PERGAMON

Expert Systems with Applications 22 (2002) 313–320

Expert Systems with Applications

www.elsevier.com/locate/eswa

A knowledge grid model and platform for global knowledge sharing☆

Hai Zhuge*

Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, People's Republic of China

Abstract

This paper proposes a knowledge grid model for sharing and managing globally distributed knowledge resources. The model organizes knowledge in a three-dimensional knowledge space, and provides a knowledge grid operation language, KGOL. Internet users can use the KGOL to create their knowledge grids, to put knowledge to them, to edit knowledge, to partially or wholly open their grids to all or some particular grids, and to get the required knowledge from the open knowledge of all the knowledge grids. The model enables people to conveniently share knowledge with each other when they work on the Internet. A software platform based on the proposed model has been implemented and used for knowledge sharing in research teams. © 2002 Elsevier Science Ltd. All rights reserved.

Keywords: Grid; Knowledge management; Knowledge sharing; Query language; Groupware

1. Introduction

In contrast to the rapid growth of information on the web, Internet users still lack effective means to store their knowledge, organize knowledge distributed on the Internet, and share knowledge with each other. Internet users have to be isolated from each other to perform tasks or to solve problems from scratch according to their respective knowledge. Although the current markup languages like resource description framework (RDF) can express some semantic knowledge in a document, so far we do not have effective means to organize, to retrieve, and to manage knowledge distributed across the Internet.

Knowledge management plays an important role in promoting innovation and productivity in a cooperative team (Dieng, 2000; Zhuge, Ma, & Shi, 1997). An Internet-based knowledge management should enable any user to store his/her knowledge at any time when he/she has generated some new knowledge like problem-solving methods and to easily get the required knowledge from the knowledge repositories distributed on the Internet for solving encountered problems. In this way, knowledge resources on the Internet can be rapidly accumulated and evolved as the common knowledge assets of the whole Internet community with the expansion of the Internet users.

Since the speed of the next-generation Internet will be over 10,000 times as that of the current one (i.e. every 10 min for information exchange will then be shortened to less than 0.06 s), ordinary users will feel that information exchange across the Internet is carried out as in one PC. This provides us the rationale to regard the whole Internet as a single computer. If so, the whole information on the Internet can be regarded as a worldwide information repository. Internet users can retrieve the required information from the repository by using a query language like the SQL. Such a repository requires an information model like the relational data model and the related operational language. Similarly, if we have a worldwide knowledge repository, then Internet users will be able to conveniently share knowledge with each other when they work on the Internet.

2. The notion of grid

The word grid is usually explicated in the grid computing literature. The grid computing refers to the technology that enables a large scale distributed computing system to carry out the controlled sharing of computing resources. Grid computing concepts were first explored in the 1995 I-WAY experiment, where high-speed networks were used to connect high-end resources at 17 sites across North America. Since then, a number of grid research projects that developed the core technologies for 'production' grids in various communities and scientific disciplines (Foster, 2000) (see http://www.gridforum.org). The grid computing can also be regarded as the problem-solving strategy of divide-and-conquer.

Nowadays, the notion of grid has been extended. A grid

☆ This work was supported by the National Science Foundation of China.
* Fax: +86-106-256-7724.
E-mail address: zhuge@ict.ac.cn (H. Zhuge).

