SUPPORTING PROJECT MEETINGS WITH CONCURRENT INTEROPERABILITY IN A CONSTRUCTION INFORMATION WORKSPACE

SUBMITTED: August 2004
REVISED: April 2005
PUBLISHED: July 2005 at http://www.itcon.org/2005/12/
EDITOR: Z. Turk

Marcus Schreyer, Dr.-Ing.
Visiting Scholar, Center for Integrated Facility Engineering (CIFE), Stanford University, USA
email: schreyer@infra.dreso.com

Timo Hartmann, Dipl.-Ing.
Software Engineer, Sofistik AG Nürnberg Burgschmietstrasse, Germany
email: timo.hartmann@sofistik.de

Martin Fischer, Associate Professor of Civil and Env. Eng. and (by Courtesy) Computer Science
Stanford University, Stanford, USA
email: fischer@stanford.edu

SUMMARY: *Based on prior research on information workspaces at Stanford the work presented in this paper describes an approach to enhance the interoperability between standard AEC applications. Beyond the ability to visualize multi-disciplinary relationships within a project model, our approach allows the automated propagation of changes to the model. This is accomplished by integrating the applications in a middleware controlled messaging infrastructure which is based on an extensible and flexible data model. Compared to today's standardized product models the proposed model enables users to relate data of discipline-specific applications much more flexibly and easily. These criteria are particularly valuable for rapid computational generation and analysis of virtual project models since the content for a model can differ from meeting to meeting depending on its participants and scope. Especially during meetings, the proposed software environment could therefore allow project participants to evaluate several project alternatives during the same iRoom meeting, enabling them to predict the effects of each variation on e.g. costs, schedule and the necessary resources. We presume that such a decision process based on the simulation and evaluation of different project alternatives as virtual models will enable meeting participants to communicate and analyze critical relationships among project information faster. Furthermore, we expect that the efficiency and comprehensiveness of the decision making process in meetings will increase. We believe that such a solution would decrease the risk inherent to complex decisions and motivate project teams to spend more time to find high quality solutions to their problems in less time.*

KEYWORDS: *information workspace, decision support software, virtual project models, virtual building, virtual construction*

1. INTRODUCTION

Decision making in project meetings requires the application of the interdisciplinary knowledge of a multi-disciplinary project team. To obtain a solution for a problem, e.g. to make up an unexpected delay or to realize change orders issued by the owner, the meeting participants contribute their knowledge to create decision alternatives, evaluate them with respect to the project goals and predict their consequences. Today, AEC professionals use sophisticated software tools to produce their discipline-specific project information in electronic form. Although software is used to generate information, the current practice in project meetings is to provide and present information in paper-based formats. Liston et al (2001) observed that communicating project information through paper-based representations limits a project team's ability to collaborate and to solve problems and make decisions. The reasons for this are:

- paper-based information is static and not interactive and therefore doesn't allow to predict the impact of changes ad hoc. Consequently, decisions are delayed to later meetings

- discipline-specific modeling, analysis and visualization tools are often too complex to be understood by the other participants of meetings
- critical temporal, spatial or cost relationships between the discipline-specific views of a project are typically not communicated in today's graphical representations
- views are often inappropriate for group use if they are not flexible towards representing information in different levels of details

A main cause for this situation is that the user interfaces for the various discipline-specific applications are often complex; they are designed for professionals from the specific discipline and incomprehensible for others. The applications allow their users to model a specific view of the building project and furthermore enable them to predict the performance of the building (e.g. structural behavior, costs, energy consumption), the building process (schedule, resources) and the organization (supply chain). However, a software that integrates or at least provides access to the data in the different discipline-specific applications in order to allow more comprehensive and interactive views on a project model is missing. Particular challenges are the interactive and dynamic use of such a project model in meetings. Existing approaches to create project models (e.g., Industry Foundation Classes, IFC) do not facilitate the establishment of the many relationships between interdisciplinary data needed to support interactive and dynamic use of project models by multi-disciplinary teams.

Researchers at the Center for Integrated Facility Engineering have proposed an interactive, multi-user and multi-application hardware and software environment called "Construction Interactive Workspace" or iRoom to enable a project team to integrate the different AEC applications used in meetings (Fischer et al. 2002). Interactive workspaces are generally used for cross-application communication of the distributed project information. They let participants of meetings use the software applications they know to present and modify their data and add interoperability between the connected applications through a central data model and a common communication layer.

The work presented in this paper builds on the prior iRoom research at CIFE and in the Department of Computer Science at Stanford University (Fox et al. 2000). It extends the current iRoom architecture and messaging infrastructure with a data model and corresponding middleware (which we call midserver) that enable users to model relationships between the discipline-specific data much more flexibly and easily than prior solutions.

1.1 Current approaches towards interoperability in the AEC industry

The striving for interoperability between discipline-specific AEC applications has a long history (IAI 1999). The simplest way to exchange information between programs is to transfer data files from one application to another. Since it is often necessary that the receiving application must be able to recognize the semantics of the data it receives, different efforts to establish specifications for software independent data models have evolved in the past. It may be desirable that such a specification is independent of software vendors and disciplines. In the last decade, much research has focused on developing methods to integrate project information and standardize how AEC data and their relationships are modeled and exchanged. Today's standardized product models for buildings, like the Industry Foundation Classes (IFC) (IAI 2000), address the interoperability problem by defining a complex object-oriented structure for the data of buildings. With the IFC, it is possible to accumulate building information in a file that can be shared easily with other applications that understand IFC objects (Kam and Fischer, 2002). For example, information about quantities can be exchanged from 3D CAD software to cost estimating software to accelerate and improve the estimating process. Applications like Graphisoft's Archicad and Timberline's Precision Estimating support such a solution on the basis of the interim IFC 2.0 standard.

However, such a file-based interoperability (Fig. 1) is sequential and doesn't reflect the iterative and dynamic workflow of the design process. For example, when an architect gives her design drawing file to a structural engineer she cannot continue to work on the part of the model she gave to the engineer while the engineer is doing his structural analysis. In this case, truly concurrent work is not possible. Because of the same reason file-based interoperability does not support the use of multi-disciplinary information in meetings.

The next level towards interoperability is a model-server architecture: A database server is placed between the project model and the AEC applications (SECOM and VTT 2002). This solution supports the management of the model data via locking and unlocking functions for parts of the data that are in use. The advantage of the model server approach over file-based interoperability is that it is far easier to create and maintain one common project model that supports the work of several disciplines. It also supports concurrent work, but only on different parts of the project at the same time. However, in project meetings, engineers need to collaborate concurrently on the same parts of a project employing their discipline-specific applications to model and analyze particular aspects of the project. Functionality is needed to synchronize the data among the applications when the project model has changed.

