Compliance Applications on the Semantic Web

Richard Hancock

richard.hancock@3kbo.com

Abstract. Limitations with the resource linking capabilities of a conventional web and mobile phone based application lead to the investigation of a semantic web [1] based architecture. The current application provides the construction industry with the ability to verify that work completed at the job site complies with mandated standards, regulations and specifications. The semantic web architecture aims to maintain the current functionality plus improve the linking of project team members, site locations, referenced documents and project tasks.

Overview

In this paper compliance is discussed primarily as it is applied in the construction industry, though the concepts could be applied to other industries. In the context of the construction industry compliance is the requirement that a piece of work is completed to a satisfactory level, as mandated in a specification, standard, law or regulation. The standards are typically maintained be a standards organization [2] and the laws and regulations by national, state and local government.

Typically an organization needs to define compliance plans based on context, for example a construction project in close proximity to the ocean may mandate that stainless steel bolts be used due to the corrosive nature of that environment.

Fully managed compliance is performed in the context of a project or a business process. In an initial design phase users specify what needs to be verified and the criteria that will form the basis of the verification. A compliance plan is produced which lists the items to be inspected, the compliance criteria and who has been allocated the inspection tasks. During the inspection process when an item can't be deemed compliant a follow up action is raised. The generated list of outstanding follow up actions is monitored, with each action being managed to a successful resolution.

Project stakeholders are typically organizations with their associated users performing project tasks according to the roles assigned to them.

The application on which this article is based, the Compliance Data Management Service (CDMS) [3], allows on-site inspections of construction work to be performed using a mobile phone web based application. For CDMS the main types of organizations are currently "Construction Contractors" and "Project Owners". The main user roles are currently "Project Designer", "Project Manager", "Checker" and "Guest".

Compliance Work Flows

The main workflows for a compliance application are:

1. **Compliance Specification** is the task of defining the steps that need to be taken to validate the state of specific items. The outcome is a “Compliance Plan” which lists “Verification Points” that need to be deemed valid or not as per a given set of “Criteria”. The “Compliance Plan” is used in the “Project Design” workflow. The “Compliance Specification” workflow is a potential candidate for implementation as a semantic wiki [4] to encourage the development of community based “Best Practice” compliance plans.
2. **Project Design** is the task of defining a project in terms of one or more “Compliance Plans” which are applied in specific contexts, deemed “Worklots”. A “Job” is the pairing of a “Compliance Plan” with a “Worklot”. The “Job” consists of a set of Verifications, each Verification matching a “Verification Point” defined in the “Compliance Plan” and applied in the specific context of the assigned “Worklot”. A Verification is the smallest unit of work that can be assigned to an individual. The individual would perform the verification using the criteria defined against the Verification Point of the Compliance Plan.
3. **Compliance Validation** is a ubiquitous activity supported by a web-based application that is accessible in the field by a mobile phone and in the office by a standard web browser. The web application provides the means to validate work or raise follow up actions.
4. **Project Management** requires close to real time reporting on outstanding follow up actions and the overall state of the project.

