PROCESS CONNECTORS: LINKING DISTRIBUTED PROCESSES IN THE CONSTRUCTION SUPPLY CHAIN

William J. O'Brien[1], Ting-Kwei Wang[2], Mohsin Siddiqui[3], Joachim Hammer[4], and Ronald Wakefield[5]

ABSTRACT

This paper reviews the architecture, approaches, and research issues associated with the development of process connectors. Process connectors are software links that support coordination of distributed processes (schedules in particular) across the many firms on a construction project. The architecture of a process connector is presented as an efficient approach for both initial setup and run-time capabilities. Process connectors include a stub for connection to each firm and a bridge between stubs. A stub represents company information, including schedules and resource assignments, and makes this information available for connection and evaluation. A bridge component validates schedule mappings between firms and includes limited capabilities for schedule evaluation and recommendation when participating firms' schedules do not correspond. Beyond the architecture, specific research issues reviewed include discovery and representation of firm specific information as well as constraint representation and propagation.

KEY WORDS

Schedule Integration, Construction Process Modeling, Decision Support, Constraint Propagation, Distributed Computing in Engineering

INTRODUCTION

The construction industry has long used specific methods to model individual processes from the perspective of a single project stakeholder, such as bar charts and critical path networks (Antill and Woodhead 1990) and discrete event simulation (Halpin and Riggs 1992). These methods work well to model a specific process. However, difficulties arise in the course of sharing information between several stakeholders when combining their processes. Schedules,

---

[1] Assistant Professor, Department of Civil, Architectural and Environmental Engineering, University of Texas at Austin, 1 University Station C1752, Austin, TX 78712-0273, USA, wjob@mail.utexas.edu

[2] PhD. Candidate, Department of Civil, Architectural and Environmental Engineering, University of Texas at Austin, 1 University Station C1752, Austin, TX 78712-0273, USA, tingkwei@mail.utexas.edu

[3] PhD. Candidate, Department of Civil, Architectural and Environmental Engineering, University of Texas at Austin, 1 University Station C1752, Austin, TX 78712-0273, USA, mohsink@mail.utexas.edu

[4] Associate Professor, Department Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611, USA

[5] Professor & Head, School of Property, Construction and Project Management, Design and Social Context Portfolio, RMIT University, Melbourne, Australia

the most common example, are generally formulated from the perspective of one party (say the general contractor) who must then verify with other parties (e.g., subcontractors and suppliers) that the schedule is valid. Traditionally, this is a time consuming task and is difficult to perform iteratively when schedules change (O'Brien et al 1995). Some commercial software attempts to speed this process by posting contractor generated schedules on the web and allowing subcontractors to confirm their availability. However, this information is largely limited to dates and primarily represents the viewpoint of the general contractor or schedule creator. Several authors have critiqued such information as too limited to represent the various constraints on-site as well as resource constraints (e.g. Choo et al. 1999; O'Brien and Fischer 2000). Various improved scheduling models have been proposed and tested in practice, including a checklist based planning approach (Ballard and Howell 1998) and IT approaches that combine constraint planning with decision support algorithms (Sriprasert and Dawood 2003).

Whatever the schedule coordination approach, existing methods do not overcome the difficulty of sharing information between firms. An electronic solution is needed that can scale to communication across several firms while also representing firms' specific worldview, including constraints. Beyond the basic need for a physical connection to data, the researchers suggest the representation problem has two aspects: First, a mapping problem where schedules have differing levels of detail. Second, a constraint representation and propagation problem where local constraints (such as resource availability) must be propagated globally. The mapping problem is subtle; consider that the research literature in scheduling suggests that different level of detail can be accommodated with hammocked or hierarchical activities (Antill and Woodhead 1990). For example, a general contractor has a master schedule and subcontractors have detailed breakouts of each activity on the master schedule. If so, then creation of mappings between schedules can be simplified to a 1:n mapping. However, the authors' research reveals examples where contractors and subcontractors approach schedules in differently, where within a larger hierarchy (e.g., 1st floor, 2nd floor) the mappings can only be reduced to an m:n mapping. Such complex mappings require a rich and flexible model to support useful coordination of firms' schedules. Representing and propagating constraints on m:n mappings requires further richness and flexibility of the base model. The researchers propose a tree-based approach (outlined below and detailed in a companion paper (Siddiqui et al. 2006)) as a flexible approach for schedule mapping and constraint propagation.

