A Framework for Collaborative Project Planning Using Semantic Web Technology

Lijun Shen[1] and David K.H. Chua[2]

Abstract

Semantic web technology has become an enabling technology for machines to automatically process and integrate information and further conduct logical reasoning over a set of asserted facts based on formalized ontology applicable to various domains. It is possible for humans and machines to share information and make decisions interchangeably through mapping and merging ontologies and combining inference rules with business rules. Adoption of semantic web technology has been increasingly seen in many enterprise applications, including the improvement of collaboration in Architecture-Engineering-Construction project management due to the fact that the highly fragmented and localized structure in the life cycle of project development sets a series of organizational, technological and culture barriers impeding project players from efficiently acquiring information and effectively resolving conflicts. Rich-semantic applications allow for transparently dealing with heterogeneous, massive-scale, and dynamic enterprise content management in a virtual organization environment with more declarative, more expressive, and more consistent ontology management built on the Internet infrastructure. In this paper, we propose a semantically described and coordinated framework for enhancing project planning and execution through making use of semantic search, integration, and analysis to better manage constraints, minimize disruptions and delays, and achieve more reliable planning and higher productivity. The system architecture is depicted in six layers including multi-agent layer, application layer, semantic rule and query layer, ontology layer, data storage layer, and peer-to-peer (P2P) layer. An illustrative example is presented to demonstrate the work flow of collaborative planning with the interaction of humans and machines to expedite constraints, resolve schedule delays, and coordinate multiple schedules among various trades.

Keywords

Semantic web, ontology, collaborative planning, constraint management, agent, P2P

Introduction

It is a common phenomenon that project performance is often impeded by poor management of constraints and inefficient coordination of schedules due to a wide range of changes and uncertainties in specifications, contracts, approvals and permits, engineering designs, method statements, prerequisite work, materials, labor, equipment, quality, safety, space, site

1 Research Fellow, Dept. of Civil Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260, (65)65164643, Email: cveslj@nus.edu.sg

2 Associate Professor, Dept. of Civil Engineering., National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260, (65) 65162195, FAX (65) 67791635, Email: cvedavid@nus.edu.sg

conditions, weather, etc. Successful project delivery relies on detecting critical constraints in the bottleneck processes and resolving them in a timely fashion causing minimum impact to the project progress (Shen and Chua, 2008). This is, however, not a trivial task considering the large number of players and issues involved. Building a collaborative work environment is important for facilitating constraint management and schedule execution in the life cycle of project development. Traditionally, collaboration takes place among a group of temporarily organized participants. With the new generation Internet-based information technology, namely semantic web, project collaboration can be extended between humans and machines, which will fundamentally change the practice of business operations.

Semantic web (Daconta, et al, 2003; Davies, et al, 2003) is a vision and technology for creating a ***data web***, which focuses on ***data*** (in contrast to HTML ***pages*** in the original web) and ***machines*** (in contrast to ***people*** of social networks in the second-generation web). It empowers computer agents and other software to automatically process and integrate information and further conduct logical reasoning over a set of asserted facts based on formalized ontology applicable to various domains. The corner stone is machine-readable metadata constructed in the form of ontology, which by definition is a formal, explicit specification of a shared conceptualization (Gruber, 1993). Ontology defines vocabularies and their relations that can be interpreted by machines for retrieving information and making reasoning to deliver best-match search results. It uses a directed labeled graph data model to describe real world complex data with simple, declarative, expressive, and consistent elements presented in linked subject-predicate-object triples. This provides a powerful means for remixing and mashing up open and legacy data in the business systems to re-create new applications and new content.

In project management, many data and documents are stored and exchanged in proprietary formats, the content of which cannot be easily accessed without special software. It is difficult, if not impossible, to search for a particular data with logical criteria or combine relevant data from various sources to support decision making. Work flow is often interrupted due to lack of information in the collaboration loop. All these problems are rather common and can be improved by adopting semantic web technology in data and metadata management, collaborate people with computer agents, and embedding semantics in the search results with an extended coverage of schedules, documents, people, constraints, and decisions.

