An Ontological Knowledge-base System for Composing Project Team Members

Yu-Liang Chi
Dept. of Information Management, Chung Yuan Christian University, 200 Chung-Pei Rd., Chung-Li, Taiwan 320, Republic of China

Keywords: Team Member Composition, Knowledge-based System, Ontology, Semantic Rules.

Abstract: In teamwork-based projects, human play a critical role in achieving project success. This study utilizes ontological approaches to build project teaming models into ontology. It helps to develop a set of logic rules for identifying semantic relationships between individuals. By following a knowledge base creation process, the factual data of project, workers, and teaming factors can be inserted into ontological knowledge base. Based on knowledge inference, reliable knowledge bases are established for selecting project team members in runtime. A case study is presented to demonstrate the effectiveness of the proposed design.

1 INTRODUCTION

Collaboration is a major feature of teamwork-based projects, which are frequently implemented by high performance project teams. Effective project teaming thus has become essential in human-side project management. In project management, project teaming refers to managing a project team with assignment of project tasks and roles (Beranek *et al.*, 2005), and the appropriate composition of the development or workplace team that performs ad-hoc project tasks. Industry experts and academic researchers continue to work on identifying factors and composition approaches for effective project teaming. While current methods and considerations are presented mostly as predefined and syntactic criteria, further consideration of the effect of derived semantic relations and facts should also be carried out. The characteristics typically considered in composing a quality team include team size, personal commitment, current workload of the individual, leadership, skill competence, years of experience, communications skill, and so on (Chen and Lin, 2004). A need for cross-functional composition with regard to the skill backgrounds of team members is recognized in projects and is multi-disciplinary in nature. Configurational and task-oriented approaches to project teaming require the composition of a team to depend on tasks of the project work (Coates *et al.*, 2007). Such tasks contribute to the technical and explicit foundations of a software project team.

A solid technical foundation alone does not guarantee a quality composition of the project team. For example, Krishnan (1998) reported that the effects of three team-related measures include not only the domain and language experience of the team, but also the capabilities of the team personnel with regard to information system product costs and quality. This is particularly true when it is recognized that culture and human or "soft" factors, for example differences in individual characteristics of preferences, also contribute to team success (Gorla and Lam, 2004). Regarding personality, the Myers-Briggs Type Indicator has been widely employed to assess software engineer personality types (Stewart *et al.*, 2005), as well as to assess the influence of team member personality namely Agreeableness, Conscientiousness, Extraversion, Neuroticism, and Openness to experience, also known as the "big-five personality factors" on individual role, social role and task accomplishment. Thus, the human-side of project management should be integrated into technological project management methods and tools.

2 DEVELOPING TEAM MEMBER COMPOSITION MODEL

Project team knowledge has numerous sources and different aspects. To build the knowledge model of

Chi Y..
An Ontological Knowledge-base System for Composing Project Team Members.
DOI: 10.5220/0003950602970300
In *Proceedings of the 14th International Conference on Enterprise Information Systems* (ICEIS-2012), pages 297-300
ISBN: 978-989-8565-10-5

project teaming, his study refers to the discipline of knowledge engineering (KE) proposed by Guarino (1995). Typical KE includes expertise gathering, knowledge model building, and knowledge representation. Detailed steps are described as the followings:

Expertise Gathering: Expertise gathering is the focus of identifying critical elements of project teaming. Observation of project teaming events identifies three primary subjects including *Project, Worker* and *Teaming factors*. Further expertise gathering are also implemented, including elicitation, analysis, and transformation are implemented.

- Property elicitation. Over 100 properties are gathered during this stage. For example, the collected properties of projects are basic features such as project's budget and skill needs.
- Analysis. A total of 35 properties are identified. New or subordinate subjects are generated by either separating existing subjects or assembling relevant subjects. For example, the subjects *Project_Type*, *Title*, *Skills* and *Personality* are subordinate to the *Teaming_Factors* subject.
- Transformation. This step transforms subjects and their dependent properties into an ontological representation. The representation is formed by a formatted expression written as *{**Concept**: Property list}*. Examples are presented below.