Concomitant with the problem of representing schedules, there needs to be an efficient mechanism for setting up software links between firms so schedule information can be rapidly exchanged. A logical implementation choice is a web services, which are broadly designed to allow rapid instantiation of specific services in a flexible and distributed manner (Alonso et al. 2003). Indeed, recent research by Law and his students demonstrates the use of web services to aid schedule monitoring and coordination, employing a variety of construction applications (Cheng et al. 2003). That said, there remains considerable development to design specific web service components as well as make architectural choices about connections between components that suit the needs for functionality, scalability, and flexibility.

The remainder of this paper describes the process connectors architecture and its functionality in response to specific requirements to support distributed process coordination, outlines an approach to mapping between processes (schedule information in particular) and discusses specific research issues relating to implementation.

PROCESS CONNECTORS ARCHITECTURE

Before discussing specific research and implementation issues, we first describe the overall process connectors architecture. As noted above, the fundamental challenge is to develop an approach that supports rich mappings between firms' internal processes. Some specific requirements include:

- Representation and links to firms' internal data and process representations
- Maintenance of mappings between firms' processes
- Scalability to coordinate processes across a large number of firms (i.e., large projects)
- Ability to provide coordination support with temporarily disconnected/off-line firms
- Support advanced decision support applications
- Support for human directed what-if analysis
- Support confirmation and validation of revised schedules

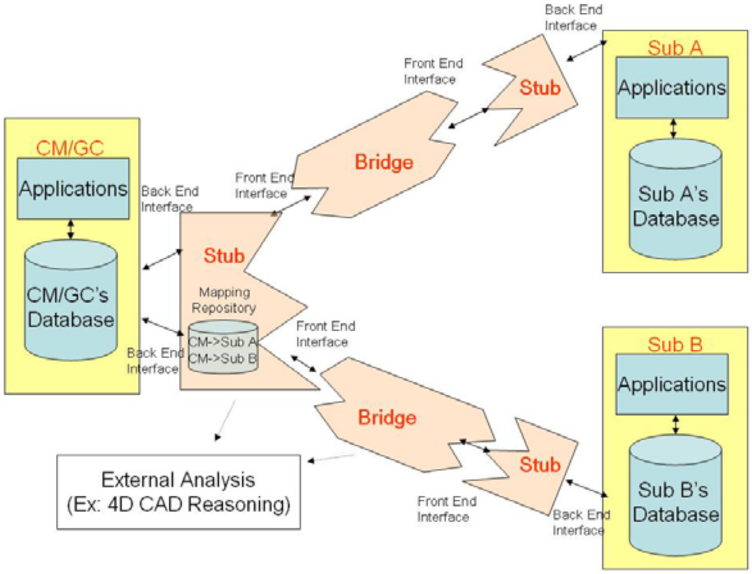

Figure 1: Process Connectors Architecture

Based on those requirements, we propose the process connector architecture as illustrated in Figure 1. Our proposed process connector architecture consists of bridge and stub components. The stubs are attached to firms' data and applications and translate this information to the internal data format of the process connector architecture. The bridge component supports mappings across firms' internal processes, performs basic analysis, and directs communications between firms (stubs). The design is intended to be modular, allowing easy implementation within a web services framework as well as support for additional components. For example, external analysis packages (e.g., 4D CAD, advanced planning and scheduling systems) can be linked to the process connectors architecture. We postulate this connection would be made at the construction manager/general contractor (CM/GC) stub where data from multiple firms would be available. However, a similar connection could be made at any stub (for example, to provide advanced cost or schedule analysis planning to a subcontractor using the process connectors data format as a common standard to link to firms applications).

Below we briefly review the functionality of each component during build-time and run-time as well as discuss the basic mapping approach with respect to the functional requirements listed above.

FUNCTIONAL COMPONENTS

Stubs

Stubs are a key component of the process connectors architecture as they are the connection point between a firm's internal data and applications and connection to other components and analysis. Stubs are key to scalability as they translate firms' process information to a data format for sharing and further processing. Stubs communicate information out of a firm as well as send information back to the firm (for example, a revised schedule for confirmation and acceptance by subcontractor management). Stubs must thus support validation, implying that beyond being a wrapper or translator, they must also contain some application logic to process information as well as an interface for setup and confirmation (or, alternately, present an interface within an existing application such as MS Project).