PII: S0957-4174(02)00019-2

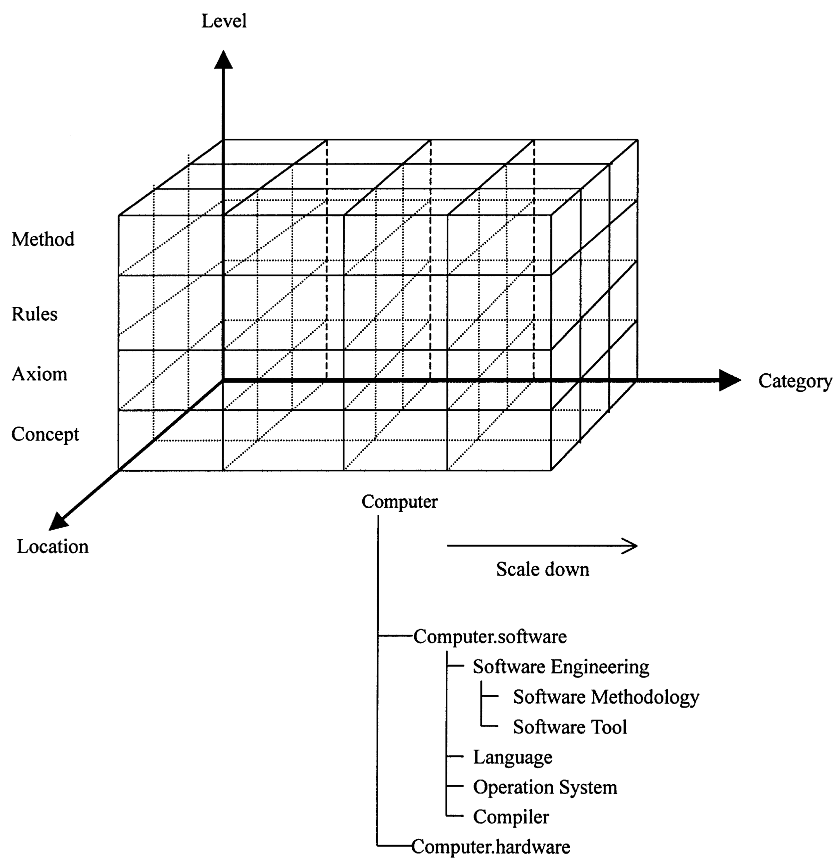

Fig. 1. Three-dimensional knowledge space.

can be computation, data, software, agents, and even people. So a grid can be regarded as an integrated mechanism that enables the control sharing of various kinds of resources. A generic grid should have three characteristics: (1) the network ability, i.e. the ability to link resources with or into the grid; (2) the inter-operability, i.e. the ability for any grid participant to use any resources to perform some tasks; and (3) the composition ability, i.e. the ability to compose grid resources to form new combined resources. A grid itself can also be regarded as a resource. We herein give the definition of the generic grid as follows.

Definition 1. A generic grid can be defined by the following two items: (1) a set of closely related objects is a grid; an object can take the form of knowledge, information, or service; (2) a set of ⟨grid, location⟩ pairing together with a set of inter-operations constitute a grid; and (3) any grid can get objects from any other grid by using these inter-operations.

3. Knowledge grid model

A *knowledge grid* is a set of well-organized knowledge together with a set of knowledge management operations. A well-organized knowledge should guarantee knowledge to be accurately identified. Coordinates are the most commonly used and efficient means to identify objects in a space. For example, we can accurately locate any place in the world according to latitude and longitude on map. If we regard the whole knowledge of human beings as a knowledge space and establish its coordinate grid, then we can accurately store and retrieve any knowledge according to its coordinates.

A *worldwide knowledge grid* (WWKG) is a three-dimensional knowledge space: (*knowledge-category*, *knowledge-level*, *location*). The former two dimensions identify knowledge content, and the third dimension identifies the locations that store knowledge. Each point in the space represents knowledge at a certain knowledge level of a knowledge category and is stored at a certain location as shown in Fig. 1.

With reference to the knowledge levels of an axiom system, we classify the knowledge space into four knowledge levels from low to high: conceptual level, axiom level, rule level, and method level. The conceptual level contains the basic concepts in the form of noun or noun phrases together with their explanations or definitions like dictionaries. The axiom level contains the commonsense knowledge of knowledge categories. An axiom takes the form of a natural language statement or a mathematical equation that describes the relationship between concepts. The rule level contains the basic rules and principles of knowledge

Table 1
Knowledge representation: a two dimensional table

	Computer.software.coding	Computer.software.methodology	Computer.software.tool
Method	Content (*x*, *y*)	Content (*x*, *y*)	Content (*x*, *y*)
Rule	Content (*x*, *y*)	Content (*x*, *y*)	Content (*x*, *y*)
Axiom	Content (*x*, *y*)	Content (*x*, *y*)	Content (*x*, *y*)
Concept	Content (*x*, *y*)	Content (*x*, *y*)	Content (*x*, *y*)

categories. A rule reflects the logical relationship between axioms. It takes the form as *IF condition THEN conclusion*. The method level contains the problem-solving methods. A problem-solving method takes the form of the problem-solution pairs. The solution can be either a multi-step problem-solving process or simply a one-step solution (e.g. a software component or an algorithm).

Each knowledge category can include several sub-categories, and each sub-category can further include several smaller sub-categories. A knowledge category together with its all-level sub-categories constitutes a *knowledge category hierarchy*. So coordinates of the knowledge-category axis are scalable because people usually concern knowledge across different knowledge levels. Except for the basic sub-categories, each coordinate of the horizontal axis can be scaled down onto a set of low-level coordinates, which can be then scaled down again or scaled up to its up-level coordinates. The top-level coordinates of the horizontal axis are the roots of all the knowledge category hierarchies. When name duplication happens between sub-categories, we should refer to the up-level categories of each to differentiate between them by denoting a sub-category as *up-level-category.sub-category*. A knowledge content can be accurately located when its knowledge category and knowledge level are given. For example, if a student wants to learn some concepts on software, then he/she can accurately retrieve the content by giving the following two coordinates: (knowledge-category = computer.software, knowledge-level = concept).

