USING MULTI-LEVEL VIRTUAL ENVIRONMENTS AS A MEDIUM FOR CONDUCTING DESIGN REVIEW THROUGH A SHARED IFC DATASET

MATTHEW E. SWARTS, HUGO A. SHEWARD
Georgia Institute of Technology, USA

ABSTRACT: For a long time the Architecture-Engineering-Construction (AEC) community has had difficulty in communicating the content of their work, not only the various specialties involved, but also to their clients. Studies (Doorst and Cross 2001; Bakhtin 1994) suggest the importance of multi-role collaborative environments in supporting design processes. We are developing a Multi Level Design Review Tool for the AEC industry which allows multiple actors to congregate and interact as agents around a central Building Model. It merges real-time virtual 3D visualization technologies with Industry Foundation Classes (IFC) to support both high levels of semantic content and seamless interoperability.

KEYWORDS: Design review, virtual environment, interoperability

RÉSUMÉ : Depuis longtemps la communauté de l'Architecture, de l'Ingénierie et de la Construction (AIC) a eu du mal à transmettre la teneur de leurs ouvrages à leur clients et aussi de rendre compte des diverses spécialités au sein de leur domaine. Des études (Doorst and Cross 2001) (Bakhtin 1994) suggèrent que les environnements multi-rôle et accomplis en collaboration sont importants. Nous avons mis au point un outil pour l'AIC qui laisse les acteurs multiples interagir autour d'un modèle central de bâtiment. Il fusionne les technologies de la visualisation que sont le temps-réel, le virtuel, et le 3D avec les fichiers de l'IFC pour supporter les niveaux hauts de contenu sémantique ainsi que l'interopérabilité intégrée.

MOTS-CLÉS : Révision de la conception, environnement virtuel, interopérabilité
1. METHODOLOGY

By using Industry Foundation Classes (IFC) data, we can provide not only a platform independent application but also an extensive set of attributes for each of the multiple objects brought in to the design review process. On the other hand we have the Unreal Engine and their real-time 3D graphics engine which provides many advantages when compared with similar systems (Busby et al. 2005).

The purpose of implementing a user interface through the Unreal game engine platform is to incorporate into the process the following capabilities:

- multi-player networkability;
- spatially responsive communication among actors;
- database accessible datasets;
- report generation to a common database;
- simultaneous activation of purpose driven subsets of data;
- client-side interface customization.

We propose that the proper integration of both can help to represent and manage the data imbedded in IFC files in more efficient ways, and can vastly improve the design review process and therefore the design outcome. Our system supports seamless and enhanced interaction among the different actors involved in the design review process and their correspondent semantic content.

2. THEORETICAL BACKGROUND

Our main interest is to provide an enhanced environment for design reviews, and an environment capable of not only providing extensive information to the reviewers, but also capable of supporting a great deal of interaction between the actor and the digital environment.

The data utilized by the application is derived from all the data elements from IFC, which have been developed specifically to contain information commonly used by the Architecture-Engineering-Construction (AEC) industry. Unlike many other data exchange formats, IFC contain far more than just geometry. The data structure of IFC contains data definitions capable of addressing most of the material properties, objects used, and physical activities preformed by a wide variety of actors in the AEC industry.

The virtual environment selected to support the interaction between building and actors is the Unreal Engine, which was selected for a wide variety of features capable of enhancing the interactive process. The Unreal Engine, known for its powerful graphics, physics, and developed workflows, is one of the most widely licensed in the game industry (Epic Games 2007). As with most game engines, the Unreal Engine has stable networking capabilities. The
typical network provides the capability of handling up to 32 logins. It even implements a voice over IP (VOIP) that is spatially distributed. The VOIP propagates sound and voice signals to distribute positions for multiple users within a space. This feature enhances the immersive effect of the virtual environment where by position of the actors is not only presented visually but also aurally through sound allocation within the space.

“In the early architectural phase it is important that the experience of the VR model is as close as possible to the real building or environment when built. A faithful rendering of the appearance of the future building is desired and the feeling of being present may be important.” (Westerdahl et al. 2006)

3. OPERATIONAL APPROACH

One of the aspects in contemporary design operations which have seen major changes lately is the design review process. These review meetings are performed several times during both the design and construction phases. The participants of these vary depending on the complexity and time availability of the actors. Now more than ever before design teams are using tools such as phone conference calls or online on-screen sharing applications, but seldom are these communication tools both flexible enough so as to address different types of users and interactive enough as to be considered effective.