We expect stubs deployed commercially will vary in complexity based on the nature of the internal applications of the firm. For prototype purposes, we posit two types of firms, each with a supporting stub: First, a sophisticated firm that maintains process information on existing commercial applications. The sophisticated firm stub wraps to existing applications while additionally providing an interface for utilization of functionality within the process connectors architecture, such as confirmation of schedule changes or initiation of a schedule reconciliation process. Second, an unsophisticated firm that uses minimal existing software. The unsophisticated firm stub provides a GUI and limited scheduling and resource management capabilities, essentially providing stand-alone scheduling software capability to the firm (picture GUI instead of firm's internal applications in figure 1). Note that we posit the CM/GC will always be a sophisticated firm – if not, it is unlikely the firm would be able to benefit from the process connectors applications.

For both the sophisticated and unsophisticated case, there is an initial, one-time setup or build-time process. In this process, the firm management must map the firm data and processes to the internal data format of the stub/process connectors architecture. For the unsophisticated case, this is essentially setting up a software application with resources, schedule templates, etc. The sophisticated firm case is more complicated as a connection must be made to existing applications and the translation must be made between data formats. We call this connection the back-end interface as shown in figure 1. The development of software applications with APIs or that can be exposed as web services, as well as the development of data standards, makes such translation steps easier as they provide a foundation to build from. When data is not available in an understood data format, automated discovery processes such as those detailed in the SEEK project (O'Brien et al. 2002) can be used to speed setup. In any case, managers must verify the translation; ideally this can be done by operations managers with limited input from technical staff. For both the sophisticated and unsophisticated firm, the build-time process need be completed only once and can be reused for many projects, limiting the overall setup burden on the firm. Of course, incremental improvements or changes can be made by reinitiating the build-time process. The authors believe the setup-once/use many times aspect of the stub provides a key aspect of scalability for the process connectors architecture.

During run-time, the stub is ideally intended to operate in a fully connected mode with the rest of the process connectors architecture. Hence a request from a bridge component for updated schedule information (received through the front-end interface in figure 1) would generate a near real-time reply from the stub (accessing firm data through the back-end interface). However, the stub should provide useful data even when disconnected. To accomplish this, data on a specific project is stored in a time-stamped XML document. The bridge can import this XML document for further processing. Generation of the XML document with firm data pertinent to the project in question is one of the key run-time capabilities of the stub. Similarly, the stub accepts an XML document in return from the bridge component and checks to see if changes are needed in the subcontractor schedule. The subcontractor management can accept/reject/propose alternatives to the proposed schedule through a GUI (a second key run-time capability).

Bridge

The bridge component connects stubs, providing a pathway for information to flow between firms. However, the bridge is more than a conduit; it provides important build-time functionality at project initiation and supports analysis of distributed processes to ensure coordination during run-time.

While stubs are generated once per firm and persist indefinitely, mappings must be made on a per project basis. It is the build-time function of the bridge component to support development of the mappings between firms' processes (the CM/GC and subcontractor following figure 1). Mappings are constructed on a pair wise basis between firms (see discussion below and extended discussion in (Siddiqui et al. 2006)). This is done by importing the XML project specific data document generated by the stub and then manually matching activities. Some automation is possible at this stage. However, as it is likely the number of activities for a subcontractor's portion of a project is relatively small (based on collected examples, fewer than 100 in almost all case and fewer than 20 in many cases), the authors do not see manual mapping as a difficult activity. Once the mappings are made and verified – that is, activities between firms are associated with each other and humans have approved the association – the mapping function generates an XML document containing the mapping information. The verified (and validated, see below) mappings are stored in a mapping repository (figure 1). The verification process is shown conceptually in figure 2, where the mapping portion of the bridge component imports the firms' project specific XML documents, conducts and verifies the mappings through a GUI, and generates a verified mapping XML document for the pair. We separate the verification stage from the initial mapping as managers at both firms may wish to verify the accuracy of the mapping (only one need conduct the initial mapping). As with build-time for the stub, the mapping process can be repeated to account for changes in at each firm.