This paper presents a semantically described and coordinated framework for enhancing project planning and execution through making use of semantic search, integration, and analysis to better manage constraints, minimize disruptions and delays, and achieve more reliable planning and higher productivity. The system architecture is depicted in six layers including multi-agent layer, application layer, semantic rule and query layer, ontology layer, data storage layer, and peer-to-peer (P2P) layer. An illustrative example is presented to demonstrate the work flow of collaborative planning with the interaction of humans and machines to track constraints, resolve schedule conflicts, and coordinate multiple schedules among various trades.

Semantic Web Technology in Project Management

The base language of semantic web is Resource Description Framework (RDF), which adopts a graph data format in contrast to relational data formats (such as most databases) and hierarchical data formats (such as XML). An RDF model is a collection of triples, each consisting of a subject, a predicate, and an object as depicted in Figure 13. Figure 1(a) asserts an individual fact that a project is named “SRC_RCC_Revamp_Y2009”. Figure 1(b) asserts multiple facts that a project has two tasks, task 1 and 2, where task 1 must finish before task 2 starts. The duration of task 1 is 3 days and task 1 has a supervisor whose full name is Alex Lee, mailbox is alex@xyz.com, and employer is company X. Figure 1(c) shows two RDF models representing two schedule files, both assigning Alex as a supervisor for some tasks. It is possible that the two schedules are acquired from two projects and stored in different formats (e.g. one in MS Project format and the other in Primavera format). The RDF provides an open standard for generating exchangeable metadata that can be merged into a single model. This is a powerful feature allowing computer agents to automatically search, integrate, and mash up data from various sources to re-create the content as new knowledge. In this case, Alex could manage his schedules more transparently without overlooking any hidden tasks using the integrated RDF model.

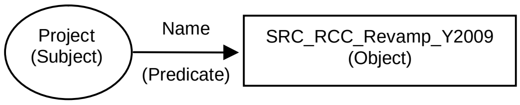

Figure 1(a). An Individual RDF Statement of a Project Model

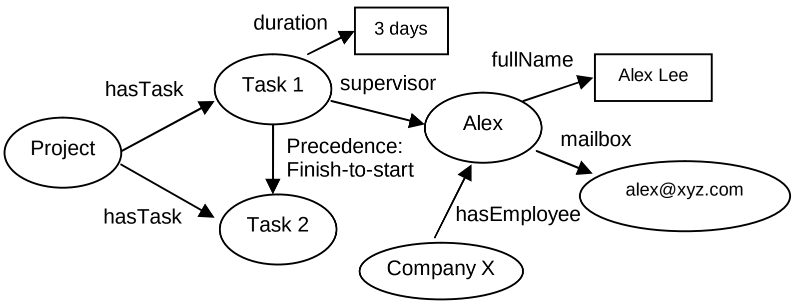

Figure 1(b). Multiple RDF Statements of a Project Model

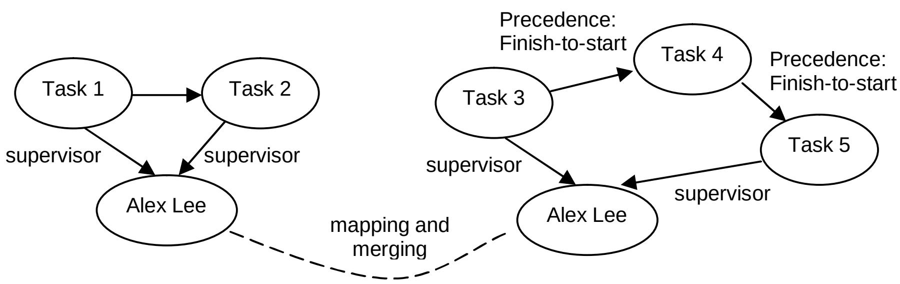

Figure 1(c). Semantic Mapping of Two RDF Models from Two Different Sources

RDF models are defined with a dedicated modeling language called RDF Schema (RDFS), which generally targets on simple ontologies. For complex ontologies requiring more vocabulary terms and formal semantics, another knowledge representation language called Web Ontology Language (OWL) is needed. OWL is built and extended from the RDF/RDFS

