Context Modeling and Measuring for Context Aware Knowledge Management

Ke Ning and David O'Sullivan

Abstract—**To fully understand and to better reuse knowledge, it's necessary to correlate knowledge with the context under which the knowledge is generated and managed. However, it remains a challenge to support context awareness in knowledge management system. This paper tries to overcome the difficulties by introducing key enabling technologies for context modeling and measuring: The knowledge context ontology correlates knowledge and its contexts through a high level activity based model is defined; The algorithm for Context similarity measurement is designed by exploiting the hierarchical domain structure defined in the context model.**

Index Terms—**Context aware, context modeling, knowledge management, similariy measure.**

I. Introduction

In the current era of knowledge economy, knowledge has become the most important assets for any kinds of enterprises [1]. Enterprises are putting more and more efforts on knowledge management (KM) in order to retain expertise of employees, to enhance customers' satisfaction, and to increase profits or revenues [2]. A key factor for KM in an organization is the development and implementation of a knowledge management system: a technological information system for managing knowledge in the organization for supporting creation, capture, storage and dissemination of knowledge [3-5]. In academia or industry, different types of KM systems have been developed and deployed, including file management systems, database management systems, document and content management systems, data warehouses and data mining tools, and intranet and extranet knowledge portals.

To fully understand and to better reuse knowledge, it's important to correlate knowledge with the context under which the knowledge is generated and managed. Without proper supporting tools, this correlating process usually happens in a knowledge worker's mind when he/she is working on a knowledge based activity. It is hard to manage and heavily relies on the knowledge worker's personal ability and expertise. As KM systems are getting more and more popular in enterprises, it's desired to support this context awareness feature in KM systems, so as to free knowledge workers to other more innovative work. Basically, a context aware KM system should be able to help user with different types of knowledge management tasks by providing relevant knowledge to users based on context. For example, when a technical person and a sales person both are searching with the same keywords to retrieve documents from a knowledge base, the results should be different for them as they have quite different backgrounds/contexts: the technical person might be interested with more technical oriented documents, while the sales person might be interested with more business oriented documents. Another example is, when you are using different devices (such as a PC, or a mobile phone) to access a knowledge base, the KM system should be smart enough to know that only small size documents should be sent to your mobile phone because of band width and memory limitations. To realize context awareness, there are many questions to be answered: How to clarify and represent context, how to acquire context during people's knowledge based activity, how to measure context to tell the similarity between two situations, and so on.

In this paper, we do not intend to provide a total solution, but try to solve the key problems: context modeling and context measuring. Firstly, we have a basic assumption that explicit computational representations of knowledge based activities provide a good way to model context. Previous work has proven and shown how methods based on such an activity concept can enhance people's collaborative knowledge work [6-10]. We refine this idea to provide knowledge context ontology to represent both the concepts of knowledge and knowledge context in an integrated semantic model. The activity concept is used to glue knowledge item and knowledge context such as people, resource and environment, in a semantic way. This allows knowledge and its generating and using context be correlated, and provides the potentials to enhance knowledge with context.

Secondly, we want to improve the context measuring methods based on our knowledge context model. Existing methods [10-11] for context aware system do not offer sophisticated mechanisms to compute context similarity, but either rely on human's interpretation of the context information to tell if two context situations are similar or not, or simply based on text matching to compare similarity. In general similarity is an important and widely used concept, Many similarity measures have been proposed in different areas, such as cosine based similarity [12], distance based similarity [13]. Usually the definitions of similarity are tied to a particular application or a form of knowledge representation. Dekang Lin [14] proposed an information-theoretic similarity measure, which is theoretically well motivated and elegantly derived, though it needs further elaboration to use in specific application. There

Manuscript received May 3, 2012; revised May 31, 2012. The work presented in this paper is partly supported by the K-NET project (EU-FP7, Grant Agreement No. 215584) and the Ireland-China Science and Technology Collaboration Research Fund (Royal Irish Academy).

Ke Ning is with High-tech Industrial Park, Nanshan, Shenzhen, China (email: ke_ning@kingdee.com).

