Moving from Information Tunnels to Configurable, User-Model Driven Environments: A Vision for Future Project Information Technologies

William O'Brien*, Raja R. A. Issa** and Ian Flood***

*CSR Rinker Assistant Professor, M.E. Rinker, Sr. School of Building Construction Gainesville, University of Florida, FL 32611-5703; wjob@ufl.edu **UFRF Research Professor, M.E. Rinker, Sr. School of Building Construction, University of Florida, FL 32611-5703; raymond-issa@ufl.edu ***Holland Associate Professor, M.E. Rinker, Sr. School of Building Construction, University of Florida, FL 32611-5703; flood@ufl.edu

Abstract

Projects have long been seen as information rich environments with associated calls for information integration to support decision-making. Yet even with advanced applications based on shared data standards, project professionals still face considerable difficulty accessing needed information in a form useful for decision making. In the authors' viewpoint, a primary reason for these difficulties is that most data and tools are designed and operated in the context of specific applications. This creates the problem of information tunnels: rich data and application areas with knowledge that is not easily transferable to other application areas. Information tunnels per se would not be a difficulty if projects had identical problems; walls between tunnels could be torn down or professionals could develop heuristic integration mechanisms that functioned with a high degree of reliability, speed, and accuracy. But each project faces unique challenges, making integration across tunnels difficult for both machines and humans. Hence, we envision information technologies that provide project professionals much greater configuration control in how they access project information, regardless of its source or tunnel. We believe user models will drive such technologies, formal structures that help capture and structure both the data and metadata that professionals use in their work. We describe a user model driven information architecture and compare how that environment might work in contrast to existing approaches.

Introduction

The intention of this paper is to outline a novel approach – configurable environments – to make information more accessible and useful to design and construction practitioners. The goal of the approach is to customize data collected from a large number of sources to help professionals solve complex business problems. The configurable environments approach combines personalization with complex collection and processing of data. "Personalization" remains a current buzzword with various approaches including custom filters, data visualization, user modeling, intelligent user interfaces, etc. Collection and processing of data is a broad concept that includes interoperability, autonomous agents, mediation, etc. At least by inspiration and example, the configurable environments (CE) approach combines elements of all the above.

Novel aspects of the CE approach are the complexity of the problem addressed (e.g., real problems in design and construction) and the ability to support users who, while professionals, are not narrow technicians or computer scientists. For design and construction users, the authors envision a toolkit that (1) provides complex analysis and visualization tools to provide what-if type capabilities, (2) draws data from multiple sources, and (3) is configurable to the needs of the user (and hence is not a static toolkit). A CE differs from current approaches that provide users a set of (rigid) applications that make use of specific sets of information. These current approaches have been characterized as "information tunnels" (Tannenbaum 2002). Further, the authors argue that such tunnels persist in an environment with data standards designed for interoperability.

The CE approach draws from current construction information technology research in visualization, data integration, and application development. Broadly, the CE approach is a specific instance of broad visions supported by information technology research and development organizations such CIB W78 (w78.civil.auc.dk) and FIATECH (www.fiatech.org). In a review of W78 research, Amor et al. (2002) note that a major theme is the "modeling of processes and products integrated with visualisation and simulation of information life-cycles" with particular current emphasis on flexible approaches to information representation and management. Similarly, the FIATECH technology roadmap calls for development of integrated systems, information on demand, and scenario-based planning systems to explore alternatives (FIATECH 2003). The authors CE approach is designed to be flexible in response to user needs while supporting complex analysis.

Related Efforts in Problem-Solving Environments

The CE approach owes a significant intellectual debt to the stream of construction IT research. However, it has also been developed from broader research in information technology, including problem-solving environments (PSEs). Specific instances of PSEs are reviewed below to support and expand the description of the configurable environments (CE) approach presented in following sections.

PSEs have been defined as "an integrated system that provides all the facilities needed to solve a target class of problems conveniently and efficiently" (Rice and Boisvert 1996). Inspired by the success of graphical user interfaces, PSEs were developed in the 1990s from a desire to make the large number of scientific computing tools more accessible and more useful. Several PSEs have been developed to help engineers and scientists solve problems; MATLAB® (www.mathworks.com) is perhaps the most successful product. Other tools are more focused in application, designed to solve a narrow, domain specific problem. The reader is referred to projects documented at Purdue (www.cs.purdue.edu/research/cse/pses/research.html) and Virginia Tech (research.cs.vt.edu/pse).

