A Conceptual Framework For Multi-Modal Interactive Virtual Workspaces

SUBMITTED: July 2001
REVISED: November 2001
PUBLISHED: December 2001 at http://www.itcon.org/2001/11/

Rob Aspin, Mr.
Centre for Virtual Environments, University of Salford, UK
email:r.aspin@salford.ac.uk http://www.nicve.salford.ac.uk

Laurent DaDalto, Dr.
CSSI, France
email:laurent.dadalto@C-S.FR http://www.c-s.fr

Terrance Fernando, Prof.
Centre for Virtual Environments, University of Salford, UK
email:t.fernando@salford.ac.uk http://www.nicve.salford.ac.uk

Enrico Gobbetti, Dr.
CRS4, Italy
email:gobbetti@crs4.it http://www.crs4.it/vvr

Mathieu Marache, Dr.
CSTB, France
email :marache@cstb.fr http://cic.cstb.fr

Mark Shelbourn, Mr.
Construct IT, University of Salford, UK
email:m.a.shelbourn@pgr.salford.ac.uk http://www.construct-it.org.uk

Souheil Soubra, Dr.
CSTB, France
email :soubra@cstb.fr http://cic.cstb.fr

SUMMARY: *Construction projects involve a large number of both direct stakeholders (clients, professional teams, contractors, etc.) and indirect stakeholders (local authorities, residents, workers, etc.). Current methods of communicating building design information can lead to several types of difficulties (e.g. incomplete understanding of the planned construction, functional inefficiencies, inaccurate initial work or clashes between components, etc.). Integrated software solutions based on VR technologies can bring significant value improvement and cost reduction to the Construction Industry. The aim of this paper is to present research being carried out in the frame of the DIVERCITY project (Distributed Virtual Workspace for Enhancing Communication within the Construction Industry - IST project n°13365), funded under the European IST programme (Information Society Technologies). DIVERCITY's goal is to develop a Virtual Workspace that addresses three key building construction phases: (1) Client briefing (with detailed interaction between clients and architects); (2) Design Review (which requires detailed input from multidisciplinary teams – architects, engineers, facility managers, etc.); (3) Construction (aiming to fabricate or refurbish the building).Using a distributed architecture, the DIVERCITY system aims to support and enhance concurrent engineering practices for these three phases allowing teams based in different geographic locations to collaboratively design, test and validate shared virtual projects. The global DIVERCITY project will be presented in terms of objectives and the software architecture will be detailed.*

KEYWORDS: *Virtual Environments, Product modelling, Construction Industry, Client Briefing, Design Review, Construction Planning.*

1. INTRODUCTION

It is a known fact that a typical construction team will comprise of 20 or more organisations, formed into a temporary project team. Such teams are likely to be a unique combination of partners for each major project. These partner organisations can also be geographically separated. They have a pressing need to follow efficient project processes, set up integrated communications infrastructures, and develop shared models of the project and the buildings they are constructing. IT systems currently used in the industry are stand-alone point-to-point applications dealing with parts of the internal operations of participants in the process. According to the Egan report (Egan, 1998), the construction industry is typically dealing with the project as a series of sequential and largely separate operations undertaken by individual designers, engineers, constructors and suppliers. There is considerable benefit to be gained from substantially integrating solutions being applied by project teams as a means of reengineering the project process. A building project can only be considered as a successful project only if the final result meets the expectation of all the stakeholders of the project. All parties need to work as a temporary virtual organisation and get involved by setting up common objectives to deliver the final product successfully.

A recent survey (Egan, 1998) showed that clients believed that significant value improvement and cost reduction could be gained by the integration of design and construction. Furthermore, at present, client-briefings and multi-disciplinary design reviews take place at different time spans and usually around 2D drawings with little or no effort devoted to the lifecycle issues. This can cause unforeseen problems creeping into the design and discovery of construction problems later in the lifecycle. The problems associated with such limited interaction between design and construction has been documented by Lahdenpera (Lahdenpera, 1995).

There is a genuine need in the construction industry to explore the use of interactive computer modelling and simulation environments to improve client briefing and design reviews. Such environments can then be used to capture the client's needs and to ensure the compatibility between the client's vision of the project and the resulting product.

Recent benchmarking reviews of IT use in briefing and design have identified serious shortcomings in the construction industry (Construct IT, 1996). IT systems being used in the industry at present are stand-alone point-to-point applications dealing with parts of the internal operations of participants in the process.

This lack of technological uptake is compounded, in Europe, by the fact that research expenditure does not reflect its economic importance – investment is limited to 0.3% of the sector’s turnover, compared with the situation in Japan where investment is of the order of 2.0-3.0% (Brussels 1997).

Furthermore, such interactive technology can be used to consider lifecycle issues such as concept and detailed design, environmental impact, space planning, facilities management, emergency evacuation, security and constructability during design reviews, involving planners, architects, designers, civil-engineers, contractors, facility managers and security personnel, to facilitate concurrent engineering. There is scope for considerable improvement in the industry’s performance, productivity and ability to meet increasingly demanding customer needs by the prudent integration of IT into the industry.

If you ask the construction industry the direct question of what do they want from information technologies, they will tell you that they want everything yesterday, at the least possible cost! Clearly this is not sufficient information to build a system for improving the processes of construction projects.