3 Real world RDF models use Universal Resource Identifier (URI) to uniquely identifying the subject, predicate, and object of a statement. For simplicity purpose here, we replace URI with short descriptive name instead.

vocabulary by adding more model semantics on classes, properties, instances of classes, relationships between classes or instances, and characteristics of those relationships (McGuinness and Harmelen, 2004). It can describe the hierarchy of a domain through sub-classing and set operation (e.g. intersection, union, complement, oneOf, and disjointWith). OWL also provides several property characteristics (e.g. transitiveProperty, symmetricProperty, functionalProperty, inverseOf, and inverseFunctionalProperty) and property restrictions (e.g. allValuesFrom, someValuesFrom, Cardinality, and hasValue) which are very useful for advanced reasoning. Both RDF and OWL are the languages of choice for semantic web.

The semantic web extends, but does not replace, the current web. It creates an additional layer of metadata in exchangeable data formats (such as RDF and OWL) from the existing heterogeneous web content so that the underlying data can be indexed and searched even though the content cannot be directly accessed. This metadata layer makes it possible for computer agents to share knowledge with humans and delegate many tasks with higher efficiency. An agent is essentially a special software component that has autonomy that provides an interoperable interface to an arbitrary system and/or behaves like a human agent, working for some clients in pursuit of its own agenda (Bellifemine et. al., 2007). Interaction and collaboration between human and agents are crucial for building semantic web applications. Generally speaking, humans are good at content creation and defining the semantics of domain knowledge, while computer agents are good at repetitive work such as metadata generation, data mining, and semantic mapping. Some agents are designed to be able to directly communicate with humans through natural language processing and semantic reasoning.

AEC project management is essentially a collaborative work flow involving multiple parties and various heterogeneous and legacy documents. Semantic web technology may help remove the obstacles in data retrieval, exchange, and integration. In the domain of project planning and control, some key improvement can be achieved by adopting semantic web technology, including: (1) constraint identification and categorization, where shared ontologies and standard templates are built for modeling constraints; (2) constraint tracking, where agents automatically monitor and react to the status of constraints; (3) communication, where many routine communication task such as request for information can be replaced by agents; (4) document management, where open-standard metadata are generated from legacy documents for semantic search and integration; (5) learning and knowledge management, where domain knowledge can be better represented with additional metadata information.

In the following section, we describe an enhanced project planning framework using semantic web technology. The focus is on constraint management and collaborative scheduling, though the principles can be adopted in other project functions such as design management and procurement.

Framework for Collaborative Project Planning

The proposed collaborative planning framework consists of six layers including, from bottom to top, peer-to-peer (P2P) layer, data storage layer, ontology layer, semantic rule and query layer, application layer, and multi-agent layer as depicted in Fig. 2.

