Short Paper

Innovation-Oriented Knowledge Query in Knowledge Grid

LU ZHEN AND ZU-HUA JIANG
Department of Industrial Engineering
Shanghai Jiao Tong University
Shanghai, 200030, P.R.C.

The knowledge grid, based on the Semantic Web, aims at providing the new platform for effectively sharing and managing versatile Web resources including information, knowledge and services. This paper is mainly concerned with the implementation of innovation-oriented knowledge query based on knowledge grid. The architecture of a knowledge query platform based on Ontology is proposed at first. Then we analyze the key techniques: knowledge ontology building, ontology reasoning, semantic register, and semantic parser. At last, a working scenario is employed to illustrate the process of knowledge query on the platform.

Keywords: knowledge query, knowledge grid, grid, innovation, knowledge management

1. INTRODUCTION

In today’s highly competitive and uncertain market environment with short product life cycles, product development must not only satisfy the quality and speed of production, but also ensure that products include innovative values [1]. In the process of innovative design, enterprises have been paying more and more attention to acquire knowledge from outside of them rather than just mining knowledge from inside. Enterprises are both the producers and consumers of knowledge [2]. Therefore, all the enterprises should cooperate with each other in the fields of knowledge supply and consumption, knowledge exchange and purchase, so that the design knowledge among enterprises could be shared among much wider extent and range. Knowledge grid, a platform for sharing and managing globally distributed knowledge resources [3], acts as a carrier for the target of knowledge supply. Based on the knowledge grid, engineers could acquire potentially useful knowledge that may give some hints or inspiration for innovative design. Therefore, how to query and acquire the knowledge for aiding engineers’ innovative design becomes a new aspect in academia. This paper proposed a framework for implementing innovation-oriented knowledge query in the environment of knowledge grid.
2. THE NOTIONS OF GRID AND ONTOLOGY

The “grid” is usually presented in the literatures about grid computing, which refers to the technology that enables a large scale distributed computing system to carry out the controlled sharing of computing resources [4]. Nowadays, the notion of resource in grid has been extended, not just computing resources. A grid 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. Zhuge H. has defined “generic grid” as follow: a set of closely related objects is a grid; an object can take form of knowledge, information, or service [3].

The “ontology” refers to a set of consensual, shared, formal descriptions of important concepts in a domain. It provides structural and semantic definitions of documents [5]. Ontology identifies classes (each characterized by properties that all elements within that class share) and organizes them hierarchically. It also includes important relations between classes and their elements.

Ontology brings “innovation” into the generic information query, and Gird supplies a platform for storing and sharing knowledge resources. Therefore, the combination of above two things will give us a potentially useful solution for the innovation-oriented knowledge query among globally distributed knowledge resources.

3. KNOWLEDGE GRID (KG)

It is mentioned in section 2 that “grid” can be regarded as an integrated mechanism which enables the control sharing of various kinds of resource (e.g. knowledge, information, service, etc.). In the same way, the “Knowledge Grid (KG)” is defined as a mechanism that includes a knowledge space for uniformly storing knowledge and knowledge operation interface for sharing and managing the knowledge resources [3]. Here, we regard the whole knowledge resources as a knowledge space and establish its coordinate grid; then we can accurately store and retrieve any knowledge according to its coordinates.

Knowledge grid enables users or virtual rolls to conveniently publish, acquire, share, manage knowledge resource; or supply knowledge service to demanders, help them in cooperative work, problem solving, decision support and knowledge innovation [6]. The architecture of knowledge grid system is illustrated in Fig. 1. Physical layer consists of five sorts of servers: (1) **Ontology Server**: maintaining a set of basic ontology knowledge service. (2) **Knowledge Server**: store design knowledge including, design cases, patents, design formulas, design rules, software for design or simulation and etc. (3) **Rule Server**: store design rules, innovation principles. (4) **Knowledge Map (K-Map) Server**: supply knowledge index service, including knowledge notes register, semantic register. (5) **Workflow Server**: store the workflow models in the enterprises. Those servers among one sub-grid of enterprises are connected to form a virtual organization, and some sub-grids of enterprises that belong to the same industry may constitute a knowledge union, which will promote knowledge sharing and exchange.
4. INNOVATION-ORIENTED KNOWLEDGE QUERY BASED ON ONTOLOGY

Innovation and innovation in design, in particular, have many interpretations [7]. Innovation, it has been suggested, is mainly concerned with the introduction of something new into a design. It should lead to results that is unexpected (as well as being valuable). In the process of product design, the engineers will be confronted with a series of key problems about product design. It is a shortcut for innovative design to get inspiration and hints from existent experience and knowledge in engineering design. The core issue is how to introduce innovation into the process of knowledge query, so that knowledge could assist engineers with innovative design, rather than the knowledge query is degraded into a search engine.

