Project Title: Integration of Product Design and Project Activities using Process Specification and Simulation Access Languages
Principal Investigator: Kincho H. Law, Stanford University
Duration: September 1, 2002 – August 31, 2005

PROJECT OBJECTIVES

The design of a product is an integrated project that must take full consideration of the processes and activities as well as resources and organizational structures involved in manufacturing and construction of the product. Product design and manufacturing requires the integration of product, process and project management information. The objective of the proposed study is to develop methodologies to facilitate product design with process simulation. Specifically, we are interested in the evaluation of the applicability of the process specification language (PSL) and to design and implement a simulation access language (SimQL) that can be used to facilitate process simulation of an engineered product. This proposed joint research effort with NIST includes:

- To integrate product design data management with process models
- To analyze the concepts of PSL for product and process information exchange with selected project management applications
- To investigate the reasoning power of PSL for consistency checking and conflict resolution of product, process and project information from various sources
- To design and implement a simulation access language for integration of engineering services involved in product design and process simulation
- To integrate PSL with SimQL, and develop a demonstration prototype for reusing and integrating results from a variety of application services

Our long term goal is to develop a distributed network-based framework to integrate product and process modeling, and virtual simulations of project activities. The results of this project should greatly enhance modeling and design of engineered product, taking into consideration downstream process simulation for the construction and manufacturing of the product. In the following, we briefly summarize our previous accomplishments on the evaluation of PSL for project management applications. We then discuss our proposed research utilizing PSL and SimQL.

2. PREVIOUS RESULTS

In the Process Specification and Simulation project previously sponsored by NIST (2000 – 2001), we have conducted a preliminary evaluation of PSL as an interchange standard for project management applications. First we have extended the PSL ontology (Schlenoff et al. 2000) based on the concepts in Vite (Figure 1). (Vite is a project and organization modeling system designed to assist in developing organizational structures and identifying potential problems with project cost, time, or quality.) We proposed a set of PSL extensions to include organization, activity and project modules.

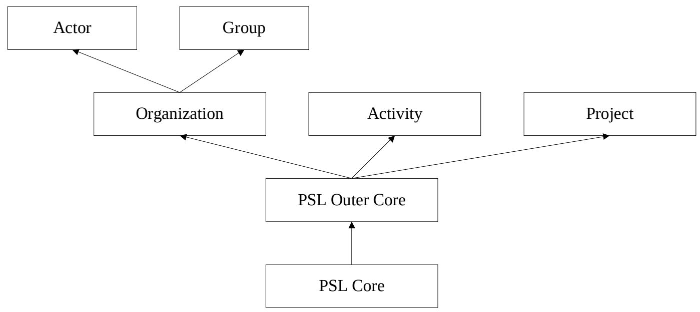

Figure 1: Suggestion of Extensions to Current PSL Ontology

To illustrate the exchange of product, process and project management information using PSL, we have developed PSL wrappers for Primavera P3[1], Microsoft Project[2], Vite[3] and 4D Viewer[4] (Figure 2). Primavera P3 and Microsoft Project are widely used commercial software for project management applications. Different technologies, such as Primavera Automation Engine, JDBC (Java Database Connectivity) and others, have been employed to implement the PSL wrappers. Using the PSL wrappers, we have successfully exchanged project information among these application software using a small sample project from Vite and the Morton Ceiling Project (which is part of the Disney Concert Hall project).

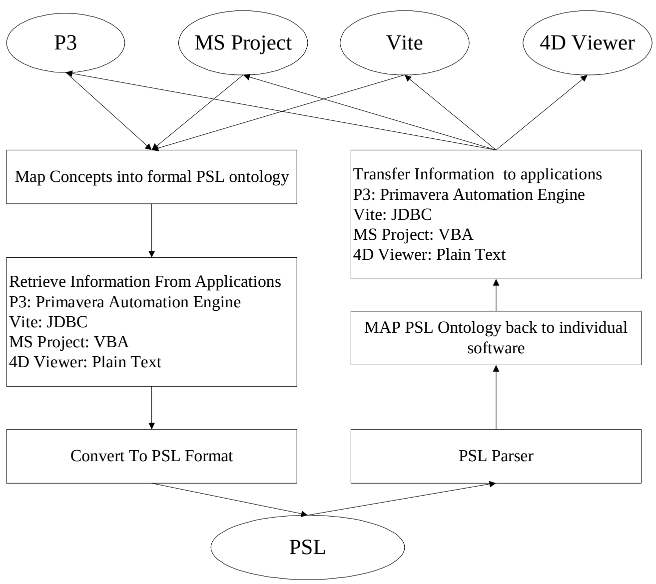

Figure 2: Information Exchange Using PSL

---

1 Primavera Project Planner (P3) is a registered trademark of Primavera Systems, Inc.
2 Microsoft Project is a registered trademark of Microsoft Corporation.
3 Vite is a registered trademark of Vite Corporation.
4 4D Viewer is a 4D software developed by Professor Martin Fischer and his research group at Stanford University.