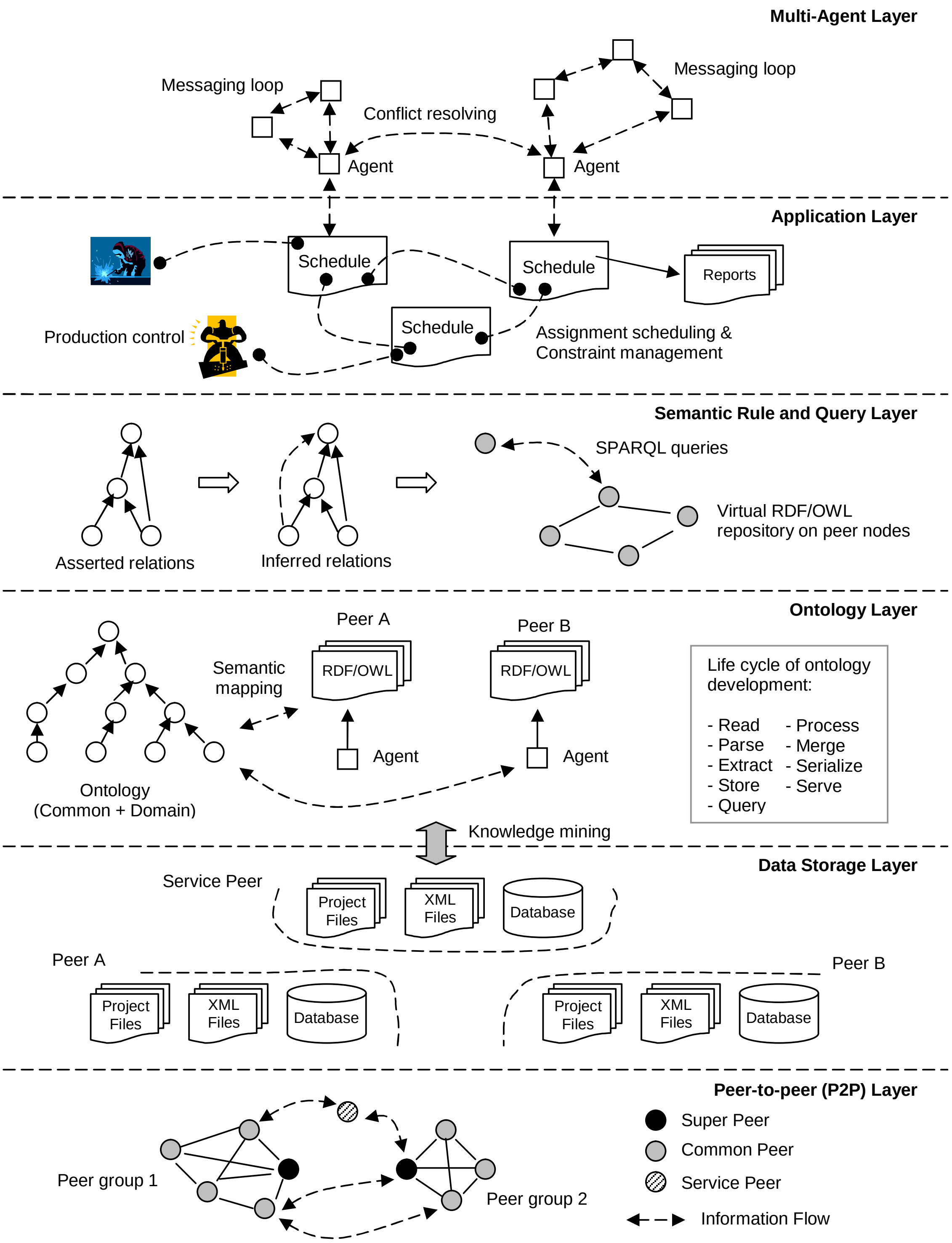

Figure 2. A framework of semantic search enabled collaborative project planning

Peer-to-peer (P2P) Layer

P2P is a distributed network topology where participants can use a variety of devices (e.g., sensors, cell phones, PDAs, laptops, workstations, servers and supercomputers) to share resources and provide services mutually as peers. It does not require centralized administration and content management as commonly seen in a client/server architecture. Peers can use any compatible devices to locate and communicate with one another independent of network addressing and physical protocols. P2P applications are becoming increasingly prevalent in applications such as file sharing, instant messaging, and distributed computing.

In a collaborative project management environment, project participants may join the collaboration loops on an ad-hoc basis from various locations using different devices. Each of them may generate a lot of data (e.g. documents, drawings, constraints, and progress) that are frequently updated. They often need to work closely and temporarily as a group for resolving problems. The P2P provides a scalable and robust infrastructure enabling the end users to create, advertise, discover, and join the peer groups and collaboration loops with much ease. Moreover, with the help of computer agents deployed on peer nodes, it is possible to further enhance the efficiency of communication and data exchange which is important for constraint tracking and conflict resolving.

The P2P layer in Fig. 2 consists of three types of peer nodes (super peers, common peers, and service peers) and two peer groups (1 and 2). A super peer acts as a manager who creates and administrates peer groups and determines which restrictive resources and content can be accessed by other peers. A common peer acts as a user with limited privilege but collectively they are the information providers and consumers. A service peer is designed for supporting roles such as directory service, data storage, and security. Information flow and collaboration flow can be established between peers and peer groups to achieve improved communication, collaboration, resource sharing, peer service, and activity monitoring.