The location coordinates are the Internet locations that store knowledge. We call the knowledge location universal knowledge location (UKL). The format of the UKL is URL/[GroupName/][UserName/[attribute]/[x,y/]. The ‘URL’ is the web site that the user can store knowledge. The ‘GroupName’ and the ‘Username’ are the names of the group or the person who contributes the knowledge. The ‘attribute’ can be either PRIVATE or PUBLIC which, respectively, represents the private knowledge that the other people cannot get and the public knowledge that can be shared by the other people. The attribute can be default if it takes the PUBLIC value. ‘*x*, *y*’ refers to the coordinates of the knowledge content, and the location refers to the whole knowledge grid in case of default. For example, ‘kg.ict.ac.cn/~zhuge/PRIVATE/computer.software,method/’ means that the software methods contributed by ‘zhuge’ are private knowledge, and ‘kg.ict.ac.cn/~zhuge/PUBLIC/computer.software,concept/’ means that the software concepts contributed by ‘zhuge’ are public knowledge. The UKLs can be ordered as location coordinates according to the time zones the corresponding URLs belong to.

A two-dimensional table, Table 1, can be used to represent knowledge in a knowledge grid. The contents of the table are the knowledge representation of the corresponding knowledge level. A knowledge grid concerns three kinds of knowledge privileges: (1) the public knowledge that can be shared by all the other knowledge grids; (2) the group private knowledge that can only be shared by users of the same group; and (3) the private knowledge that can only be used by its owner.

4. Knowledge grid representation

Since knowledge grids may be distributed onto different platforms, a worldwide knowledge grid should enable users to share knowledge across platforms. For this purpose, we adopt the XML (www.w3.org/TR/REC-xml) or the XML-based markup language (semantic web) to represent knowledge grids. The worldwide knowledge grid contains the meta-information of all the knowledge grids: *the user name*, *the related UKL*, and *the group name*. A knowledge grid can operate on any other knowledge grid according to the meta-information. The following are the samples of the worldwide knowledge grid and the knowledge grid.

⟨WorldwideGrid⟩
  ⟨UserName$_1$⟩ ⟨UGL$_1$: LocalGridName: Group⟩ ⟨/UserName$_1$⟩
  ……
  ⟨UserName$_n$⟩ ⟨UGL$_n$: LocalGridName: Group⟩ ⟨/UserName$_n$⟩
⟨/WorldwideGrid⟩

⟨LocalGridName⟩
  ⟨Public⟩
    ⟨Method⟩
      ⟨CategoryName⟩ {MethodSet} ⟨/CategoryName⟩
      …
      ⟨CategoryName⟩ {MethodSet} ⟨/CategoryName⟩
    ⟨/Method⟩
    ⟨Rule⟩
      ⟨CategoryName⟩ {RuleSet} ⟨/CategoryName⟩
      …

```
      ⟨CategoryName⟩ {RuleSet} ⟨/CategoryName⟩
    ⟨/Rule⟩
    ⟨Axiom⟩
      ⟨CategoryName⟩ {PrincipleSet} ⟨/CategoryName⟩
      …
      ⟨CategoryName⟩ {PrincipleSet} ⟨/CategoryName⟩
    ⟨/Axiom⟩
    ⟨Concept⟩
      ⟨CategoryName⟩ {ConceptSet} ⟨/CategoryName⟩
      …
      ⟨CategoryName⟩ {ConceptSet} ⟨/CategoryName⟩
    ⟨/Concept⟩
  ⟨/Public⟩
⟨/LocalGridName⟩
```

For example, the concepts of software engineering can be represented as follows:

```
⟨Concept⟩
  ⟨Computer.Software.SoftwareEngineering.Object Orientation⟩
    {object: definition; class: definition: encapsulation: definition;…; inheritance: definition}
  ⟨/Computer.Software.SoftwareEngineering.Object Orientation⟩
⟨/Concept⟩
```

5. Knowledge grid operation language

KGOL is a set of basic operation statements for creating knowledge grids, and then for sharing knowledge and managing knowledge. We can either separately use these statements or compose them as a program to implement an application. In the following statements G_i represents the name of a knowledge grid; K_i is a set of knowledge (i.e. concepts, commonsense knowledge, rules, or methods); the 'Condition Expression' is a Boolean expression that specifies the coordinates of the knowledge content as well as the other constraints, the capital words are the retained by the KGOL; and, the blanket '[…]' means the content within the blanket by default.

