©2007 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE
Application of Protégé and SPARQL in the Field of Project Knowledge Management

Hai Dong
Curtin University of Technology
hai.dong@cbs.curtin.edu.au

Farookh Khadeer Hussain
Curtin University of Technology
hussain.farookh@cbs.curtin.edu.au

Elizabeth Chang
Curtin University of Technology
elizabeth.chang@cbs.curtin.edu.au

Abstract

Protégé is a set of open-source ontology design software developed in Stanford Medical Informatics. SPARQL (Protocol and RDF Query Language) is recommended by W3C, to represent the RDF (Resource Description Framework) graph – a set of triples that consist of a subject, a predicate and an object as the basic expression of data stored in OWL-based knowledge base.

In this paper, we propose an ontology-based project knowledge management methodology, by means of Protégé and SPARQL, to solve the issues in project management activities. By introducing a set of new ontology notations, we present the conceptual model of our ontology to realize the function of knowledge management in project organizations. Following that, we realize the prototype in Protégé and validate it by means of SPARQL. Finally we make comments on our project and plan our future work.

1. Introduction

One typical feature of current project organizations is that people in the organizations are geographically dispersed [3]. With the increase of project outsourcing, project groups and its members are probably located in different areas, from different cultural background and even speaking with different languages [2]. Knowledge management activities among these groups and members could be challenged by the cultural issues and geographical issues.

OWL (Web Ontology Language) is a web language which intends to interpret the web information into the machine-readable content with semantics [4]. Protégé-OWL which is a set of open-source ontology design software developed in Stanford Medical Informatics [5]. SPARQL (Protocol and RDF Query Language) is recommended by W3C, to represent the RDF (Resource Description Framework) graph – a set of triples that consist of a subject, a predicate and an object as the basic expression of data stored in OWL-based knowledge base [6].

In this paper, we propose an ontology-based project knowledge management methodology, by means of Protégé-OWL and SPARQL. By introducing a set of new ontology notations, we present the conceptual model of our ontology to realize the function of project knowledge management. Following that, we realize the prototype in Protégé-OWL and validate the prototype by means of SPARQL. Finally we make comments on our project and plan our future work.

2. Notation system for ontology representation

Before we represent the ontology model for project knowledge management, a notation system is introduced in this section. The notation system utilized in the ontological representation is based on Chang et al. [1]’s work, which consists of three basic notations as Table 1. Although in the past we usually used to employ UML to represent ontology model, due to its complex symbols categories, UML cannot efficiently help people better understand the shared knowledge. This notation system simplifies the symbols and its symbols are closer to the principle of ontology which is the combination of shared concepts and relationships between concepts [1]. (Table 1)

<table>
<thead>
<tr>
<th>Table 1. Ontology notation system</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ontology Notation</strong></td>
</tr>
<tr>
<td>----------------------------------</td>
</tr>
</tbody>
</table>
3. Project knowledge management ontology

In this section, we propose the project knowledge management ontology, which consists of the hierarchy of project organization domain concepts and the ontology of employee and its sub-composition, project ontology and criterion ontology.

3.1. Hierarchy of Project Organization Domain Concepts

In a Project Organization Domain, the Project Organization concept can be seen as a combination of Employee concept and Project concept. Employee also consists of:

- **CEO** who is responsible for managing all projects in Project Organization;
- **Director** who is responsible for managing the projects which belong to his/her department in the Project Organization;
- **Manager** who is responsible for managing the projects which belong to his/her division in each department;
- **Personnel** who are responsible for the implementation of arranged projects.

The graphical view of Hierarchy of Project Organization Domain Concepts is shown in Fig. 1 through the use of the ontology notation.