Nevertheless, discussions with potential users, both in the UK and Europe through Special Interest Groups (SIG) and meetings with all project stakeholders set up by the user groups, have shown the need to go beyond the current use of CAD packages (and some VR toolkits) that display static “walkthroughs” and aim for a more direct and visual interaction with the design related data. The combination of Virtual Environments and Simulation Environments would therefore contribute in:

- improving the co-ordination and communication between the different project partners and stakeholders around a visual, and thus intuitive, 3D representation of the planned construction;

- evaluating the design, earlier in the process, in regards to different constraints (architectural, technical, financial, environmental, etc.) since VR tools allow the design team to quickly gain insight, resulting in high quality feedback on the project;
- displaying what-if scenarios during the detailed design phase, in order to assess the proposed solutions from different technical points of view (acoustic, thermal, lighting, etc.);
- bridging the gap between design and engineering on one hand and construction on the other hand.

2. WHAT THE STAKEHOLDERS WANT

The stakeholders questioned for the requirements included clients, architects, contractors, sub-contractors and different engineers. The stakeholders represented a cross section of the European construction industry. The information was gathered in a number of different ways. The user groups associated with the project gave their wish list of the requirements they would like to see in the DIVERCITY system. This was then tested with groups from industry to verify that the requirements given were indeed the wishes of the industry. The full wish list for the system is too large for this paper, so the main points that all stakeholders agreed upon will be highlighted.

2.1. Requirements capture

The researchers adopted Boehm's spiral development model (Boehm, 1988). However, to manage the large scale and evolving industry requirements, Boehm's recommendation for a "win-win spiral model" (Boehm, 1996) were taken on board, where he advises on creating three critical milestones, i.e. (i) lifecycle objectives; (ii) lifecycle architectures; and (iii) operational capability.

A number of techniques were used in order to communicate the complex top-level industry requirements and progressively break them down into more detail. In particular the following approaches were adopted: Vision statement; Stakeholder perspectives; Use cases and Systems requirements.

The vision and the stakeholder perspectives provided the lifecycle objectives for the industry and for DIVERCITY, respectively. Life cycle architectures were captured, using UML and scenario based development (Booch 1998; Kruchten 2000). Top-level architectural diagrams and some functional decomposition diagrams were developed to define the scope of DIVERCITY, and demonstrate how it fits into the larger industry requirements.

2.1.1. Construction industry use case

The construction industry was represented in the project as a use case to determine the scope of requirements that were needed for the project. The diagram below (fig. 1) shows this use case. The spheres show the high level processes that are undertaken in a typical construction project. The spheres have two different colours to them. The purple spheres are the processes that are being undertaken by the DIVERCITY project, and the orange spheres show the other processes that are not being supported. High-level requirements for all the spheres have been captured for the project with more detailed information being gathered for the spheres that are being supported by the project.

2.1.2. High level system requirements

Since DIVERCITY is an evolving environment, rather than developing Boehm's "initial operational capability" (Boehm, 1996), the project decided to define top-level technical requirements, from the users perspective. It is a short description of: (i) the hardware platforms the industrial users prefer; (ii) the need for DIVERCITY to interact with other construction applications (COTS-driven process); and (iii) how the users perceive the system to work.

2.1.3. Business specific requirements

The specific aspects being covered by the DIVERCITY project have been categorised into three separate software workspaces called "client briefing", the "design review" and "construction planning".

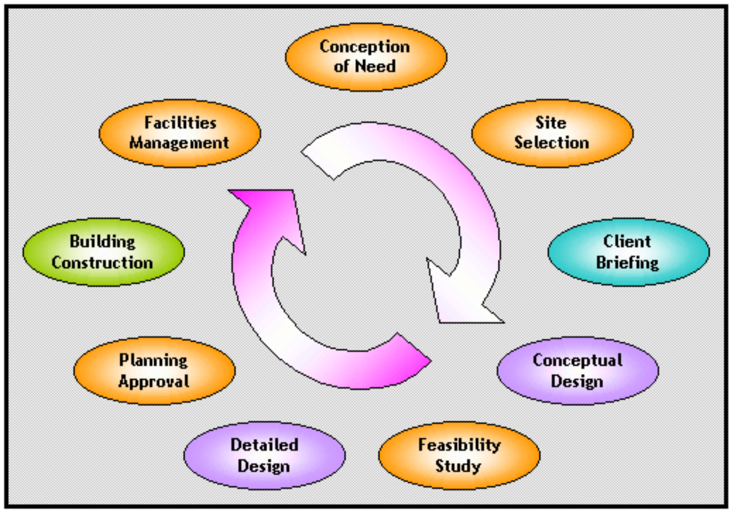

FIG 1: Construction Project Lifecycle use case diagram

Figure 2 (over the page) shows the processes of how the DIVERCITY system will handle the different simulation environments that are to be developed. The background system also needs to be developed in order for the specific business requirements can be fulfilled within the DIVERCITY system.

2.1.4. Procurement path

The procurement path for any given project is chosen from a number of differing types. Discussions on the most relevant path that would encompass the most practices of the European partners was decided to be the "Design and Build" approach. This meant that the project consortium would base its case studies around projects with a design and build approach initially, and once it had been proven that this approach could be supported by the DIVERCITY system then other procurement paths would be tested.

2.1.5. Plugins

As already highlighted above the construction industry uses many bespoke software and hardware systems, this means that it is difficult for these systems to communicate and pass information between them. Stakeholders would like to see a system that they can plugin their existing software and hardware to enable them to work more collaboratively with other project stakeholders.

2.1.6. Simulations

The project stakeholders highlighted many different simulations that they would like to be able to show clients in the DIVERCITY system. They include structural; lighting; acoustic; thermal; equipment, finishes and furniture specifications; budgetary models; environmental impacts; buildability; site layouts and access; waste management; maintenance schedules; etc.

