Project Management Scientific Journal (https://damaacademia.com/pmsj/)
Volume 1, Issue 5, pp.01-10, May 2019
Published by: Dama Academic Scholarly & Scientific Research Society (www.damaacademia.com)

A Simulation Access language and Framework with Applications to Project Management

Jinxing Cheng
Department of Civil and Environmental engineering
Stanford University

Abstract

As computer programs become ever more complex, software development has shifted from focusing on programming towards focusing on integration. This trend is also seen in the design and construction industry, where there is an increasing need to integrate and reuse commercial software tools. Although many software applications (e.g., Microsoft Project, Microsoft Excel, the Primavera Project Planner, and AutoCAD) are commonly available for construction engineering and project management, it remains a laborious process to integrate and coordinate these tools to work together and to support decision making. To name a few, the sheer volume and complexity of the tools and the information generated by them, their scattered distribution, and the lack of interoperability are among the challenges in integrating, coordinating, and reusing these tools. This thesis first discusses the potential applications of the Process Specification Language (PSL) for project management applications. Initiated by the National Institute of Standards and Technology (NIST), PSL is emerging as a standard exchange language for process information in the manufacturing industry. This thesis discusses how PSL can be used for exchanging information among project management software applications in the construction industry. The potential applications of PSL in consistency checking and constraint scheduling are also explored. Specifically, a formal mechanism is proposed to perform consistency checking on project information from different computer tools. Furthermore, the use of PSL for checking conformity of project schedules to scheduling constraints is illustrated. This thesis presents a simulation access language (SimAL) and framework for project management applications. The SimAL language and framework integrate legacy project management applications, coordinate different tools, manage the information flow among them, and bring their functionalities online. The prototype of the SimAL framework has been implemented based on PSL for data exchange and a flow-based software composition infrastructure for software integration. Using the prototype, users can simulate scenarios and build up new services from the existing tools. The potential applications of the SimAL language and framework are demonstrated using three illustrative examples. This first example illustrates the use of SimAL to incorporate online information in project management. The second example illustrates how to use SimAL to compare different scenarios in project management. The third example demonstrates how to extend the functions of legacy software applications (e.g., AutoCAD ADT and the Primavera Project Planner) by integrating them to provide new services. Finally, this thesis presents a question answering system to query the information in different project management applications. A prototype question answering system has been built and tested to illustrate the potential usefulness of such a system for project management applications.

Keywords: *Simulation Access Language, Project Management*

1.0 INTRODUCTION

As computer programs become ever more complex, software development has shifted from focusing on coding toward focusing on integration, as illustrated in Figure 1.1 [98]. In parallel to this trend, there is another shift of software development from standalone applications toward distributed, Web-based or Web-enabled services. As a result, future software will be based more and more on the composition and integration of existing software components.

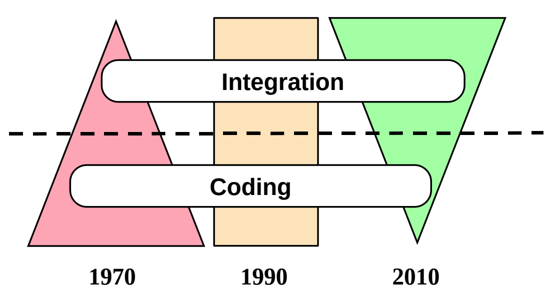

Figure ***Error! No text of specified style in document.****.1: The Trend of Software Development*

Let us take the project management field in the construction industry as an example. There are many software tools (e.g., Microsoft Project, Microsoft Excel, the Primavera Project Planner, and AutoCAD) that are commonly available for construction engineering and project management. These standalone application tools are mature and widely adopted. However, integrating and coordinating these tools to work together and to support decision making remains a laborious process. As an example, resource discrepancy occurs frequently in construction project management. When procurement is being delayed, what options are available and which option is the most appropriate to recover the lost time? Should human resources be added to accelerate the remaining tasks or should extra fees be spent to expedite the delivery? What adjustments are necessary for each option? When making a decision, we also have to consider the potential impacts on project schedules, costs, resources, and organization. To simulate possible impacts and to support decision making would involve using different tools. As illustrated in Figure 1.2, one may use the Primavera Project Planner (P3) or Microsoft Project to schedule the project, Vite SimVision to simulate project organization, Timberline's Precision Estimating to estimate project cost, and 4D Viewer or other CAD tools to view the models. A framework that would allow dynamically integrating and coordinating application tools to simulate the impact of resource allocation on a project and to review "what-if" scenarios can significantly enhance the decision-making process.