1. *Create-statement*: CREATE G_1: ⟨Knowledge-Levels, Knowledge-Category-Hierarchies⟩,…,G_n: ⟨Knowledge-Levels, Knowledge-Category-Hierarchies⟩ AT UKL_1, UKL_2,…,UKL_n;
2. *Get-statement*: GET K_1,…,K_m FROM G_1,…,G_m [AT UKL_1, UKL_2,…,UKL_n] [WHERE ⟨Condition Expression⟩];
3. *Put-statement*: PUT K_1,…,K_m TO G_1,…,G_m AT UKL_1, UKL_2,…,UKL_n [WHERE ⟨Condition Expression⟩];
4. *Delete-statement*: DELETE K_1,…,K_m IN G_1,…,G_m AT UKL_1, UKL_2,…,UKL_n [WHERE ⟨Condition Expression⟩];
5. *Browse-statement*: BROWSE G_i AT UKL_i [WHERE ⟨Condition Expression⟩];
6. *Log-statement*: LOG G_1,…,G_m AT UKL_1, UKL_2,…,UKL_n ON/OFF G [WHERE ⟨Condition Expression⟩].
7. *Open-statement*: Open G_1,…,G_m AT UKL_1, UKL_2,…,UKL_n TO G [WHERE ⟨Condition Expression⟩].
8. *Join-statement*: Join G_1,…,G_m AT UKL_1, UKL_2,…,UKL_n INTO G [WHERE ⟨Condition Expression⟩].

The meaning of create-statement is to create a knowledge grid G_i at location UKL_i with the given knowledge level and the knowledge category hierarchy. We represent the knowledge levels in the order from low to high level as ⟨$level_1$,…,$level_n$⟩, and represent a category hierarchy as category[sub-$category_1$[sub-$category_{11}$,…, sub-$category_{1i}$], …,sub-$category_n$[sub-$category_{n1}$,…,sub-$category_{nj}$]].

The meaning of the get-statement is to get knowledge from G_i at location UKL_i that satisfies the condition expression and then assign to K_i. The location can be default if we do not know its address. For example, GET K FROM ZhugeGrid WHERE CATEGORY = computer.software AND LEVEL = concept. A get-statement can be embedded in another get-statement for complex query by replacing the UKL with another query statement, for example, GET K1 FROM GET K2 FROM ZhugeGrid WHERE CATEGORY = computer.software AND LEVEL = concept. Besides giving a pair of coordinates of the knowledge grid in the condition part, the KGOL also enables users to limit the search scope by giving the keywords of the required knowledge in the condition part.

The meaning of the put-statement is to put the knowledge K_i to the point in the knowledge grid G_i at the location UKL_i. The condition determines the constraint relationship between K_i and G_i. For example, 'PUT {inheritance} TO ZhugeGrid AT kg.ict.ac.cn/~ZHUGE/computer.software,-concept/ WHERE (computer.software, concept) NOT INCLUDE inheritance' is to put a concept set 'concepts' to the point (CATEGORY = computer.software, LEVEL = concept) in the knowledge grid 'ZhugeGrid' at the location 'kg.ict.ac.cn/~ZHUGE'.

The delete-statement is for deleting the out-of-date knowledge or the redundant knowledge K_i at the place that satisfies the condition in the knowledge grid G_i at location UKL_i. Only the private knowledge is allowed to be deleted, and can only be performed by its author. The common knowledge is not allowed to be deleted because the knowledge grid is sharable by all users and become a common asset once it has been logged onto the worldwide knowledge grid. For example, DELETE {goto} IN

ZhugeGrid AT kg.ict.ac.cn/~ZHUGE/computer.software.programing,concept/.

The browse-statement enables us to see a view of the knowledge grid G_i at UKL_i that satisfies the given condition. The 'Condition Expression' defines a view of G_i. For example, 'BROWSE ZhugeGrid AT kg.ict.ac.cn/~Zhuge WHERE CATEGORY = ⟨computer.software.programming, computer.software.methodoglogy⟩ AND LEVEL = ⟨rules, method⟩' is to browse a view of knowledge grid 'ZhugeGrid' defined by two coordinates 'computer.software.programming' and 'computer.software.methodoglogy' at the knowledge category axis and two coordinates 'rule' and 'method' at the knowledge level axis at location 'kg.ict.ac.cn/~Zhuge'. The condition part also enables users to input keywords to limit the scope of the view that has been defined.

The log-statement is to log the knowledge grid G_i at location UKL_i on or off the worldwide knowledge grid at the point that satisfies the given condition. For example, 'LOG ZhugeGrid AT kg.ict.ac.cn/~Zhuge ON G WHERE CATEGORY = computer.software AND LEVEL = method' is to log the knowledge grid 'ZhugeGrid' at location 'kg.ict.ac.cn/~zhuge' onto the point that satisfies the condition 'CATEGORY = computer.software AND LEVEL = method' in the worldwide knowledge grid G. Users cannot visit a knowledge grid that has been logged off.