Most PSE provide a useful but static toolkit to help solve a well-understood problem. Rice and Boisvert (1996) note that future PSEs will need to become more flexible to support a broader range of problems, and note an alternate definition of a PSE is the sum of a user interface, libraries, knowledge base, and integration. In this alternate definition, the knowledge base and integration aspects provide the user a more flexible toolkit than that provided by a static library with a visualization component. The range of environments classified as PSEs is large, although a review suggests that even flexible PSEs are designed for highly specific, generally technical applications. Concomitantly, most PSEs are designed for users such as engineers who want to bring a range of tools to a problem. In this sense, the integration and knowledge base components refer to a PSE's ability to seamlessly connect its various library components with visualization tools and aid users in the selection of the appropriate library tools and interpretation of analysis results.

It is useful to review current applications that fall within the definition of a PSE:

- The Environmental Decision Support System (EDSS, http://www.emc.mcnc.org/EDSS/EDSSPage.html#Mode20Management) is an example of a powerful and extensible PSE. With a focus on air quality issues, it provides a library of analysis and simulation components to provide what-if analysis and visualization. The EDSS also provides a superset of its components called the Strategy Development Tool to help technical users evaluate the impact of air pollution control strategies (see Figure 1). The EDSS further offers an API from which further tools can be integrated and adaptors to specific data sets can be built. The EDSS is an environment that is self-contained and useful as a toolkit for sophisticated scientists and engineers working on air pollution issues.

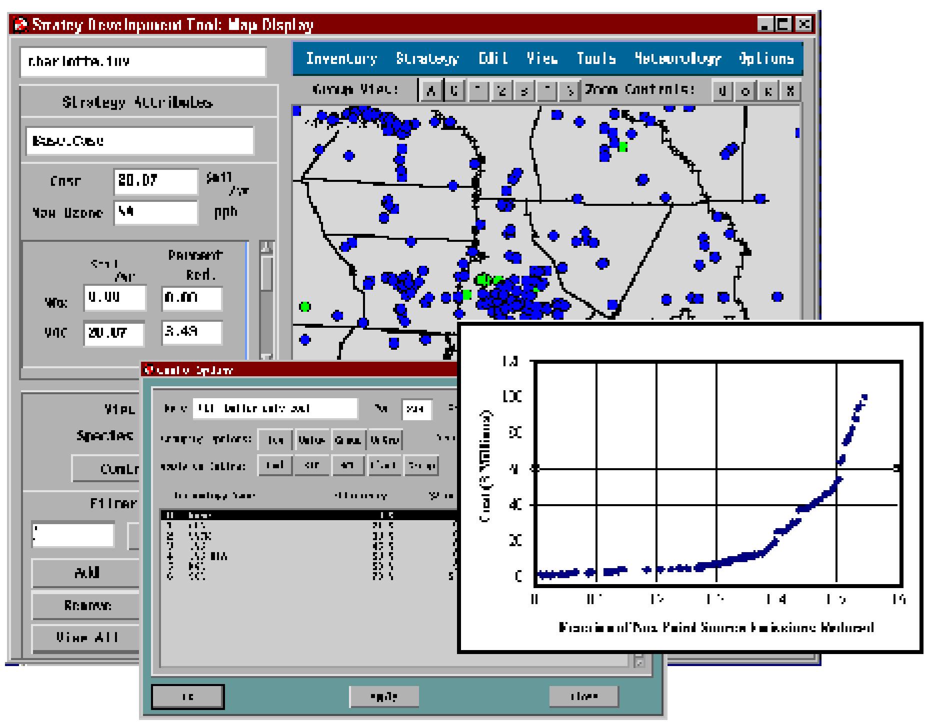

Figure 1. The EDSS Strategy Development Tool environment, combining analysis, data entry, and visualization.

- In sharp contrast to EDSS, the CoMotion™ environment built by MAYA VIZ (www.mayaviz.com) is a designed as a multi-user, collaborative environment to support shared visualizations and manipulation of rich data for decision support. As a broad environment, CoMotion has been customized to support a variety of applications including military, logistics, and healthcare. (A generalized CoMotion desktop is shown in Figure 2.) As a PSE, CoMotion combines business logic, simulation, and visualization tools in a self-contained environment. Furthermore, the CoMotion environment can access external data sources. However, once instantiated for a specific application, the environment is rigid in terms of accessing new data or supporting new analysis or visualization tools.