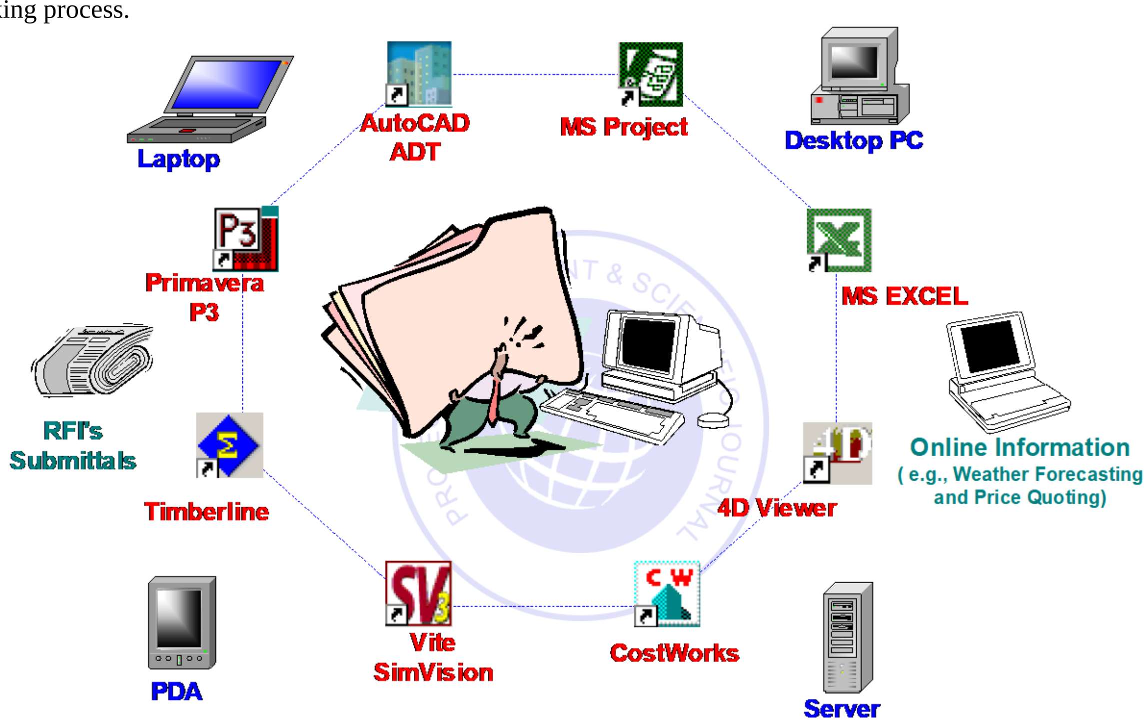

*Figure **Error! No text of specified style in document.**.2: Applications in Construction Project Management*

Among the key issues for integrating and coordinating application tools for simulation is the issue of data and software interoperability. It is not unusual that project data is being re-entered from one application to another. Typical engineering and project management application tools generate large volumes of information that are not easily shared among the applications. Project information is created from different sources in a variety of formats. In addition, project activities are often performed and managed in a geographically distributed fashion. For example, in a construction project, the construction site, the regional office and the company headquarters are often located in different cities and states. Furthermore, different tools are employed at the site and in each office. Ubiquitous access, by means of which project and company personnel can review project information regardless of time and location, becomes important. The sheer volume and complexity of the tools and information, coupled with their scattered distribution and the lack of interoperability, makes any attempt to coordinate and reuse the tools and information a daunting task.

1.1 Research Objective

The key research question to be addressed in this thesis is: How to integrate and coordinate existing COTS (commercial off-the-shelf) tools as well as publicly available information sources to support project management and decision making tasks. We first address the issue of data integration among different project management tools. We then address how to invoke and coordinate these tools for simulation. In addition, the system needs to allow users to query and update project information in different tools and to compare different scenarios.

The objective of this research is, thus, to develop a high-level simulation access language and infrastructure, which would allow users to easily integrate and coordinate standalone applications and to conduct simulation of project scenarios without the detailed knowledge of the network communication and application software. The high-level language should be able to invoke and transfer data among the tools and to query and update project information. With the simulation access language and infrastructure, the functionalities of each individual application can be extended. The users can also choreograph the execution of the tools to potentially support workflow management and decision making applications.

1.2 Related Research

The issues of data and software integration are not new. There have been many standards that have been proposed for data exchange and software interoperability. To review all previous and current approaches and developments is beyond the scope of this thesis. In this section, we briefly summarize some of the related work.

1.2 Data Integration and Exchange

The need for efficient data management and exchange in computer aided building engineering has been a subject of active research and development for quite some time. A detailed review of data integration and exchange can be found in a recent book by Eastman.

Direct Translator: Traditionally, pre- and post-processors are developed to translate data between two application programs using a mutually agreed upon exchange format. The major disadvantage of using pre- and post-processors between programs are the potentially large number of translators. To provide data translation among *n* applications, *n(n-1)* translators have to be developed. Thus, the direct translator approach does not scale as the number of applications increases. Furthermore, extensive software maintenance of the translators could be costly.

Centralized Database: Using a common database can significantly reduce data redundancy and the number of translators for data exchange. Each application generates, stores and retrieves the information according to the database schema. A centralized database has traditionally been considered to be one of the most effective methods for achieving interoperation. There have been many research efforts attempting to define design information about a construction project and to store the information in a single repository. Using a common database allows easy integration of a new tool with other design tools within an organization. The problem with this approach is that to define a common schema for different applications, even within the same domain, can be quite difficult. This approach also does not provide the flexibility to support collaboration among multiple disciplines and organizations.