In addition to information exchange among different project management applications, we have developed an integrated service infrastructure (Liu et al. 2001). As illustrated in Figure 3, we use Oracle 8i Relational Database as the persistent store of project information. The active mediator acts as an intelligent bridge that connects various devices with the database, while PSL acts as a process interchange standard through which various applications communicate with each other.

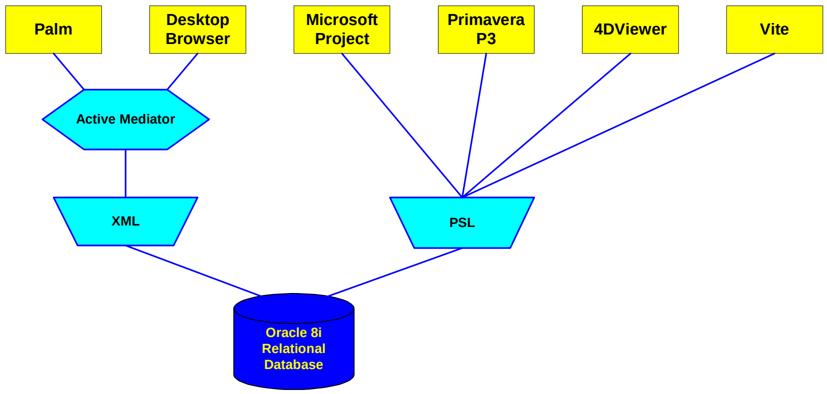

Figure 3: An Integrated Service Infrastructure

There have been many ontology standards proposed for product and process modeling and project management applications. Examples include STEP (ISO 1994), IFC (IAI 1997), and XML-based standards such as ifcXML (Liebich 2001) and aecXML (IAI 2002). A preliminary investigation has been attempted to develop ontology mapping between PSL and XML-based standards. Figure 4 shows the translation process between PSL files and XML files. The translation has been successfully tested on a few sample projects.

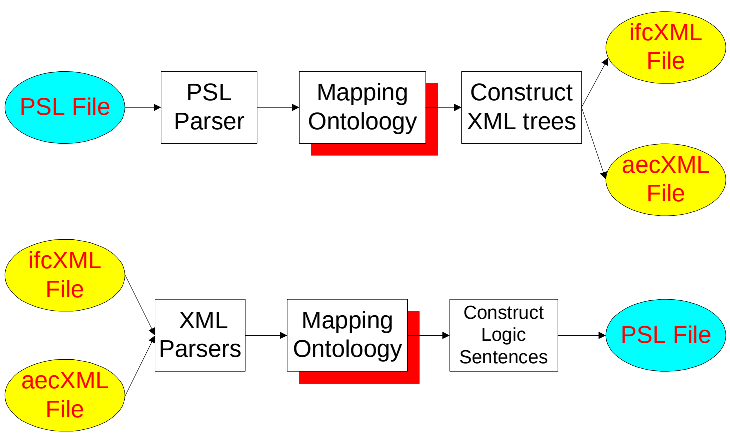

Figure 4: Mapping Process Between PSL and XML

One major difference between PSL and other ontology standards is the logic structure of PSL. PSL is based on first order logic and situation calculus and has more expressive power than most other ontology standards, such as STEP, ifcXML or aecXML. An initial research has shown that PSL could be useful for checking consistency of product and project information.

3. PROPOSED RESEARCH

Our previous research has demonstrated the potential of PSL as an information exchange standard for project management applications. In this proposed research project, we intend to continue the study of the PSL ontology and logical reasoning with PSL. In addition, we plan to design and implement a simulation access language (SimQL) and to develop a prototype distributed platform utilizing PSL and SimQL for workflow management applications. The project will include a thorough investigation of the current PSL and SimQL. Selected applications, such as Vite, Primavera P3, Microsoft Project and others, will be employed to facilitate our investigation.