David O'Sullivan is with National University of Ireland, Galway, Ireland(email: dos@nuigalway.ie)

is no easy way to use directly existing methods to compute similarity between two structures as defined in our context model. Our method are mainly inspired from Prasana's work [15], which proposed to exploit hierarchical domain structure to compute element similarity and provided a sophisticated analysis and evaluation. We adapted its idea of measuring similarity between two nodes in a hierarchical tree and designed our own algorithm based on it to compute context similarity.

In the following we represent the knowledge context ontology and the algorithm for context similarity measurement.

II. Knowledge Context Ontology

A. *Integrated Model of Knowledge and Context*

The knowledge context ontology provides an integrated model for knowledge and context. As shown in Fig. 1, it defines a few fundamental concepts: (1) Activity, (2) Knowledge Item, (3) Actor, (4) Resource, and (5) Environment. The basic idea of the ontology is that a knowledge item is contextualized by being involved in an activity, which also glues other knowledge context such as actors, resources, and environment related with the activity. Meanwhile, when a knowledge item is involved in an activity, it is also part of the knowledge context, as it is now becoming the background of other knowledge items.

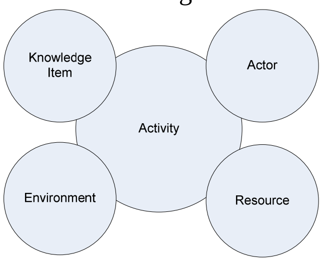

Fig. 1. The Fundamental concepts of knowledge context ontology

The ontology builds on the Dublin core [16], Friend-Of-A-Friend [17] ontologies to describe standardized properties such as titles, descriptions, and e-mail addresses. Other properties are also defined for these concepts as well as their relationships. It is extensible to define more properties to each concept as the ontology is described in RDF/OWL language [18].

With all these knowledge context elements defined, a Context Situation can be characterized by an activity and its related actor, resource, knowledge item, and environment. Fig. 2 shows an example, in this context situation, a "Maintain Hoist B" activity is happening, with these related context elements: actors "David" and "Tom", resource "Menu of Hoist", knowledge item "form1", time "time3", and location "location3".

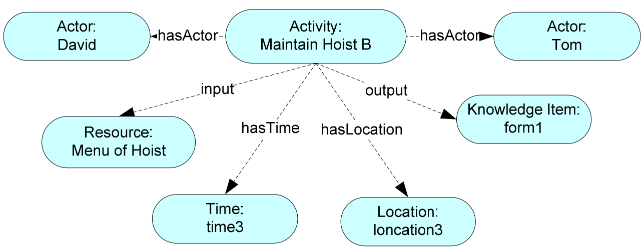

Fig. 1. An example of context situation

A. *Domain Specific Extension of the Ontology*

The knowledge context ontology is developed for domain independent usage. However, it also supports domain specific extension to meet different industrial requirements, without necessitating a system redesign.

The extension is realized by defining domain specific sub class for the fundamental concepts. Fig. 3 show an example of extension of the activity concept: Operation, Sale, and Service are defined as sub classes of Activity; and Corrective Maintenance and Regular Maintenance are defined as sub classes of Service. By doing so we are now able to define individuals of domain specific concepts, such as to specify an activity is actually a Regular Maintenance Activity, but not other types of activities. Accordingly, domain specific knowledge and context can be captured by the ontology.

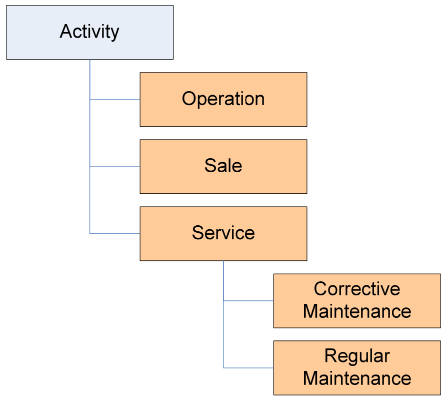

Fig. 3. An example of domain specific extension of the knowledge context ontology