Neutral File: Another approach is to develop industry-wide neutral file formats for specific application domains. In the neutral file approach, a translator needs to be developed for each application, so that the application can read information from and write results to files in a standard format. Consequently, only *n* translators need to be developed to provide interoperation among *n* applications. Early work on neutral files focuses on the data exchange on CAD and graphical data, such as DXF [64] and IGES. As engineering companies are increasingly seeking ways to integrate their applications, many neutral file standards have been developed by standards organizations and industry consortia. One major problem confronting integration of software applications stems from the challenges of specific data formats for exchange and sharing. The International Organization for Standardization (ISO) has been actively pursuing the development of STEP. STEP (the Standard for the Exchange of Product Model Data) is a product data integration standard to facilitate information exchange among different applications. STEP is based on the EXPRESS language [50], which enables STEP to provide an unambiguous, computer interpretable representation of product data. EXPRESS is a data definition language that is used to represent the structure of data and any constraints that may apply to the data. Examples of product models developed using STEP for the building and construction applications include CIMsteel, the steel model, and the roofing system. Software tools are commercially available to integrate STEP product models with databases and other application programs.

Another notable effort in the building and construction industry is the development by the International Alliance of Interoperability (IAI) which aims at developing a set of industry foundation classes (IFC) as a universal library of commonly defined objects throughout the lifecycle of a facility, from design to operation and maintenance [46]. IFC is a data representation standard developed specifically for defining product data for architectural and construction applications. Based on EXPRESS, IFC is designed to exchange data among Architecture, Engineering, Construction and Facilities Management (AEC/FM) applications. While the earlier IFC modules focused primarily on product data, attempts have been made to extend IFC from product modeling to support data for cost estimating and project management purposes.

Recently, XML (eXtensible Markup Language) has been fast becoming a de facto infrastructure standard for data exchange because of its extendibility, hierarchical (object) structure and the vast support by computer software and hardware vendors. XML can be used as an object representation format. XML includes a meta-markup language that consists of a set of rules for creating semantic tags used to describe data [104]. Many software programs now adopt native XML support features. Desktop applications such as Microsoft Office and AutoCAD as well as database programs such as Microsoft SQL 2000, IBM DBMS, and Oracle support XML data. The benefits of using an XML-based standard instead of an ASCII-based standard are that XML is (1) a published standard by W3C.org; (2) becoming the standard meta language for data interchange across the computer industry; (3) object-oriented (supporting advanced software development concepts); (4) readable; and (5) extendible. Widespread adoption of XML as the language for data exchange has led to the explosion of software tools that use and manipulate data. In addition, a new breed of native XML-based databases has started to emerge in the market place.

With the emerging popularity of XML, XML schemas have been proposed as ontology standards in the building and construction industry with ifcXML [59] and aecXML [47] being the two most popular XML schemas. IfcXML is an XML version of the IFC and is a fairly extensive schema (with over 400 pages) designed to enable the exchange of IFC data in an alternative XML format. In ifcXML, tags have been defined for various stages and purposes in the project life-cycle, such as product modeling, cost estimating, scheduling and maintenance. For example, *WorkSchedule*, *ScheduleTimeControl*, and *RelSequence* elements have been defined in the project scheduling domain. AecXML is a similar effort which was initially proposed by Bentley Systems in 1998 and is now also part of the IAI (International Alliance of Interoperability). AecXML provides XML-based schemas to describe information specific for data exchange among participants involved in the design, construction, and operation of buildings, plants, infrastructure, and facilities.

Current standards, such as STEP and IFC, focus on the exchange of product data. The need to integrate software applications to support processes and activities has become increasingly important. In project management, enterprise modeling, and manufacturing applications, the activities and the constraints on their occurrences need to be represented. For example, data integration occurs in business process reengineering, where enterprise models integrate processes, organizations, goals, and customers. Even when applications use the same terminology, they often associate different semantics with the terms. This clash over the meaning of the terms prevents the seamless exchange of information among the applications. Interoperability of process oriented applications must deal with the issue of the differences in terminology and representations now found in most of the project management and enterprise software applications. The Process Specification Language (PSL) has been designed to facilitate correct and complete exchange of process information among manufacturing systems [67, 80]. PSL is developed using KIF (knowledge interchange format), which is based on first-order predicate logic. Included in these applications are scheduling, process modeling, process planning, production planning, simulation, project management, workflow, and business process reengineering.

In this research, we focus on the integration of project management applications and project models. As a point of departure, PSL, an emerging international standard for process information exchange, is adopted as the information exchange language for project management applications and the usage of PSL is extended to illustrate the potential of PSL for consistency checking and constraint management. Furthermore, ifcXML is employed for the exchange of data on the product models describing a facility.