Innovative thinking is a dialectical unity of the deductive thinking and convergent thinking. The latter is the premise of the former. The level of innovation lies on the degree of the deductive thinking. No matter the deductive thinking or the convergent thinking, it is a helpful way to solve the engineering design problems with the method of association and analogy, which come from the all kinds of connections. Therefore, if we could build a repository of design experience, design knowledge, design principle, design cases and patents; solving the innovative design problems would be simplified as the process of searching the connections between the problems and solutions. This way is named Ontology method in the aspect of innovation engineering [8].

This paper proposed a way of knowledge query based on Ontology, which build the connections between the design problems and solutions using Ontology, and aiding engineers with the innovative design. The platform consists of: knowledge browser, design problem analysis, knowledge query based on Ontology, query parser, query dispatch and other modules, all of which pave the way for enterprises’ technical innovation. The framework of the platform is illustrated in Fig. 2:
At first, through the knowledge browser, users submit a semi-structure formed knowledge query, which is the description of current design problems or bottlenecks. On the basis of the design experience and knowledge, engineers analyze the causality among the design problems and gain the hierarchy of sub-problems, from which the query parser generate a series of query plans; then the query dispatch will execute the query plans and return the results for users.

5. ANALYSIS OF SOME KEY TECHNIQUES

5.1 Ontology

Ontology is the semantic basis for all databases in grid. According to the domain characteristics of the engineering design, this paper proposed a set of ontology which is oriented by innovative design for engineering. Fig. 3 illustrates one part of the ontology.

It mainly described three concepts involving in the domain of engineering design: problem, solution and method.

- **Problem** describes some problems emerging from the process of engineering design. For example: “There is excessive oil consumption in diesel engine”. This problem consists of some common attributes: “Problem ID”, “Problem Name”, “Situation”, “Explanation” and “Remark”; the detail descriptions of problems are “Parameter” and “Degree”, the former is the object of the problem – “oil consumption”, while the latter is the modifying degree of the former – “excessive”. As to the “Parameter”, it also consists of “Parameter ID” and “Parameter Name. In addition, “Parameter” belongs to one certain “Part”; here the “part name” is “diesel engine”. The majority of problems emerging in the engineering design could be described in the form mentioned above, which will bring convenience to solutions’ searching and correlation’s evaluation.
Solution describes the solutions for design problems. They are the accumulation of design experiences, cases and knowledge during the process of engineering design among enterprises. As to the above problem, there is a solution that consists of a series of methods. For example: “Prevent oil penetration in combustion chamber”, “Decrease oil temperature in combustion chamber”, “Prevent oil vapor penetration in crank shaft section” and etc.

Method describes one detail method for problems. For example, “Prevent oil penetration in combustion chamber” is a method, which consists of some common attributes: “Method ID”, “Method Name”, “Constraint” and “Remark”; the detail descriptions of method are “Action” and “Parameter”, the former is the action of the method – “Prevent”, the latter is the object of the action – “Oil penetration”. As to “Parameter”, it also includes “Parameter ID” and “Parameter Name.” In addition, “Parameter” belongs to one certain “Part”; here, the “part name” is “Combustion chamber”. The majority of methods for solving the problems in the engineering design could be described in the form mentioned above, which will bring convenience to methods’ searching.

5.2 Ontology Reasoning

As to knowledge query, the core is retrieving the knowledge according to the descriptions of keywords of knowledge demands (e.g. parts, parameters involved in design problems). It is mentioned that those keywords are derived from the predefined Ontology. So, system could make some ontology reasoning based on rule. Some rules in this paper are shown as follow:

Rule 1: If the “Part” involved in a problem contains the keyword: “para_1”, the parent-class of the “para_1” should be added into the original query.

For example, a user want to solve a problem about “How to reduce temperature of oil”, the involved “Part” is “oil”; the solution about “reduce temperature of liquid” would be also recommended to the user, because the “liquid” is the parent-class of “oil”.
Here, it should be noted that those hierarchical relationships among the concepts are predefined in Ontology.

**Rule 2:** If the “Part” involved in a problem contains the keyword: “para_1”, the sibling-classes of the “para_1” should be added into the original query.

In above example, the solution about “reduce temperature of water” would be also supplied to the user, because the “water” is a sibling-class of “oil”, they shared the same parent-class “liquid”.