{**Project**: *Project_Description, PM, Budget, Number_of_HR, Skill_Need, PM_Skill_Need, Length_of_Time, has_Type, Qualified_Basic_Worker, Watch_List, …* }

{**Worker**: *Age, Gender, Seniority, Salary, has_Title, has_Skill, member_of, Hostile_to, has_Experience, …*}

Building Knowledge Model: A knowledge model is based on an abstract view of the task domain, and can be used as an intermediary between the real world and information systems. Two type of relations including "*is-a*" and "*has-a*" are developed. The "*is-a*" denotes an inheritance relationship between two concepts. For example, the *Teaming_factors* concept has sub-concepts such as *Title, Skills and Personality*. The "*has-a*" denotes the "*part-of*" relation between two concepts. These properties stipulate a schema for describing the concepts. Users can employ these properties to contribute their factual knowledge to the knowledge base or to obtain implicit knowledge via inference mechanisms.

Furthermore, two types of property's content including "Asserted property" and "Inferred property" need to be defined during model building. The asserted properties provide the basis of inference engine to deduce new knowledge. On the other hand, the content of inferred property is implicit, but can be obtained by inferring factual knowledge via a reasoned. The inferred property plays a critical role for rule-based reasoning.

Knowledge Representation: This study employs Web Ontology Language (OWL) as the notation and formalism for representing the knowledge to be stored in ontology. After constructing the team composition model, OWL is utilized for knowledge representation. OWL is highly appropriate for representing structured knowledge using classes and properties organized in taxonomies.

3 CREATING RELATIONSHIPS USING SEMANTIC RULES

Horrocks and Patel-Schneider (2004) have reported several limitations and issues of OWL in syntax and computation, particularly in relationships between roles chains using rules, causing indeductibility, logical undecidability, by embedding the word problem in inferences. The rules apply the syntax "*Antecedent* ➔ *Consequent*". Both antecedent and consequent are conjunctions of atoms of the form $atom_1 \wedge .. \wedge atom_n$, where a variable is indicated by a question mark (e.g., *?x*). The semantic rules are used to extend the power of the ontological approach to identify semantic relationships between instances.

This study utilizes the *Project_Type* concept to manage characteristics of typical historical projects as best practices. Several properties used in rules development are detailed below. The *PT_Skill_Need* and *PT_PM_Skill_Need* properties are used to indicate the skills needed by workers and project managers, respectively. Furthermore, the *PT_Personality_Need* property describes preferred personality types for performing the project. These properties can be used to develop rules to connect other concepts such as *Worker* to obtain candidate members for a project team. The following five rules are examples developed in this study.

Rule-1 is used to identify qualified team members with reference to best practice. Rule-2 helps identify candidate project manager(s) based on the qualified workers with reference to best practices. Rule-3 is applied to group senior workers as candidate team members. Rule-4 adds the *PT_Personality_Need* property to deduce whether the qualified workers possess the preferred personalities. Rule-5 examines whether a qualified worker is hostile to someone then both of them will be inserted into the *Watch_List* property of the project.