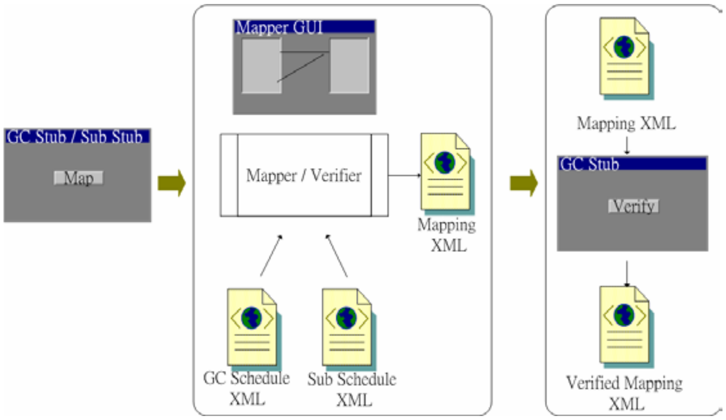

Figure 2: Mapping construction and verification process

Once mappings have been verified during build-time, they can be used during runtime to support various analysis functions to ensure coordination of the distributed processes. The first such analysis function is a validation process that ensures dates and related constraints for sets of mapped activities are not in conflict (e.g., scheduled dates may not coincide and hence the mapped schedules are invalid). Validation is an automatic process; if all matches are valid, then the verified XML mapping document is also recorded as validated. If the matches are invalid, several options are possible. First, the bridge can note what parts of the mappings are invalid and send a report to the affected firms via an XML document (a function of the stub is to process such a document). Second, the bridge can conduct analysis and propose recommendations for a new schedule. This requires a second phase of the validation process where precedence constraints are mapped between subcontractor activities; here, we use the precedence constraint information of the CM/GC as a guide. Overall, we expect the analysis performed in the bridge to be limited to relatively simple changes such as a right shift and validation of subsequent mappings. Thus a third capability of the bridge is to support data collection for external analysis (e.g., the LEWIS system that works combines 4D analysis and subcontractor resource constraints (Sriprasert and Dawood 2003)). Analysis systems such as LEWIS would need only one connection to the process connectors architecture; the bridge and stub components could relay proposed schedule changes to affected firms for confirmation or rejection. Additionally, the bridge component could re-validate the proposed changes of the analysis tools as an additional check; this may be particularly useful if the analysis tool focused on a subset of firms in the project.

It is useful to note that analysis activities in the bridge can be triggered both manually and automatically. Once initial mappings have been verified in the build-time phase, a validation process would be launched automatically to provide feedback to the project participants. Similarly, changes in schedule or other constraints at any one firm could automatically launch the validation process; if invalid, participants could manually access further analysis tools for recommendations.

MAPPINGS

A central component of the process connectors architecture and functionality is representation of mappings between processes. Mappings provide the basis for analysis on and, hence, coordination of, distributed processes. The mapping approach is described in more detail in a companion paper (Siddiqui et al. 2006), so discussion is limited here to a few central elements.