The importance of social interaction and collaboration regarding the outcome of the design process has been explored and validated by several researchers (Brereton et al. 1997, Analyzing), even though not much has been explored regarding the use of digital media and the amount of information resources that digital media can bring to the review process. The impact of rich information environments and their influence in the design process is well documented (Brereton et al. 1997, Collaboration). Plume and Mitchell (2005) have shown the effects of extensive interoperability in AEC multidisciplinary environments.

The intention of the Multi Level Design Review Tool is that of creating digital environments capable of generating domain specific environments while at the same time supporting remote design team interaction. “As the design team negotiates the problem space, each designer makes bids to have issues they think important discussed by the team. Having called focus on an issue, the other designers might engage in the focus, adding ideas towards a partial solution” (Brereton et al. 1997, Analyzing).

We have developed the concept of user specific environments. These are digital environments which are built by using information explicitly related to a specific domain of expertise, but have a common base used by the entire design review team. This common base as well as the user specific environments is extracted from IFC exchange files.
Due to the extent and complexity of the information contained within IFC exchange files, we decided to pre-process a set of packets of geometric data in order to improve the efficiency of the data access process. Through these IFC views, we need only process a subset of the information and not the entire data set. The specific data entities contained in each of the proposed views have been selected so they can be combined as needed to provide the data required to construct each of the user environments. These user environments, even though they are constructed using the same geometrical information, can display different sets of information regarding specific fields of expertise within the AEC industry. Even though the IFC views will be pre-processed, specific combinations of these are only performed at the time the user(s) logs in to the Multi Level Design Review Tool. We have developed three approaches regarding the information display capabilities of the Multi Level Design Review Tool: the global, the local, and the shared display of model information.

4. SYSTEM ARCHITECTURE

The review session is initiated when the client establishes the basic conditions for the meeting in the server based application. These define the uploading of the IFC file containing the building data and the initial setup for the meeting structure into the session manager. This will determine the number and therefore the required semantic content of each of the environments.

After reading the IFC input, the parser object extracts all the information required to construct the base environment and to provide all the building information required by the different domains included in the meeting. The system stores an Unreal map of the parsed data in the database. This will develop in the creation of the base environment containing the geometry of the project and the attributes and object content databases.

Once the pre-processing and generation of the mapping of IFC into Unreal objects has been completed, the session manager will distribute the different instances of the environment to the Unreal game environment. In order to maintain the computational efficiency of the module, we have developed an attribute inspector which handles the queries coming from each of the environments by associating the Global Unique Identifier (GUID) of the attributes related to it, which are then saved in the database of the system.

When a user logs into the system, their information will be confirmed and their domain specifications set. Since each environment is pre-processed for each domain, an instance of the prepared environment will load on the client machine. When other users log into the system, they each receive a viewable instance of the shared environment that is specific to their domain. Objects within each of the environments will be shared across all users with data replication and reliability maintained by the server.
5. INTERACTION PROTOCOLS AND THE CONCEPT OF VIEWS

Design meetings occurring in physical environments are arranged through a complex organization, where by using both verbal and non-verbal language, the participants generate a structure for communication and information sharing. By using the Unreal Engine as a base, we are capable of not only providing stable verbal communication, but we can also recreate the perception of sound localization in space.

In the same manner in which a meeting conducted in the physical world, the position of leader in the Multi Level Design Review Tool is acquired by the meeting organizer. The data element selected (Figure 2) by the leader is visible and highlighted in all the screens of the active users in the meeting, classifying
certain information as general data and other as domain specific data. We define three different classifications for the data displayed in the module: Global, Local, and Shared.

**FIGURE 2. SCREENSHOT SHOWING EXEMPLAR SELECTION RESULTS.**

IFC definitions from a model are broken into subsets of views which can be attached to a particular individual’s virtual representation. On logging into the system, one may select from a preset of views which define certain roles in the AEC process, or may select a set of preferred views, all of which is tied to the login and stored. A base set of geometry entities is viewed by default.

Using the dataset, a list of explicitly defined spaces, each related to a position in the virtual environment, is used in selecting a dynamic entry point into the map space.

Once in the environment, there are three main activities, namely: inspect, query, and annotate. Inspecting the environment and model is simply a matter of moving and looking around. Entities may be selected within the environment directly through the screen. A selected entity’s attributes are queried from a database running independently and connected via networking scripts. The attributes are displayed on the screen. Two sets of attributes are listed: general and specific. General attributes are ones which all users view. Specific attributes are domain specific for the view or set of views which the user currently has chosen. Finally, annotations may be added to a selected entity. Annotations may be added on the general or the specific but is always recorded to the external database for later review.
There are two modes of review. The first mode is a roam mode. The second is a leader or guided mode. In roam mode individuals are able to freely move around the space to make queries, add annotations, or enter verbal discussion with others nearby. Entering and leaving discussions is performed simply by spatially entering or leaving the area of discussion. Users may elect to view the names, professions, or type of views of other participants. A color laser pointer system, along with basic avatar pointing animations is used to denote what speakers are referencing.