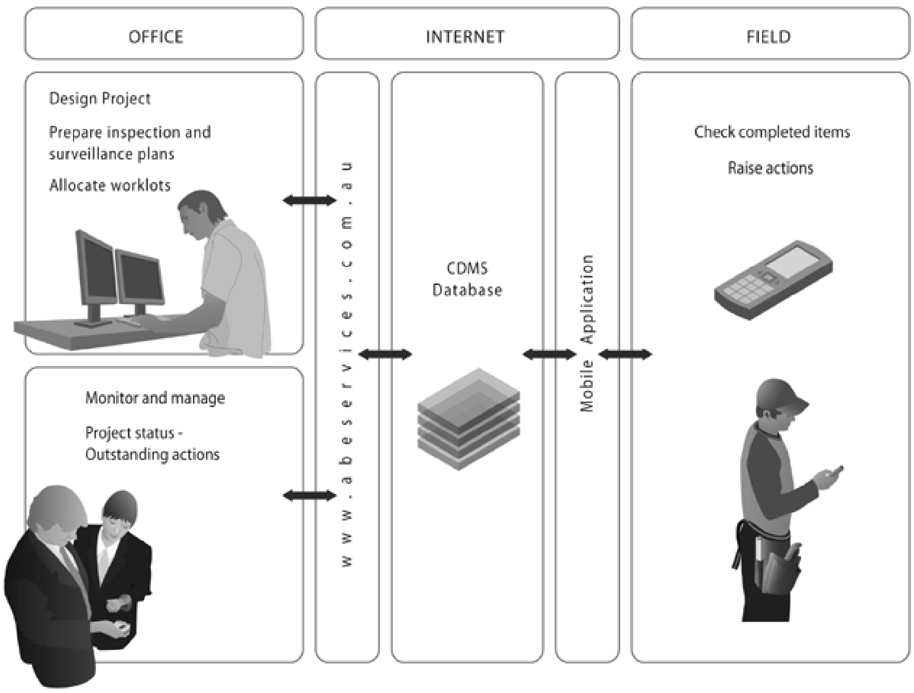

Figure 1: Compliance Application Overview

Figure 1 above provides a simplified view of the compliance workflows as currently implemented in the CDMS application, where the “Compliance Specification” and “Project Design” workflows are merged, since “Compliance Specification” is currently a closed activity, not available to the public communities.

Ontology Engineering

Defining OWL [5] ontologies suitable for compliance applications draws on information from the following sources:

1. Concepts unique to a compliance application, the central ones being "Compliance Plan", "Verification Point", Criterion/Criteria and "Follow up Action".
2. The vocabulary of the specific problem domain.
3. Information gleaned from on line sources that can be referenced by compliance plans. This includes legislation, regulations and standards, work classifications and industry groupings, product specifications, project specifications, geographical information, land title and address information.
4. Existing OWL ontologies, especially those describing the relationships between people, projects, collaborating on line communities, published articles, thesaurus, workflows and geographical locations.

The ontology engineering effort uses existing ontologies where possible.

Existing Ontologies

The following existing ontologies provide a base for the compliance application to link together people, projects and on line resources.

- Dublin Core Metadata Initiative [6] provides meta data concepts for describing documents.
- FOAF (Friend of a Friend) [7] provides the concept of a Person and personal relationships.
- SIOC [8] provides the concept of an on line User acting in a Role and uses FOAF.
- Geo [9] represents latitude, longitude and altitude information.
- SKOS [10] provides specifications and standards to support thesauri, classification schemes, subject heading systems and taxonomies.

The Compliance Ontology

The compliance ontology needs to provide the ability to describe an agent, human or otherwise, verifying that a work task has been performed satisfactorily or that an entity is in an appropriate state in relation to its current context. If not a follow up action is raised that then needs to be resolved before the work task or entity can be deemed to comply with a required standard or specification. These concepts could not be found in existing ontologies. A draft Compliance Ontology is based on consideration of the ontologies above plus consideration of the following:

- Project Vocabulary [11] specifies a vocabulary for describing projects independently of the project domain. It includes definitions for Project, Task, Agent.
- Work Flow Ontology [12] with Tasks, states and state transitions.

Architectural Goals

The main architectural goals resolve down to supporting the RDF/OWL ontologies with typical CRUD (create/read/update/delete) operations plus SPARQL [13] query support.

A rapid application development framework for building a web interface on top of the semantic web layer, rendering content for both desktop and mobile phone browsers is required.

Additional desired framework features include support for:

- OWL inference support,
- SWRL [14] rules and
- GRDDL [15] and migrating existing data sources to RDF/OWL.
- A flexible single sign on mechanism, allowing participating organizations to work together on specific projects.

A long-term goal would be SPARQL Update [16] support.