Data Storage Layer

The data storage layer refers to a collection of files and databases distributed on the peer nodes. It can be classified into three categories - unstructured data, semi-structure data, and structured data – based on machine-processability. Unstructured data does not have a data model or has one that is not easily usable by a computer program. Unstructured data include unstructured text (such as the body of Word document, PDF, email, HTML pages, and ASCII files) and unstructured binary (such as drawings, images, audio, video, and many proprietary formats). Semi-structured data contains tags or other markers to separate semantic elements and hierarchies of records and fields within the data. For instance, XML is a popular semi-structured data widely used for data exchange between heterogeneous data models. Structured data include most databases (relational, object) that have formal data structures.

The techniques for creating metadata are different from one data model to another. For unstructured text, natural language processing can be employed but at present this technique is still at early stage. Some unstructured binary data can be annotated with standard tags, e.g. EXIF for digital camera images. XML can be easily transformed to RDF/OWL representation. Databases require customized codes to generate metadata and ontology.

Ontology Layer

The ontology layer involves a series of knowledge management tasks such as recognizing concepts, defining taxonomies, defining classes and properties, defining relations, mapping and merging multiple ontologies, resolving name conflicts between standard ontologies and domain ontologies, and publishing shared ontologies (Léger et al, 2007). Typically, the life cycle of ontology development is composed of read, parse, extract, store, query, process, merge, serialize, and serve. The computer agents deployed on peers can help automatically generate RDF/OWL data such as assignment schedules and constraint checklists containing detailed information of tasks, precedence relationships, resource requirements, organizations, and constraints. The real data will be retrieved and assembled from various sources on peer nodes through semantic mapping of ontology. The ontology layer facilitates improving the interoperability of heterogeneous, massive-scale, and dynamic enterprise content management in a virtual organization environment.

Semantic Rule and Query Layer

The semantic rule and query layer provides automated machine reasoning and SPARQL (Prud'hommeaux and Seaborne, 2008) query language for RDF services. Inference is a process of deriving new data from existing data that someone already knows. The RDFS and OWL specifications define a set of rules for making inference on class and property relations. These rules can be used to derive implicit relations such as dependency and conflict of processes. The reasoning enables advanced search functions which cannot be implemented by a keyword based search engine. It also supports machine learning to improve search results when more asserted facts are available through merging ontologies from various sources. On the other hand, the SPARQL allows for querying across diverse data sources of RDF models in native formats or generated via middleware.

Application Layer

The application layer provides user interface and functionality for project management services. A look-ahead planning tool with enhanced constraint management to improve schedule reliability is employed (Shen & Chua, 2008). Semantic web technology helps constraint management in terms of (1) improving interoperability of documents, drawings, schedules, and databases; (2) facilitating information sharing with extensible metadata; (3) enhancing integration of local data with external virtual data repository among peers; (4) allowing automated constraint tracking and decision making using computer agents.

Multi-Agent Layer

The multi-agent layer contains general and special purpose software agents delegating humans to communicate, negotiate, and collaborate with other peers. Agents can perform simple tasks such as sending a message to request information, or complex tasks such as resolving a schedule conflict by re-assigning multiple processes and constraints. Agents can mimic human decision via machine reasoning. The presence of agents may greatly reduce human intervention in some routine tasks and automated many processes in the collaboration loops.

Implementation

The implantation of the framework employs primarily open standards and open source technologies based on the Java programming platform. The P2P layer is developed on JXTA technology (Sun Microsystems, 2008), which is a set of open and generalized P2P protocols that are independent of programming language, network transport protocols, and deployment platforms. JXTA provides six standard protocols: peer discovery protocol, peer information protocol, peer resolver protocol, pipe binding protocol, endpoint routing protocol, and rendezvous protocol. Peer functions are programmed as services and modules, which can be assigned to each peer depending on its roles in the peer group. The database on peer node is

MySQL. The ontology authoring and editing tool is Protégé (Horridge, 2009), which can export ontologies to a variety of formats including RDF(S), OWL, and XML schema. Protégé supports inference and visualization of ontology through third-party plug-ins. The semantic web applications and services are developed on the Jena platform, which provides a programmatic environment for RDF/RDFS, OWL, SPARQL, and includes a rule-based inference engine. The multi-agents are programmed on the JADE (Java Agent DEvelopment Framework) platform, which complies with the FIPA agent communication language specifications. All the above functions are integrated into the application layer where the graphical user interface is a research prototype for look-ahead planning and constraint management.