With the extension, a context situation can be further characterized with more domain specific information, or in more detail. Suppose a domain specific extension has been defined for the example shown in Fig. 2, now the context situation can be described in more domain specific detail, as shown in Fig. 4: "Maintain Hoist B" is actually a Hoist Maintenance activity (sub class of Activity), "David" is a Service Engineer (sub class of Actor), "Tom" is a Customer (sub class of Actor), "Menu of Hoist" is a Menu (sub class of Resource), "form1" is a Service Request Form (sub class of Knowledge Item), "time3" is in Office Hours (sub class of Time), and "location3" is in Edinburgh (sub class of Location).

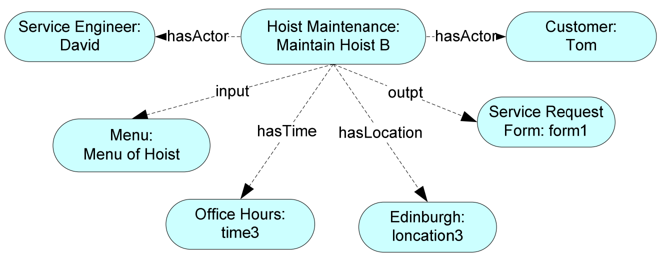

Fig. 4. An example of context situation in domain specific concepts

By defining sub classes of the fundamental concepts and sub classes of the sub classes, it actually forms a hierarchical tree which represents a hierarchical structure of a business domain. In the next section, the algorithm for context similarity measure is developed by exploiting this hierarchical class tree.

III. Exploiting Hierarchical Domain Structure to Compute Context Similarity

A. Similarity between Two Context Situations

As discussed in the previous section, a context situation is characterized by an activity and its related context elements such as actor, resource, knowledge item, and environment. Or formally, a context situation C can be defined as a set of context elements, which include one and only one activity element (E_1) , a set of Actors (E_2) , a set of elements (Resources or Knowledge Items) input to the activity (E_3) , a set of elements (Resources or Knowledge Items) output from the activity (E_4) , and a set of Times (E_5) and Locations (E_6) (without loss of generality, here only time and location are used to represent environment. Other environment elements can be added when needed), as shown in equation (1):

$$C = \begin{Bmatrix} \mathrm{E}_1 : \text{Activity} \\ \mathrm{E}_2 : [\text{Actors}] \\ \mathrm{E}_3 : [\text{Resources or Knowledge Items}] \\ \mathrm{E}_4 : [\text{Resources or Knowledge Items}] \\ \mathrm{E}_5 : [\text{Times}] \\ \mathrm{E}_6 : [\text{Locations}] \end{Bmatrix} \quad (1)$$

Then the similarity between two context situations C_1 and C_2 can be divided into similarity measures of these 6 elements, as shown in equation (2):

$$sim(C_1, C_2) = \sum_{i=1}^{6} w_i sim_{E_i}(E_i(C_1), E_i(C_2)) \quad (2)$$

where $sim(C_1, C_2)$ is the similarity measure between C_1 and C_2, w_i is the weight of the element set ($\sum_{i=1}^{6} w_i = 1$), $sim_{E_i}(E_i(C_1), E_i(C_2))$ is the similarity between two element sets of C_1 and C_2 (the value is a real number between 0 and 1).

The weight of each context element can be the same in a simple mode, or a bigger weight can be applied to the more important element in a more complex mode. In the latter case, a domain expert or a group of experts could be asked to provide the weights; or a learning algorithm could be applied to adjust the weights based on user feedback.

In the following we will see how the similarity between two element sets $sim_{E_i}(E_i(C_1), E_i(C_2))$ is computed by exploiting the hierarchical class tree.