- Autodesk® Revit® (www.autodesk.com/revit) is an object-oriented CAD tool for design and construction. Revit can be considered a PSE as it enables individuals and teams to use analysis and visualization tools to evaluate design alternatives. In particular, Revit uses context-specific logic to maintain consistency among the parametric components of its product model. This enables users to make changes to, for example, a wall location and the model updates all the geometry of adjoining walls to reflect the change. Taken from parts libraries, the parametric model components also enable rapid generation of quantities for cost estimating. The Revit environment has some ability to import and export data from other CAD application and to spreadsheets. There is also an API to enable other developers and sophisticated users to link other applications and data sources with Revit. However, Revit is largely a static, self-contained environment.

Figure 2. A sample CoMotion™ desktop. Windows represent various analysis and visualization tools in the environment.

Limitations of Problem-Solving Environments

The PSE paradigm provides powerful tools to users. Commercial environments such as MATLAB®, CoMotion™, and Autodesk® Revit® demonstrate the sophistication PSEs have reached in terms of user friendliness and ability to meet users needs. Similarly, non-commercial PSEs such as EDSS demonstrate the depth of analysis tools available to users in an integrated environment. However, the PSEs reviewed above also reveal limitations. First, PSEs are not easily customized to include new tools or capabilities. The CoMotion shell is a tool within which specific PSE applications can be built, but such customization takes considerable effort by experts. Second, PSEs are largely self-contained environments and do not easily integrate with heterogeneous data sources. While there are APIs, binding a legacy source requires expertise with both the data format of the PSE and the source. Third, as environments designed to aid in the solution of specific classes of problems, their analysis and visualization components are not easily applied to related problems that do not fall neatly within the designer's intent.

These three limitations – limited customization, limited integration with heterogeneous sources, limited application to related problems – make application of PSEs to design and construction problems problematic. The truism that construction projects are unique has, perhaps, been misused by some practitioners to avoid adoption of software tools that aid generalized classes of design and construction problems. But the uniqueness of construction problems does pose problems for PSEs as general analysis and visualization toolkits may not be fully applicable (or fully support) project-specific problems.

Consider as an example the problem of managing space on construction sites. This is a complex problem and a long stream of research has been performed to classify types of space use and/or propose algorithms and tools to aid practitioners (e.g., Akinci et al. 2002; Riley 2000; Riley and Sanvido 1997; Thabet and Beliveau 1997; Tommelein et al. 1991; Tommelein and Zouein 1993). Much of this research is application oriented and, with some reworking, could provide the core of an analysis library and knowledge base for a site space management PSE. But an application of these tools would have to be carefully mapped to the needs of specific sites. For example, the work of Riley and Sanvido (1997) is specific to multi-story buildings. Applying their research to other facility types could lead to problematic results. More broadly, each of the existing research projects above makes several explicit and implicit assumptions about the problem being addressed. Without making these assumptions explicit in the PSE, users are likely to find analysis tools that make recommendations inappropriate for their specific site needs. Of course, such a critique is true for application of any formula or tool, but the facility provided by an integrated environment such as a PSE invites abuse.

Exacerbating the problems that stem from tools of limited applicability is the lack of easy customization. While software applications provide users some ability to filter results, customize menu bars, etc., it is generally impossible or impractical for users to make substantive changes in the capabilities provided to them. Hence applications like PSEs reinforce either flawed use of analysis tools or lack of use.

Customization by programmers is also difficult. Beyond the basic difficulties of complex code, a core difficulty stems from the "user models" adopted by programmers. User models are derived from the cognitive science community and are broadly described as the mental picture a user has of how a particular software application should work (Payne 2003). The user model concept has a long history and a broad range of interpretation (Payne 2003; Rich 1979); for the authors' purposes, however, the definition above suffices. Ideally, software should be designed with the user model in mind. Articulating user models and making them explicit to programmers is a well understood problem in the software development world (Cooper and Reimann 2003; Kulak and Guiney 2000; Winograd 1996). As it is difficult to articulate a user model, it is also difficult to articulate variations in a user model and hence direct programmers to effectively customize software for specific users and specific needs. The complexity of tasks and proliferation of professional roles in the design and construction industry appears to make customization all the more difficult.