Illustrative Example

Here is a simple example to illustrate how the proposed framework helps collaborative planning and decision making. Fig 3(a) shows an original schedule with a 2-day delay caused by some prerequisite constraints (denoted by dots prior to tasks). The bottleneck constraints can be determined by key constraint analysis (Chua and Shen, 2005). In this case, C5 must be expedited for 1 day in order to reduce project duration for 1 day. The computer agent searches the RDF to find out what upstream processes affect constraint C5 and who should be contacted. This may trigger several messaging loops between humans-agents or agents-agents before the confirmation is obtained that C5 can be expedited 1 day. To further reduce project duration for another day, both constraints C3 and C5 need to be expedited for 1 day as shown in Fig. 3(c). Consequently, a similar negotiation process is initiated. However, for this time, C3 cannot be expedited so an alternative solution must be adopted. The computer agent uses inference engine to determine that if a task on the critical path can be expedited 1 day, then the project duration can be shortened as well. After multiple rounds of negotiation, the computer agent finally achieves the objective to reduce project duration for 2 days through accelerating task T4 for 1 day and expedite constraint C5 for 2 days in total as shown in Fig. 3(d). All affected processes are updated with the latest schedule information.

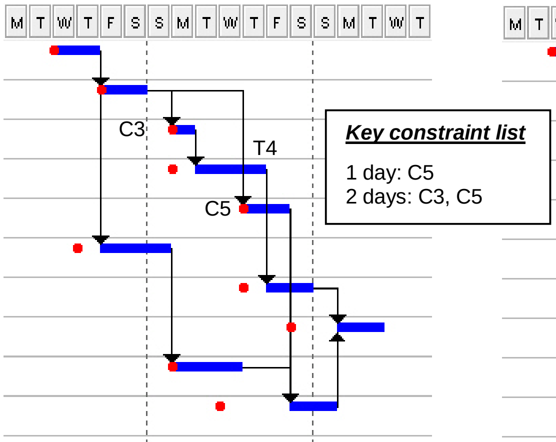

Figure 3(a). Original schedule with 2-day delay

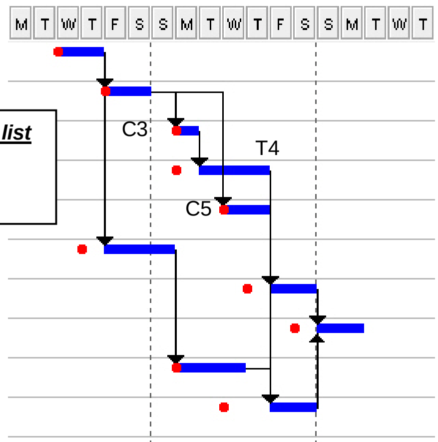

Figure 3(b). Expedite key constraint C5 for 1 day

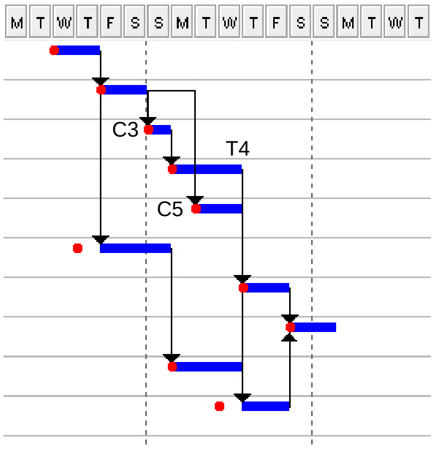

Figure 3(c). Expedite key constraints C3 and C5 for 1 day, respectively

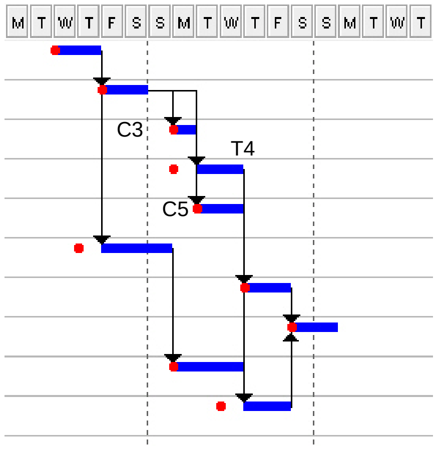

Figure 3(d). Expedite key constraint C5 for 1 day, and accelerate task T4 for 1 day

Conclusions

In this paper we presented a collaborative project planning framework using semantic web technology. The framework consists of six layers including: peer-to-peer layer for decentralized user and user group management; data storage layer for deploying unstructured project files, semi-structured XML files, and structure databases; ontology layer for knowledge representation and ontology development; semantic rule and query layer for machine reasoning and search in a virtual RDF/OWL repository on peer nodes; application layer for constraint management and production control; and multi-agent layer for advanced messaging and collaboration to delegate human roles. The framework as a whole targets on improving collaboration, communication, and decision making in project planning and control through the interaction of humans and machines. With the advancement of Internet-based semantic web technology, the proposed framework is implementable and may directly or indirectly change the practice of project management and collaboration in the future.