Obviously these all require expertise that this project consortium does not have, and would require a budget to complete larger than the one allocated by the EU. The expertise within the consortium means that 3D lighting, acoustic and thermal, site layout and access, and buildability simulations will be supported.

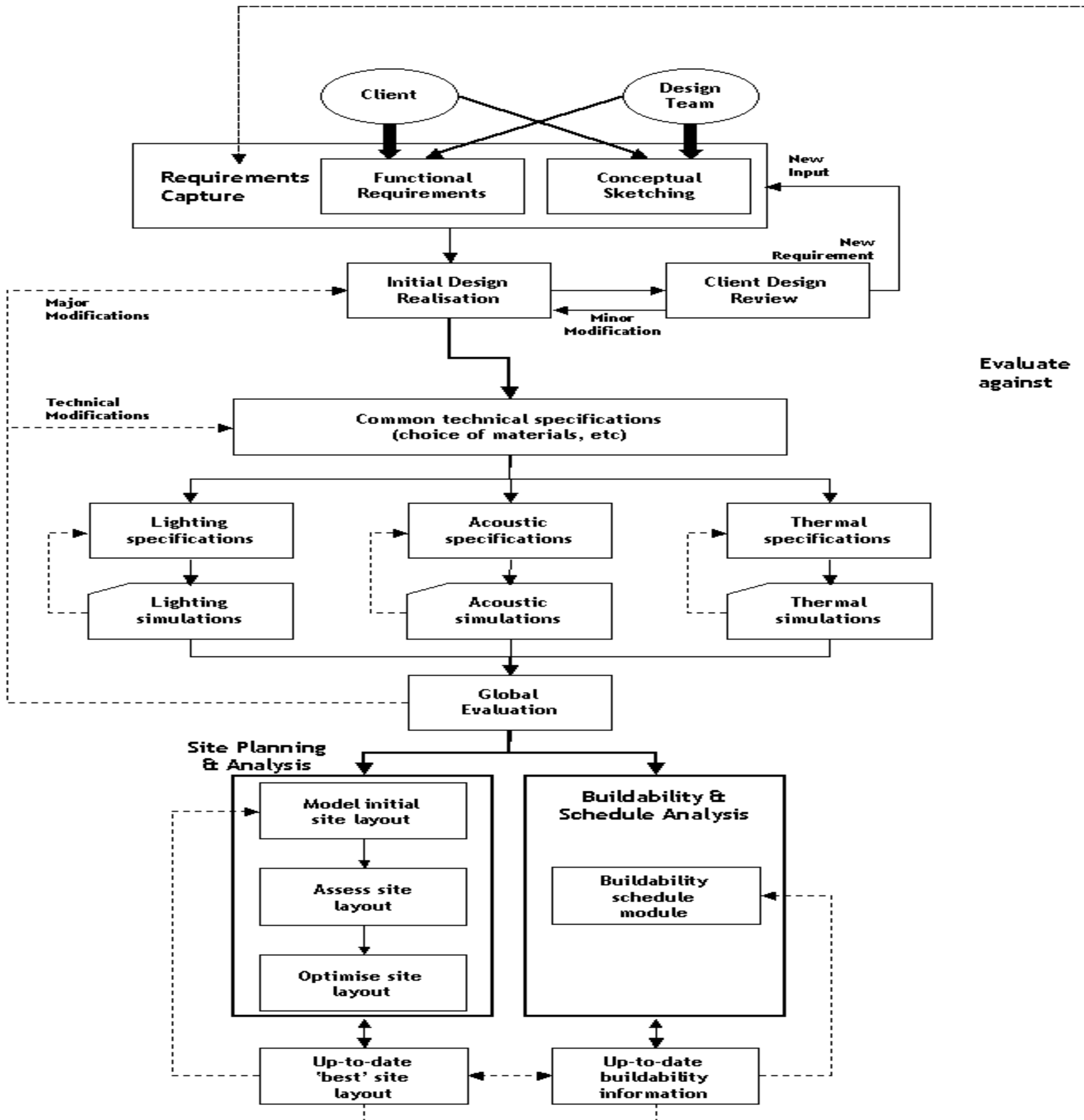

FIG 2: DIVERCITY project process map for the business specific use cases

2.1.7. What if scenarios

Through the use of the simulations the users wish to be able to undertake what if analyses of different building positions on the site; different positions of spaces within the building; different lighting, thermal, acoustic properties of spaces; to name a few. Different versions of the design at this stage should be able to be viewed and stored with comments on why a particular aspect of the design has been left out or included.

2.2. What current systems offer

Historically CAD programs were created for building models. In contrast, VR programs were created to display models. Originally CAD information could only be entered and viewed in a 2D form. However, as CAD software increased in sophistication the model could be viewed and then edited in a 3D format. Future developments of CAD include so called 4D CAD (3D + time), which will enable the CAD model to be viewed at each stage of the development lifecycle. Nevertheless, the prime purpose of CAD software remains the entry of data for model creation. VR surpasses CAD by being able to place users inside the model, allowing them to interact directly with the objects they are viewing rather than through a 2D computer interface. Many 3D solid model CAD systems allow the user to rotate an object but only within VR can the user walk around the object, stop, touch it, manipulate parts of it, or even enter it. CAD software can be linked, via an interface, to VR software, in order to import data models into the virtual world. This data adds a powerful new way to understand and interact with CAD data. Furthermore, VR allows the user to improve the visual appearance of the products. This can be done by applying texture mapping, surface properties and, with sophisticated programs, varying lighting conditions. VR can also allow the user to define kinematic behaviour of objects to demonstrate the operation of a particular product.