Furthermore, today's product models often require a high data overhead that can make the data exchange time-consuming. Product models become unwieldy when model servers are used to perform iterative changes to the project data and to provide rapid analyses of the modified model in project meetings. The somewhat rigid structure of the IFC data model often requires the transfer of more data then necessary to maintain the integrity of the data that is transferred, although only small parts of the model did actually change and would need to be updated. These restrictions currently result in the sequential exchange of relatively large sets of data between IFC-compliant applications (Fig. 1). Additionally, extending the relations beyond the already existing object relations defined in the IFCs is complex and requires a huge effort, which can hardly be done during a meeting. To allow interoperability in a multi-disciplinary software environment where many applications interact with another, a less restricted way to define relations between data objects or data objects and the applications that can modify them is necessary.

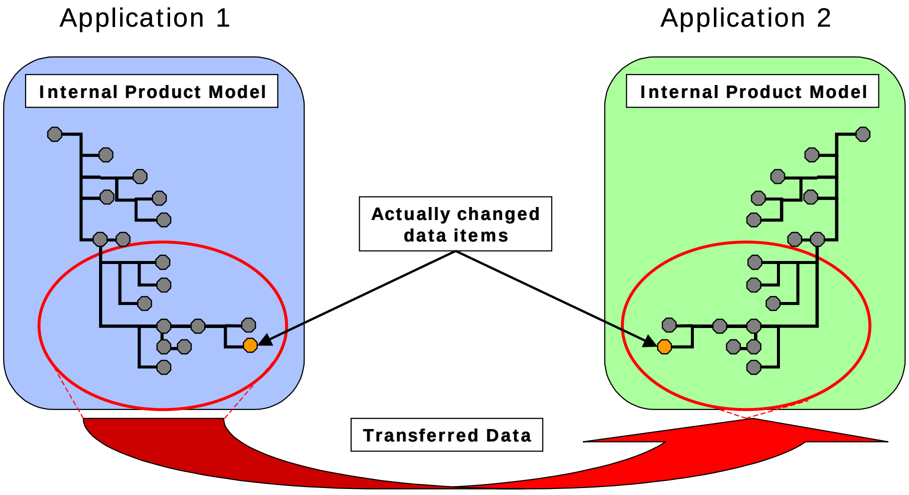

FIG. 1: Restrictive product models often result in a sequential data exchange caused by a large data overhead.

1.2 Requirements for Concurrent Interoperability in Project Meetings

To allow interdisciplinary interaction and ad hoc and rapid computational analyses, we and others (e.g., BLIS 2002) argue that concurrent interoperability should be one of the next goals in AEC software development. Concurrent interoperability requires that information fragments that are distributed in different applications are connected with each other through a flexible data model. This data model does not only need to contain data about the building itself. It also must contain metadata that describes which applications control which data objects if objects with the same meaning are used within more than one application. Furthermore, the data model must be flexible, because the content of the project model differs from meeting to meeting since it depends on the participants and purpose of a meeting. In addition, the software environment must offer the flexibility to infer the relations between the objects of the project model regardless of the object naming scheme used by applications contributing to the project model and regardless of the existence of a particular object in the project model.

To keep the federated data consistent, a general communication layer is necessary which enables applications and the model server to communicate changes to data objects among the different applications. Current approaches for interoperability, e.g. IFC Model Servers, do not offer such a two way communication or metadata inference. To develop such a server, the connected applications have to offer APIs to enable the server to address the data objects internal to the applications. Depending on the available API, this can be achieved by directly accessing the internal data objects using standards like SOAP or, as in our case, by programming software extensions for standard applications which act as clients in the communication layer of the workspace.

To accelerate and simplify the information exchange between participating applications the amount of data that has to be transferred after each change to the project model should be minimized. One approach is to use event-driven messages for the communication between the applications and the middleware server instead of transferring parts of the project model.

Considering the requirements for the project data model to support multi-disciplinary meetings, we found that the data model has to be much less rigid in its structure than the IFC. This would enable professionals to build a specific data model for the information needed during a meeting. In project meetings, it is often not necessary to use the whole information database for a project. A data model and software environment that also work when only parts of a building's data are modeled results in a smaller and easier to build database that may make it easier for expert professionals to work with interoperable data during meetings. A practical solution for concurrent interoperability in project meetings therefore requires an easy to build, extensible and flexible project data model and corresponding software.

1.3 Practical Motivation

The motivation of our research is to contribute to the development of an interactive and interdisciplinary software environment that enables project meeting participants to communicate and analyze critical relationships among project information on the basis of a computer model. Furthermore, the decision making process in meetings should become more effective and comprehensive when alternatives like e.g. changed schedules or designs can be simulated and evaluated as virtual projects. Compared to paper-based decision processes in today's meetings, interactive project models could reduce decision delay dramatically and ensure that everyone in the meeting understands the context and impact of decisions made.

At CIFE, current investigations try to quantify the impact of such a computational meeting environment on the efficiency and effectiveness of the time spent in AEC project meetings. According to observations made in traditional paper-based meetings and on projects supported by Interactive Workspaces (IWs) Liston et al. (2001) predict that updated views and visualizations in IWs could be designed specifically for comparing scenarios, enabling "what if" discussions far beyond what is currently possible (Fig. 2). In the paper, the authors divide the tasks discussed in project meetings in four groups:

- Descriptive: Questions about who, what, where, when and how
- Explanative: Questions to explain the rationale of project documentation, typically beginning with "Why…"
- Evaluative: Comparing project information to the project goals, typically "Does this meet the XYZ requirement ?"
- Predictive: Questions to predict the impact of changes to the current project, starting with "What if…" or "What happens to…"

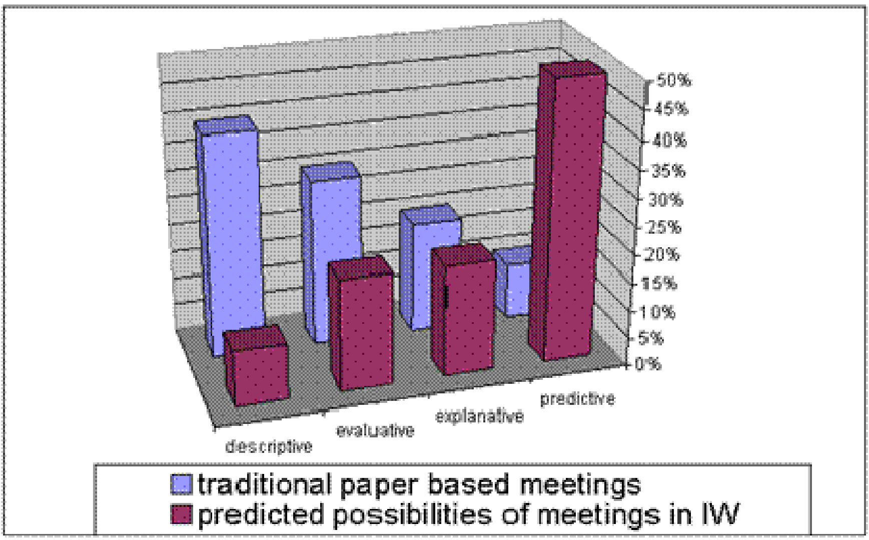

FIG. 2: Time spend in meetings (Liston et al 2000).