The main challenges in realizing a working system could come from, to name a few: 1) the complexity of technologies involved in each layer and sub-systems, 2) a variety of heterogeneous standards, many of which are not stabilized yet, 3) proprietary data formats, which require special software and plug-ins to handle, 4) difficulties in building ontologies for various domains, 5) integration with existing systems and data models, and 6) implementation of agent-based project functions to be deployed on various peer devices. Our future work will concentrate on enhancing ontologies in several domains (e.g. constraint management and design interface management) and develop core project functions to support automated workflow for communication, decision-making, and execution.

Acknowledgment

This work was funded by Singapore MOE Academic Research Fund (AcRF) Tier 1 under grant number R-264-000-236-112.

Reference

Bellifemine et. al. (2007). Developing Multi-Agent System with JADE. John Wiley & Sons, Ltd.

Beckett D. (2004). RDF/XML Syntax Specification (Revised). http://www.w3.org/TR/REC-rdf-syntax/

Chua, KHD and Shen LJ (2005). "Key constraint analysis with Integrated Production Scheduler." Journal of Construction Engineering and Management, July issue.

Daconta M. C., et al (2003). The Semantic Web: A guide to the future of XML, Web services, and knowledge management. Wiley Publishing, Inc.

Davies J., et al (2003). TOWARDS THE SEMANTIC WEB: Ontology-driven Knowledge Management. John Wiley & Sons Ltd

Gruber T.R. (1993). "A translation approach to portable ontology specifications", Knowledge Acquisition, vol. 5, 1993, pp. 199-220.

Horridge M. (2009). A Practical Guide To Building OWL Ontologies Using Prot´eg´e 4 and CO-ODE Tools. The University Of Manchester

Jena website. http://jena.sourceforge.net/documentation.html

Leger et. al. (2009). "Semantic Web Take-Off in a European Industry Perspective", Semantic Web for Business. Pp 1-29.

McGuinness D. and Harmelen F. V. (2004). OWL Web Ontology Language Overview. http://www.w3.org/TR/owl-features/

Prud'hommeaux E. and Seaborne A. (2008). SPARQL Query Language for RDF. http://www.w3.org/TR/rdf-sparql-query/

Shen, L.J., and Chua, K.H.D. (2008). An investigation of critical chain and lean project scheduling. 16th International Conference on Lean Construction, July 16-18, 2008, Manchester, United Kingdom, 679-690.

Sun Microsystems, Inc. (2007). JXTA Java Standard Edition 2.5: Programmers Guide.