Rule-1: Project(?x) ∧ has_Type(?x, ?y) ∧ PT_Skill_Need(?y, ?z) ∧ Same_Skill_Worker(?z, ?a) ➔ Qualified_Basic_Worker(?x, ?a)
Rule-2: Project(?x) ∧ has_Type(?x, ?y) ∧ PT_PM_Skill_Need(?y, ?z) ∧ Same_Skill_Worker(?z, ?a) ∧ has_Title(?a, Project_Manager) ➔ PM(?x, ?a)
Rule-3: Project(?x) ∧ has_Type(?x, ?y) ∧ PT_Skill_Need(?y, ?z) ∧ Same_Skill_Worker(?z, ?a) ∧ Seniority(?a, ?b) ∧ swrlb:greaterThan(?b, 5) ➔ Qualified_Advanced_Worker(?x, ?a)
Rule-4: Project(?x) ∧ has_Type(?x, ?y) ∧ PT_Skill_Need(?y, ?z) ∧ Same_Skill_Worker(?z, ?a) ∧ PT_Personality_Need(?y, ?b) ∧ has_Personality(?a, ?b) ➔ Quality_Intensive_Worker(?x, ?a)
Rule-5: Project(?x) ∧ has_Type(?x, ?y) ∧ PT_Skill_Need(?y, ?z) ∧ Same_Skill_Worker(?z, ?a) ∧ Hostile_to(?a, ?b) ➔ Watch_List(?x, ?b)

4 CASE STUDY

Before implementing this case study, known facts (instances) of concepts must be identified. For example, instances about workers, including *age*, *salary*, and *skill*, must be given into the asserted properties. Some instances regarding the example scenario are detailed below. Table 1 lists known instances of the *Project_Type* concept. Each instance involves three known property values, such as skills required of the project manager, skills required of workers, and the personalities preferred by the project. These instances are considered to represent the best practices for future projects.

Table 1: Instance samples of the *Project_Type*.

Type	PM Skills*	Member Skills*	Personalities**
BPM	PC; PMC; PP	BM; CM; SAD; DP	High_A; High_E
ERP	PC; PMC; PP	BM; CUT	High_C; High_E
GCM	PP	CM; UAT;DP	Low_N; High_C
HRM	PP	QA; TR	Low_N; High_C
MES	PMC; PP;	RD; TR; SAD	High_A; High_E
PLM	PP; CM	BM; CM; CUT	High_O; High_C

*Skills. BM: business modeling; CM: configuration management; CUT: coding and unit test; DP: deployment; PC: project closure; PMC: project monitor and control; PP: project planning; and etc.
**Personalities. A: Agreeableness; C: Conscientiousness; E: Extraversion; N: Neuroticism; O: Openness

Table 2 lists partial instances of the *Worker* concept. The row headings indicate the property names for each instance. A worker comprises eight known property such as title and gender. The *has_Skill* property presents a list of skill items. The symbol '×' indicates that a worker has this corresponding skill. In the *Personality* property, the symbols *N, A, C, E, and O* denote Neuroticism, Agreeableness, Conscientiousness, Extraversion and Openness respectively. Furthermore, the symbol '+' represents positive psychological power, while the '-' indicates negative psychological power. Total 17 persons are identified for the following experiments.

Table 2: Instance samples of the *Worker*.

Name	Allen	Alvin	Cindy	Eric	Leon	Mavis	Phil	Stan	Ted
Title	IC_L.	IC	BC	PM	PM	IC_L.	PM	BC	IC
Gender	M	M	F	M	M	F	M	M	M
Salary	48k	52k	40k	70k	160k	50k	120k	67k	70k
Seniority	7	3	2	2	3	8	5	8	7
has_skills									
BM			×					×	
CM		×							
UAT			×		×				
Hostile_to	-	Ian	-	Flying	-	Jeff	Stan Ted	Phil	Phil Eva
Personality									
N	+		+			+	+		
O				–		–	–		