3.1 PRODUCT, PROCESS AND PROJECT MANAGEMENT APPLICATIONS AND PSL ONTOLOGY

As part of this research, we will study the concepts and ontology of PSL and compare with application services. We have had some preliminary investigation on the concepts related to integration of product information with project scheduling. An integrated project will include not only product and process information but also information on cost, resource, organization and other information. For example, MS Project and Primavera P3 include detailed scheduling information and some rudimentary resource and cost information. Other project management applications, such as Timberline's Precision Estimating[5] and R.S. Means' CostWorks[6], focus on cost estimating. Table 1 shows the information in some of these tools.

Table 1: Concepts in selected applications

Software	Information In the Application
Primavera P3, and Microsoft Project	Detailed scheduling information, rudimentary cost and resource information
Vite	Detailed project organization information, rudimentary cost and scheduling information
4D Viewer	Product 3D model information and project scheduling information
Precision Estimating	Cost estimating information
CostWorks	Cost estimating information of standard tasks based on quantity, location and other factors.

For an integrated project, we often need to work with multiple applications, each provides different functionalities. In this work, our plan is to extend PSL to handle life cycle information including product, process, schedule, cost, resource, organization and other information (Figure 5). The PSL parser that we have developed so far can only parse predefined PSL terms. We plan to investigate the possibility to build a generic PSL parser in Java using a parser generator. A parser generator is a tool that reads a grammar specification and converts it to a Java program that can recognize matches to the grammar. Java Compiler Compiler (JavaCC) is one of such popular parser generators for use with Java applications (SUN 2002).

---

5 Precision Estimating is a registered trademark of Timberline Software Corporation.
6 CostWorks is a registered trademark of R.S. Means Company, Inc.

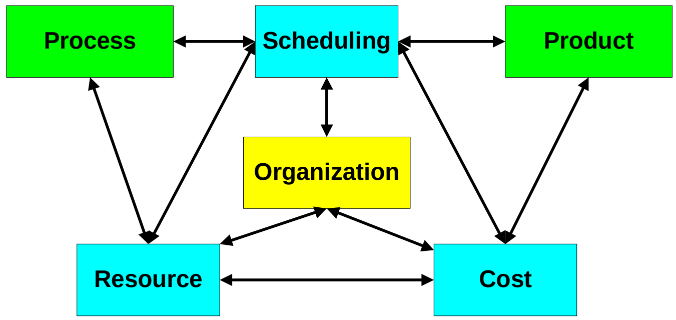

Figure 5: Information in an Integrated Project Model

3.2 MAINTAINING CONSISTENCY USING PSL

A project usually involves information from different sources. For example, as shown in Figure 6, project team members may use Primavera Project Planner (P3) or Microsoft Project to schedule the project, Vite to simulate the project organization, Timberline's Precision Estimating to estimate project cost, and 4D Viewer to view the product model and the progress of the project. One key issue is to ensure as much as possible consistency among the data generated from such diverse sources of application services.

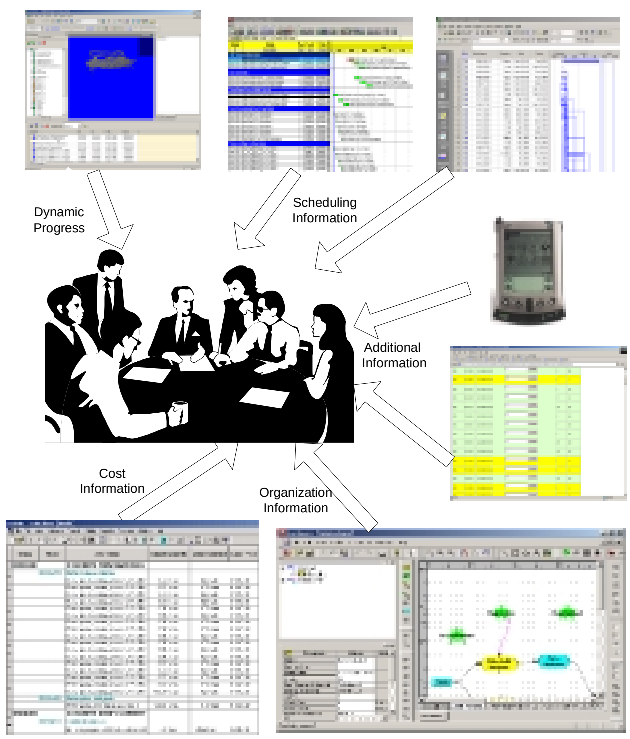

Figure 6: Project Management Services

We have initiated research to study checking consistency using PSL, where we focus on conflicts in product design information and project schedules. In practice, conflicts also would arise from other types of information. For example, project cost and duration are not independent. Different information, such as cost information in Precision Estimating and scheduling information in Microsoft Project, may not be consistent. Further examples for sources of potential conflicts may include versions, scheduling, cost, resource and organization as well as conflicts among the different sources of information.

