ is the named graph; and Cxt_i is a set of facts and the situation within which those facts are believed to be true. When a new collaborative entity is created or changed, a graph Cxt is created and stored into CM together with the situation.

A set of context query services are developed based on Jena API[5] which utilizes SPARQL as the query language providing simple select query, simple combination and inference query.

2 Collaborative Association Discovery

RDF query languages such as SPARQL allow the discovery of all resources that are linked to a particular resource by an ordered set of specific relationships. However, it cannot be used directly to query the relationships between entities. This section sets out to implement semantic association discovery in RDF-named graphs for context-aware collaborative environments based on the notion of "semantic association

Received 2006-05-25.

Foundation item: The National Natural Science Foundation of China (No. 90412009).

Biographies: Wang Guiling (1978—), female, graduate; Shi Meilin (corresponding author), male, professor, shi@csnet4.cs.tsinghua.edu.cn.

identification and discovery"[6] which is used to determine the semantic association among resources in an RDF graph.

2.1 Spread activation algorithm

The spread activation (SA) technique is one of the most frequently adopted processing frameworks for semantic networks. It has been successfully deployed in information retrieval applications. Recently some interesting systems have begun to use SA to process ontology[7-8]. Favoring the idea, we envision it to be an interesting choice of knowledge processing algorithms for the RDF dataset.

The pure spread activation model is quite simple. Given a network data structure consisting of nodes connected by links, the processing process starts from one or several nodes and spreads to other nodes until some restriction conditions are satisfied. There are many ways of spreading the activation over the network. In its simplest form, the SA algorithm computes the input value of node j using the following formula:

$$I_j = \sum_i O_i w_{ij}$$

where I_j is the total input of node j; O_i is the output of unit i connected to node j; w_{ij} is a weight associated with the link connecting node i to node j. Please refer to the survey paper[9] for the detailed description of the SA technique. There are two points in the SA algorithm. One is the initial value of the start nodes, the other is the weight associated with the link between nodes.

In the following, we will discuss how to calculate the initial value of the nodes and the strength value of the edge in the context of collaborative environments.

2.2 Context graph and resource importance

In our approach, ontologies and their instances are represented as RDF datasets. The background graph stores the URI of the named graphs and the description of the entities. The named graphs can be seen as the description of the context or situation of those entities. For different entities, the context may intersect at different nodes. For instance, Alice and Bob may attend an identical task at different times and different places. Therefore, the named graphs describing the activity intersect at the common task. In fact, every entity in background graphs intersects other entities through their named graphs. Thus, the entities in the background graphs are linked together forming a new graph called a context graph (see Fig. 1). To do so, nodes of certain classes in the background graph are first selected and inserted into the context graph and then edge is added to link two nodes if there are common entities in the named graphs.

The edges in the context graph have no direction. In general, those nodes in context graph with more degrees are more important than others. Different from traditional graphs, the degree of nodes is dependent on the weight of edges. For example, the relationship that two persons work together on a task is more important than the relationship that two persons work in a common place.

Based on this idea, we give the following formula to calculate the resource importance in a context graph.