The problems of operating with user models suggests that software development for the design and construction industry is an iterative affair with applications being slowly improved over time within their establish domain (e.g., scheduling, estimating, accounting, CAD, etc.). This reinforces the development of "information tunnels" (Tannenbaum 2002) with integrated sets of data and applications designed to serve narrow specialties. However, these specialized applications do not directly support many project analysis needs. For example, managing site space allocation requires knowledge of scheduling, estimating, and CAD/spatial data at a minimum. The need to draw from multiple data sources or information tunnels to answer relevant business questions on projects is a general problem and has been argued by many (e.g., a recent paper by Froese et al. (2000) diagrams the information integration problems faced by professionals nicely).

The presence of information tunnels has led to calls for data standards to support interoperability (e.g., the Industry Foundation Classes (IAI 1996)). However, data standards are a partial solution. Several authors have recently argued that the wealth of legacy applications and associated variation in level of detail (O'Brien et al. 2002) and range of applications (Amor and Faraj 2001; Turk 2001; Zamanian and Pittman 1999) means that development and adoption of relevant data standards to support all project participants is unlikely. Hence information tunnels are likely to persist even with significant adoption of data standards. Consequently, the adoption of tools like PSEs that require data from multiple sources will remain problematic.

Configurable Environments

The authors propose development of configurable environments (CE) as an evolution of the PSE paradigm. CEs provide the analysis and visualization capabilities of the PSEs while adding the ability to integrate information from multiple, heterogeneous sources (mitigating the problems of information tunnels). CEs also provide customization abilities not native to statically configured applications. The CE vision makes aggressive use of user models to direct configuration and customization.

Returning to the example above of managing space on sites, it is useful to consider how a CE would operate in contrast to existing tools. First, a specific instantiation of a CE will be customized to the needs of an existing class of individual via a user model. A site superintendent, for example, would utilize a CE built from a superintendent user model. The superintendent may further customize the CE by accessing preferences in the user model. The CE provides the superintendent an integrated view of the site layout, schedule, resource utilization, and procurement and delivery schedule. The superintendent may access analysis tools to investigate the ramification of different schedules on site space use. Current 4D CAD tools (with some setup difficulty) provide much of the analysis and visualization capabilities envisioned for the site space management CE. Consider though the important differences: Through the CE, the superintendent has access to up-to-date procurement information from a variety of suppliers. Also, the CE provides the information to the superintendent at a level of detail suitable for decision-making (e.g., the shipment will arrive on date x as opposed to a detailed bill-of-materials that may be more suited to estimating or production planning). Further, the CE provides all the information in a visualization environment customized to the superintendents needs.

The data integration and customization aspects of CEs are considerably sophisticated and difficult to implement for a given application or PSE. Further, once implemented, they remain difficult to customize for individual users who may need different levels of detail of information, have different information integration needs and privileges, etc. However, the advent of automated approaches to information integration, together with advances in user modeling supported software make it possible for us to concretely envision an architecture and implementation approach for CEs.

A high level CE architecture is shown in Figure 3. Reading from right to left, existing legacy sources for information (e.g., application data, databases) are accessed through a value-added wrapper. The wrapper translates the data format of the legacy source into the internal format CE. The wrapper may have a value added component of simple mediation tasks (Wiederhold 1998) and may or may not be native to the CE. Several translators are being developed for applications. Toolkits such as SEEK, in development by the authors (O'Brien et al. 2002), provide extensive capabilities to semi-automatically configure wrappers and provide an advanced query interface to the legacy source. Alternate approaches to organizing data for flexible access by humans (King 2003; Lucas and Senn 2002) may also provide descriptors and wrappers that the CE can access.