2.3. What are we doing

Researchers have identified the need for an integrated construction environment, which acts as a project repository, during all stages of the lifecycle (Kiviniemi, 1999; Aouad et al., 1997; Alshawi, 1996; Aish, 1999; Grassi, 1999; Sawhney, 1999). However, this environment has been difficult to implement and no commercial solutions exist.

Furthermore, the state of the art concerning software tools supporting the early and detailed design phases show that the existing tools lack of 3D real-time inspection features expected by users to test and compare different design alternatives does not exist.

In that context, the DIVERCITY consortium aims to develop innovative workspace technologies for the briefing and design phases of the life cycle and evaluate the results on live projects. This will allow the improvement of the co-ordination and communication between the different project partners and stakeholders around a visual, and thus intuitive, 3D shared conceptual model of the planned construction. In the future, these workspace technologies will be extended to other phases of the construction lifecycle, in an aim to create an integrated construction environment.

2.4. Performance measures

The DIVERCITY project is aiming to validate the results of this project by comparing two similar projects. One project will use the system and the other will use the traditional methods of the project lifecycle. This will be the most difficult aspect to the project. User evaluations will provide feedback on the usefulness of the different modules in the system. This information will help in deciding the success of the project, but it will not be until the system has been used on a number of construction projects that the real benefits of the system may be realised.

3. SYSTEM REQUIREMENTS

The user groups in the DIVERCITY consortium highlighted their “wish-list” as one of the first tasks to be completed in the project (DIVERCITY, 2000). Each group represented a stakeholder in the project and listed what they would like to see from a virtual workspace that would help in the client briefing and design activities of a construction project. The results of these discussions were collated together and presented to the technical developers who then analysed them and described to the users what was capable in the time and budgetary constraints.

3.1. DIVERCITY client application

The DIVERCITY Client Application (DCA) must fulfil several requirements. Some of the requirements are general; others are related to the nature of the workspaces that the DIVERCITY project is seeking to address. These general requirements for the DIVERCITY system relate to both user requirements and the technical demands of a VR system.

Research into user demand has shown that the system will need to be configurable in that each stakeholder within the construction process should be able to customise their own application workspace for the role they fulfil within the design project. The ability to integrate proprietary and 3rd party software with the DIVERCITY system is also seen as an important feature.

VR technology imposes several demands on the system architecture. Issues relating to the rate at which data can be accessed and the appropriateness of the data to the 3D rendering process is important if the illusion of presence, coupled with effective interaction, is to be achieved. Obviously the system should be capable of displaying the virtual environment, on a variety of display devices (CAVE, Reality Room, etc) and should also allow for the use of input devices that support the interactive nature of the environment.

3.2. Workspaces

Each of the three DIVERCITY workspaces has their own particular requirements of the system. These are in addition to the general system requirements, which will support the domain within which the workspaces are developed. More detail is given in the next sections.

3.2.1. Client Briefing

Client Briefing is predominately concerned with the communication between the client and design team/designer. The workspace must aid the designer in capturing the clients' needs and allow the designer to present design concepts to the client. Clients' needs are typically captured as text documents, often supported by sketches made by the designer in the presence of the client. DIVERCITY aims to develop 3D interactive tools to support this process, so the provision of sketching tools within the system is important. However, many of the requirements will still need to be expressed as text, so the DIVERCITY system should also support the annotation of the design sketches with notes that can illustrate the clients' needs.

To allow the designer to present their concepts to the client the ability to display early stage designs within a 3D environment is desirable. Many construction CAD tools already have the ability to display static "walkthroughs" of a design, however, little interaction is usually possible. Within the DIVERCITY system, the ability to make minor changes from within these presentation environments is seen as important. The ability to make changes to the design is possible in some CAD walkthrough systems, however, this must normally be achieved from within the design tool, and then re-imported to the display system. Potential end-users of the DIVERCITY system have suggested that the ability to make these changes within the display environment is a feature they would like to be developed.

It is neither desirable, or within the scope of the DIVERCITY project, to develop a completely new CAD tool for the construction industry. However, designers will need to progress from the sketches, made as a result of their discussions with the client, to the designs they wish to present to the client. Typically, this would be done using an existing CAD package of the designers' preference. The sketches should form a starting point for their design development and therefore there is a need to be export these to a CAD tool for a more complete design to be finalised. The more complete design needs to be brought back into the DIVERCITY system from the CAD package. It is therefore important that the DIVERCITY application can both read and write at least one common CAD format.

3.2.2. Design Review

The Detailed Design Review is an important phase in the design/construction process where the inputs are represented by a rather precise architectural design (usually drawings on a 1:100 scale) and the outputs are precise definitions of all the technical domains related to the design (e.g. structural design, heating and thermal, lighting, acoustic, fire safety, etc.).

Current software tools supporting detailed design review already exist. Nevertheless, these exiting tools suffer from two important limitations:

- Lack of 3D-real time inspection features. Consequently, members of the project team spend too much time trying to (i) understand the project information and (ii) describe this information to one another;
- Discontinuities between the different software tools. This makes the re-use of the results of one phase in the design process as an input for another phase practically impossible.

Therefore, the Design Review workspace within DIVERCITY looks at enhancing reviews by combining Product Modelling technologies with Simulation Environments in order to allow project teams to visualise and to interact (in real-time) with the project on a multidisciplinary basis. Continuous design is a major feature here. This means that the architectural design resulting from the conceptual design phase can be fed into the Design Review workspace and that the detailed design can be fed to the Construction planning workspace without any data loss therefore bringing important improvements to the overall process.