2.0 SOFTWARE INTEGRATION AND INTEROPERABILITY

One issue when different participants or organizations engage in collaborative activities is interoperability between the applications and infrastructure services within an organization and among the collaborating parties. The heterogeneity of application tool employed within and among engineering companies creates a demand for an interoperability solution to achieve software integration. There have been many approaches for developing software integration, ranging from localized integration and client-server integration to Web-based integration and distributed integration.

Localized Integration: The most primitive method for software integration involves integrating software tools locally on a machine. Integration can also be accomplished through the API (Application Protocol Interface) provided by the application programs. An application can provide software interfaces that allow other applications to communicate directly with the application. Examples of software interfaces include RA (Primavera Automation Engine) in the Primavera Project Planner and VBA (Visual Basic for Application) in Microsoft Project.

Client-Server Integration: The client-server integration model aims at leveraging the capabilities of typical corporate networks that consist of many low-end computers and a few dedicated servers. Typically, a project repository, either in neutral files or a centralized database, resides on a server. All applications communicate to the server to access the information. Most Web-based (intranet or extranet) portals developed for construction project management applications employ client server models. In essence, most of the current Web-based tools focus on providing a common project repository of data and tools.

Distributed Integration: A distributed system integration model deals with integrating applications on different (often heterogeneous) computers over private or public, local or wide area networks. Examples of mechanisms that are commonly used to support distributed applications include Remote Procedure Call (RPC) [10], messaging [52], and distributed shared memory. With the continuing proliferation of the Internet and Web-based technologies, there have been many research and development efforts to "publish" and to "support" independent applications as Web services. Conceptually, a typical Web service architecture consists of three entities: service providers, services brokers, and service requesters.

- Service providers develop Web services, register them with service brokers, and publish them on the Web.
- Service brokers act as bridges between service providers and service requesters; they also maintain detailed lists of published Web services.
- Service requesters search the brokers' lists, find the required services, and send requests to the corresponding service providers.

The development of Web services is motivated by a need to represent information, retrieve and update data, and reuse services provided by other parties over the network. Currently, some of the features constituting Web services are as follows:

- The basic principle of Web services is loose coupling; in other words, components depend less on the implementation of the others. Web services are not Remote Procedure Calls (RPCs) [10] or Common Object Request Broker Architecture (CORBA) [75]. RPC is primarily designed for tightly bounded but geographically distributed systems, while the principle behind Web services is loose coupling. In CORBA messages are manipulated by instantiating objects; however, document-style messages are used to communicate among Web services.
- Web services communicate by passing messages structured in XML and packaged according to the Simple Object Access Protocol (SOAP) [13]. XML can be used to represent data in self-describing, platform-independent text, while SOAP provides a simple protocol to create complex self-contained messages.
- Web services describe themselves using descriptive languages such as WSDL (Web Services Description Language) [28] and support their own discovery using mechanisms such as UDDI (Universal Description, Discovery and Integration) [7].

To integrate distributed services over the Web, a data standard needs to be employed, so that results can be reused by other applications. Network communication issues, such as asynchronous messaging, also need to be addressed [12]. Furthermore, mechanisms for invoking and terminating applications over the network have to be provided [62]. Many emerging languages can assist in reusing Web services and conducting business transactions [21]. Examples of some of the previous and current efforts are:

- XLANG, an extension of Web Service Description Language (WSDL), aims at facilitating the orchestration of services [89]. XLANG uses WSDL to describe the service interface of each participant. In XLANG, the basic constituents of a process definition are actions. In addition to the inherited WSDL actions (request/response, solicit response, one way, and notification), XLANG adds two new actions: timeouts and exceptions. A service with a behavior represents an interaction with other services; therefore, orchestrating services can be achieved through sequencing the actions of the services.
-
- WSFL, an XML-based language, describes Web service compositions as part of a business process definition [58]. There are two basic ways to compose Web services using WSFL: (1) a flow model where the basic constituents are activities, represented by nodes in a linked graph and each activity is associated with a service provider for the execution of the process and (2) a (global) business collaboration model to facilitate interactions between business partners.
-
- BPEL4WS, a product of the merger of WSFL and XLANG, provides the formal specification of business processes and business interaction protocols [3]. BPEL4WS supports two distinct usage scenarios: implementing executable business processes and describing non-executable abstract processes. As an executable process implementation language, BPEL4WS is used to define a new Web service by composing

a set of existing services. For the second role, BPEL4WS supports modeling the behavior of business protocols.

- The ebXML language is a set of specifications that enables enterprises to conduct business over the Internet [34]. In other words, ebXML defines a framework allowing enterprises to find each other and to conduct business based on well-defined XML messages. BPML, a meta-language for modeling business processes, provides an abstracted execution model for collaborative and transactional business processes [8]. BPML represents business processes as the interleaving of control flow, data flow, and event flow. BPML and ebXML are complementary standards for business processes. While ebXML allows users to specify the public interface of their business processes, BPML provides a standard way to describe the corresponding private implementations.

- DAML-S is another ontology that has been developed to assist in automating Web service tasks (e.g., discovery, composition, invocation, and monitoring) [5]. DAML-S defines an ontology, within the framework of the DARPA Agent Markup Language, for Web services.