The open-statement is to open the access privilege of the whole or a part of the knowledge grid $G_1,\ldots,G_m$ at UKL_1, $UKL_2,\ldots,UKL_n$, respectively, to another knowledge grid G under the given condition. This statement enables different knowledge grids to form a temporal group so as to share some particular knowledge. The join-statement is to join knowledge grids $G_1,\ldots,G_m$ at UKL_1, $UKL_2,\ldots,UKL_n$, respectively, into another knowledge grid G. The condition of carrying out this operation is that the privilege of the operator is higher than that of G_i and G or that they agree to open to each other.

6. Enhancement knowledge grid model

6.1. Establish abstraction relationship

In order to enhance the effectiveness of using knowledge, we can establish abstraction relationship between knowledge contents at all knowledge levels. Abstraction hierarchy at different knowledge levels can help people to understand complex objects, to effectively use knowledge, and to solve problems. The problem-solving methods using the abstraction hierarchy are based on the following problem-solving principle: if a problem has a solution at low-abstraction level then it must have a solution at high-abstraction level, and if a problem does not have a solution at high-abstraction level then it must not have a solution at low-abstraction level. According to this principle, when we are not able to solve a problem at low-abstraction level, we can transform it onto the high-abstraction level and find the solution with the knowledge at the high-abstraction level, and finally transform the solution down to the low-abstraction level by considering knowledge at the low-abstraction level (Zhuge et al., 1997).

We can add two retained words IS-HIGHER-THAN and IS-LOWER-THAN to the condition part of the KGOL so as to use the abstraction relationship to query the required knowledge. For example, GET K FROM ZhugeGrid WHERE CATEGORY = computer.software AND LEVEL = method AND IS-HIGHER-THAN method1.

6.2. Establish similar relationship

We can also establish similar relationship at these knowledge levels. This enables users to carry out analogy between the existing knowledge (e.g. problem-solving methods) and the new problem, and then to make full use of the existing knowledge (e.g. to use the existing solution to a problem that is similar to the new problem) to solve the current problems (Zhuge et al., 1997). We can add a keyword IS-SIMILAR-TO to the condition part of the KGOL so as to use the similar relationship to query the required knowledge. For example, GET K FROM ZhugeGrid WHERE CATEGORY = computer.software AND LEVEL = method AND problem IS-SIMILAR-TO problem1.

We can use the XML-link to represent the abstraction relationships and the similar relationships. After we add the abstraction and similar relationships to the knowledge grid, we need to build a maintenance mechanism to maintain these relationships after carrying out knowledge grid operation.

6.3. Consider factors of time and inexactness

Some knowledge concerns factors like time and inexactness. These factors can be regarded as a part of knowledge content. So we can attach a knowledge generation time and a certainty factor (CF) to the knowledge content as the following form: knowledge-content (time, CF). Accordingly, we need to add the relevant keywords like IS-AFTER, IS-BEFORE, and IS-ON in the condition syntax of the KGOL. We can express the time factor and CF in the condition part of the KGOL to operate knowledge grid according to the given time factor and certainty constraints. For example, GET K FROM ZhugeGrid WHERE CATEGORY = computer.software AND LEVEL = rule AND Time IS-AFTER 1996 AND CF > 0.8.

7. Relationship between knowledge grid and semantic web

In recent years, the notion of the semantic web has been proposed and is regarded as the next-generation web (Heflin & Hendler, 2001; Hendler, 2001; Maedche & Staab, 2001; McHraith, Son, & Zeng, 2001). It is for providing the