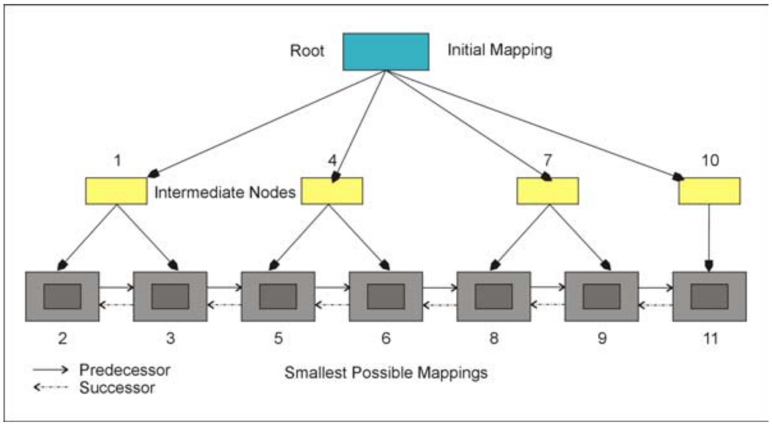

Figure 3: Tree based index for mapping between firms' processes

As firms have a variety of constraints and operate with different levels of detail, it is important that mappings have a rich and flexible representation. A tree representation is used as it allows decompositions of process data to multiple levels of representation. This is shown in figure 4, where there is a mapping hierarchy. The root node associates processes at the most aggregated level of detail (i.e., project to project) with further decomposition at intermediate nodes (e.g., at the second level in the tree, first floor to first floor) down to the smallest possible decomposition. In the context of schedules, the smallest possible decomposition contains sets individual activities in each firms' individual schedule that cannot be further split apart. Smallest possible decomposition mappings can be 1:1, 1:n, or m:n associations. Additional information (e.g., resource constraints) can be added to each node or leaf as needed, allowing flexible representation of constraints that can vary depending on the detail available at each firm. The tree based structure also makes possible reasoning based on different levels of aggregation. For example, it may be useful for analysis tools to make use of mapping data at intermediate nodes (for example, a 3D/4D tool may use data collected by floor as well as smallest possible decompositions).

It is useful to note some further definitions the authors have developed to support logical construction and operations on mappings. First, we separately define a controlling firm and a coordinating firm. A coordinating firm contains the master schedule and hence the logical association between smallest possible decompositions or leaf nodes on the tree. The coordinating firm (typically the CM/GC) thus defines the predecessor and successor information shown in figure 3. In contrast, a controlling firm limits changes within each smallest possible decomposition. For example, a smallest possible decomposition may be invalid because the dates for each firm do not coincide. The controlling firm would indicate, as a first pass, that the other firm make changes to its schedule. The controlling and coordinating firms do not have to be the same. A CM/GC may maintain the master schedule and hence coordinate the overall progression of activities, but individual subcontractors with resource constraints may control schedule changes for specific activities.

DISCUSSION AND RESEARCH ISSUES

The authors believe the process connectors architecture meets the many requirements for implementation outlined above. It is a modular design and amenable to implementation in a web services framework. This provides many tools to support development and deployment. Ultimately, however, web services is just an implementation choice. Keys to scalability stem from the splitting the architecture into many stubs attached to each firm (serving many projects) and a bridge component for construction, maintenance, and analysis of mappings. This split limits the build-time activities (stubs need be generated only once) while providing firms a generalized gateway to share process information. Similarly, it allows assembly of collections of firms on a per-project basis using the functionality of the bridge components. Other aspects of the design, such as provision for analysis when firms are off-line by storing process data in XML documents as well as provision for connection to higher-level analysis tools increases both the power and scalability of the architecture to meet the diverse needs of the construction industry.

To-date the authors have constructed the basic components of the process connectors architecture and defined the specifications for the XML documents that contain the internal data flows. Full implementation, verification and testing within a web-services framework is underway. The authors have little doubt the architecture can be fully implemented without major problems. However, several research questions remain that pertain to the power and generality of the process connectors architecture. We briefly discuss these below:

DISCOVERY AND REPRESENTATION OF FIRM SPECIFIC INFORMATION

There are many firms on a project, with an associated variety of information systems. The SEEK project led by authors O’Brien and Hammer (O'Brien et al. 2002) provides promise that physical connection and translation is possible for heterogeneous systems. However, this is limited to data and simple business rule translation. Broader procedures and complex constraints are more difficult to represent. Insofar as they can be codified, it should be possible to represent them, but the extent to which they can easily be represented and utilized within a process connectors or similar framework is unknown. Reuse and abstraction of firm specific process information is a broad question that extends beyond this study; our work will help gain insight into this problem.

More specifically, we can test the extent to which a generalized ontology and supporting specification language for deployment can be generalized for application across sectors of the construction industry. While much current thinking in standards development suggests that it is impossible for a standard to completely scale to all firms (e.g., Turk 2001), it is also recognized that a shared ontology is necessary for communication between firms. Specific applications can have more powerful and general shared ontologies within their niche. For example, Smith and Becker (1997) present a scheduling core ontology that they have developed across multiple applications, and (Cheng et al. 2003) demonstrate that the Process Specification Language (Schlenoff et al. 2000) can be used for a variety of construction process information. A basic research problem we address is thus construction and evaluation of a scheduling and constraint process ontology, testing the ability of standardized ontology to represent a variety of firms' process data and associated constraints.