3.0 CONCLUSION

In summary, there have been many significant developments in recent years to help build autonomous Web services. Web services have broad applications, ranging from real-time price quoting to workflow management and enterprise application integration [18]. The objective of this research is to extend current Web service models not only to allow integration of business or engineering applications but also to provide facilities for simulation using standalone and Web-based applications.

The focus of this research is to develop a simulation access language (SimAL) and infrastructure framework for the access of project management applications to support simulation and decision making. The effort follows closely the development of SimQL, which consists of infrastructure environment and a descriptive language to execute and reuse simulation results [99, 100]. The SimQL language includes a schema language and a query language, which handle model creation and data query respectively.

The SimQL schema language allows users to register wrappers, to create models, and to update models. Furthermore, the SimQL query language allows users to query information based on the created models. Users can query information from the results using a SQL-like SELECT statement. SimAL uses a Flow-based Infrastructure for Composing Autonomous Services (FICAS) [60-62].

In FICAS, a service composition language, CLAS (Compositional Language for Autonomous Services), which was derived from the Composition Language for Autonomous Megamodules (CLAM) [79] has been provided to specify and to invoke a megaservice. In addition to the primitives for invoking services, CLAS also provides elements for asynchronous control, such as WHILELOOP, LOCAL and BRANCH elements. SimAL modifies CLAS and extends its developments to support simulation of engineering and project management activities.

Reference

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

[2] B. Akinci, M. Fischer, R. Levitt, and R. Carlson. "Formalization and Automation of Time-Space Conflict Analysis," *Journal of Computing in Civil Engineering*, 16(2):124-134, 2002.

[3] T. Andrews, F. Curbera, H. Dholakia, Y. Goland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic, and S. Weerawarana. *BPEL4WS Specification: Business Process Execution Language for Web Services Version 1.1*, http://www-106.ibm.com/developerworks/library/ws-bpel/, 2003.

[4] I. Androutsopoulos, G. D. Ritchie, and P. Thanisch. "Natural Language Interfaces to Databases -- An Introduction," *Journal of Natural Language Engineering*, 1(1):29-81, 1995.

[5] A. Ankolekar, M. Burstein, J. R. Hobbs, O. Lassila, D. L. Martin, S. A. McIlraith, S. Narayanan, M. Paolucci, T. Payne, K. Sycara, and H. Zeng. "DAML-S: Semantic Markup for Web Services," *Proceedings of the International Semantic Web Working Symposium*, Stanford, CA, 2001.

[6] J. Antill and R. Woodhead. *Critical Path Methods in Construction Practice*, 2nd Ed., John Wiley & Sons, Inc., New York, 1970.

[7] N. Apte and T. Mehta. *UDDI: Building Registry-based Web Services Solutions*, Pearson Education, 2003.

[8] A. Arkin. *Business Process Modeling Language*, Business Process Management Initiative, 2002.