Software Architecture

The software architecture consists of the following:

- IkeWiki [17] provides a semantic-wiki for developing the SKOS based thesaurus and glossaries plus community designed compliance plans. IkeWiki uses Jena to store the RDF/OWL ontologies. Most of the ontologies identified above are packaged with IkeWiki ready to be loaded into the underlying Jena [18] RDF triple store.
- Jena is the RDF Triple Store, using the Postgresql [19] relational database to persist RDF /OWL ontologies to relational tables.
- Joseki [20] provides a SPARQL query interface to the ontologies held in Jena
- ActiveRDF [21] and Ruby On Rails [22] to provide the web application framework, connecting via the Joseki SPARQL query interface to the RDF/OWL ontologies held in Jena.
- XSLT [23] scripts for rendering XML based standards documents and legislation to RDF.

Promising aspects of the above architecture are:

- The IkeWiki semantic-wiki for the development of the SKOS based thesaurus and glossaries plus community designed compliance plans.
- The ActiveRDF and Ruby On Rails integration with the semantic-wiki via the Joseki SPARQL interface.

Areas for further investigation are:

- The best method for updating the Jena RDF Tripe Store from the ActiveRDF/Ruby On Rails web application. Currently SPARQL provides read-only access. The proposed SPARQL Update [16] specification and support for its implementation by both Joseki and ActiveRDF would provide an ideal solution.

- Enhanced support for mobile phones within the Ruby On Rails framework. The ideal implementation would provide within Ruby On Rails functionality equivalent to that available with the Java JSP WALL[24] tag library.

Conclusions

The potential star performer is the semantic-wiki. While IkeWiki still needs to be evaluated further, the potential for developing SKOS based thesaurus, glossaries and compliance plans as part of an open community effort holds promise for creating best practice compliance documents.

The implementation of the SPARQL Update specification is seen as a positive step for simplifying the creation of semantic web applications.

References

1. Semantic Web, see http://www.w3.org/2001/sw/
2. Standards Organization, see http://en.wikipedia.org/wiki/Standards_organizations
3. Compliance Data Management Service, see http://www.abeservices.com.au/
4. Semantic wiki , see http://en.wikipedia.org/wiki/Semantic_wiki
5. OWL – Web Ontology Language, see http://www.w3.org/2004/OWL/
6. Dublin Core Metadata Initiative, see http://www.dublincore.org/
7. FOAF - Friend of a Friend, see http://www.foaf-project.org/

8. SIOC – Semantically Interlinked Online Communities, see http://sioc-project.org/ontology/
9. Geo - Basic Geo (WGS84 lat/long) Vocabulary, see http://www.w3.org/2003/01/geo/

10. SKOS - Simple Knowledge Organisation Systems, see http://www.w3.org/2004/02/skos/

11. Project Vocabulary, see http://dannyayers.com:88/xmlns/project/index.htm

12. Work Flow Ontology, see http://www.w3.org/2005/01/wf/

13. SPARQL - SPARQL Query Language for RDF, see http://www.w3.org/TR/rdf-sparql-query/

14. SWRL - A Semantic Web Rule Language Combining OWL and RuleML, see http://www.w3.org/Submission/SWRL/

15. GRDDL - Gleaning Resource Descriptions from Dialects of Languages, see http://www.w3.org/TR/grddl/

16. SPARQL/Update A language for updating RDF graphs, see http://jena.hpl.hp.com/~afs/SPARQL-Update.html

17. IkeWiki - the Semantic Wiki for Knowledge Engineers, see http://ikewiki.salzburgresearch.at/

18. Jena – A Semantic Web Framework for Java, see http://jena.sourceforge.net/

19. Postgresql, see http://www.postgresql.org/

20. Joseki - A SPARQL Server for Jena, see http://www.joseki.org/

21. ActiveRDF, see http://www.activerdf.org/

22. Ruby On Rails, see http://www.rubyonrails.org/
23. XSLT - XSL Transformations Version 2.0, see http://www.w3.org/TR/xslt20/
24. WALL - the Wireless Abstraction Library see http://wurfl.sourceforge.net/java/wall.php