Leader mode is akin to NetMeeting in a 3D virtual environment, where one individual presents, while the rest participate as listeners. A leader mode presentation is initiated by a user, who may define the presentation as open or closed, by selecting from a list of participants to have access to the presentation. Once the leader mode session is started, others may join. Control of the session may be requested, but must be accepted by the current leader. Once entering the session, each participant in the leader mode observes through the perspective of the leader. The leader is virtualized so that any set of views may be chosen independently of the actual leader’s set of views.

The leader has the ability to move through space, query, and annotate. Others in the leader mode session view these actions on their display. They do not have control of the leader’s virtual laser pointer, which is solely owned by the current virtualized leader position. However, they keep control over their own local virtual laser pointer, and may independently query and annotate, unknown to others.

Roam mode and leader mode may be performed concurrently through the use of channels. The world channel is used for the spatially distributed VOIP of roam mode. Separate channels are used for specific leader based presentations. Leaders or users in roam mode may elect to view each other, but no interaction, other than visual, is allowed. Typically, though, those in a leader mode session should elect not to be distracted by others in roam mode within the environment or by other separate leader mode sessions.

6. IFC DATA HANDLING

The Industry Foundation Classes (IFC) also known as ISO Step part 21 is an ongoing effort by the International alliance for Interoperability. Its intention is to provide a neutral data exchange format for the AEC industry.

The IFC schema provides definitions for numerous entities related to the AEC industry. Being able to carry information that goes well beyond the geometrical description of objects, it also carries information about ownership, materiality, construction phases, and R-values, to mention a few. This provides a data model that can be shared by the entire industry rather than having separate models for sharing in each domain set (Figure 3).
The complexity and the extent of the data contained in IFC files, which is of extreme importance at the time of supporting interoperability, might be considered excessive when implementing other types of operations.

To extract the building elements that have been identified as critical for the implementation of the application, we identify the elements by pairing their names within the schema definition. After locating the position within the IFC file by following the syntax structure defined for IFC files, we extract the data elements associated to its definition.

The syntax defined for the structure of IFC files clearly identifies the relationships between objects and attributes. This depends on the location and values for each location of the string line that defines the IFC entity.

The structure of the entity definition organizes the information by a system of file instantiations. In the case of building entities, the first value within the entity definition is reserved for a 20 character GUID identifier identified as ifcGloballyUniqueId.

Following the GUID definition and separated by commas, we find all the attributes defined for the entity. These may either contain explicit data for the entry or be in the form of specific reference calls to lines of code within the IFC file itself (Figure 4).

**FIGURE 3.** (A) SHOWS THE CURRENT DATA SHARING IN AEC, (B) IFC SHARED PRODUCT MODEL APPROACH.
In figure 4 the “call to line” are calls that make specific reference to locations within the IFC file. In the example these are defined as follows:

- call to line 42;
  
  #42=IFCOWNERHISTORY(#41,#2,$,.NOCHANGE.,$,$,$,0);

- call to line 55;
  
  #55=IFCLOCALPLACEMENT(#46,#54);

- call to line 70;
  
  #70=IFCPRODUCTDEFINITIONSHAPE($,$,(#58,#69)).

The purpose of the application is to provide domain specific information for each of the users in the form of one digital environment capable of generating multiple user views. Due to the fact that there is no way to predetermine which fields of expertise will need to be represented during a design review, there is little justification to load the entire set of building model data into the application.

Data extracted from the IFC file is imported to a relational data base in which, for efficiency purposes we separate the data into the Unreal base environment data base, the attribute data base and the object content data base. Within each of these the following class level semantic information will be contained.

- geometric relationships/references;
- general purpose attributes;
- building construction attributes;
- building systems attributes;
- engineering attributes.

Access to the class level information is managed through the attribute inspector which during runtime can query the database in order to display attributes and object content if required by the user by associating the Unreal object with the attributes by using the objects’ GUID.

7. CREATION OF IFC OBJECT INSTANCES IN UNREAL

The Unreal Engine uses a scripting language called Unrealscript. The partial implementation of the IFC schema in Unrealscript was used to store data retrieved from the database. The International Alliance for Interoperability (IAI) IFC Schema website was used as reference in creating IFC classes.