The first case uses instances of *Project_Type* as a reference for best project practices. For example, when a project is newly created, the decision makers identify the project has having typical features like the *BPM*. As shown in Figure 1, a user selects the *BPM* as a known fact in the *has_Type* property. This property value is initially the only factual knowledge associated with the new project. After firing the JESS rule engine, the project obtained five inference results as presented within inferred properties. The rules engine utilizes known facts of the *BPM* to provide for the computational needs of Rule-1 to 5. For example, Rule-1 is applied to identify qualified workers using the instances in the *PT_Skill_Need* property of the *BPM*, including *BM, CM, SAD* and *DP*. A total of nine workers were inferred into the *Qualified_Basic_Worker* property. Rule-2 deduced two qualified project managers such as *Eric* and *Leon* for this new project. Rule-3 deduced four candidate workers for *Qualified_Advanced_Worker* property. These workers are all highly qualified and each had over 5 years of working experience. Rule-4 treats preferred personalities as noted criteria in the property of the *BPM*. A total two workers were recommended inside the *Quality_Intensive_Worker* property. These workers have at least one personality item conforming to *Agreeableness, Extraversion*, or both. Finally, Rule-5 contributed five workers to the *Watch_List* property. These workers may have interpersonal relationship issues based on the record of the *Hostile_to* property of the *BPM*.

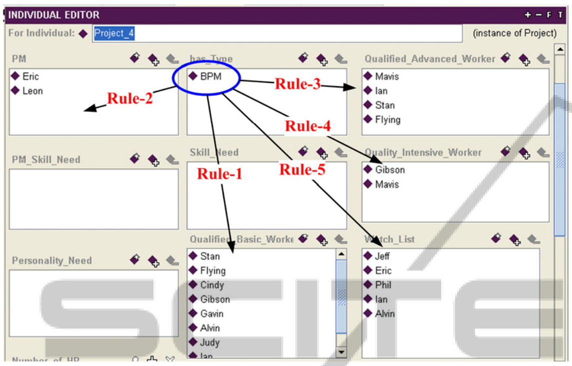

Figure 1: Using the instance of *Project_Type* as a reference to infer candidate team members.

5 DISCUSSION AND CONCLUSIONS

This study employs OWL as the notation for representing knowledge to be stored in the ontology. SWRL rules are applied to infer semantic relationships of instances. Once ontology and rules are used for knowledge representation, it is possible to stipulate practical facts as factual knowledge. The experimental results demonstrate that the proposed design can support the system for identifying appropriate project teams. Additionally, the proposed design stresses that the system can be continually maintained by factual knowledge providers rather than system developers. The inference mechanism then helps establish a new and complete knowledge base for maintaining system reliability. Consequently, the combination of semantic rules and ontologies can manage intricate information such as the project teaming problem mentioned in this study.

ACKNOWLEDGEMENTS

The author would like to thank the National Science Council of the ROC for financially supporting this research under Contract No. NSC 99-2410-H-033 -028 -MY3.

REFERENCES

Beranek, G., Zuser, W., and Grechenig, T. (2005). Functional group roles in software engineering teams. *ACM SIGSOFT Software Engineering Notes*, *30*(4), 1-7.

Chen, S. J., and Lin, L. (2004). Modeling team member characteristics for the formation of a multifunctional team in concurrent engineering. *IEEE Transactions on Engineering Management*, *51*(2), 111-124.

Coates, G., Duffy, A. H. B., Hills, W., and Whitfield, R. I. (2007). A preliminary approach for modeling and planning the composition of engineering project teams. *Proceedings of the Institution of Mechanical Engineers*, *221*(7), 1255-1265.

Gorla, N., and Lam, Y. W. (2004). Who should work with whom: Building effective software project teams. *Communications of the ACM*, *47*(6), 79-82.

Guarino, N. (1995). Formal ontology, conceptual analysis and knowledge representation. *International Journal of Human-Computer Studies*, *43*(5-6), 625-640.

Horrocks, I., and Patel-Schneider, P. F. (2004). Reducing OWL entailment to description logic satisfiability. *Journal of Web Semantics*, *1*(4), 345-357.

Krishnan, M. S. (1998). The role of team factors in software cost and quality: An empirical analysis. *Information Technology and People*, *11*(1), 20-35.

Stewart, G. L. Fulmer, I. S., and Barrick, M. R. (2005). An exploration of member roles as a multilevel linking mechanism for individual traits and team outcomes. *Personnel Psychology*, *58*(2), 343-365.