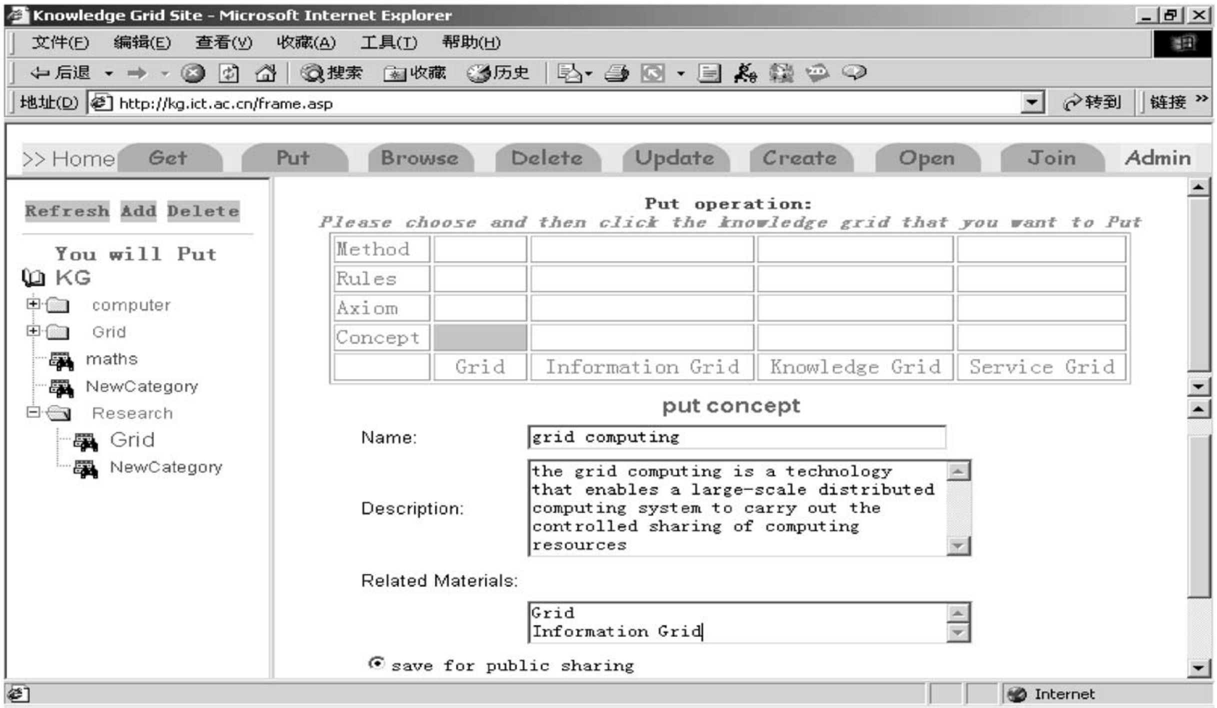

Fig. 2. The 'put concept' operation interface of the knowledge grid.

services based on the machine-understandable web resources, because the current search engine does not know the content of the HTML-based web pages and the current web pages cannot reflect its machine understandable semantics. The semantic web currently focuses on the markup languages such as RDF, OIL, and DAML. The XML-based RDF (see www.w3c.org/rdf) defines the machine-understandable semantics of web resources by the object-attribute-value model (Decker et al., 2000; Klein, 2001). The RDF schema (RDFS) enhances the representation ability of the RDF by providing the means to define the vocabulary, the class-based structure and the constraints for expressing the metadata about web resources. As the extension of the markup language, the approach for representing knowledge in document by extending the RDFS has been proposed (Broekstra et al., 2001). Ontology inference layer (OIL) is an extension of RDFS through the well-defined syntax in XML based on the document type definition (Fensel et al., 2001). DAML is the extension of web ontologies to allow rules to be expressed within the languages (Hendler & McGuinness, 2000). Ontologies are regarded as the key to support information exchange across various networks. The ontology of a particular domain establishes a common understanding between people. It usually contains a hierarchy of concepts of a domain and describes each concept's crucial properties through an attribute-value. WordNet (www.cogsci.princeton.edu/~wn) can be regarded as a kind of ontology at the concept level. People have developed assistant tools for the creation and management of ontologies (Fensel et al., 2001). Besides, approaches for representing knowledge within documents were proposed such as the web knowledge base (WebKB; Martin & Eklund, 2000) and the frame-based simple HTML ontology extensions (SHOE; Heflin & Hendler, 2001).

The common point between the semantic web and the knowledge grid is that both are based on the Internet. The main difference is their intentions. The main intent of the semantic web is to establish the information sharing mechanism across the web based on the idea of enabling the document to be machine understandable. But, the intents of the knowledge grid are for knowledge sharing management, and finally for helping Internet users to efficiently solve problems. Currently, the semantic web does not provide any means to manage and retrieve knowledge. Our strategy is to use the latest research results on the semantic web to update our knowledge representation approach used in the knowledge grid model. The knowledge grid can be regarded as a kind of knowledge web above the semantic web.

8. Implementation

Based on the proposed knowledge grid model, we have developed a knowledge grid platform to support distributed knowledge sharing and management. The platform is available at http://kg.ict.ac.cn. Figs. 2 and 3 show the 'put concept' and the 'get concept' operation interfaces of the knowledge grid, respectively. The left portion of the interface is a scalable coordinate hierarchy of the category axis. A cooperative team can unify the coordinates by editing the