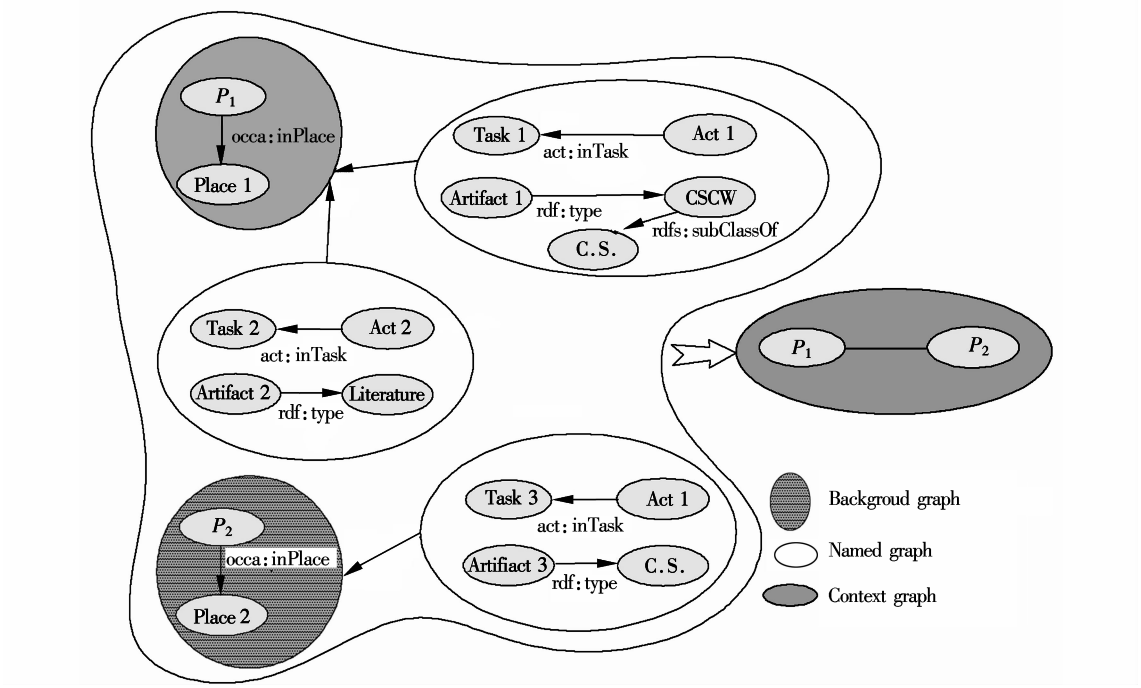

Fig. 1 Context graph

Definition 2 Resource importance

Let r be a resource in context graph, N_e be the total number of entities in its named graphs, N_c be the number of common entities with other resources, $\mathrm{wt}(e)$ be the weight of common entities in the named graphs, then the importance of the resource is calculated as

$$R(r) = \sum_{e \in ng} \frac{N_c}{N_e} \mathrm{wt}(e)$$

It is very important to calculate the weight of each edge in context graphs.

2.3 Weight assignment

Weight assignment is application dependent. We only consider the following weight assignment method for discovering the association among persons.

Joint weight If two persons take part in a common activity frequently, the relationship between them is very close. It is possible to assign a weight to the link between two person instances using the statistical value of the joint activity. Suppose that two persons have taken part in n activities. We use vectors of n dimension to distinguish one person from another. Tab. 1 gives an example of P_1 and P_2 with $n = 4$.

Tab. 1 Person by activity coordinate

Person	Activity 1	Activity 2	Activity 3	Activity 4
P_1	3	2	0	1
P_2	2	0	1	0

P_1 has coordinates $(3, 2, 0, 1)$ and P_2 has coordinates $(2, 0, 1, 0)$. The weight of the edge between P_1 and P_2 is calculated by normalizing their correlation coefficients as follows.

Definition 3 Joint activity weight

$$\mathrm{wt}(e_{P_1, P_2})_{\mathrm{activity}} = \frac{\sum_{k=1}^{n} (P_{1_k} - \bar{P}_1)(P_{2_k} - \bar{P}_2) + 1}{2\sqrt{\sum_{k=1}^{n} (P_{1_k} - \bar{P}_1)^2 \sum_{k=1}^{n} (P_{2_k} - \bar{P}_2)^2}}$$

where n is the number of activities, $\bar{P}_1 = \frac{1}{n} \sum_{k=1}^{n} P_{1_k}$, $\bar{P}_2 = \frac{1}{n} \sum_{k=1}^{n} P_{2_k}$.

The $\mathrm{wt}(e)_{\mathrm{activity}}$ ranges from 0 to 1. The greater the weight, the more closely related the two persons. Taking the example of Tab. 1, the result is 0.702 26.

The weight of the joint task simply counts how frequently two individuals are performing activities for the same task. If individuals frequently work together on tasks, they will have a stronger relationship than individuals rarely working together.

The joint task weight of an edge e is defined as follows.

Definition 4 Joint task weight

$$\mathrm{wt}(e)_{\mathrm{task}} = 1 - \frac{1}{\#t}$$

where $\#t$ is the number of common tasks.

When considering entities of an artifact, some issues arise. Two persons may not work on a common artifact, but they may work on the same type of artifact. Fig. 1 depicts this situation. P_1 and P_2 have modified Artifact 1 and Artifact 3. They are both C. S. related documents. Although the artifacts are different from each other, they often have similar interests in the C. S. domain. The artifact instances with their superclass and subclass form a hierarchy. Those lower in the hierarchy can be considered to be more specialized instances than those higher in the hierarchy. The intuition is assigning more weights to the edge with more specific common artifacts because they convey more meaning than general common artifact types.

Let $\#A$ be the number of joint artifacts, H_A be the position of the artifact in its hierarchy H and $|H|$ be the height of the artifact hierarchy. The class/instance at the top has value 1 and the value of the lower class/instance adds one to each layer. The weight of the edge between two persons with a common artifact is given by the following definition.

Definition 5 Joint artifact weight

$$\mathrm{wt}(e)_{\mathrm{artifact}} = 1 - \frac{1}{\#A(H_A/|H|)}$$

User-defined weight When considering the relationship between two persons, the interests of the users are important. Persons with similar interests may have closer relationships than others. As no interests of users are taken into account in the joint weight formula, we give the following measure to make up for the deficiencies. Our method, called user-defined weight in this paper, adopts Aleman-Meza's context weight to named graphs.