**Rule 3:** If the “Part” involved in a problem contains the keyword: “para_1”, the child-classes of the “para_1” should be added into the original query.

According to above rule, the solution about “reduce temperature of gasoline” would be also recommended to the user, because the “gasoline” is a child-class of “oil”.

Sometimes, as to a question, a great deal of solutions will be found based on ontology reasoning. How to optimize the query, and find much more correlative solutions for users’ query? It is the core issue to measure the correlative relationship between the possible solution and the queried problem. This paper proposed a “Correlation” value to reflect the correlative degree of each solution with the queried problem. The “Correlation” value is derived from two concepts’ relationship in the predefined Ontology. For instance, the relationship value between the “oil” and “water” is 0.6, which is pre-assigned in Ontology by knowledge experts; then, as to the problem about “How to reduce temperature of oil”, the “Correlation” value of the solution about “reduce temperature of water” is 0.6. In general, the parent-class has higher “Correlation” value; the child-class takes the second place; the sibling-class is lowest.

With the “Correlation” value, the possible solutions found by knowledge query system could be ranked according to the value. Users could assign a threshold, the solution with higher “Correlation” value could be recommended to users. In above example, the solution about “reduce temperature of liquid” is more prior to be delivered to the user than “gasoline” and “water”.

### 5.3 Semantic Register

In knowledge grid, ontology is the basis of the semantic query; so it is essential to build a bridge between the two categories. Here, the semantic register acts as that bridge. The main task of the semantic register is building the relationships between the concepts, attributes of ontology and tables, fields of databases. The flow for semantic register is illustrated as follow:

**Step 1:** Each table in grid database is regarded as a GDBS (Grid Database Service). So, firstly, users should register a ServiceData which includes some meta-information: physical address, version, current state and a logical name.

**Step 2:** GDBS defines its register tuple including: tables in the databases, fields of tables, the mapping from tables to concepts, the mapping from fields to attributes and service’s logical name mentioned in the previous step.
Step 3: The ontology service module will validate the above register tuple. If pass, the register is finished; otherwise, return the error information.

According to the ontology shown in Fig. 3, the following Table 1 is the corresponding semantic register table:

<table>
<thead>
<tr>
<th>GDBS ID</th>
<th>Grid Database</th>
<th>Table Name</th>
<th>Semantic note</th>
<th>Register of fields mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>GS1</td>
<td>Nova_Solution</td>
<td>Solution _Table</td>
<td>Solution</td>
<td>Sola_ID→ Design_Solution.Solution_ID → Sola_Name → Design_Solution.Solution_Name → Design_Solution.Remark → ...</td>
</tr>
<tr>
<td>GS2</td>
<td>Nova_Solution</td>
<td>Solution _Problem</td>
<td>Solution</td>
<td>Prob_ID→ Design_Solution.Solution_ID → Prob_Name → Design_Solution.Problem_Name → Design_Solution.Problem.Remark → ...</td>
</tr>
<tr>
<td>GS3</td>
<td>Nova_Solution</td>
<td>Solution _Component</td>
<td>Solution</td>
<td>Comp_ID→ Design_Solution.Solution_ID → Comp_Name → Design_Solution.Component_Name → Design_Solution.Component.Remark → ...</td>
</tr>
<tr>
<td>GS4</td>
<td>Nova_Problem</td>
<td>Problem _Table</td>
<td>Problem</td>
<td>Prob_ID→ Design_Problem.Problem_ID → Prob_Name → Design_Problem.Problem_Name → Design_Problem.Problem.Remark → ...</td>
</tr>
<tr>
<td>GS5</td>
<td>Nova_Problem</td>
<td>Problem _Parameters</td>
<td>Problem</td>
<td>Para_ID→ Design_Problem.Problem_ID → Para_Name → Design_Problem.Parameter_Name → Design_Problem.Parameter.Remark → ...</td>
</tr>
<tr>
<td>GS6</td>
<td>Nova_Parameters</td>
<td>Parameter _Table</td>
<td>Parameter</td>
<td>Para_ID→ Parameter_ID → Para_Name → Parameter.Remark → ...</td>
</tr>
</tbody>
</table>
sign problems; then, it will be converted into the query plans which are executable in the physical layer, such as SQL-Query or XQuery (for XML documents) etc.

The flow of semantic parsing is listed as follow:

**Step 1:** According to the emerging problems, engineers analyze the causalities among the problems and gain a causality chart, named “fishbone chart”, to describe the causality hierarchy for the problems.