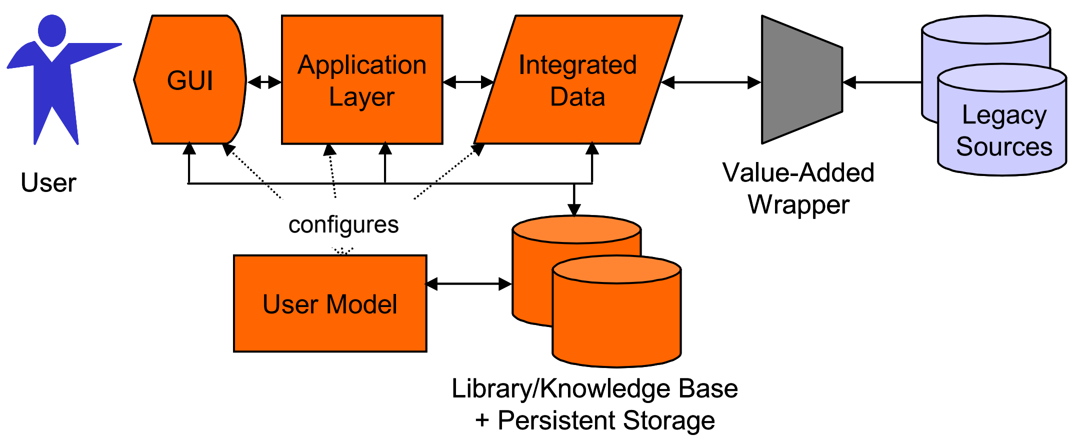

Figure 3. Configurable Environments high-level architecture.

The integrated data component of the CE provides an organized view of the data specific to the needs of the user (e.g., superintendent, project manager, estimator, etc.) Depending on availability of data from the legacy source, the integrated data component may or may not cache data drawn from legacy sources. Rather, the integrated data component provides organization on top of legacy data. (The separation between the wrapper and data component is subtle and there is some overlap between their roles. Each is called out separately as the wrapper component may not be native to the CE while the integrated data component is a necessary part of the CE.) Shen et al. (2003) discuss a specific implementation of an integrated data component tied directly to a graphical user interface (GUI) to support queries.

Beyond Shen et al.'s (2003) model, the application layer provides analysis and query capabilities on top of any mediation tasks contained in the wrapper and data components. The application layer is supported by the library of tools and knowledge base as in a PSE. However, the application layer also supports broad information queries by users to take advantage of the CE's links to external data sources. The user accesses the CE through the GUI. The GUI component supports a variety of visualization tools and is customized to the needs of the user. Broadly, the GUI will likely look much like existing PSEs.

Underlying the GUI, application layer, and integrated data components is the user model component. The user model contains specific information about the class of user (e.g., superintendent) and may be further customized to the specific needs of an individual. The salient distinction of CEs compared to PSEs and other applications is that the user model directs configuration of the components of the CE. Thus a site management CE for a superintendent may provide a different level of detail and different analysis capabilities than does a site management CE for a trade foreman. For example, a superintendent may wish to view broad areas needed to look for space conflicts among trades whereas the foreman may wish to view details of specific spaces to layout production plans.

Configuration by user models is technically challenging but the real difficulty lies in specification of the model. Specification has dual challenges: Understanding the needs of humans and translation of these needs to formal abstractions. There is an increasing body of professional knowledge about development of user models to guide programmers (e.g., Cooper and Reimann 2003), although the problem cannot said to be reduced to a simple knowledge elicitation formula. Domain understanding is necessary for useful specification of user models. Similarly, much of the user modeling performed to date is used to help humans create specific implementations rather than used to autonomously direct substantive configuration of applications. (And much autonomously directed configuration to-date makes relatively trivial changes to the core application.) Research to define requirements for user model abstractions is a necessary perquisite to further development, although papers in the field of user modeling show increasing sophistication (e.g., see the proceedings edited by Bauer et al. (2001)).

Conclusions

This paper has presented configurable environments as an approach to supporting users handle complex information processing and management tasks on projects. The CE approach provides data integration capabilities as an alternative to the information tunnels currently facing users. CEs also are highly customizable via the application of user models to direct configuration. In a simple sense, the CE approach is a natural evolution of the PSE paradigm to support the needs of projects. Similarly, CEs are a specific approach developed from broad visions for computing in design and construction.

However, CEs also represent a paradigm shift away from specific application development to broader research about environments to support human decision-making. Elements of such a shift are present in calls for flexible systems, ontology development, and mapping of transactions and processes. But at the same time there remains considerable research in creating applications to optimize and automate solutions to specific problems. There is a lack of corresponding research to broadly understand problem context, human roles, and associated formalisms. Making the CE approach a reality will take considerable research about users, and hence a basic implication of the approach is that there is a need to reemphasize the role and value of documenting, understanding, and formalizing human information processing and decision-making needs in the design and construction domain.