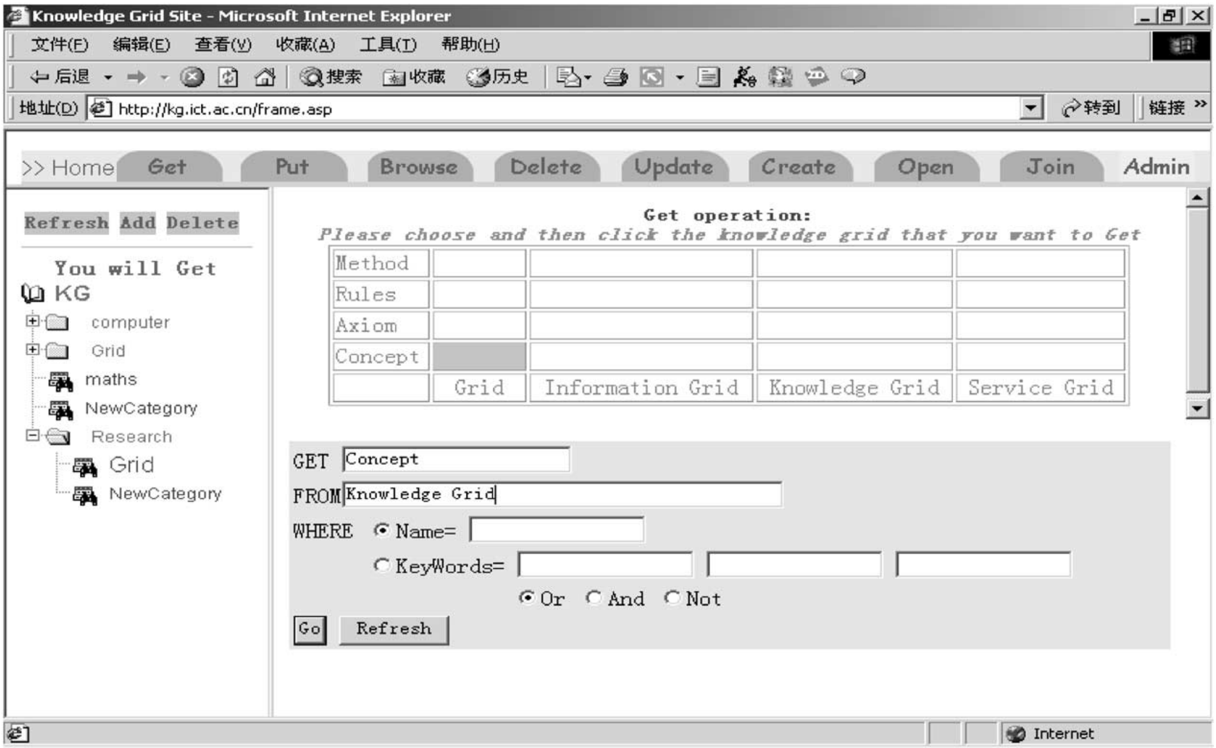

Fig. 3. The 'get concept' operation interface of the knowledge grid.

hierarchy. For example, a software development team can set the following five coordinates at the category axis: analysis, design, programming, testing, and maintaining according to the stages of the structured software development methodology. They can also add coordinates such as a team-affair after the creation of the knowledge grid.

Any user can login as either a team member or a personal user and then create his/her own knowledge grid. A user can click on the operation batten and then select the suitable knowledge content to be operated by clicking on the rectangle representing the knowledge point of the knowledge space to start the knowledge operation. For example, a user can put his/her problem-solving method at the coding stage into his/her knowledge grid by clicking the 'Put' operation button and then clicking the rectangle at the coordinate point: (Coding, Method).

9. About information grid and service grid

With the implication of the knowledge grid, we can establish an information grid model to accurately identify information on the web. Information can be either data or documents of different types on the web. A *worldwide information grid* is a three-dimensional information space: (information-category, information-level, location). The former two dimensions identify information content, and the third dimension identifies the locations that store information. Each point in the space represents information at a certain information level of an information category and is stored at a certain location. Similar to the knowledge category, information category refers to the classification of the information content. Information can be leveled according to the scale of information granularity or the types of information. For example, information granularity can be defined as five scales from small to large: *data*, *paragraph*, *page*, *section*, and *file*. We can use the markup languages of the semantic web like RDF to represent the content of information so as to make it machine understandable and to be able to cross platform. Similar to the KGOL, we can develop an information grid operation language to manage the information grid.

We can also establish a service grid model to enable users to organize and accurately identify services on the web. A worldwide *service grid* is a three-dimensional service space: (service-category, service-level, location). The former two dimensions identify service region, and the third dimension identifies the locations that store services. The coordinates of the service-category axis are the classifications of the functions of services. The service-level axis contains four coordinates from low (close to hardware) to high (close to users): *the system level*, *the middleware level*, *the application interface (API) level*, and *the application level*. Each point in the space represents services at a certain service level of a service category and is stored at a certain location. Similar to the KGOL, we can develop a service grid operation language to manage the service grid.