For the mapping process we began from the instance level objects such as IfcWallStandardCase and worked up through parent classes to IfcRoot. In a second pass, each subsequently defined variable type was implemented along with a path up through its parent classes. IfcRoot is not the base class for all IFC classes so IfcBase was introduced. IfcBase’s parent is the Object class for Unrealscript. IfcBase was used to isolate the IFC implementation from other
packages within the Unreal Engine. EXPRESS language TYPEs and ENUMs were defined in IfcBase as structs and enums. In this way all children of IfcBase were able to use these IFC constructs.

EXPRESS language LISTs and SETs were implemented as arrays. Static array definitions were primarily used for both lists and sets. While dynamic arrays are available in Unrealscript, they are not replicated in network sessions. This posed an issue when data in a multi-user session needed verification. As a result large static arrays were used where dynamic arrays were needed. Though this slowed performance, it allowed for the mechanisms of network reliability and communication to function properly.

The EXPRESS language WHERE rules and some of the underlying functions were primarily implemented as getters. However, some functions required lengthy or specific measures to ensure the proper outcome. Unrealscript functions do not return arrays, so some functions had to be constructed so as to return a set of variables to be combined into an array later.

The integration of IFC classes onto the Unreal Engine allows for loading and linking each IFC object and attributes specified by the particular view from the database. Once the environment is loaded, interaction with object data is replicated back to the database and propagated among all relevant user views.

8. RELATED WORK

Despite the fact that there have been efforts to provide online multi-user visualization platforms, many of these are built under the shared desktop premise. Most of these allow for a limited interaction between the data elements displayed by the meeting leader and the rest of the meeting members. Most of these have been specifically developed to support business type meetings and are unable to support the level of interaction and complexity as well as the multifaceted characteristic of design reviews.

Even though within IFC certain viewers are capable of implementing rendering and lighting, such as in the case of the Octaga Modeler (Figure 5), most of them have been designed having IFC developers in mind as the end users. These can be considered to be not appealing or realistic enough to imply a virtually immersive experience in their representation. Most of them display the information as information trees, and many map the entity information to the IFC schema. Needless to say networking capabilities have not being included in the implementation requirements of these.

Virtual environments for design reviews such as mWorld (Dias et al. 1997) are efforts developed specifically to support collaborative design development. Research efforts for collaborative environments have addressed issues of collaborative environments using computational and virtual environments (Craig
and Zimring 2002; Kalay 1998), but few have analyzed them as supporting interoperability across the entire AEC industry.

Most IFC viewers have been developed with the intention of IFC developers as end users. Despite IFC viewers, such as the Octaga Modeler, having rendering and lighting simulation capabilities implemented, most of them have limited capabilities regarding advanced rendering capabilities, and might be considered sub-par when compared to the rendering capabilities of many contemporary game engines that are available. Information display in IFC viewers is basically organized according to the present information referenced by the IFC entities and schema (Figure 6), which might not be understood by users unfamiliar with it.

**Figure 5. Octaga Modeler, Presentation of an IFC File.**

**Figure 6. IFC Engine Viewer, Data Presentation of an IFC File.**
9. CONCLUSION AND FUTURE WORK

It has been demonstrated (Moloney and Amor 2003) that virtual environments have a place in design review. They have also demonstrated the possibility of articulating the representation of two disparate building representations of a building in a game engine environment, but the system is mainly oriented to architectural assessment, our system tries to implement a platform independent review environment which can not only provide user friendly information environments throughout the AEC industry but also to support the instantiation of the building model coming from any party within the industry.

We are currently researching on the reporting capabilities of the Multi Level Design Review Tool. Not only should the process of generating online documents be capable of informing the meeting participants of decisions made during the review but it should also provide specific information regarding each of the user views, domain specific data as well.

The other area of extension under evaluation is related to what seems to be a natural extension for our tool, which is the possibility of generating design changes within the virtual environment and writing these changes into a valid IFC file. These kinds of extensions will require for the Multi Level Design Review Tool to be capable of performing changes in two different data domains.

Changes to the geometrical properties of the objects contained within the virtual environment through a mesh based representation will require translation of the IFC solid description capabilities either through solid or boundary based descriptors. In the same manner in which changes to the geometry will be implemented, we plan to establish a direct mapping between possible changes to object attributes in the virtual environment and the further translation of these into a standard IFC file format.

ACKNOWLEDGEMENTS

We would like to thank the IMAGINE Lab in the College of Architecture at Georgia Tech for the use of their facilities and expertise.

REFERENCES