Three key simulation features are addressed: the lighting, thermal and acoustic properties of the design. In order to display the results of these simulations the display component of the system will need to be able to integrate the results with the project data. In addition the system should also allow users to manipulate the simulation parameters, so that what-if scenarios can be evaluated from within the 3D environment, rather than having to use an external system.

3.2.3. Construction Planning

Construction Planning allows the designers to determine how the project will be built. Assessments about the order of construction, site safety, and location of the temporary site facilities are all operations which may be performed by this workspace. This implies requirements on the system to have a concept of time that may be linked to the design, as well as being able to set up various simulations and position objects within the 3D environment.

4. DESIGN CONCEPT

Two requirements have had a significant impact on the design of the DIVERCITY Client Application (DCA):

- the need for the application to be extensible, so that 3rd party developers, as well as DIVERCITY partners can extend the functionality of the system ;
- the ability to allow the users to configure the system for the role they are fulfilling within the project.

These two conditions both suggest that the design of the DCA needs a modular approach, which will allow the inclusion of additional modules without significant impact on the existing system. In the case of configurability this should also allow the removal of unused modules. One approach could be to design the system as a set of interlinked applications, each application implements some functionality and can be used to perform operations on either a data file or some form of database. However, to achieve good display rates and latency free interaction, VR requires that the functionalities are more tightly integrated. The tools used within the virtual environment must be developed as part of that environment, using the same data structures as the environment itself.

For this reason the high-level design of the DIVERCITY Client Application is a framework, which will support the integration and operation of the various modules. This may be thought of as a plugin based system, similar to many advanced applications such as Netscape and Internet Explorer. The plugin modular design takes advantage of the use of dynamic libraries common to most modern operating systems. In the DIVERCITY Client Application, almost all functionality will be implemented as plugin modules. The DCA forms a framework that allows the user to select which plugin modules they wish to use and then ensures that the modules operate correctly within the framework. This design offers great flexibility, not only does it address the users requirement to have a configurable system, it also allows for the development of additional plugin modules, which extend the functionality of the DIVERCITY System.

One important issue raised by the use of a plugin architecture is where and how the runtime data is managed. Obviously all the modules will need to access it in some way. Equally, if the data is maintained within one of the plugin modules, how can others depend on that data being available within the system.

The approach the DIVERCITY Client Application takes to solve this runtime data issue is to provide a special form of plugin module interface, a Design Structure Layer (DSL) from which modules that are purely responsible for the management of runtime data are inherited. Each DSL is used to maintain a single data structure of a specific format. For instance one DSL may contain a CAD design description of the project, another may be used to manage GIS data relating to the site, a third may be used to maintain a rendering scene graph for the 3D Display. All could be available within the system at the same time. However, many of the components within these DSLs are common, that is within any set of DSLs there may be several representations of the same design element. For instance, the same wall may be in both a CAD DSL and within a DSL that maintains a rendering scene graph. If one of these representations is changed all of the others are then dated. Furthermore, if two of the different representations are changed at the same time how could the differences be resolved.

To manage the updating of the different design representations, and to ensure that conflict resolution can take place the control of the DSLs is carried out by a sub-component of the DIVERCITY framework, the Shared Design Space (SDS). The SDS is responsible for managing the operation of the DSLs, it also maintains a collection of design elements, which allow key features of the design, such as walls, windows, etc, to be linked across DSLs. By granting plugin modules exclusive access to a design element and registering when changes have been enacted, the SDS can then ensure that conflicts do not happen and that any DSL can receive notification of the changes that take place on design elements for which they maintain a representation. The SDS can then also act as a single entry point for any module that needs access to the runtime data.

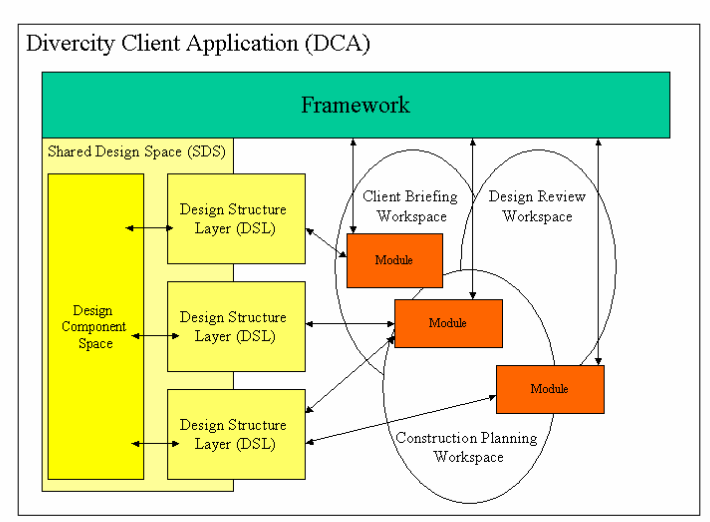

FIG. 3: High-Level Conceptual Design.

While the main communication process between modules will use the SDS modules, with modules receiving notification of change to Design Elements they have expressed interest in, communication will also be permitted directly with each other. However, it is not anticipated that direct inter-module communication will be used for anything other than simple synchronisation and state notification.

5. SYSTEM ARCHITECTURE

Much of the functionality of the DIVERCITY Client Application (DCA) will be developed within plugin modules. Due to the transitory nature of these modules inclusion within the system, it is impossible to define a single system architecture for them. However, the framework that supports the modules represents a complex system, for which a detailed design must be created to ensure that it meets the needs of the application.