3.2. The ontology of employee and its sub-compositions

In a project organization, the Employee Ontology is defined as the conceptualization of the Employee who has an Employee Position in the organization and is identified by an Employee Name as well as has Responsibilities which include some Projects. (Fig. 2)

![Employee Ontology](image)

**Figure 2. Employee Ontology**

We present the Employee Ontology as the combination of the ontology name and a tuple where the elements of the tuple can be complex elements as defined below:

Employee [Employee Position, Employee Name and Responsibilities] where:

- **Employee Position** is a unique identification of Employee in a project organization.
- **Employee Name** is a unique identification of Employee Position in a project organization.
- **Responsibilities** is an aggregation of Projects which Employees should take part in. Different Employee Positions are in correspondence with different Responsibilities.

The four sub-compositions of Employee Ontology – CEO Ontology, Director Ontology, Manager Ontology and Personnel Ontology inherit all the relations from Employee Ontology and the only difference is the scopes of the inherited concepts’ properties.

3.3. Project Ontology

In a project organization, the Project Ontology is defined as the conceptualization of the concept of Project that is identified by Project Code, is shown Date Started, is responsible to Employee and is evaluated by Project Status. (Fig. 3)
3.4. Criterion Ontology

In project organization environments, the Criterion Ontology is defined as the conceptualization of the concept of Criterion that is identified by Criterion No., is shown Date Logged, is responsible to Employee and is determined by Criterion Status which has the attributes of Criterion Completeness, Criterion Clarity and Criterion Importance. (Fig. 4)

We represent the Criterion Ontology as the combination of the ontology name and a tuple where the elements of the tuple can be complex elements as defined below:

Criterion [Criterion Name, Criterion No., Date Logged, Responsible Persons, Criterion Status, Criterion Completeness, Criterion Clarity and Criterion Importance] where:

‘Criterion Name’ usually refers to a Criterion itself. In project organization environments, a Criterion Name is seen as a unique identification for Criterion.

‘Criterion No.’ is the mixture of numerical symbols and alphabetic symbols, which also can be seen as the unique identification for Criterion. The use of Criterion No. mainly focuses on the storage of Criteria’s records in databases, which is beneficial to the pick-up and the storage of Criteria’s documentations.

‘Date Logged’ refers to the date when a criterion has been mutually agreed between an evaluating person and an evaluated person.

‘Responsible Persons’ is an aggregation of Employees who are relevant to a Criterion.

‘Criterion Status’ is a sub-tuple of the Criterion tuple, which uses quantitative means to determine the
extent to which a criterion has been completed or delivered up on the mutually agreed Criterion. It consists of three elements – Criterion Completeness, Criterion Clarity and Criterion Importance.

‘Criterion Completeness’ is an element of Criterion Status, which qualifies the extent of task completion according to its corresponding Criterion.

‘Criterion Clarity’ is an element of Criterion Status, which qualifies the extent whether a Criterion is mutually agreed between an evaluating person and an evaluated person or not in a Project. Its scope is as below:

‘Criterion Importance’ is an element of Criterion Status, which expresses the importance of a Criterion in a Project.

4. Application of Protégé and SPARQL to realize project knowledge management ontology

The whole implementation and validation process involves four steps: firstly realizing the project knowledge management ontology by means of Protégé-OWL; secondly creating instances for the built ontology concepts in Protégé-OWL; thirdly querying the ontology instances by means of SPARQL; finally comparing the query results with the ontology instances entered in the second step to validate the ontology. Thus, if the results completely match the data stored in the OWL-based knowledge base, the ontology is validated.

In the first step, we build the ontology concepts, the relationships between these concepts and the constraints of the relationships. The graphical view of Hierarchy of Project Organization Concepts in Protégé-OWL is shown in Fig. 5.

After building the ontology model, by survey and collecting data from City of Melville council (www.melville.wa.gov.au) – a real project organization, we create the instances for the ontology concepts based on the council management structure and the hypothesis of project task distribution. (Fig. 6)

Figure 5. Hierarchy of ontology concepts in Protégé-OWL

The code below is to query the data from Employee Relationship Ontology which is a tree-like management structure, including CEO, Director, Manager and Personnel. Query results are displayed in Fig. 7.