First, a region of interests is defined as a subset of classes (entities) and properties of OWL ontology. The region may vary at class level, property level and instance level. Details of region definition at class level and property level are discussed in Ref. [10]. The specification of instance restrictions is similar to that of class level. A region R_i is described as an XML segmentation and has a weight r_i.

The calculation of user-defined weight is similar to that of joint activity weight. Let N be the total number of entity, property and instance elements in user-defined regions R (from R_1 to R_m, suppose they do not intersect with each other), P_k be the total number of

components in P's named graphs that are identical to the k-th element in the user-defined region R, and r_i be the weight attributed to that region R_i. Then a person P can be denoted as a vector of N dimensions, where each element in this vector equals the product of P_k and r_i.

Definition 6 User-defined weight

$$\mathrm{wt}(e_{P_1,P_2})_{\text{user-defined}} = \frac{\sum_{k=1}^{N}(r_iP_{1_k}-\bar{P}_1)(r_iP_{2_k}-\bar{P}_2)+1}{2\sqrt{\sum_{k=1}^{N}(P_{1_k}-\bar{P}_1)^2\sum_{k=1}^{N}(P_{2_k}-\bar{P}_2)^2}}$$

where $\bar{P}_1 = \frac{1}{N}\sum_{k=1}^{N} r_iP_{1_k}, \bar{P}_2 = \frac{1}{N}\sum_{k=1}^{N} r_iP_{2_k}$.

Till now, we have defined the weight of the edge between two persons with joint tasks, joint activities and joint artifacts. With these definitions, the overall weight of edge with joint entities, denoted by $\mathrm{wt}(e)$, is defined as follows:

$$\mathrm{wt}(e) = k_1\mathrm{wt}(e)_{\text{activity}} + k_2\mathrm{wt}(e)_{\text{task}} + k_3\mathrm{wt}(e)_{\text{artifact}} + k_4\mathrm{wt}(e)_{\text{user-defined}}$$

where k_1, k_2, k_3 and k_4 are specific with applications such that $k_1+k_2+k_3+k_4=1$.

2.4 Implementation

Based on the discussion above, we can apply the SA algorithm to search collaborative associations in *ContextGraph* for collaborative environments. The algorithm to find the most associated collaborators for a given person p is as follows:

1) Let *currP* be the current person object. Initialize a queue called *AssociationQueue* with one person object p and set its value to $R(p)$.

2) While *AssociationQueue* is not empty and the stop condition (e.g., the number of traversed persons has reached a maximum value) is not satisfied:

① Fetch a person object as *currP* from *AssociationQueue* which has a maximum importance value.

② For every edge e of *currP* in the *ContextGraph*:

Let *destP* be the end of e, and add *destP* to *AssociationQueue* or update it if it is already in *AssociationQueue* to value $R(destP) + \mathrm{wt}(e)R(currP)$.

3) Return person objects in *AssociationQueue* and their value associated with *currP*.

3 Case Study

To demonstrate our ideas, a prototype system called CCS (collaborative context space) is developed as shown in Fig. 2. CCS is a new extension to LAGrid[11] service-oriented middleware. It is based on LAGrid middleware where CCS integrates several collaborative tools and supplies context services in context manager including context query service and collaborative association discovery service.

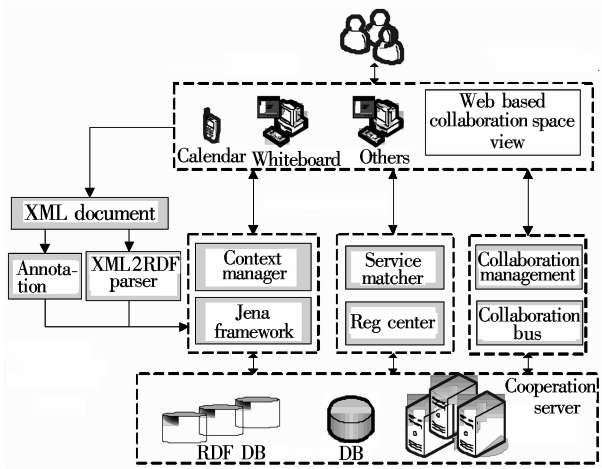

Fig. 2 The architecture of CCS

In CCS, users can not only query context information based on certain fields, but also can track the collaboration results based on complex relationships between the collaborative entities in a very flexible and exact way. When a query results in overload, CCS uses the context query language to make some restrictions. Moreover, users in CCS can always discover new opportunities to cooperate with others when their working context changes.

4 Related Work

In COBRA[12], context query is based on an ontology model and supports context reasoning. Our approach is to describe context as named graphs based on RDF dataset. Then we can memorize the context of each entity over time for later track which is not supported in COBRA.