B. Exploiting Hierarchical Class Tree to Compute Similarity

As discussed previously, for each fundamental context element, sub classes could be defined to meet domain specific requirements, which actually form a hierarchical tree. Fig. 5 shows an example of domain specific extension of the Activity concept. Three sub classes of Activity are defined: Maintenance, Marketing, and Meeting Organization. Furthermore, Air Compressor Maintenance, Hoist Maintenance, and Winch Maintenance are defined as sub classes of Maintenance. a_1, a_2, a_3, and a_4 are individuals (a_1 and a_2 are instances of Hoist Maintenance, a_3 is instance of Winch Maintenance, a_4 is instances of Meeting Organization). This forms a tree, where the root node is the fundamental concept, and the leaf nodes are the most specific sub classes or the individuals.

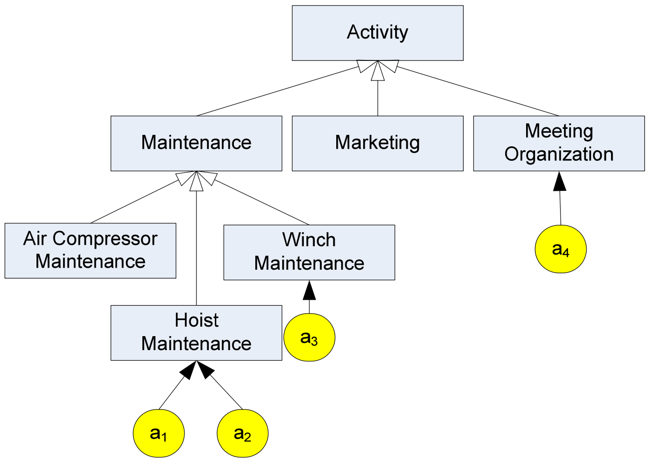

Fig. 5. Hierarchy tree of context element

When talking about similarity between the individuals, a human being can intuitively tell that a_1 and a_2 are quite similar with each other, as they are all instances of Hoist Maintenance, which is a quit specific type of activity; a_1 and a_3 are not that similar, as they are instances of different classes: Hoist Maintenance and Winch Maintenance, although they are all sub classes of Maintenance, which give them certain similarity; a_1 and a_4 are even more different, as not only they are instances of different classes (Hoist Maintenance and Meeting Organization), but also the classes have no similarity at all.

Interestingly, the same conclusion can be drawn by simply exploiting the tree structure: a_1 and a_2 are quite similar with each other, as their common ancestor node Host Maintenance is in a very low level; a_1 and a_3 are not that similar, as their common ancestor node Maintenance is in a not that low level; a_1 and a_4 are even more different, as their common ancestor node is the top level root node. This provides an automatic mechanism to compute similarity between two instances. According to Prasana's work [15], a similarity measure between two nodes a, b in a hierarchy tree can be formally defined as equation (3):

$$sim(a,b) = \frac{2 * depth(LCA(a,b))}{depth(a) + depth(b)} \quad (3)$$

where LCA (a, b) means the Lowest Common Ancestor of a and b, depth(x) means the depth of the node x in the hierarchy tree (for the example in Fig. 5, depth("Activity")=0, depth("a1")=3).

Let's go back to the Fig. 5 example again with the computation results. You can see that it's quite matching with our intuitiveness:

$$sim(a_1,a_2)=\frac{2*2}{3+3}=0.67$$

$$sim(a_1,a_3)=\frac{2*1}{3+3}=0.33$$

$$sim(a_1,a_4)=\frac{2*0}{3+2}=0$$

$$sim(a_4,a_5)=\frac{2*0}{2+2}=0$$

$$sim(a_1,a_1)=\frac{2*3}{3+3}=1$$

C. Algorithm for Context Similarity Measurement

For the similarity measure of the Activity element E_1, as there is one and only one activity instance in each context situation, we can simply use equation (3) to compute $sim_{E1}(E_1(C_1),E_1(C_2))$.

For other elements Ei (i=2 to 6) which might have more than one occurrence in a context situation, similarity is not between two instances, but two set of instances. In this case, the following algorithm is used to compute $sim_{E_i}(E_i(C_1),E_i(C_2)), i=2\text{ to }6$.