Since all the terms in PSL are formally defined using first order logic, it is possible to employ existing logic-based reasoning tools, such as Otter, to reason about the information in the knowledge base (Figure 7). Otter (Organized Techniques for Theorem-proving and Effective Reasoning) is a resolution-style theorem-proving program for first order logic with equality (McCune 1994). Such tool typically will "reason" until it finds some conflicts or when no more conclusions could be inferred (Wos and Pieper 2000). As shown in Figure 7, to check the consistency of the project information from heterogeneous sources, we first need to use the PSL wrappers to retrieve project information from different applications. We then need to convert PSL files into the format that the logic-reasoning tool can understand. In the previous research, we have developed a PSL to Otter translator for project scheduling of product data. A more robust and generic translator, however, is needed to bring PSL to the logic-based reasoning tools.

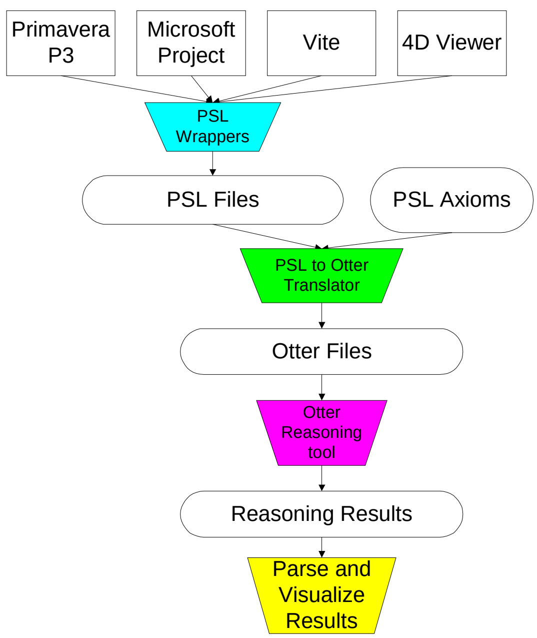

Figure 7. Consistency Checking Using PSL

310 [hyper,136,44,85] after_start(ID110,ID160,TUTO).
333 [hyper,310,44,252] after_start(ID160,ID160,TUTO).
361 [hyper,333,47,333] $F.

Figure 8: Reasoning Results in Logic Sentences

One other non-trivial issue is to translate and visualize the reasoning results. The reasoning results are usually shown in logic sentences (Figure 8). To be of practical use, the results need to be presented in such a way that it is easy to understand by the project managers. One example would be to parse the results and to report, for example cyclic activity dependency relationships as shown in Figure 9. We plan to develop some presentation tools for the reasoning results.

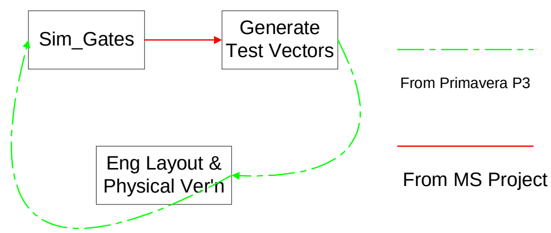

Figure 9: Cycle in Dependency Relationships

3.3 INTEGRATING PSL WITH SIMQL

SimQL (A Simulation Access Language) has the potential to standardize and improve the reusability of simulation tools (Wiederhold et al. 1998). The driving idea behind SimQL is that the simulation services can be written in any language using any technique, but should be able to be accessed through a consistent language independent wrapper.

We envision that the SimQL system includes three layers (Figure 10):

- Simulation Tool Layer, which includes various simulation tools, wrappers and local query engines.
- Control Layer, which provides central control over the simulation tools and provide services to other new applications
- Application Layer, where user can query simulation results and build new simulation applications by integrating simulation results

That is, we need to design and implement query commands to invoke services, monitoring the simulation and parsing and translating the simulation results among application services.

3.3.1 SIMQL MODEL SCHEMA AND LANGUAGES

For the proposed SimQL system, we first need a model schema to wrap the simulation results from the simulation tools, a query language to query the simulation results, and a logic language to control the data among various services.