The work presented in ONTOCOPI[7] describes a system for identifying community of practices (COPs) in an organization. Though the spread activation algorithm is utilized in their work as is done in our work, the weight of edge is given by hand in ONTOCOPI instead of being calculated automatically according a quantitative formula as in our work.

Another interesting work related to ours was presented in Ref. [13]. The event logs are specified based on an XML schema. Our work specifies event as an RDF dataset and the schema is based on OWL ontology. New event properties and new event class of specialization can be added into OCCA, which is impossible in Ref. [13]. Semantic association is also

studied in Ref. [6]. It calculates a weight for a relation instance using simple addition operation. The method of calculating weight in our work is to denote each entity as a multidimensional vector and measure the similarity between two entities.

5 Conclusion and Future Work

This paper presents an approach for query context memory and finding collaborative association in a collaborative environment. The work reported is a unique attempt to encourage more research on context awareness and context sharing in the area of CSCW, and to use semantic web technology to build a test bed for collaboration.

Although only one test case has been evaluated qualitatively due to the limitation of existing RDF data collection in collaboration domain, it suffices to show the significance of applying semantic web technologies to collaborative application. In the future, we will perform quantitative evaluation on real world data.

References

[1] Wang Guiling, Jiang Jinlei, Shi Meilin. A context model for collaborative environment[A]. In: *The* 10*th International Conference on Computer Supported Cooperative Work in Design* (*CSCWD*2006)[C]. Nanjing, China, 2006. 77 - 82.

[2] Berners-Lee Tim, Hendler James, Lassila Ora. The semantic web: a new form of web content that is meaningful to computers will unleash a revolution of new possibilities [J]. *Scientific American*, 2001, **284**(5): 34 - 43.

[3] McGuinness Deborah L, van Harmelen Frank. OWL web ontology language overview [EB/OL]. (2004-02-10) [2006-04-10]. http://www.w3.org/TR/2004/REC-owl-features-20040210/.

[4] The protege ontology editor and knowledge acquisition system[EB/OL]. (2006-04-14) [2006-05-10]. http://protege.stanford.edu/.

[5] Jena—a semantic web framework for Java[EB/OL]. (2005-10-07) [2006-04-10]. http://jena.sourceforge.net/.

[6] Sheth Amit, Aleman-Meza Boanerges, Arpinar I Budak, et al. Semantic association identification and knowledge discovery for national security applications[J]. *Journal of Database Management*, 2005, **16**(1): 33 - 53.

[7] Alani Harith, Dasmahapatra Srinandan, O'Hara Kieron, et al. Identifying communities of practice through ontology network analysis[J]. *IEEE Intelligent Systems*, 2003, **18**(2): 18 - 25.

[8] Rocha Cristiano, Schwabe Daniel, Aragao Marcus Poggide. A hybrid approach for searching in the semantic web [A]. In: *Proceedings of the* 13*th International World Wide Web Conference*[C]. New York: ACM Press, 2004. 374 - 383.

[9] Crestani F. Application of spreading activation techniques in information retrieval[J]. *Artificial Intelligence Review*, 1997, **11**(6): 453 - 482.

[10] Aleman-Meza B, Halaschek C, Arpinar I, et al. Context-aware semantic association ranking[A]. In: *Proceedings of SWDB*'03[C]. Berlin, Germany, 2003. 33 - 50.

[11] Wang Guiling, Li Yushun, Yang Shengwen, et al. Service-oriented grid architecture and middleware technologies for collaborative e-learning[A]. In: *IEEE International Conference on Service Computing* (*SCC* 2005)[C]. Orlando, Florida, USA, 2005. 67 - 74.

[12] Chen Harry. An intelligent broker architecture for context-aware systems[D]. Baltimore County: Computer Science Department of University of Maryland, 2003.

[13] Wil M P, Van Der Aalst, Reijers Hajo A, et al. Discovering social networks from event logs[J]. *Computer Supported Cooperative Work*, 2005, **14**: 549 - 593.

协作环境中的上下文查询和关联发现

王桂玲　姜进磊　史美林

(清华大学计算机科学与技术系,北京 100084)

摘要:提出了上下文记忆模型以及进行上下文查询和关联关系发现的方法. 上下文查询方法基于RDF数据集和SPARQL语言. 为了进行协作关联关系的发现,提出了一种将RDF具名图转换为"上下文图(context graph)"的方法,首先用统计分析的方法对"上下文图"中节点重要性以及边的权值进行定义,然后将激活扩散算法(spread activation)应用在该上下文图中. 最后给出了该方法在协作上下文空间(CCS)系统中应用的框架实例,该系统用来支持协作环境中的上下文记忆查询和协作关联关系发现.

关键词:本体;上下文记忆;语义网;语义关联;激活扩散算法

中图分类号:TP391