[9] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*, Addison Wesley Ltd, 1999.
[10] A. D. Birrell and B. J. Nelson. "Implementing Remote Procedure Calls," *ACM Transactions on Computer Systems*, 2(1):39-59, 1984.
[11] B. C. Bjork. "Basic Structure of a Proposed Building Product Model," *CAD*, 21(2):71-78, 1989.
[12] A. Bosworth. "Developing Web Services," *Proceedings of the International Conference on Data Engineering*, Heidelberg, Germany, pp. 477-481, 2001.
[13] D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. Nielsen, S. Thatte, and D. Winer. *Simple Object Access Protocol (SOAP)*, W3C Note, http://www.w3.org/TR/SOAP, 2000.
[14] E. J. Breck, J. D. Burger, D. House, M. Light, and I. Mani. "Question answering from large document collections," *Proceedings of AAAI Fall Symposium on Question Answering Systems*, North Falmouth, MA, pp. 26-31, 1999.
[15] E. J. Breck, J. D. Burger, L. Ferro, L. Hirschman, D. House, M. Light, and I. Mani. "How to Evaluate Your Question Answering System Every Day and Still Get Real Work Done," *Proceedings of LREC-2000, the 2nd International Conference on Language Resources and Evaluation*, Athens, Greece, 2000.
[16] P. Brothner. "Kawa: Compiling Scheme to Java," *Proceedings of Lisp Users Conference*, Berkeley, CA, 1998.
[17] L. D. Brown. *Managing Conflict Among Groups*, Prentice-Hall, Inc., 1995.
[18] M. Burner. "The Deliberate Revolution Transforming Integration With XML Web Services?," *ACM Queue*, 1(1):28-37, 2003.
[19] J. Burstein, K. Kukich, S. Wolff, C. Lu, and M. Chodorow. "Computer Analysis of Essays," *Proceedings of NCME Symposium on Automated Scoring*, 1998.
[20] C. Callison-Burch and P. Shilane. *A Natural Language Question and Answer System*, Stanford University, Stanford, CA, 2000.
[21] S. Chandrasekaran, G. Silver, J. Miller, J. Cardoso, and A. Sheth. "Web Service Technologies and their Synergy with Simulation," *Proceedings of the 2002 Winter Simulation Conference (WSC'02)*, San Diego, CA, pp. 606- 615, 2002.
[22] J. Cheng, B. Kumar, and K. H. Law. "A Question Answering System for Project Management Applications," *Advanced Engineering Informatics*, 16(4):277-289, 2002.
[23] J. Cheng and K. H. Law. "Using Process Specification Language for Project Information Exchange," *Proceedings of the 3rd International Conference on Concurrent Engineering in Construction*, Berkeley, CA, pp. 63-74, 2002.
[24] J. Cheng, P. Trivedi, and K. H. Law. "Ontology Mapping Between PSL and XML-Based Standards For Project Scheduling," *Proceedings of the 3rd International Conference on Concurrent Engineering in Construction*, Berkeley, CA, pp. 143-156, 2002.
[25] J. Cheng, M. Gruninger, R. D. Sriram, and K. H. Law. "Process Specification Language for Project Information Exchange," *International Journal of Information Technology in Architecture, Engineering and Construction*, 1(4):307-328, 2003.
[26] J. Cheng, K. H. Law, and B. Kumar. "Integrating Project Management Applications as Web Services," *Proceedings of the 2nd International Conference on Innovation in Architecture, Engineering and Construction*, Loughborough University, UK, 2003.
[27] J. Cheng, K. H. Law, and B. Kumar. "Online Collaboration of Project Management Applications," *Proceedings of the 11th International Workshop of the EG-ICE*, Weimar, Germany, 2004.
[28] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana. *WSDL. Web Services Description Language*, W3C Note, World Wide Web Consortium, http://www.w3.org/TR/wsdl, 2000.
[29] A. Diekema, X. Liu, J. Chen, H. Wang, N. McCracken, O. Yilmazel, and E. D. Liddy. "Question Answering : CNLP at the TREC-9 Question Answering Track," *Proceedings of Proceedings of the 9th Text REtrieval Conference (TREC-9)*, National Institute of Standards and Technology, Gaithersburg, MD, pp. 501-510, 2000.
[30] A. M. Dubois and F. Parand. "COMBINE Integrated Data Model," *Proceedings of CIBSE National Conference*, Manchester, UK, pp. 96-108, 1993.
[31] C. L. Dym and R. E. Levitt. *Knowledge-Based Systems In Engineering*, McGraw-Hill, Inc., 1991.
[32] C. M. Eastman. *Building Product Models: Computer Environments Supporting Design and Construction*, CRC Press, 1999.
[33] D. Echeverry, W. Ibbs, and S. Kim. "Sequence Knowledge for Construction Scheduling," *Journal of Construction Engineering and Management*, 117(1):118-130, 1991.