SELECT DISTINCT ?CEO ?Director ?Manager ?Personnel
WHERE
{
?CEO :manage1 ?Director.
?Director :manage2 ?Manager.
?Manager :manage3 ?Personnel
}
ORDER BY ?CEO ?Director ?Manager ?Personnel

Figure 7. Query results of Employee Relationship Ontology

The code below is to query the data from Employee Ontology, including employee position, employee name and responsible projects. Query results are displayed in Fig. 8.

SELECT DISTINCT ?Employee ?Employee_Name

Figure 6. Instances of Project Organisation Ontology
WHERE

Employee :holdsa Employee_Name.
Employee :has Responsibilities.
Responsibilities :include Responsible_Project.

ORDER BY Employee

Figure 8. Query results of Employee Ontology

The code below is to query the data from Project Ontology, including project name, project code, date project starts, value of project completion extent and responsible people. Query results are displayed in Fig. 9.

SELECT DISTINCT Project, Project_Code, Date_Started, Project_Status, Responsible_People
WHERE

Project :isIdentified Project_Code;
:isShownDateStarted Date_Started;
:isResponsible Responsible_People;
:isEvaluated Project_Status.
Date_Started = dateStarted;
Project_Status = isShownProjectStatus Project_Status.
Responsible_People :involve Responsible_People.

ORDER BY Project

Figure 9. Query results of Project Ontology

The code below is to query the data from Criterion Ontology, including criterion name, criterion code, its belonged project, date criterion are logged, value of criterion completion extent, value of criterion clarity extent, value of criterion importance extent and responsible persons. Query results are displayed in Fig. 10.

SELECT DISTINCT Project, Criterion_No, Date_Loged, Criterion_Completeness, Criterion_Clarity, Criterion_Importance, Criterion_Name, Criterion_Reponsible_People
WHERE

Project :isDivided Criterion.
Criterion :isIdentified Criterion_No;
:isShownDateLoged Date_Loged;
:isDetermined Criterion_Status;
:isResponsible RP.
Date_Loged = dateLoged;
Criterion_Status :hasAttribute Criterion_Completeness;
:hasAttribute Criterion_Clarity;
:hasAttribute Criterion_Importance.
Criterion_Completeness :isShownCriterionCompleteness Criterion_Completeness.
Criterion_Clarity :isShownCriterionClarity Criterion_Clarity.
Criterion_Importance :isShownCriterionImportance Criterion_Importance.
RP :involve Criterion_Reponsible_People.

ORDER BY Project, Criterion

Second International Conference on Systems and Networks Communications (ICSNC 2007)
0-7695-2938-0/07 $25.00 © 2007 IEEE
By means of SPARQL, we can find that most query results match the instances we enter in the ontology within Protégé-OWL, which reveals that the ontology realizes most functions of proposed project knowledge management. In other words, this ontology is validated by experimental method.

6. Conclusion and possible future works

In the paper, we observe the problematic situation of knowledge management activities in project organizations. To solve the issues, we utilize the theory of ontology into the field of project management and design a solution based on an ontology notation system from Chang et al.’s works. To realize the function of this ontology, we introduce Protégé-OWL and SPARQL to build and validate the ontology prototype.

The benefits of this project are concluded as below:

- It realizes the function of tracking and tracing projects completion status from the perspective of project management, which is beneficial to improving the competitive ability of project organizations.
- It simplifies the procedure of project knowledge management activities and identifies own task and assessment criteria for each member in a project organization. Thus, the methodology may save the time and the cost of projects completion and help personnel self-evaluate.
- It adopts quantitative methodology to measure the project completion status, which is effortlessly understood by organizational management.

The limitation of the project is concluded as below:

- The ontology is not tested in practice, and thus we cannot validate its actual contribution to project knowledge management activities in project organizations.

Based on the issue in the current project, in the future we will implement the methodology in the environment of real project organizations to validate its functions. In addition, we will add more contents to the methodology based on different implementation situations.

7. References