The DCA framework is responsible for providing the core services that allow for the configuration of the modules and the operation of them during the runtime of the application instance. Its two main responsibilities are to maintain and enable the plugin modules within the system, and to provide support for the centralized runtime Shared Design Space (SDS). This is intended to provide flexibility both in the users selection of tool sets, and the range and type Design Structure Layers (DSLs) available.

For instance, several rendering modules could be implemented which would support differing rendering modalities, such as stereoscopic, desktop, and CAVE, the users could then choose which one they wish to use for their workspace.

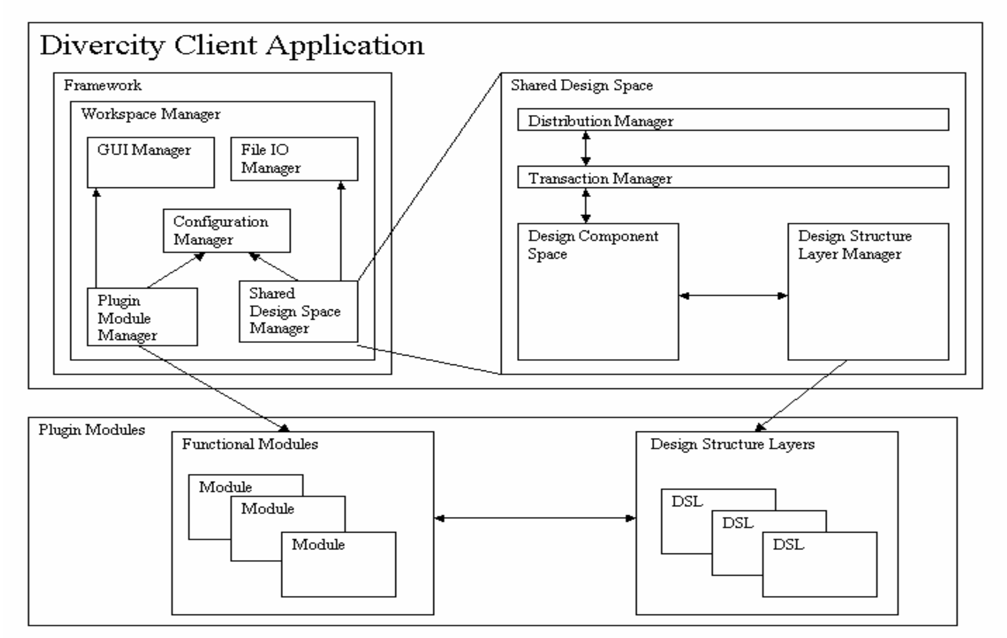

FIG. 4: Detailed Design of the software framework.

5.1. Application framework

The DCA Framework is the part of the application that binds everything together. This consists on one super manager, the Workspace Manager, which initialises the system, starts the other high-level system managers, and then co-ordinates the interaction between these management components.

The GUI manager instantiates an empty GUI framework and provides services to modules which allow to them present their own GUI widget within the main application GUI. By starting the GUI manager first the DCA may display start-up information to the user, and allow configuration of the DCA plugin modules when the Configuration Manager is started.

Following the initialisation of the GUI Manager the Configuration Manager is started. This loads the system configuration files, populating a data structure that may be queried by all application modules. In particular the Configuration Manager will load set-up information for the plugin modules and the Design Structuring Layers.

The Module Manager is responsible for the management of the dynamic modules that implement the DIVERCITY applications selected functionalities. Its main role is to load and initialise the users choice of modules, creating the functional workspace defined by the user. A secondary role is to provide services to the plugin modules such as allowing them to create user interfaces, using the GUI Manager, and allowing them to gain handles to other modules and the Design Structuring Layers within the Shared Design Space. The Module Manager also provides a module interface from which all plugin modules have to be inherited. This interface defines the manner in which the module is initialised forcing the module developer to implement the functionality necessary for the Module Manager to control the plug-in module.

5.2. Shared Design Space

The Shared Design Space (SDS) provides the DCA with a runtime data repository that preserves multiple semantic representations of the same data set. This is achieved by maintaining several Design Structure Layers (DSLs), each of which maintains a single semantic data set. The Design Component Space (DCS) then provides a mechanism to link key design features (called Design Elements), such as a window, table or wall, etc, across some or all of the available DSLs, provided they contain a structure that relates to the key design feature. This system is intended to provide a managed, structured form of a common construct used in many VR applications;

the maintenance and synchronization of multiple scene graphs, each of which defines a different semantic structure to the same data set. However, unlike the common implantations of multiple scene graphs, the Shared Design Space aims to allow the synchronization of data across different data structures. This enables the DIVERCITY application to maintain multiple data structures in the native format of the source data. For instance, the application could contain one DSL, which maintains a native CAD format that defines the construction semantics of the design, and one DSL, which would maintain native OpenGL Optimizer scene graph definitions of the rendering semantics of the design. Changes made to either DSL will be propagated and translated to the other ensuring synchronization across the entire data set.

The Design Structure Layer manager provides four interfaces for implementations of the DSLs to inherit from.