Initial experiments in the construction information workspace at CIFE showed that IWs together with visualization techniques can be used to establish focus between participants from multiple disciplines and organizations. However, support to create and evaluate design/schedule alternatives as virtual models immediately in meetings was still missing.

The goal of our research is therefore to enhance interoperability in this interactive workspace to enable the creation and discussion of alternative product, process and organizational models during meetings. This would enable us to investigate to what extent IWs can help to answer predictive questions, decrease the decision risk and allow project teams to spend more time to find high quality solutions to their problems in less time.

2. INTEROPERABILITY IN INTERACTIVE WORKSPACES

The Interactive Workspaces Project (Winograd et al 1999) is exploring new ways for people to work together in technology-rich spaces with computing and interaction devices on many different scales (Fox et al. 2000). In collaboration with CIFE, the prototype of an integrated, technology-rich work environment for construction called the iRoom (CIFE 2002) was developed over the last few years. Such computational environments to support collaborative work are being developed for various industries and are supposed to become economically and operationally practical for widespread use within a few years (Liston et al. 2001).

One of most significant characteristics of the iRoom are the large-format displays that facilitate shared views of applications and information. Since the three screens of the CIFE iRoom configuration provide many more pixels and much more screen area than even large conventional monitors, large amounts of information can be shown in close proximity. The information workspace also supports direct interactions with the information that is shown on various connected information devices. See (Johanson et al. 2001b) for a detailed description of the technology infrastructure for the iRoom.

The iRoom design is based on the following key concepts:

- Use multiple applications and multiple displays to visualize different views of a single model simultaneously
- Use multiple applications and multiple displays to allow the user to compare alternative states or versions of a model simultaneously
- Separate the control of a visualization from the applications that hold the project model, enabling visualization of data from multiple applications

iRoom technology includes two kinds of technology: “view” and “glue”. The computers and large screen displays support different simultaneous views of a project. The “glue” is an internet-enabled software infrastructure that allows the multiple applications to share data and control. The inter-application glue is enabled by the CIFE iRoom database and various software applications that extract data from and send data to the other project applications. Custom applications provide the specific glue that links AEC project modeling and analysis applications and enable a user to create views from different applications that are coordinated, i.e., showing the same time or physical object in all views on all screens.

To give the reader an understanding of the current possibilities using integrated workspaces for AEC decision making, the following chapter describes the established hardware and software components of the CIFE iRoom.

2.1 Overview of the CIFE iRoom Environment

To meet the specific needs of the construction industry, the CIFE iRoom uses standard AEC project modeling and analysis applications, such as Autodesk’s Architectural Desktop (ADT) (Hartmann 2003), MS Project, MS Excel, Commonpoint’s 4D CAD Modeler, etc. to display and manipulate the project data. These applications run in the iRoom just as they do on a desktop computer. The large interactive displays just make it easier for project teams to view and discuss project information simultaneously.

The physical configuration for the CIFE iRoom consists of three SMART boards, large touch screens with their projectors arranged from the ceiling (Fig. 3). A wireless mouse and a keyboard are usually placed on the central meeting table. Three ordinary Pentium III Windows Desktop PCs or Notebooks are connected to the projectors and to the SMART boards and usually run the applications. A fourth PC is designated as the server machine, which also runs the PointRight server software. PointRight enables users to use one mouse and keyboard across the three screens to control all three computers connected to the SMART boards (Johanson et al 2001a).

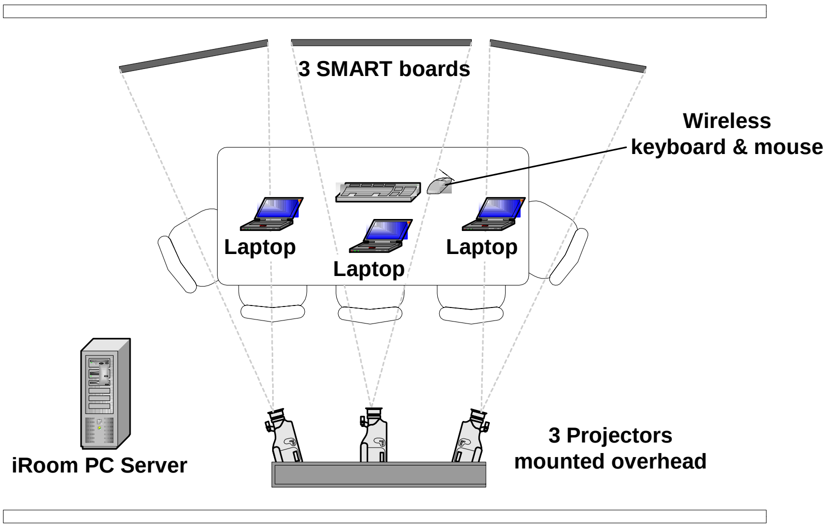

FIG. 3: A sketch of the actual iRoom layout at Stanford's CIFE

Besides the capabilities of the iRoom to display information on three SMART boards, the most important aspect of this technology is the IP-based message passing system (Fox et al. 2000). It enables all the applications of the iRoom to send messages to or receive messages from other applications running in the iRoom environment. Fig. 4 shows how messages are exchanged within the iRoom through a central service called the Event Heap (Stanford CS).

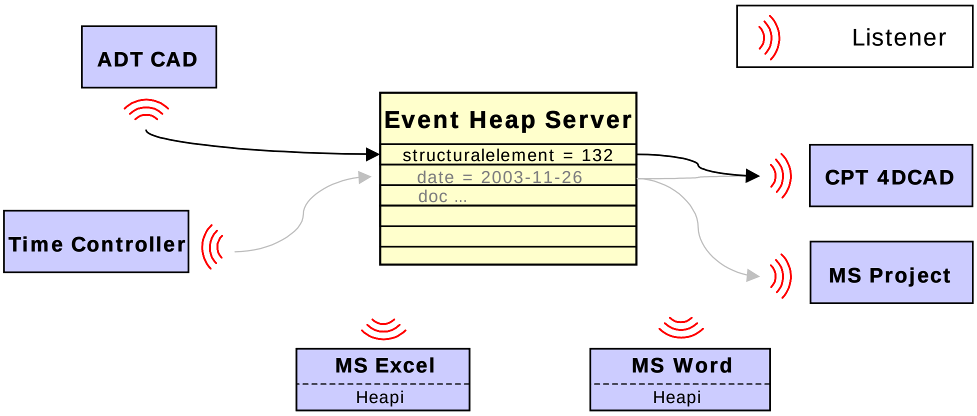

FIG. 4: Schema of the IP-based Message Passing System between the iRoom applications

To enable commercial applications to access these messages they had to be extended with a software component called event heap listener. A listener observes the event heap constantly enabling its corresponding application to react to messages that concern it. Since the communication is IP-based, the C++ Code of the listeners can easily be adapted and compiled to any respective operating system platform.