[34] B. Eisenberg, A. Grangard, and D. Nickull. *ebXML Technical Architecture Specification v1.0.4*, Organization for the Advancement of Structured Information Standards, http://www.ebxml.org/specs/ebTA.pdf, 2001.
[35] Fatdog. *XQEngine Introductory Tutorial*, Fatdog Software, http://www.fatdog.com/tutorial.html, 2002.
[36] C. Fellbaum and G. Miller. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*, MIT Press, 1998.
[37] J. Fowler. *STEP for Data Management, Exchange and Sharing*, Technology Appraisals Ltd., UK, 1995.
[38] G. Frank. "A General Interface for Interaction of Special-Purpose Reasoners within a Modular Reasoning System," *Proceedings of Question Answering Systems, Papers from the 1999 AAAI Fall Symposium*, North Falmouth, Massachusetts, pp. 57-62, 1999.
[39] T. Froese, M. Fischer, F. Grobler, J. Ritzenthaler, K. Yu, S. Sutherland, S. Staub, B. Akinci, R. Akbas, B. Koo, A. Barron, and J. Kunz. "Industry Foundation Classes for Project Management-A Trial Implementation," *Electronic Journal of Information Technology in construction*, 4:17-36, 1999.
[40] GAO. *Computer Aided Building Design*, GAO, 1978.
[41] F. K. Garas and I. Hunter. "CIMSteel (Computer Integrated Manufacturing in Constructional Steelwork) - Delivering the Promise," *Structural Engineering*, 76(3):43-45, 1998.
[42] H. Garcia-Molina, J. D. Ullman, and J. D. Widon. *Database Systems: The Complete Book*, Prentice Hall, 2001.
[43] M. R. Genesereth and R. Fikes. *Knowledge Interchange Format Reference Manual - Version 3*, Report No. CSD-Logic-92-1, Department of Computer Science, Stanford University, Stanford, CA, 1992.
[44] E. F. Gould. *Managing the Construction Process: Estimating, Scheduling, and Project Control*, Prentice Hall, 2002.
[45] E. Hovy and C. Y. Lin. *Automated Text Summarization in SUMMARIST*, MIT Press, 1999.
[46] IAI. *Industry Foundation Classes*, International Alliance for Interoperability, Washington, DC, 1997.
[47] IAI. *AecXML*, International Alliance for Interoperability, http://www.aecxml.org, 2002.
[48] IGES. *Initial Graphics Exchange Standard (IGES), Version 5.1*, National Bureau of Standards, Gaithersburg, MD, 1991.
[49] ISO. *Industrial Automation Systems-Product Data Representation and Exchange*, International Organization for Standardization, 1989.
[50] ISO. *EXPRESS Language Reference Manual: External Representation of Product Definition Data*, ISO DIS10303 Part 11, 1991.
[51] P. Jacobs and L. Rau. "SCISOR: Extracting Information from On-line News," *Communications of the ACM*, 33(11):88-97, 1990.
[52] V. Karamcheti and A. A. Chien. "A Comparison of Architectural Support for Messaging in the TMC CM-5 and the Cray T3D," *Proceedings of ISCA 95*, Santa Margherita, Italy, pp. 298-307, 1995.
[53] D. Klein and C. Manning. "Parsing with Treebank Grammars : Empirical Bounds, Theoretical Models, and the Structure of the Penn Treebank," *Proceedings of the 39th Annual Meeting of the ACL*, Toulouse, France, pp. 330-337, 2001.
[54] K. H. Law and M. K. Jouaneh. "Data Modeling for Building Design," *Proceedings of the 4th Conference on Computing in Civil Engineering*, ASCE, pp. 21-36, 1986.
[55] K. H. Law, D. L. Spooner, and M. K. Jouaneh. "Abstraction Database Concepts for Engineering Modeling," *Engineering with Computers*, 2(2):79-84, 1987.
[56] K. H. Law, T. Barsalou, and G. Wiederhold. "Management of Complex Structural Engineering Objects in a Relational Framework," *Engineering with Computers*, 6:81-92, 1990.
[57] S. M. Lewandowski. "Framework for Component-Based Client/Server Computing," *ACM Computing Surveys*, 30(1):3-27, 1998.
[58] F. Leymann. *Web Services Flow Language (WSFL 1.0)*, IBM Corporation, http://www4.ibm.com/software/solutions/webservices/pdf/WSFL.pdf, 2001.
[59] T. Liebich. *XML Schema Language Binding of EXPRESS for ifcXML*, International Alliance for Interoperability, 2001.
[60] D. Liu, K. H. Law, and G. Wiederhold. "Data-flow Distribution in FICAS Service Composition Infrastructure," *Proceedings of the 15th International Conference on Parallel and Distributed Computing Systems*, Louisville, KY, 2002.
[61] D. Liu, J. Peng, K. H. Law, G. Wiederhold, and R. D. Sriram. "Composition of Autonomous Services with Distributed Data Flows and Computations," *submitted to ACM Transactions on Internet Technology*, http://mediator.stanford.edu/papers/FICAS.pdf, 2003.