**Step 2:** On the basis of Ontology reasoning, a series of semantic queries are generated in the form of Q3 query from above “fishbone chart”.

**Step 3:** Since the Q3 query and Semantic register information of databases are based on the same semantic register table, it will not be difficult for each Q3 query to be converted into a series of SQL query plans, named Single Term Query Plan.

**Step 4:** Each Q3 query is corresponding to one series of query plans, which could be emerged into the Merged SQL Query Plan.

### 6. A WORKING SCENARIO

This section will present a working scenario of the innovation-oriented knowledge query platform based on knowledge grid.

For example, one certain enterprise has been confronted with such a problem: “The oil consumption of diesel is excessive”. As to that design bottleneck, the enterprise should bring out an innovative solution to solve it and overcome that obstacle in the process of engineering design.

![Knowledge Browser on Innovation-Oriented Knowledge Grid](image)

**Fig. 4.** The fishbone chart for the design problem.
At first, engineers should analyze the problem based on their design experience and general knowledge. They may think about: what are the causes of the problem? What are the sub-problems, if they had been solved, then it would result in solving the final problem. In the same way, as to the sub-problem, engineers could make the further analysis to find sub-sub-problems and so on. The Fig. 4 is the example of the fishbone chart for the above problem:

According to the fishbone chart, we could gain:

Design Problem: There is excessive oil consumption of diesel
Design Sub-Problem: How to reduce temperature of oil
Design Sub-Problem: How to prevent oil penetration of crank shaft
Design Sub-Problem: How to reduce temperature of wall of tank
Design Sub-Problem: How to increase efficiency of water cooling system

The italic words with underline denote the key words in the design problems. According to those key words, a semantic based Q3 query expression will be generated as to each design problems. Here, a Q3 query example, corresponding to the final design problem, is shown as follow:

q3:content [ q3:prefix (idk: http://nova-design-kdg.org/schemas)
q3:variable{ x1 a idk:Design_Solution,
x2 a idk:Design_Problem,
x3 a idk:Parameter,
x4 a idk:Part } ]
q3:pattern [ x1.idk:Solution_ID,
x1.idk:Solution_Name ]
q3:constraint [ x1.idk:Solve = x2,
x2.idk:Object = x3,
x3.idk:Parameter_Name = “oil consumption”,
x3.idk:Belongto = x4,
x4.idk:Part_Name = “diesel” ]

Table 2. The process of the Q3 query’s semantic parsing.

<table>
<thead>
<tr>
<th>Q3</th>
<th>Single Term Query Plan</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1.idk:Solution_ID</td>
<td>Select GS1.Solu_ID from GS1</td>
</tr>
<tr>
<td>x1.idk:Solution_Name</td>
<td>Select GS1. Solu_Name from GS1</td>
</tr>
<tr>
<td>x1.idk:Solve = x2</td>
<td>Select * from GS2 where GS2.Problem_ID = GS5.Problem_ID</td>
</tr>
<tr>
<td>x2.idk:Object = x3</td>
<td>Select * from GS5 where GS5.Parameter_ID = GS6.Parameter_ID</td>
</tr>
<tr>
<td>x3.idk:Parameter_Name = “oil consumption”</td>
<td>Select * from GS6 where GS6.Parameter_Name = “oil consumption”</td>
</tr>
<tr>
<td>……</td>
<td>……</td>
</tr>
</tbody>
</table>

For sub-problems, we could also gain the Q3 queries in the same way. So, we could get five Q3 queries. Due to the limit of space, the other four Q3 queries are omitted here. As to each Q3 query, a series of Single Term Query Plans would be generated through
the semantic parser. During the parsing process, it needs the support of the semantic register table, based on which all the expressions in Q3 query are parsed one by one. Continue above example:

The above Single Term Query Plans will be merged into the Merged SQL Query Plan, which is shown as follow:

```sql
Select GS1.Scheme_ID, GS1.Scheme_Name
from GS1, GS2, GS5, GS6
     and GS5.Parameter_ID = GS6.Parameter_ID and GS6.Parameter_Name = "oil consumption"
```

The following task: transform above Merged SQL Query Plan over the multiple distributed databases into the query aiming to different GDBSs, which are located in different points among the grid. As to above example, the detail of the query dispatch process is shown as follow:

1. Resolve the query of users, and generate a Single_Relation for each RefName:
   - Single_Relation_1 (The Table “Solution_Table” corresponding to GS1)
   - Single_Relation_2 (The Table “Solution_Problem” corresponding to GS2)
   - Single_Relation_3 (The Table “Problem_Parameter” corresponding to GS5)
   - Single_Relation_4 (The Table “Parameter_Table” corresponding to GS6)

2. Generate query expressions for each Single_Relation:
   - Single_Relation_1: ```select Solution_Table.Solution_ID, Solution_Table.Solution_Name
                   from Solution_Table```
   - Single_Relation_2: ```select Solution_Problem.Solution_ID, Solution_Problem.Problem_ID
                     from Solution_Problem```
                     from Problem_Parameter```
   - Single_Relation_4: ```select Parameter_Table.Parameter_ID, Parameter_Table.Parameter_Name
                     from Parameter_Table```

3. Generate the query expression for Merged_Relation:
   ```select Single_Relation_1.Scheme_ID, Single_Relation_1.Scheme_Name from Single_Relation_1, Single_Relation_2, Single_Relation_3, Single_Relation_4
   where SingleRelation_1.Scheme_ID = Single_Relation_2.Scheme_ID and
       SingleRelation_2.Problem_ID = Single_Relation_3.Problem_ID and
       SingleRelation_3.Parameter_ID = Single_Relation_4.Parameter_ID and
       Single_Relation_4.Parameter_Name = "oil consumption"
```

At last, system executes the above dispatched query expression, and return the final results to the users in the form as Fig. 5, in which some design knowledge or cases as to the design problem: “There is excessive oil consumption in diesel engine”, are list in the left part of the knowledge browser. That knowledge will give some hints and inspiration to engineers and aid them with the innovative design.

As shown in Fig. 5, the design knowledge as to each design problem or sub-problem is divided into three categories: Precise, General and Analogous, which reflect the correlation degree between the knowledge and the problem. For example, as to the problem:
INNOVATION-ORIENTED KNOWLEDGE QUERY IN KNOWLEDGE GRID

611

Fig. 5. The result of knowledge query.

“how to reduce the temperature of oil”, the knowledge in Precise category is corresponding to the problem exactly; the General category is just related to the problem at some aspects, such as “how to reduce the temperature of liquid”; the Analogous category includes the cases with much lower correlation than above.

7. DISCUSSION AND SUMMARY

Information query on Web, Knowledge Grid and CAI are three hotpots in academia. In each aspect, plenty of research achievements have been emerged these years. The work presented in this paper is the multidisciplinary research over the above three aspects. It proposed a new idea and method named “innovation-oriented knowledge query based on knowledge grid”.

Comparing with the previous work of research work of other scholars, the major difference lies in: This paper focus on knowledge query in grid, just one aspect of knowledge grid; while majority of scholars have been focused on the generic architectures of knowledge grid: e.g. Fran Berman [9] and Cannataro [10] have proposed their frameworks of knowledge grid respectively, which have capabilities of inference and question-answering through the data mining of gird. Zhuge Hai also proposed his architecture of knowledge grid in 2001 [3], and published the first monograph “knowledge grid” in this aspect [11]. In the application aspects, Wu Zhaohui proposed the architecture of knowledge base grid, and constructed TCM-Grid [12] (Traditional Chinese Medicine Grid) which set an example for the applications of the knowledge grid.

This paper bears significant relationships with aforementioned efforts. However, the following characteristics distinguish our efforts from the past efforts: (1) We aim to address knowledge management just in the area of engineering design, rather than discuss generic knowledge of human beings. (2) We bring the “innovation” into the knowledge
query, rather than information query on Web such as search engine. (3) We propose an innovation-oriented knowledge query platform model, and the knowledge query would support enterprises’ innovative design activities.

Innovation is becoming more and more crucial in engineering design. Moreover, the knowledge grid is a trend for KM (Knowledge Management). Since the knowledge query is an important module in knowledge supply system, this paper focused on the innovation-oriented knowledge query in knowledge grid, which will pave the way for developing innovation-oriented knowledge supply grid system in future.

REFERENCES


Lu Zhen (镇璐) is a doctor candidate at the Department of Industrial Engineering, Shanghai Jiao Tong University, P.R.C. His current research interests include: knowledge management, knowledge flow and knowledge supply. He has published 8 papers in referred International conferences and journals.
Zuhua Jiang (蒋祖华) is a professor at the Department of Industrial Engineering, Shanghai Jiao Tong University, P.R.C. His current research interests include: knowledge management, cooperative design, concurrent engineering. He has published 50 papers in referred International conferences and journals, including 10 SCI journal papers and 25 EI journal papers.