For example, in Fig. 4 Autodesk's ADT CAD sends a message to the event heap to highlight information about a geometrical *structural element* object. This message is picked up by two applications, whose listeners were developed to react to messages sent by ADT. In the example, CPT's 4D CAD picks up the message to highlight the respective building component, and Microsoft Project picks up the message to highlight corresponding construction activities as defined in the database. Table 1 lists the main categories of software services in the iRoom.

TABLE 1: Software layers and selected services of the CIFE iRoom (Stone et al 2002)

Visualization and interaction layer

AEC Applications: AutoDesk's ADT, Commonpoint's 4D Modeler, Microsoft Project, and Excel have been customized to interact in the iRoom.

Viewers: The CIFE iRoom includes three custom-made data viewers for concisely displaying table, tree and document views of information in the database. The AEC applications also function as viewers.

Controllers: Controllers send events, e.g., the Time Controller is a custom controller used to link all applications to a common date.

Messaging layer

Event Heap: The general coordination mechanism for an iRoom is called the Event Heap, or eHeap. Messages called "Events" can be posted and retrieved by programs running in the iRoom. The Event Heap itself runs on the iRoom server, and Event Heap clients run on all machines that send or receive events.

Multibrowsing: These events provide a general mechanism for passing Windows commands between machines.

PointRight: A program that enables a user to control all iRoom computers with one common mouse and one keyboard. PointRight runs as a server on the iRoom server and as a client on all of the other machines.

Database and management layer

XML Database: Commercial database supplied by eXcelon, which includes a web-based set of tools for managing it.

Import-Export Utility: Program for translating between the database and native file formats of the AEC modeling applications.

Room Controller: A Java applet that allows the user to control the display on the SMART boards.

At the beginning of the CIFE iRoom project in late 1999, it was planned to build the initial version of a construction information workspace on top of the semantic information exchange standards for the exchange of product data, like STEP and IFC. They seemed to offer the best starting point for information sharing (Fischer and Froese 1996). These information exchange standards can be powerful once a group of organizations has aligned their IT environments to them because a range of analyses based on the same information can be done very quickly (Kam and Fischer 2002). However, experience in applying them (see introduction and Froese et al. 1999) has shown that this is, so far, not a scalable solution because very tight coordination and working relationships are needed to make sure that all the semantics are aligned between software vendors and users in various organizations. Furthermore, until today there exist only a small number of IFC implementations since the implementation of this comprehensive model requires a significant effort for a software company. As outlined in the introduction and above, today semantic information standards with their large data overhead make it difficult to develop an Information Workspace which addresses the need for flexibility towards the data model, since during project meetings most of the times only a specific subset of the comprehensive project model is required. A solution that lies between the comprehensive but restrictive information standards and a semantic-poor graphical information sharing approach is needed.

2.2 Meetings in the Interactive Workspace

In a CIFE iRoom meeting, the heterogeneous project views like schedules, the 3D and 4D models, MS Excel cost tables, etc. are represented as electronic views on the screens. Instead of working with paper copies of the agenda, the participants focus on the displayed electronic agenda. If a participant selects an item in one application, the relevant project information is displayed and highlighted in all corresponding applications on the other screens (Liston et al. 2000). The highlighting functionality enables participants to easily recognize critical relationships between related items in the views of other applications like 4D view, MS Project Gantt chart view, Excel cost tables, or Excel resource tables. Another CIFE iRoom functionality is the synchronization of views according to a common date or view point. I.e., if a user sets a date in one application (e.g., in a Gantt chart view) other date-sensitive applications (e.g., the 4D software) adjust their views to reflect the new date. The visualization technology "highlighting" and the temporal and spatial synchronization methods support the identification of problems and help the team to quickly understand the constraints and rationale of the displayed information. The project team can spend more time performing predictive tasks and thus can more efficiently compare and evaluate design or schedule alternatives.

In (Fischer et al. 2000), first research results were publicized on the impact of interactive workspaces on the time spent in meetings. They show that with paper based media professionals spend 40% of their time in meetings on describing project information, while only 10% is spent on predictive tasks. However, these predictive tasks are important for generating high quality solutions.

3. EXTENSIBLE IROOM MIDDLEWARE

To explore alternative solutions during an iRoom meeting, data changes in the different applications have to affect the project model and consequently lead to an update of the views on the model in the other applications as well. To quickly view the impacts of modifications of the project model, a sequential data exchange of the applications to the project model, each followed by an update of the whole database, would be very inefficient and for larger models (in our experience a model with over 500 3D components and related schedule, cost, and other information is a large model) very slow. Thus, such a solution to support concurrent interoperability between applications in interactive workspaces wouldn't be a practical alternative in project meetings. An alternative to a sequential data exchange is a message-based solution, realized with a three tier software architecture including a middle server that manages the relations between the objects of the model as well as the relations between the different data models of the applications. The implementation of this architecture in the CIFE iRoom required changes in the database layer and an extension of the messaging system. These changes are described in the following section.

3.1 Project Data Repository

As explained at the beginning, an important technical requirement for concurrent interoperability is a fast and targeted exchange of messages between the applications. With the intention to keep the amount of data that has to be exchanged as small as possible, we realized during our research that only a small subset of the overall model actually has to be shared between the applications. Depending on the number of different objects (CAD objects, cost items, activities etc.) that are necessary for the actual iRoom model the size of the database could consequently be reduced significantly. On the other hand, the shared database approach to connect the federated models in the different applications requires storing additional information about the relationships between objects and between objects and the corresponding applications.

If e.g. one of the applications changes a data item within the iRoom data repository, the middle tier software (midserver) has to inform all affected other participating applications about the change. This can only be achieved by storing additional object-to-application relationship information, which enables the midserver to comprehend the redundancies between data items of the iRoom database and the corresponding applications, which also store these items within their data repositories. For example in the current iRoom environment at CIFE two applications are able to model geometrical building data: 4D CAD and ADT. If a geometrical CAD object is changed in one application, the midserver uses the object-to-application relationships to communicate this change to all other affected applications instead of forcing them to reload the whole model. These relationships are established by mapping the data objects provided by the applications to their respective iRoom object, which each has a unique ID within the iRoom data repository.