1) The Design Structure Layer interface defines the minimum functionality that a DSL must implement to allow the Design Structure Layer manager to maintain and interact with it. This interface must be implemented by all DSLs. Each DSL will extend the DSL interface to provide its own interface to the native design data it contains. The remaining three interfaces are optional and implemented in addition to the Design Structure Layer interface. A DSL may implement just one of these additional interfaces or any combination of them, however the system requires at least one implementation of each. These additional interfaces are used to synchronise the DSLs and ensure that at least one DSL can provide a render-able scene.
2) The High-Level Geometric Supplier (HLGS) interface implements support for the synchronization of the DSLs. The DSL that implements this interface will act as the main synchronization agent within the Shared Design Space. As such it will be notified first of any modifications to the design data and produce a High-Level Geometric Element which describes the Design Element in a format understood by all DSLs. This High-Level Geometric Element will then be broadcast to all other DSLs that contain a representation of the Design Element so that they can update their native format data.
3) The Tessellation Supplier interface allows one DSL to act as a tessellator creating render-able polygonal representations from a High-Level Geometric Element.
4) The Rendering Supplier interface will allow one DSL to provide a 3D render-able structure that can be used by the rendering module. This render-able structure could be simply the connected set of all polygonal representations, but it is more likely to be a composite of both there High-Level geometry and the polygonal data, maintained within an off-the-shelf scene graph such as SGI's OpenGL Optimizer.

Within the Shared Design Space, four management modules exist: the Design Structure Layer Manager, the Design Component Space, the Distribution Manager and the Transaction Manager.

1) The Design Structure Layer Manager ensures that the DSLs necessary to handle the project data are available and coordinates their operation.
2) The Design Component Space maintains the collection of Design Elements that are used to link the common components of the DSLs.
3) The distribution manager responsible for distributing the data to multiple users, some of whom wish to access the data from remote locations.
4) The Transaction Manager mediates all changes within the data structures. It is responsible for ensuring that Design Elements are only modified by one functional module at any time. This is to ensure that no data is over written or changed by two modules at the same time. This will be achieved by locking a Design Element, granting exclusive access to a single module, during the modification stage and unlocking the Design Element when the changes have been committed. Once a change has been committed the Transaction Manager ensures that all the DSLs connected to the changed Design Element are notified of the change type and the DSL in which the change was made, this allows the connected DSLs to update their own representation. One consideration with the change mechanism is how the DSLs receive the information about the change that has take place. Notification of the change type uses an event mechanism which describes the type of change event, whether the Design Element was created, destroyed, or modified, and even some information about what sort of modification has to take place, transformation, geometric, polygonal, etc. However, many changes could take place and not all of them can be anticipated when developing the system architecture. One way to communicate the exact parameters of the

changes could be to directly query the DSL in which the original change took place. However this would mean that all DSLs would need to be able to comprehend all other DSLs. Not only is this impracticable, an exponential growth of the DSL translators would ensue, it would also mean that existing DSLs would need to be modified to parse the new data format. The alternative is to have a common data representation that may be understood by all.

Within the DIVERCITY Client Application many data formats can be supported, each one represented by a single DSL. However the are some particular data representations that relate to the nature of the application. Obviously, since this is an interactive 3D application a render-able runtime data structure is necessary. Furthermore, the DIVERCITY source data is typically CAD. CAD data formats typically use high level geometric constructs (geometric primitives manipulated with some form of Computational Solid Geometry - CSG) to describe a design. In order to present the render-able image the high level geometry must be converted to a render-able form. This means that we have two essential DSLs that the DCA must contain for any configuration; A High-Level Geometric DSL, and a render-able DSL. The system then has three basic data types that may be used to pass information within the application; High-Level Geometric Data, Render-able Data, and a intermediate format containing polygonal information which is generated from the high-level geometric data and used to populate the render-able data. Provided the High-level geometric data has a well defined format this may be used as the common format for passing information about changes to design elements within the DCA.

To accommodate these essential data mechanisms within the DCA the Shared Design Space (SDS) defines three interfaces that accompany the DSL interface. The High-Level Geometric Interface defines the operations a DSL must support for it to the systems registered High Level Geometric DSL, this is supported by a High-Level Geometric Element class interface which is used to define a class that can encapsulate this form of information about a Design Element.

5.3. Handling Large Tessellated Models

Complex synthetic scenes composed of millions of graphics primitives are rapidly becoming commonplace in many application domains, and architectural scenes such as the ones targeted for DIVERCITY make no exception to this rule.

The most successful radiosity technique for dealing with complex scenes is currently hierarchical radiosity (Steven, 93, Hanrahan et al., 1991; Schroeder, 1996).

The algorithm constructs a hierarchical representation of the form factor matrix by adaptively subdividing planar patches into subpatches according to a user-supplied error bound.

By treating interactions between distant patches at a coarser level than those between nearby patches, the algorithms reduces the cost from quadratic to linear in the number of subpatches used. However, since an initial transport link has to be computed from each of the original patches to all others, the cost is also quadratic in the number of input polygons. Volume clustering methods (Sillion, 1995), such as the one implemented in the base DIVERCITY lighting module, combat this problem by grouping input patches into volume clusters. While these methods avoid the initial quadratic transport link step, handling the light incident on a volume cluster is a difficult problem and most presented solutions are more suitable to handling unorganised sets of polygons rather than continuous curved surfaces (Willmott , 99).

Moreover, the need to maintain consistency in the multi-resolution representation of illumination forces the algorithm to traverse each input polygon

This fact has forced current radiosity systems to limit themselves to simple (or simplified) graphics scenes. In most cases, this limitation forces application users to manually remove details (e.g. by specifying fixed low polygon count tessellations of curved surfaces) from the designed scenes.