SimQL Model Schema

The SimQL model schema needs to be sufficiently rich to include all the information about a specific application domain. XML (Young 2001) is probably the most convenient language for representing the model schema since XML format can easily be displayed on a web browser and can be parsed and reused by other applications. We have begun to investigate the potential of aecXML and ifcXML as the model schema to wrap the simulation results. Our initial result shows that ifcXML is a promising XML standard for the application domain.

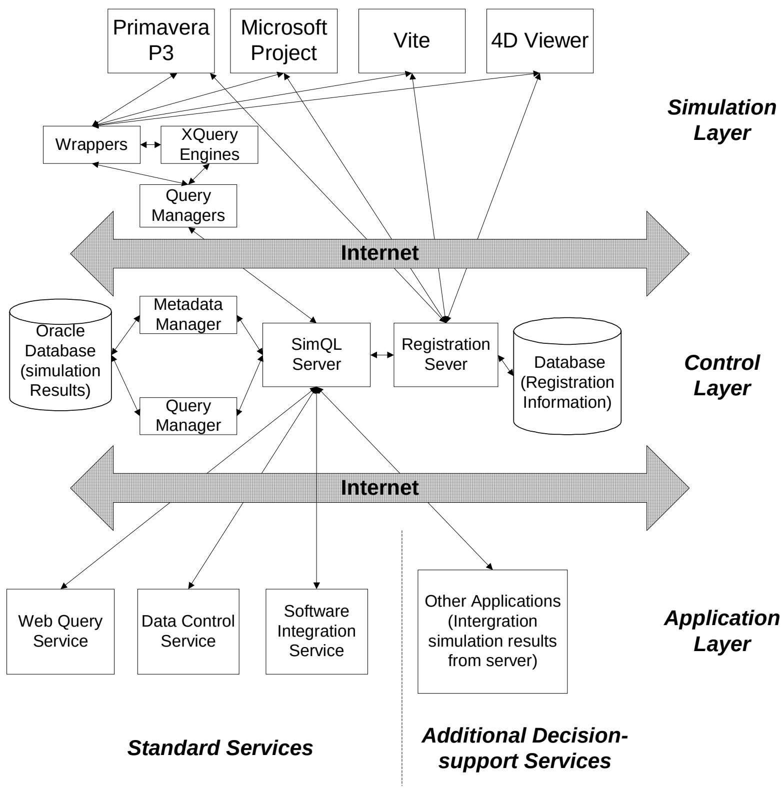

Figure 10: Proposed System Framework of SimQL

SimQL Query Language

The SimQL query language is intended to communicate with the SimQL server from the applications. The proposed SimQL query language includes two parts:

- Communication commands
- XML query language

The communication commands include basic commands to communicate with the SimQL server, such as initiating the session, terminating the session, and other tasks. The XML query languages are used to query information from the XML files. Based on our initial investigation on the various XML query languages, such as XQL (Robie 1999), XML-QL (W3C 1998), LOREL (Abiteboul et al. 1997), Xpath (W3C 1999) and Xquery (W3C 2001), Xquery appears to be the most appropriate language for our research purpose.

Xquery is an XML query language, which has been defined jointly by the XML Query Working Group and the XSL Working Group at W3C (World Wide Web consortium). XQuery is designed to be broadly applicable across all types of XML data sources. Many vendors have initial implementation of Xquery even thought Xquery is still a working draft.

Language For Data Control
One important issue is to ensure data from various sources are consistent and are in valid state. PSL, which is based on first order logic and situation calculus, can potentially play a key role in this area. As we have discussed earlier, we are experimenting with XML to PSL translator and PSL to Otter translator. After we convert information into Otter formats, we can use Otter to reason on the knowledge base. The challenge is how to integrate the SimQL system together with the conflict resolution system.

3.3.2 SIMQL SYSTEM FRAMEWORK

Simulation Layer
In our current design, the Simulation Layer includes the following components: Simulation Tools, Wrappers, Xquery Engines and Query Managers. The SimQL Wrappers act as bridges between the query managers and simulations tools. A local Xquery engine and a query manager are employed for each simulation tool. As in most projects, queries typically focus on the information of a specific object, such as the specific product activities, duration of activities, or the cost of equipment. We can use local Xquery engine to retrieve the relevant information (instead of the entire model).