In order to reflect explicit semantic interdependencies among data objects, additional object-to-object relationships became a part of the database schema. Most changes to data items of the project model will cause changes to other related data items. On the basis of object-to-object relationships, a midserver should be able to automatically execute and propagate these changes throughout the different applications connected to the iRoom. An example for this behavior would be that if the resource parameter in the resource model has been changed by one application, the duration of the assigned activity will not only be adjusted internally by e.g. MS Project, but also communicated to the schedule model and then automatically updated in all related applications.

A very important requirement for decision support in meetings is to be able to focus the view on specific levels of detail. For example, the briefing of owners generally requires much less detailed information, where else a meeting between site engineers to coordinate their work probably requires very detailed information about the most recent schedules, resources and spatial situation on site. A common way to achieve views with different levels of detail on the same project database is to structure the data elements hierarchically. The object oriented hierarchy in our database allows us e.g. to easily accumulate objects from lower hierarchical levels and to display them accordingly. To implement our prototype software we used the eXtensible Markup Language (XML) to create the data model for our test case example. We tried to group the different objects like building components, cost items, resources or construction activities in sub domains. These sub domains typically represent the different disciplines which may be involved in an interactive workspace meeting, like the design domain storing the building component objects, the schedule model storing the construction activity objects, the cost model storing the cost item objects or resource model with the available resource objects) [see Fig. 5].

In order to keep our data model flexible towards different scopes and levels of detail we decided for an approach using metadata and data mapping strategies instead of using a semantically predefined model like IFC. This decision has been mainly made due to two shortcomings of the IFC model. First of all an IFC conform implementation of the data repository would have been extremely time consuming because of the large IFC data scheme. Most of the data stored within an IFC model is not required within the data scheme intended to support a meeting. So we could reduce our implementation effort to a great extend by using our own custom tailored project model data scheme. The second reason for not using IFC objects for our database was the fact that there are at the moment almost no AEC applications which offer an IFC interface. Thus we would have had to implement interfaces to an IFC based iRoom data repository. This again would have been more time consuming due to the large IFC data base scheme than creating a new scheme which is tailored for the representation of the data needed within an interactive workspace meeting.

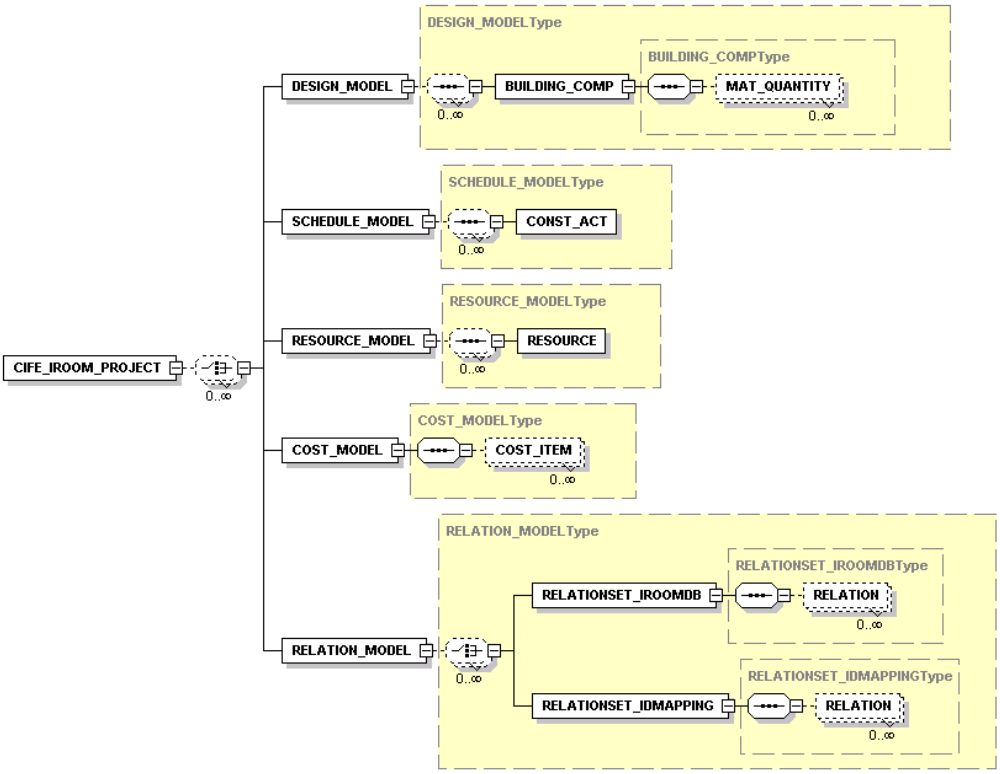

FIG. 5: Schema of the project database model for the enhanced CIFE iRoom, including separately modelled object relationships.

The requirement for flexibility also affects the different relationships that have to be established within the data model. Therefore we created a relation model with explicit relationship objects within our database scheme (Fig. 5). For the implementation of our prototype we defined an instance to describe the mutual object-to-object relations (RELATIONSSET_IROOMDB) and another for object-to-application relations (RELATIONSET_ID-MAPPING). Further instances can be defined accordingly. Besides the opportunity to reduce redundancies in the data model the chosen structure minimizes the effort to parse the database when the midserver tracks the relations.

3.2 Midserver

The communication between the different integrated iRoom applications is performed via messages. Applications send their messages to a central event heap stack which is located on the server. There, the midserver analyzes the content of the messages and uses them to infer the relations in the database. The result of this inference is the generation of new messages, this time issued by the midserver to the applications, to adjust their data accordingly. In order to be able to parse the database and infer the relation model additional intelligence has to be provided. For example if ADT sends a message to change a data item within the iRoom data repository, data of other applications like MSProject are affected as well; messages have to be send to these affected applications in order to perform an automatic data update of the applications' internal data repositories. We could have implemented this intelligence separated in all of the applications. Nevertheless this would lead to functional redundancies within some of the applications. For example the functionalities that would have had to be implemented for ADT and CPT 4D would largely be similar, concerning both applications focus mainly on geometrical building component objects. Thus we decided to implement this additional intelligence centrally in an independent middle tier layer (Fig. 6).

In the example of Fig. 6 the CAD application Architectural Desktop sends the command to highlight all data items within the iRoom applications related to a particular building component. The midserver picks up this message from the event heap. First, it searches for the related ID of the object within the iRoom database. This ID is used to parse the XML data repository's object-to-object relation objects for all items which are related to the specific building component. These related other object types can then be mapped again to the unique ID within other iRoom applications by interpreting the object-to-application items of the data repository. With this information the midserver is able to create new messages and send them to the event heap, where they can be picked up by the corresponding applications. The listener component of an application now only has to interpret and execute the formalized statement in the message. In our example CPT 4D, MS Project and Excel pick up messages from the event heap sent by the midserver to highlight building components, activities and resources related to the selected building component objects in ADT.