Willmott et al. (Willmott, 1999) recently improved this situation, presenting a hierarchical radiosity algorithm that works on multi-resolution meshes, picking the level of simplification appropriate to each transfer of radiosity between solution elements. Our work will is based on the technique introduced by Willmott and colleagues.

In our approach, a pre-processor constructs a multi-resolution representation of highly tessellated models by recursively grouping nearby patches. The result of pre-processing is made visible to the radiosity solver in the form of instances of an abstract data type representing multi-resolution models as collections of hierarchical patches. Each of the hierarchical patches are able to respond to queries from the solver in bounded form (i.e. by providing expected values and interval bounds). The solver uses the bounds to estimate solution error in order to choose the appropriate resolution. The solution is stored in a irradiance vector map, whose resolution depends only on the solvers required accuracy. The irradiance vector map is then used at rendering time to compute the colour of each input polygon.

6. CONCLUSIONS

This paper has described some of the R&D effort made within the framework of the on-going EC funded Divercity project aiming to develop innovative workspace technologies for the briefing and design phases of the life cycle and to evaluate the results on live projects.

After a presentation of the methodology and tools for User Requirements capture, the software architecture proposed by the project consortium was detailed. By its configurable and extensible features and its support to a STEP based data warehouse (that acts as the project repository during all stages of the design / construction life cycle), this framework represents an adapted infrastructure for the development of innovative Visualisation and Simulation tools for the construction industry through the Divercity project and beyond.

Future research will aim to complete the development of the framework and to validate the approach on real construction projects.

The Divercity consortium is composed of building industry representatives from Denmark (COWI), Finland (Evata) and France (SPIE) and technology providers from France (CSTB and CSSI), the UK (University of Salford), Italy (CRS4) and Finland (VTT) and researchers from Denmark (University of Aalborg). The collaboration between construction industrials (representing end-users) and technology providers and researchers has proved to be very fruitful in the design and implementation of an innovative system adapted to the needs of the construction industry.

7. REFERENCES

Aish, R., (1999). "Migration from an Individual to an Enterprise Computing Model and its Implications for AEC Research", Berkeley-Stanford CE & M Workshop: Defining A Research Agenda, Stanford, 1999.

Alshawi, M. (1996). "SPACE: integrated environment" Internal Paper, University of Salford, July 1996.

Aouad, G., Marir F., Child T., Brandon P. and Kawooya A. (1997). "Construction Integrated Databases- Linking design, planning and estimating, The OSCON approach"; International Conference on the Rehabilitation and Development of Civil Engineering Infrastructures. American University of Beirut, 52-60.

Boehm, B.W., (1988). "A Spiral Model of Software Development and Enhancement", Computer, May 1988.

Boehm, B.W., (1996). "Anchoring the Software Process", IEEE Software, July 1996.

Booch, Grady, et al (1998). "UML Users Guide", Reading, MA: Addison Wesley Longman, 1998.

Brussels (1997). "The Competitiveness of the Construction Industry Brussels", 04.11.1997, COM (97) 539 final.

Construct IT (1996) Benchmarking Best Practice Report – Briefing and Design, Construct IT Centre of Excellence, University of Salford, UK: University of Salford, ISBN 1-900491-33-8.

DIVERCITY (2000). DIVERCITY User Requirements – Issue 1 – User requirements – Stakeholder Perspectives, Use Cases – New Process Model Definitions, Stsyem Architecture Requirements, Internal Project Document, available on request, visit www.e-DIVERCITY.com for more details.

Egan (1998). "Egan Report: Rethinking Construction", UK Department of Environment, http://www.m4i.org.uk/publications/rethink/5.htm

Grassi, M.D., Zorgno, A.M., (1999). “The Virtual Construction Enterprise”, Berkeley-Stanford CE&M Workshop, Stanford 1999.

Gortler, S., Schröder, P. Cohen, M. and Hanrahan, P. (1993). “Wavelet radiosity”, Proceedings of SIGGRAPH 93, 221-230.

Hanrahan, P., Salzman, D. and Aupperle, L. (1991). “A rapid hierarchical radiosity algorithm. Computer Graphics (SIGGRAPH '91 Proceedings).

Kruchten, Philippe, (2000), “The Rational Unified Process – An Introduction”, Second Edition, Addison-Wesley, 2000, ISBN 0-201-70710-1.

Kiviniemi, A., Laitinen, J., Lautanala, M., (1999). “Defining a Research Agenda for AEC”, Berkeley-Stanford CE&M Workshop, Stanford 1999.

Lahdenpera, P (1995). Reorganizing the building process, VTT Publications 258, Technical Research Centre of Finland, Espoo, Finland, 210p.

Sawhney, A., (1999). “Research and Development Plan for the AEC Industry”, Berkeley-Stanford CE&M Workshop, Stanford.

Schroeder. P., (1996). “Wavelet Radiosity: Wavelet Methods for Integral Equations”. In ACM SIGGRAPH '96 Course Notes – Wavelets in Computer Graphics, 143-165.

Sillion, F., Drettakis, G. and Soler. S. (1995). “A clustering algorithm for radiance calculation in general environments”, Eurographics Rendering Workshop 1995, 196-205.

Soubra, S., Coudret, F., Lombardo, C. and Marache. M. (2001). “VR for Construction: the Divercity approach”, VRIC.

Turk Z. (1991). Integration of Existing Programs Using Frames, CIB Seminar Computer Integrated Future, 16-17 September, Eindhoven, Netherlands.

Willmott, A., Heckbert, P. and Garland, M. (1999). “Face Cluster Radiosity, Rendering Techniques '99”, Eurographics, 293-304.