10. Summary

The proposed knowledge grid model and system provide us a means to collect, represent, share, and manage knowledge for distributed Internet application. It is an effort towards a knowledge web that can provide knowledge sharing and other intelligent services rather than the current web's information service. Although the implementation of the knowledge web depends on the standardization of the knowledge categories to some extent; at the current stage we can make use of the library classification method to identify knowledge categories. Besides, in the cooperative teamwork applications like team software development, team members can make consensus on the classification of knowledge categories and knowledge levels before carrying out cooperation.

The ongoing work includes four aspects. First, we are going to add a knowledge acquisition service that could automatically acquire some knowledge especially the concepts of a certain discipline from the current HTML-based web pages and then put them to the corresponding knowledge grids. Second, we are going to develop an effective means to refine the raw knowledge provided by users or acquired by the knowledge acquisition service. This concerns techniques for consistent checking, redundant knowledge elimination, and knowledge generalization. Third, we are going to enhance knowledge management services, which could raise work efficiency of a cooperative team, stimulate innovation, and enable the whole knowledge of a team to be accumulated and evolved during the cooperation process in its lifetime. Fourth, we are going to investigate the mechanism for integrating related services to form a new service, and the approach for integrating the knowledge grid with the information grid and the service grid so as to provide Internet users a uniform and multi-level service. Fifth, we are going to investigate the uniform semantics of the generic grid and then to form a grid methodology for modeling complex systems.

Acknowledgements

The author thanks Professor G. Li and Professor Z. Xu for their supports of various aspects in setting up the Knowledge Grid research group. Thanks also go to all members of our research group especially Li Yanyan and Liu Jie for their efforts in the coding task.

References

Broekstra, J., Klein, M., Decker, S., Fensel, D., Harmelen, F. V., & Horrocks, I. (2001). Enabling knowledge representation on the web by extending RDF schema. *Proceedings of the 10th International WWW Conference*, May 2001, Hong Kong, available at http://www.cs.vu.nl/~frankh/abstracts/www01.html.

Decker, S., Melnik, S., Harmelen, F. V., Fensel, D., Klein, M., Broekstra, J., Erdmann, M., & Horrocks, I. (2000). The semantic web: The roles of XML and RDF. *IEEE Internet Computing*, *September/October*, 63–74.

Dieng, R. (2000). Knowledge management and the Internet. *IEEE Intelligent Systems*, *May/June*, 14–17.

Fensel, D., Harmelen, F. V., Horrocks, I., McGuinness, D. L., & Patel-Schneider, P.F. (2001). OIL: An ontology infrastructure for the semantic web. *IEEE Intelligent Systems*, *16* (2), 38–45.

Foster, I. (2000). Internet computing and the emerging grid. *Nature*, *December*, 7.

Heflin, J., & Hendler, J. (2001). *A portrait of the semantic web in action*, *16* (2), 54–59.

Hendler, J. (2001). Agents and the semantic web. *IEEE Intelligent Systems*, *16* (2), 30–37.

Hendler, J., & McGuinness, D. (2000). The DARPA agent markup language. *IEEE Intelligent Systems*, *15* (6), 72–73.

Klein, M. (2001). XML, RDF, and relatives. *IEEE Internet Computing*, *March/April*, 26–28.

Maedche, A., & Staab, S. (2001). A ontology learning for the semantic web. *IEEE Intelligent Systems*, *March/April*, 72–79.

Martin, P., & Eklund, P. W. (2000). Knowledge retrieval and the world wide web. *IEEE Intelligent Systems*, *May/June*, 18–25.

McHraith, S. A., Son, T. C., & Zeng, H. (2001). Semantic web services. *IEEE Intelligent Systems*, *March/April*, 46–53.

Zhuge, H., Ma, J., & Shi, X. (1997). Analogy and abstract in cognitive space: A software process model. *Information and Software Technology*, *39*, 463–468.

Hai Zhuge is a professor at the Institute of Computing Technology, Chinese Academy of Sciences. He was an associate professor and a postdoctoral research fellow at the Institute of Software, Chinese Academy of Sciences. He received a PhD in Computer Science from Zhejiang University, China, in 1992. He visited the National University of Singapore, the City University of Hong Kong, and the Tasinghua University several times as a senior research fellow from 1995 to 2000. His current research interests include knowledge management, grid methodology, problem-oriented model base system, component reuse, cognitive-based software process model, analogical reasoning, interoperation model for group decision, and web-based workflow model. He is now the principle investigator of the China Knowledge Grid project as well as three national grants. His publications appear mainly in IEEE Transactions on systems, Man, and Cybernetics, Information and Management, Decision Support System, Journal of System and Software, Knowledge-based Systems, International Journal of Cooperative Information Systems, Information and Software Technology, and Chinese Journal of Advanced Software Research.