CONSTRAINT REPRESENTATION AND PROPAGATION

Closely linked to evaluation of an ontology to represent firms' process information is definition of the needed detail to support coordination of distributed processes. As noted above, firms can have mappings at the smallest possible decomposition that are m:n associations of individual activities. Limits to analysis at this level have not yet been defined. More broadly, the translation and abstraction from firm specific processes will necessarily entail a loss of information. As we move from basic schedule information to richer constraints such as resources, a basic research challenge is understanding how to best represent constraints for further processing. There are some practical guides as to what constraints are important (e.g., Ballard and Howell 1998), but it is less clear how to formally represent them or how to propagate them across firms. As most scheduling research has dealt with individual schedules, we enter new territory with a schedule mapping approach.

CONCLUSIONS

This paper presents an architecture that supports distributed process coordination among firms in the construction supply chain, in particular CM/GCs and subcontractors. We believe that the approach can be generalized to multiple tiers in the supply chain, where, for example, each subcontractor would act as a coordinating firm for its immediate suppliers and so on down the supply chain. The proposed modular architecture promises to be scalable both in implementation and in ability to represent a variety of process information for coordination, although ultimate power and generality of process representation and manipulation remains an open research issue. Beyond the specifics of the architecture, a novel contribution is the tree based mapping approach. Mappings, and the process connectors implementation of them, provides firms the ability to maintain their internal process representations instead of imposing a single representation on all project participants. Hence process connectors proffer the promise of speedier and robust adoption than what is possible with monolithic systems of similar capability.

ACKNOWLEDGEMENTS

The authors wish to thank the National Science Foundation for its support of this research under grants CMS-0542206 and CMS-0531797. We also thank Dr. Nashwan Dawood of the University of Teesside for his insights.

REFERENCES

Alonso, G., Casati, F., Kuno, H., and Machiraju, V. (2003). *Web Services: Concepts, Architectures, and Applications*, Springer, New York.

Antill, J. M., and Woodhead, R. W. (1990). *Critical Path Methods in Construction Practice*, Wiley, New York.

Ballard, G., and Howell, G. (1998). "Shielding production: essential step in production control." *ASCE Journal of Construction Engineering and Management*, 124(1), 11-17.

Cheng, J., Law, K. H., and Kumar, B. "Integrating Project Management Applications as Web Services." *Proceeding of the 2nd International Conference on Innovation in Architecture, Engineering and Construction*, Loughborough University, UK, 10 pages.

Choo, H. J., Tommelein, I. D., Ballard, G., and Zabelle, T. R. (1999). "WorkPlan: constraint-based database for work package scheduling." *ASCE Journal of Construction Engineering and Management*, 125(3), 151-160.

Halpin, D. W., and Riggs, L. S. (1992). *Planning and Analysis of Construction Operations*, Wiley, New York.

O'Brien, W. J., and Fischer, M. A. (2000). "Importance of capacity constraints to construction cost and schedule." *ASCE Journal of Construction Engineering and Management*, 125(6), 366-373.

O'Brien, W. J., Issa, R. R., Hammer, J., Schmalz, M., Geunes, J., and Bai, S. (2002). "SEEK: Accomplishing enterprise information integration across heterogeneous sources." *ITcon - Electronic Journal of Information Technology in Construction - Special Edition on Knowledge Management*, 7, 101-124.

Schlenoff, C., Gruninger, M., Tissot, F., Valois, J., Lubell, J., and Lee, J. (2000). "The Process Specification Language (PSL): Overview and Version 1.0 Specification." *NISTIR 6459*, NIST.

Siddiqui, M., O'Brien, W. J., and Wang, T. "A mapping based approach to schedule integration in heterogeneous environments." *Proceedings of the Joint International Conference on Computing in Building and Civil Engineering*, Montreal, Canada, 10 pp.

Smith, S. F., and Becker, M. A. (1997). "An Ontology for Constructing Scheduling Systems." *Working Notes of the 1997 AAAI Symposium on Ontological Engineering, AAAI Press*, 10 pages.

Sriprasert, E., and Dawood, N. (2003). "Multi-constraint information management and visualization for collaborative planning and control in construction." *ITcon, Special Issue on eWork and eBusiness*, 8, 341-366.

Turk, Z. (2001). "Phenomenological foundations of conceptual product modeling in architecture, engineering, and construction." *Artificial Intelligence in Engineering*, 15(2), 83-92.