References

Akinci, B., Fischer, M., Levitt, R., and Carlson, B. (2002). "Formalization and automation of time-space conflict analysis." *ASCE Journal of Computing in Civil Engineering*, 6(2), 124-135.

Amor, R., Betts, M., Coetzee, G., and Sexton, M. (2002). "Information technology for construction: recent work and future directions." *ITcon*, 7, 245-258.

Amor, R., and Faraj, I. (2001). "Misconceptions about integrated project databases." *ITcon*, 6, 57-66.

Bauer, M., Gmytrasiewicz, P. J., and Vassileva, J. eds. (2001). *Lecture Notes in Computer Science 2109: User Modeling 2001: 8th International Conference (UM 2001)*, Springer-Verlag, Heidelberg.

Cooper, A., and Reimann, R. (2003). *About Face 2.0: The Essentials of Interaction Design*, Wiley, Indianapolis, IN.

FIATECH. (2003). "Strategic Overview: Capital Projects Technology Roadmap Initiative (version 1)." FIATECH, March 2003, 70 pages.

Froese, T., Yu, K., Liston, K., and Fischer, M. (2000). "System architectures for AEC interoperability." *Construction Information Technology 2000, Proceedings of CIT 2000 -- the CIB-W78, IABSE, EG-SEA-AI International Conference on Construction Information Technology*, Reykjavik, Iceland, 362-373.

IAI. (1996). "End user guide to Industry Foundation Classes, enabling interoperability in the AEC/FM industry." International Alliance for Interoperability (IAI).

King, J. (2003). "Future watch: taming data complexity." *ComputerWorld*, June 30, 2003.

Kulak, D., and Guiney, E. (2000). *Use Cases: Requirements in Context*, ACM Press, New York.

Lucas, P., and Senn, J. (2002). "Toward the universal database: U-forms and the VIA repository." *MTR-02001*, MAYA, March 2002, 10 pages.

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.

Payne, S. J. (2003). "User's mental models: the very ideas." *HCI Models, Theories, and Frameworks: Towards a Multidisciplinary Science*, J. M. Carroll, ed., Morgan Kaufmann, San Francisco, 135-156.

Rice, J. R., and Boisvert, R. F. (1996). "From scientific libraries to problem-solving environments." *IEEE Computational Science & Engineering*, 3, 44-53.

Rich, E. (1979). "User modeling via stereotypes." *Cognitive Science*, 3, 329-354.

Riley, D. (2000). "The role of four-dimensional (4D) models in trade sequencing and production planning." *Proceedings of Construction Congress VI: Building Together for a Better Tomorrow in an Increasingly Complex World*, Orlando, Florida, February 20-22, 1029-1034.

Riley, D. R., and Sanvido, V. E. (1997). "Space planning method for multistory building construction." *ASCE Journal of Construction Engineering and Management*, 123(2), 171-180.

Shen, Z., Issa, R., O'Brien, W., and Flood, I. (2003). "A proposed design component and schedule data integration model for construction." *Proceedings of the 10th European Group for Intelligent Computing in Engineering Workshop*, Delft, The Netherlands, July 3-4 2003, 66-75.

Tannenbaum, A. (2002). *Metadata Solutions: Using Metamodels, Repositories, XML, and Enterprise Portals to Generate Information on Demand*, Addison-Wesley, Boston.

Thabet, W. Y., and Beliveau, Y. J. (1997). "SCaRC: Space-constrained resource-constrained scheduling system." *ASCE Journal of Computing in Civil Engineering*, 11(1), 48-59.

Tommelein, I. D., Levitt, R. E., Hayes-Roth, B., and Confrey, T. (1991). "SightPlan experiments. Alternate strategies for site layout design." *Journal of Computing in Civil Engineering*, 5(1), 42-63.

Tommelein, I. D., and Zouein, P. P. (1993). "Interactive dynamic layout planning." *ASCE Journal of Construction Engineering and Management*, 119(2), 266-287.

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

Wiederhold, G. (1998). "Weaving data into information." *Database Programming and Design*, 11(9).

Winograd, T. ed. (1996). *Bringing Design To Software*, Addison-Wesley.

Zamanian, M. K., and Pittman, J. H. (1999). "A software industry perspective on AEC information models for distributed collaboration." *Automation in Construction*, 8, 237-248.