[62] D. W. Liu. *A Distributed Data Flow Model For Composing Software Services*, Ph.D. Thesis, Department of Electrical Engineering, Stanford University, 2003.
[63] M. P. Marcus, B. Santorini, and M. A. Marcinkiewica. "Building a large annotated corpus of English: the Penn Treebank," *Computational Linguistics*, 19(2):310-330, 1993.
[64] R. Mayer and J. Linden. *Using Iges, Dxf and Cals for Cad/Cam Data Transfer: The Complete Guide to Data Transfer Standards*, Management Roundtable, 1992.
[65] W. W. McCune. *Otter 3.0 Reference Manual and Guide*, Report No. ANL-94/6, Mathematics and Computer Science Division, Argonne National Laboratory, 1994.
[66] K. McKinney and M. Fischer. "Generating, Evaluating and Visualizing Construction Schedules with CAD Tools," *Automation in Construction*, 7(6):433-447, 1998.
[67] C. Menzel and M. Gruninger. "A Formal Foundation for Process Modeling," *Proceedings of Formal Ontology in Information Systems*, Ogunquit, Maine, pp. 256-269, 2001.
[68] P. Naur and J. Backus. "Report on the Algorithmic Language ALGOL 60," *Communications of the ACM*, 3(5):299-314, 1960.
[69] NRC. *The 1984 Workshop on Advanced Technology for Building Design and Engineering*, Building Research Board, NRC, 1985.
[70] NRC. *The 1985 Workshop on Advanced Technology for Building Design and Engineering*, Building Research Board, NRC, 1986.
[71] NRC. *The 1986 Workshop on Integrated Data Base Development for the Building Industry*, Building Research Board, NRC, 1987.
[72] S. K. Park, K. I. Lim, and E. D. Kim. "A STEP-based Integrated Structural Design System for Steel Framed Buildings," *Proceedings of the 8th International ASCE Conference on Computing and Building Engineering*, Stanford, CA, pp. 788-795, 2000.
[73] F. Pereira. *Logic for Natural Language Analysis*, Technical Note 275, SRI International, 1983.
[74] E. Pitt and K. McNiff. *java.rmi: The Remote Method Invocation Guide*, Addison Wesley, 2001.
[75] A. Pope. *The CORBA Reference Guide: Understanding the Common Object-Request Broker Architecture*, Addison Wesley, 1998.
[76] J. Robie. *XQL Tutorial*, Software AG, http://ibiblio.org/xql/xql-tutorial.html, 1999.
[77] E. Roman, S. W. Ambler, and T. Jewell. *Mastering Enterprise JavaBeans*, 2nd Ed., John Wiley & Sons, 2001.
[78] J. Roy and A. Ramanujan. "Understanding Web Services," *IT Professional*, 3(6):69-73, 2001.
[79] N. Sample, D. Beringer, L. Melloul, and G. Wiederhold. "CLAM: Composition Language for Autonomous Megamodules," *Proceedings of the 3rd International Conference on Coordination Models and Languages*, Amsterdam, Netherland, pp. 291-306, 1999.
[80] C. Schlenoff, M. Ciocoiu, D. Libes, and M. Gruninger. "Process Specification Language: Results of the First Pilot Implementation," *Proceedings of the International Mechanical Engineering Congress and Exposition*, Nashville, Tennessee, pp. 529-539, 1999.
[81] C. Schlenoff, M. Gruninger, and M. Ciocoiu. "The Essence of the Process Specification Language," *Transactions of the Society for Computer Simulation*, 16(4):204-216, 1999.
[82] I. Schröder. *Ingo's Collection Of POS Taggers*, http://nats-www.informatik.uni-hamburg.de/~ingo/icopost/, 2002.
[83] J. Siméon. *Galax Implementation of XQuery*, XQuery Implementation Panel, XML 2001, Orlando, http://db.bell-labs.com/galax/, 2002.
[84] A. Singh and D. A. Vlatas. "Using Conflict Management for Better Decision Making," *Journal of Management in Engineering*, 7(1):70-82, 1989.
[85] A. Singh and H. M. Johnson. "Conflict Management Diagnosis at Project Management Organizations," *Journal of Management in Engineering*, 14(5):48-63, 1998.
[86] Sourceforge. *Sourceforge Xquench Project*, Open Source Development Network, http://sourceforge.net/projects/xquench/, 2002.
[87] STEP. *ST-Developer*, STEP Tools Inc., Rensselaer Technology Park, Troy, NY, 1997.
[88] T. L. Thai. *Learning DCOM*, O'Reilly & Associates, 1999.
[89] S. Thatte. *XLANG: Web Services For Business Process Design*, Microsoft Corporation, 2001.
[90] D. Vanier. "Product Modeling: Helping Life Cycle Analysis of Roofing Systems," *Proceedings of the Life Cycle of Construction IT Innovations*, Stockhelm, Sweden, pp. 423-235, 1998.
[91] M. Vasconcellos and M. Leon. "SPANAM and ENGSPAN: Machine translation at the Pan American Health Organisation," *Computation Linguistics*, 11(2-3):122-136, 1985.

[92] Vite. *SimVision Help*, Vite SimVision Help Manual, Vite Corporation, 2002.
[93] W3C. *XML-QL: A Query Language for XML*, World Wide Web Consortium, http://www.w3.org/TR/NOTE-xml-ql/, 1998.
[94] W3C. *XML Path Language (XPath) Version 1.0*, World Wide Web Consortium, Recommendation 16, 1999.
[95] W3C. *XQuery 1.0: An XML Query Language*, W3C Working Draft 20, 2001.
[96] Webcor. *Ronal McDonal Housing Expansion Quarterly Project Review*, Webcor Builders, April 2003.
[97] WebGain. *Java Compiler Compiler (JavaCC) - The Java Parser Generator*, http://www.webgain.com/products/java_cc/, 2001.
[98] G. Wiederhold. *Strategic Uses of Information Technologies*, Presentation at Stanford Graduate School of Business, Stanford, CA, 1996.
[99] G. Wiederhold, R. Jiang, and H. Garcia-Molina. "An Interface Language for Projecting Alternatives in Decision-Making," *Proceedings of AFCEA Database Colloquium*, San Diego, CA, 1998.
[100] G. Wiederhold and H. Garcia-Molina. "SimQL: an Interface for Integrating Access to Simulations into Information Systems," *Proceedings of DARPA-JFACC Symposium*, San Diego, pp. 259-262, 1999.
[101] W. A. Woods. "Progress in Natural Language Understanding: An Application to Lunar Geology," *Proceedings of AFIPS Conference*, pp. 441-450, 1973.
[102] L. Wos and G. W. Pieper. *A Fascinating Country in the World of Computing: Your Guide to Automated Reasoning*, World Scientific Publishing Company, Singapore, 2000.
[103] P. Wyckoff, S. W. McLaughry, T. J. Lehman, and D. A. Ford. "TSpaces," *IBM Systems Journal*, 37(3):454-474, http://www.almaden.ibm.com/cs/TSpaces, 1998.
[104] M. J. Young. *Step by Step XML*, Microsoft Press, 2001.
[105] R. Zajac. "Towards Ontological Question Answering," *Proceedings of ACL Open Domain Question Answering Workshop*, Toulouse, 2001.