Since in the midserver architecture all communication is directed to a central component, it was extremly important that we provided a stable and reliable implementation of our software. On the other hand the implementation effort for the commercial AEC applications became relatively small. With the integration of the midserver we established a clear and open software architecture into the iRoom environment which allows developers to integrate new applications with only small adjustments to the already provided components. Another advantage is the significant reduction of the required requests to the data repository since parallel and redundant database requests are avoided. Further, in order to highlight related objects within different applications large parts of the data repository do not have to be parsed at all. An access of the objects stored in the domain models is e.g. only necessary if one application commits changes to one of the shared objects within the iRoom data repository.

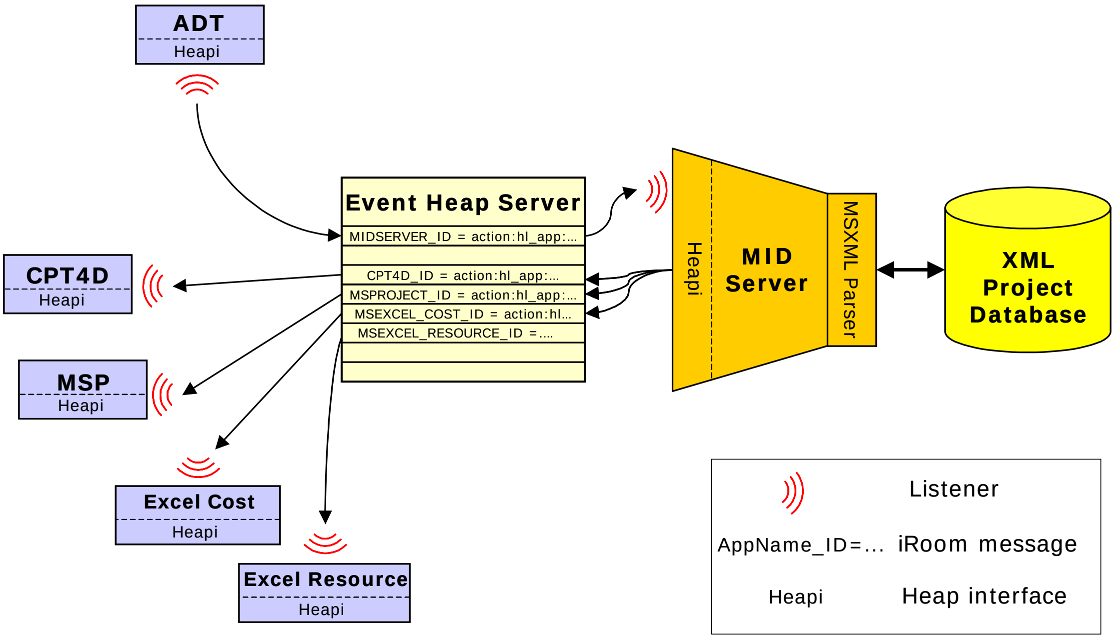
ADT
Heapi
CPT4D
Heapi
MSP
Heapi
Excel Cost
Heapi
Excel Resource
Heapi
Event Heap Server
MIDSERVER_ID = action:hl_app:...
CPT4D_ID = action:hl_app:...
MSPROJECT_ID = action:hl_app:...
MSEXCEL_COST_ID = action:hl...
MSEXCEL_RESOURCE_ID =....
Heapi
MID Server
MSXML Parser
XML Project Database
Listener
AppName_ID=... iRoom message
Heapi Heap interface

FIG. 6: Three-tier architecture of the latest CIFE iRoom.

4. TEST SCENARIO

To validate our assumptions about the database model and test the established communication system, we modeled a test scenario based on an ongoing construction project. We selected a large shopping mall in Emeryville, CA. During this project, the construction company DPR Inc. already made extensive use of computer models to develop strategies that allowed them to meet the strict schedule and cost requirements that emerged after hazardous materials were discovered. From the many problems that resulted from this discovery, we constructed a scenario of a project meeting where descriptive, explanative, evaluative and predictive tasks need to be solved with support of the capabilities of the new CIFE iRoom prototype. To build the virtual project model we used the 3D CAD model of the building constructed in Autodesk's ADT software and the general contractor's MS Project schedule and CPT 4D model. Furthermore, we modeled the labor and equipment resources and costs for our test scenario as spreadsheets in MS Excel. We limited the data on the foundations of one of the buildings, where the hazardous materials were found. This kept the amount of project information we needed to handle reasonable and allowed us to comprehend the reaction of the database model on changes in the applications. The participants of our fictitious meeting are the project manager of the general contractor, the subcontractors for excavation, piling, steelwork and utilities and the investor (ref. as client) of the shopping mall.

4.1 Describing and Explaining the Problem Context

We assumed that in the first place the project manager wants to inform the meeting participants about the problem and its consequences on the project. Typical descriptive tasks are discussed first like "Where on the site have the materials been discovered?" or "Which tasks in the schedule are directly affected?" and "Who is involved in these tasks?" To answer these questions, the project manager uses the multiple screens of the iRoom environment to show the affected area on the 3D CAD model and, by clicking on the respective piling area, to highlight the corresponding activities in the schedule, which is displayed on another screen.

He assumes that the removal of the hazardous materials will result in a four week delay of the whole schedule if no changes to the construction process are made. Furthermore, the project manager states that the discovery of the polluted soil will increase the foundation costs due to a change proposal of the excavation subcontractor by about 14%.

At this point, the client wants to have an explanation for these additional costs. To explain his changed cost estimate, the excavation subcontractor uses the wireless mouse of the iRoom and opens his initial and actual cost estimates and schedules on the iRoom screens. He then selects the foundation zones affected by the delay and highlights the related activities and cost items in the schedules and cost estimates to compare them in front of the other meeting participants. The multi-screen view of the iRoom environment allows him to explain easily that the reason for the higher costs is the increased duration for the excavation work.

For descriptive and explanative tasks the advantages using the iRoom environment in meetings are that the iRoom helps the participants to achieve focus on the currently discussed tasks. Since there is no need for them to work through stacks of paper to get an idea of a problem, they are enabled much earlier to contribute their knowledge to solve the task. Furthermore, the visualization of relationships between data objects modeled in the project database help to explain the interdependencies of the different spatial, cost and organizational views on the project. But what actually happens on the messaging layer of the iRoom when these tasks are discussed?