Input : Two sets to be compared
$A=[a_1,a_2,...,a_n]$, $B=[b_1,b_2,...,b_m]$
Output : similarity measure s.
Begin
s=0
If set A is not null
For each element a_i in set A
Compute its similarity with each element in set B
Select the maximum result as the similarity value s_i
Add s_i to s
Return s=s/(the number of set A elements)
Else
Return s=1
End

Code 1 – Algorithm for similarity measure between two sets

The following shows an example of similarity measure between two Actor sets. As show in Fig.6, Service Manager, Commercial Manager, Customer, and Service Engineer are defined as sub classes of Actor; a_1 and a_2 are instances of Customer; a_3 is instance of Service Engineer.

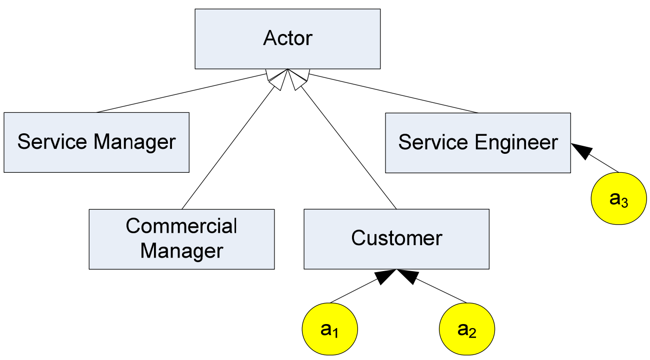

Fig. 6. Example of context similarity measure between two sets

Suppose there are 5 different Actor sets:

$S_1=[a_1,a_2]$ $S_2=[a_1,a_2,a_3]$ $S_3=[a_1]$ $S_4=[a_1,a_3]$
$S_5=[a_2,a_3]$

The similarity measures between different sets are computed as following, which are also quite matching with our intuitiveness:

$$sim\ (s_1,s_2)=\frac{1+1}{2}=1$$

$$sim\ (s_2,s_1)=\frac{1+1+0}{3}=0.67$$

$$sim\ (s_1,s_3)=\frac{1+0.5}{2}=0.75$$

$$sim\ (s_1,s_4)=\frac{1+0.5}{2}=0.75$$

$$sim\ (s_4,s_5)=\frac{0.5+1}{2}=0.75$$

D. Analysis

Note that the results produced by the algorithm are, in general, asymmetric (see the example results of Fig. 6). The reason is that when computing the similarities of the element a_i in set A with elements in set B, only the max result is selected; in the other way round when computing the element b_i in set B with elements in set A, the max result could be different. In case symmetry is needed, the symmetric similarity score between set A and B can be defined to be the average of the two asymmetric scores $sim\ (A,B)$ and $sim\ (B,A)$.

The algorithms depend on a well defined hierarchical tree. For example, in the extreme case where there is no sub class defined for a fundamental context class, the similarity between each individuals of it will have only two values: 1 (when comparing one individual with itself) or 0 (when comparing one individual with others), which obviously is not a good measure. Fortunately, in most cases enterprises can easily define at least a tree of two levels for each fundamental context class, which is already enough for our method to generate distinguishable results (Fig. 6 is a two level tree, and the similarity results between the five example sets are distinguishable).

Meanwhile, the algorithm also requires the individuals of each fundamental context class are modeled as the instances of the most specific sub classes as possible. Or from the tree point of view, the individuals should be put close to the leaf nodes as much as possible. For example, in the extreme case where all individuals are modelled as instances of the fundamental class, the similarity between each individual will also have only two values. This actually poses a requirement for enterprises to provide as much as possible fine-grained context rather than course-grained context. One possible way to refine the context grain during the context extracting process is to define more domain specific rules to infer fine-grained context from course-grained raw context.

IV. CONCLUTION

This paper proposed an activity centric approach for context aware knowledge management. It allows legacy KM systems to benefit from context awareness. This is realized through a knowledge context model which provides an integrated representation of knowledge and its generating and using context. Based on this model, the KM system can be enhanced by providing knowledge to users through context similarity measure. Specifically, our work contributes in tow folds:

1. The knowledge context ontology provides an integrated representation of knowledge and its generating and using context. It correlates knowledge and knowledge contexts through a high level activity concept. It is a generic model but also can be extended with domain specific concepts.
2. The context similarity measurement method exploits the hierarchical class tree defined in the domain specific knowledge context model to compare similarity between context situations, which provides better results than those existing text matching based methods.