Control Layer
The SimQL Server maintains the overall control on the execution of simulation tools. It maintains the necessary registration information about simulation tools in the database and is responsible to invoke the relevant information tool upon request. Other roles of the SimQL server could include maintaining backup or providing historical records about a project. Other applications may analyze the data to explore how the project evolves.

4. Summary of Proposed Research and Research Tasks

This proposed research attempts to build an integrated framework to support distributed services related to product design development, process simulation and project management. In this proposal, we have outlined the overall system architecture for the integrated framework. Specifically, we focus on the applicability of PSL as a standard specification language to describe product and process models and to extend PSL to include not only scheduling information but also resources, cost and organization information. Furthermore, we plan to design and implement a simulation access language to standardize and improve reusability and access of simulation tools.

The proposed research tasks for the three-year period are:
- To analyze the concepts of PSL ontology for product and process information exchange (3-6 months).
- To evaluate the use of PSL (and XML-based standards) for selected project management application services and build PSL wrappers to validate the results (9-12 months);
- To evaluate the use of PSL for consistency checking and conflict resolution on engineering projects (6-9 months);
- To design and implement a simulation access language (SimQL) (6-9 months);

- To integrate PSL with SimQL to support product design activities, process simulation and project management applications (12-15 months).

5. Personnel

The principal investigator of this project will be Professor Kincho H. Law of Civil and Environmental Engineering at Stanford University. In addition, a PhD student, Mr. Jim Cheng will be involved and served as a research assistant to this proposed project. Prof. Gio Wiederhold has agreed to provide the necessary help and consultation related to SimQL. Last but not least, we plan to actively collaborate with researchers at NIST and anticipate active collaboration with the PSL development group.

6. References

Abiteboul, S., Quass, D., McHugh,J., Widom, J. and Wiener J.(1997), "The Lorel Query Language for Semistructured Data", International Journal on Digital Libraries, 1(1):68-88, April 1997

Genesereth M.R. and Fikes R. (1992), "Knowledge Interchange Format Version 3.0 Reference Manual", Computer Science Department, Stanford University.

IAI (1997), "Industry Foundation Classes", Specification Volumes 1-4, International Alliance for Interoperability, Washington DC 1997.

IAI (2002), "AecXML", International Alliance for Interoperability, http://www.aecxml.org (Accessed: 15 March 2002).

ISO (1994), 10303-1:1994, "Product data representation and exchange: Part 1: Overview and fundamental principles".

Liebich T. (2001), "XML schema language binding of EXPRESS for ifcXML", MSG-01-001(Rev 4), International Alliance of Interoperability, 2001.

Liu, D., Cheng, J., Law, H.K., and Wiederhold G. (2001), "Ubiquitous Computing Environment for Engineering Services", CIFE Summer Program 2001, CIFE, Stanford University, 2001

McCune, W.W(1994), "Otter 3.0 Reference Manual and Guide", ANL-94/6, Mathematics and Computer Science Division, Argonne National Laboratory, 1994.

Robie J.(1999), "XQL Tutorial", Software AG, http://ibiblio.org/xql/xql-tutorial.html (Accessed: 25 April 2002).

Schlenoff, C., Gruninger M., Tissot, F., Valois, J., Lubell, J., Lee, J.(2000), "The Process Specification Language (PSL): Overview and Version 1.0 Specification", NISTIR 6459, National Institute of Standards and Technology, Gaithersburg, MD.

SUN (2002), " Java Compiler Compiler(JavaCC) - The Java Parser Generator", Sun Microsystems, http://www.webgain.com/products/java_cc/ (Accessed: 25 March 2002).

W3C(1998), "XML-QL: A Query Language for XML", World Wide Web Consortium, 1998, http://www.w3.org/TR/NOTE-xml-ql/ , (Accessed: 25 April 2002).

W3C(1999), "XML Path Language (XPath) Version 1.0", World Wide Web Consortium, Recommendation 16 November 1999.

W3C(2001), "XQuery 1.0: An XML Query Language", W3C Working Draft 20, December 2001

Wiederhold, G., Jiang, R., and Garcia-Molina, H. (1998), "An Interface for Projecting CoAs in Support of C2", Proc.1998 Command & Control Research & Technology Symposium, Naval Postgraduate School, Monterey CA, June 1998, pp.549-558.

Wos, L. and Pieper, W.G., "A Fascinating Country in the World of Computing", World Scientific Publishing Company, Singapore, 2000.

Young, M.J. (2001), "Step by Step XML", Microsoft Press, 2001.