A message of the action-type "highlight" is sent to the midserver when the project manager clicks on the respective foundation area in ADT. The message contains information about which data object of the ADT model has been selected and the name of the sending application. The midserver then checks the project database for data objects that are related to the foundation objects defined in the message. It assembles the new "highlight" messages and addresses them to the different applications. Now, the applications' listeners receive the messages, and the heap interfaces initiate the highlighting of the respective data objects in the application windows. Fig. 7 illustrates these actions. A message from ADT causes one or several other messages, e.g., from the midserver to a MS Project schedule.

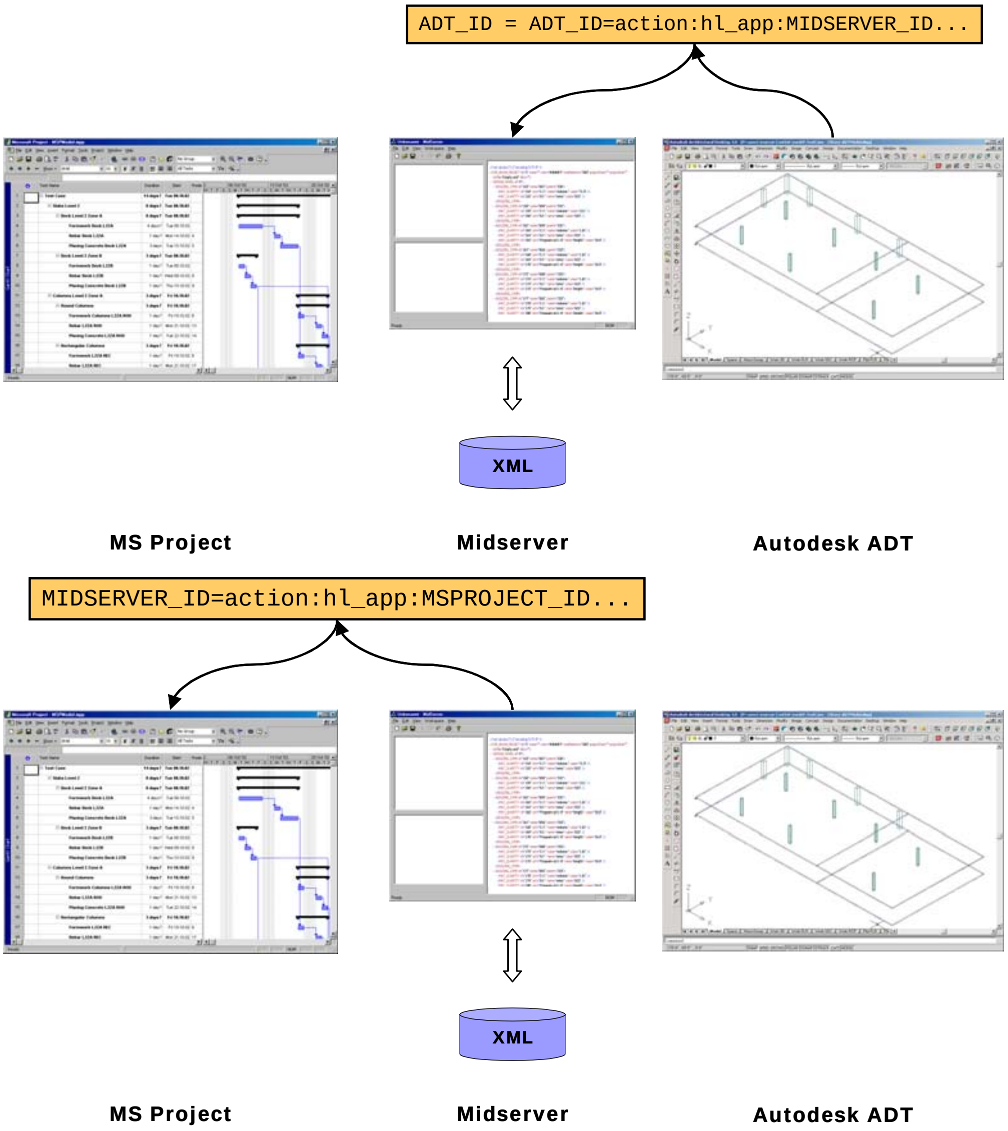
ADT_ID = ADT_ID=action:hl_app:MIDSERVER_ID...
XML
MS Project
Midserver
Autodesk ADT
MIDSERVER_ID=action:hl_app:MSPROJECT_ID...
XML
MS Project
Midserver
Autodesk ADT

FIG. 7: Messages between the iRoom applications are directed by a midserver with an underlying XML project database.

4.2 Predicting and Evaluating different alternative solutions

After the participants in the meeting are aware of the problem and its complications, the project manager asks the piling subcontractor whether his crews could work overtime to make up some of the delay. The subcontractor agrees to increase the working hours for his piling crews in the resource sheet from 8 to 10 hours per day, if he will be paid the additional costs for the overtime work. These changes to the working hours per day on the resource sheet in MS Excel are then submitted to the iRoom respectively to the midserver as explained in 3.2. The corresponding “change” message from MS Excel causes a response from the midserver to MS Project to update the related activity durations for piling. Additionally, the midserver sends “change” messages to the cost sheet in another MS Excel window to consider the hours for overtime work in the labor costs. After all the views on the project data are updated by midserver messages, the project team gets the information from MS Project that the predicted end of the project will now be two weeks earlier than before. The client learns from the updated cost estimate as a decision basis, that the cost increase would be roughly $ 4500.

Further the rebar subcontractor proposes to increase the crew strength for the reinforcement of the foundations and slab-on-grade to the project team. After reviewing the 4D model to check for spatial conflicts with other crews working on the foundation, another 10 days can be gained.

As can be seen from this scenario, the model-based decision process allows the project participants to evaluate several project alternatives during the same iRoom meeting, enabling them to predict the effects on costs, schedule and the necessary resources.

5. CONCLUSIONS

With the proposed database structure and the midserver architecture, it is possible to implement concurrent interoperability in interactive workspaces. Our approach enhances the functionality for interoperability beyond the visualization of relationships within the project model, enabling the propagation of changes to the model from one application to all other affected applications.

By modeling relationships explicitly and introducing the capability to define different types of relationships we could extend the initial CIFE construction information workspace. Furthermore, we developed a corresponding midserver that infers the project-specific relationships from the types of relationships defined by the users and from the project model. We also extended the iRoom application interfaces for the new messaging format.

These changes resulted in an extensible and flexible project data model for the iRoom, which adaptive structure can be adjusted to the specific scope of a project meeting. Compared to the IFC approach, we stored in the iRoom database only data objects that are shared between applications. Altogether, this leads to a lean database structure that simplifies the database model creation and, from the technical perspective, increases the performance of operations in IWs because the parsing process is faster. Another advantage of our approach is that unlike the IFC model server the CIFE prototype already integrates popular AEC applications.