Based on the model and method, we have developed a prototype and tested in three small and medium enterprises in Europe [19]. The results up to now are promising: with the support of the system, the searching for knowledge resources is more accurate and relevant; the management of collaborative knowledge-based activity is more productive; the efforts required for knowledge management are reduced.

Regarding future work, as the context similarity measure algorithms rely on a well defined classification tree of the knowledge context model, we are working to use fuzzy clustering method to help refine the classification tree based on the available historical data.

REFERENCES

[1] P. F. Drucker, *Post Capitalist Society*. New York: Harper Business, 1993.

[2] T. H. Davenport and L. Prusak, *Working Knowledge: How Organizations Manage What They Know*. Boston, Massachusetts: Harvard Business School Press, 2000.

[3] M. Alavi and D. E. Leidner, "Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues," *MIS Quarterly,* vol. 25, pp. 107-136, 2001.

[4] R. Chalmeta and R. Grangel, "Methodology for the implementation of knowledge management systems," *Journal of the American Society for Information Science and Technology,* vol. 59, pp. 742-755, 2008.

[5] R. Maier, *Knowledge Management Systems: Information And Communication Technologies for Knowledge Management*, 3 ed. Berlin: Springer, 2007.

[6] D. Georgakopoulos, *et al.*, "An overview of workflow management: From process modeling to workflow automation infrastructure," *Distributed and Parallel Databases,* vol. 3, pp. 119-153, 1995.

[7] R. Y. C. Hill, C.Jones, S. L. Kogan, "Beyond Predictable Workflows: Enhancing Productivity in Artful Business Processes," *IBM Systems journal,* vol. 45, 2006.

[8] A. Cozzi, *et al.*, "Activity management as a web service," *IBM Systems journal,* vol. 45, 2006.

[9] D. G. P. Moody, M. J. Muller, J. Tang, T. P. Moran, "Business activity patterns: A new model for collaborative business applications," *IBM Systems journal,* vol. 45, 2006.

[10] A. Hyung Jun, *et al.*, "Utilizing knowledge context in virtual collaborative work," *Decision Support Systems,* vol. 39, p. 563, 2005.

[11] M. Millie Kwan and P. Balasubramanian, "KnowledgeScope: managing knowledge in context," *Decision Support Systems,* vol. 35, pp. 467-486, 2003.

[12] M. J. MCGILL, *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.

[13] R. Rada, *et al.*, "Development and application of a metric on semantic nets," *Systems, Man and Cybernetics, IEEE Transactions on,* vol. 19, pp. 17-30, 1989.

[14] L. Dekang, "An Information-Theoretic Definition of Similarity," presented at the Proceedings of the Fifteenth International Conference on Machine Learning, 1998.

[15] G. Prasanna, *et al.*, "Exploiting hierarchical domain structure to compute similarity," *ACM Trans. Inf. Syst.,* vol. 21, pp. 64-93, 2003.

[16] DC, "The Dublin Core Metadata Initiative," http://dublincore.org/, 2010.

[17] D. Brickley and L. Miller, "FOAF Vocabulary Specification," ed. http://xmlns.com/foaf/spec/, 2010.

[18] OWL, "OWL Web Ontology Language," [online]. available: http://www.w3.org/TR/owl-features/, 2010.

[19] K. Ning, *et al.*, "A Service Oriented Framework for Context Aware Knowledge Enhancing," presented at the MCPL 2010, 2010.

Ke Ning is a researcher with Kingdee Research since 2012. He was a researcher with National University of Ireland, Galway since 2006. He received a Ph.D degree from Tsinghua University, Beijing, China in 2004. His main research interests include semantic web, knowledge management, and social web service.

David O'Sullivan is Director of Research, CIMRU, National University of Ireland, Galway since 2001**.** His research interests include knowledge management, innovation management.