So far, we have concentrated mainly on establishing a clear and stable software framework and on defining a central schema for the data shared by the applications that are connected to the iRoom. The implemented solutions are intended to be prototypes to validate our assumptions and test the established communication system. However, especially the creation of the data model still requires a significant effort without respective software tools. Therefore, iRoom applications like editors for the database to support the creation and maintain the integrity of the project model which haven't been subject of our research, yet, should be addressed in subsequent projects. Nevertheless the new framework can already be used to set up simple iRoom scenarios with data from actual construction projects. Built upon earlier research at CIFE about analysis of project model information by visualization techniques, our approach demonstrates a way to allow the ad hoc creation of virtual project alternatives during project meetings.

6. OUTLOOK

In our view the next milestone of the iRoom environment will be to develop the experimental prototype into software that can be used to support real world meetings. To accomplish this, it will be important to integrate additional popular AEC applications into the environment. Additionally an editor to support the creation and maintenance of larger project data models will have to be provided for iRoom users.

We further think of services that support the configuration, versioning, tracking, and organization of numerous project alternatives to avoid “information overflow” in meetings and to help the participants to focus on the solution of the important issues. In the end, the application of the interactive workspace on real world projects will enable us to evaluate the effects on project meetings predicted in (Liston et al 2000).

Regarding semantic applications for interoperable functionality, the iRoom architecture will require research to formalize and implement interdisciplinary knowledge into the environment. Typically, such semantic interdependencies are not modeled in today's discipline-specific applications. The feature based approach (Staub-French 2002) to formalize a cost estimator's rationale is an example for such an advanced data model with more sophisticated semantic relationships.

7. ACKNOWLEDGEMENTS

The authors would like to acknowledge the Center for Integrated Facility Engineering (CIFE) in Stanford, the German Research Society (DFG) and the Institute for Computer Science in Civil Engineering at the Technical University of Munich for their support of our work. We thank Calvin Kam, John Kunz, Kathleen Liston and Vibha Singhal for sharing their knowledge about the concept and technology of the CIFE iRoom with us and

Dean Reed and Peter Allen from DPR Inc. for providing us access to information, issues and the site of the Bay Street project in Emeryville, CA.

8. REFERENCES

BLIS (2002) “Building Lifecycle Interoperable Software.” http://www.blis-project.org

CIFE Interactive Workspaces Group. (2002). *iRoom Documentation.* Stanford University, Center for Integrated Facility Engineering. http://cife.stanford.edu/IROOM/documentation/document-frames.htm

Fischer, Martin; Stone Maureen; Liston, Kathleen; Kunz, John; Singhal, Vibha (2002). "Multi-stakeholder collaboration: The CIFE iRoom." Invited Paper and Keynote Address, Proceedings CIB W78 Conference 2002: Distributing Knowledge in Building, Aarhus School of Architecture and Centre for Integrated Design, Aarhus, Denmark, pp. 6-13.

Fischer M. (2001). Scalable Enterprise Systems - Designing Construction Information Workspaces: Making Information Work for Project Teams. *2001 NSF Design, Service and Manufacturing Grantees and Research Conference*, January 7-10, 2001, Tampa, Florida

Fischer M., Froese T. (1996). Examples and Characteristics of Share Project Models, *Journal of Computing in Civil Engineering*, ASCE, 10(3), 174-182.

Fischer M., Garcia C., Kam C., Kunz J., Liston K., Schreyer M., Singhal, V. (2002) Virtual Design and Construction Case Study, *CIFE Summer Program 2002*, CIFE, Stanford University.

Fox A., Johanson B., Hanrahan P. and Winograd T. (2000). Integrating Information Appliances into an Interactive Workspace., *IEE Computer Graphics and Applications*, 20(3), 54-65.

Froese T., Fischer M., Grobler F., Ritzenthaler J., Yu K., Sutherland S., Staub S., Akinci B., Akbas R., Koo B., Barron A. and Kunz J. (1999). Industry Foundation Classes for Project Management - A Trial Implementation, *ITCON (Electronic Journal of Information Technology in Construction)*, Vol. 4, 17-36.

Hartmann T. (2003). “*Integration of a Three Dimensional CAD Environment into an Interactive Workspace.”* Master Thesis. Lehrstuhl für Bauinformatik, TU München / CIFE, Stanford University, 2003

IAI. (1998). “International Alliance for Interoperability.” http://www.iai-international.org

IAI. (2000). *IfcXML*. http://www.iai-international.org/iai_international/ Technical_Documents/IFCXML.htm

Johanson B., Hutchins G., Winograd T. (2001a). *PointRight: Pointer/Keyboard Redirection for Interactive Workspaces.* Stanford University. http://graphics.stanford.edu/~stone/steelcase/pointright_ubicomp.PDF

Johanson, B., Ponnekanti, S., Kiciman, E., Sengupta, C., Fox, A. (2001b). System Support for Interactive Workspaces, *CS-TR-2001-0,* Stanford University, Stanford, CA, http://swig.stanford.edu/semipub/iwork-sosp.pdf

Kam, Calvin, and Fischer, Martin (2002). Product Model & 4D CAD. *CIFE TR 143*, Stanford University

Liston K., Fischer M., Kunz J. (2000). Designing and Evaluating Visualization Techniques for Construction Planning, *Eighth International Conference on Computing in Civil and Building Engineering (ICCCBE-VIII),* R. Fruchter, F. Pena-Mora and K. Roddis (Eds.), August 14-17, 2000, Stanford University, 1293-1300. http://www.stanford.edu/group/4D/workspace/ papers/ASCE2000_Visualization.pdf

Liston K., Fischer M., Winograd, T. (2001). Focused sharing of information for multi-disciplinary decision making by project teams., *ITcon (Electronic Journal of Information Technology in Construction)*, Vol. 6, 69-81, http://www.itcon.org/2001/6/paper.pdf

Schreyer M., Hartmann T., Fischer M., Kunz J. (2002). CIFE iRoom XT Design and Use. *CIFE TR 144*, Stanford University

SECOM and VTT. (2002). *IFC Model Server Development Project.* http://cic.vtt.fi/projects/ifcsvr/

Stanford University - Computer Science. *“Interactive Workspaces – Event Heap Documentation“*. http://graphics.stanford.edu/projects/iwork/software/htmlpages/ documentation.html

Stone M., Liston K., Fischer M. (2002). *The CIFE iRoom*. http://graphics.stanford.edu/~stone/pubs/CIFE/iRoom to Go/CIFE-iRoom.pdf

W3C. *Document Object Model (DOM).* http://www.w3.org/DOM/

Winograd T., Hanrahan P. (1999). *Stanford Interactive Workspaces Project.* http://graphics.stanford.edu/projects/iwork/