Virtual Collaborative Building Design Environment Using Software Agents

Ibrahim Fahdah
University of Nottingham
England
evxif@nottingham.ac.uk

Walid Tizani
University of Nottingham
England
walid.tizani@nottingham.ac.uk

Abstract

This paper describes an architecture for virtual collaborative building design system based on software agents. The environment helps the design team to work collaboratively and concurrently on a centralised shared model and carries out all necessary communication and data exchange electronically. The environment has been implemented in a prototype application as server-client model using .NET technologies. Virtual reality is used for visualisation and to allow for intuitive interaction with the designers. Software agents are used to carry out communication and design activities. The current implementation has shown the potential of the used technologies to support a practical virtual collaboration.

Keywords

Collaborative design, Virtual reality, Agent technology, Engineering design, Steel structures.

1. INTRODUCTION

The engineering design process for a project is carried out collaboratively. It is primarily concerned with specifying the 'product' that best fulfils the client's brief, and that ensures safety during construction and use. This process requires interactions between the disciplines of architecture, building services, structural engineering and site construction. The communication during the design process has a major bearing on the overall cost and quality of the completed project with knock-on effects on downstream issues spanning all project stages.

The benefits of following concurrent and collaborative design practices within a building design environment are now widely recognised [1]. Project teams are encouraged to work together more closely and to exchange project information in a more structured way. Collaborative engineering attempts to advance the design activities by maximising concurrency and collaboration in practice [2].

The advances in information technology enhance the capability for creating real-time and virtual collaborative design environments. These technologies are not yet fully exploited and are promising to deliver real improvements to the design process. The use of these technologies for a collaborative system can be applied to three relevant areas: virtual reality, engineering design, and communication.

Virtual reality technology can assist in bridging the gap between the various discipline-based representations of information by providing a common and an intuitive representation of the end-product.

Within virtual reality environments, considerable emphasis is usually placed upon the visual aspect or interface to the buildings design through 3D visualisation tools, and less so upon its underlying behaviour. However, virtual reality, in the context of building design, is based on the idea of virtually producing the appropriate behaviour and visualisation of a building before, during and after the buildings construction. Therefore, virtual reality in its fullest sense cannot be fully realised without also simulating the buildings behaviour through the relevant supporting processes and underlying data structures.

Product and process modelling is an active research area. There are attempts to create standard product models for building industry. Such attempts include the IFC and CIS [3,4]. The primary aim of developing these standards is to allow different software to exchange data.

Agent technology is reported to be well-suited for use in applications that involve distributed computation or communication between components [5]. It is widely recognised as a promising paradigm for the future engineering design systems [6]. Research centres have already applied the technology to concurrent engineering, collaborative engineering design and, planning and control systems [6].

This paper reports on an investigation into using virtual reality and software agents for collaborative design. Virtual reality is used to allow for the modelling of virtual prototypes while the software agents are used for both design automation and communications.

2. VIRTUAL COLLABORATIVE DESIGN ENVIRRONMENT

A prototype software for collaborative design and appraisal of multi-storey steel buildings has been implemented. The feasibility of conducting a 'virtual collaborative building design' has been investigated together with the methodologies and techniques that will enable such a design. The term 'virtual building' refers to the virtual representation of a building that behaves in a realistic way from the engineering point of view.

The overall system architecture is client/server architecture with centralised shared resources. This type of architecture is very common for multi-user applications. The main advantage of client/server architecture is providing a secure central data repository. That helps reducing data fragmentation and increase data integrity.

The physical structure of the system is a central server, maintaining the design models, and many workstations where designers can access the model data via a network. The virtual structure of the system is a three-layer model: design layer, communication layer and data layer. Figure 1 shows a macro view of the system architecture.

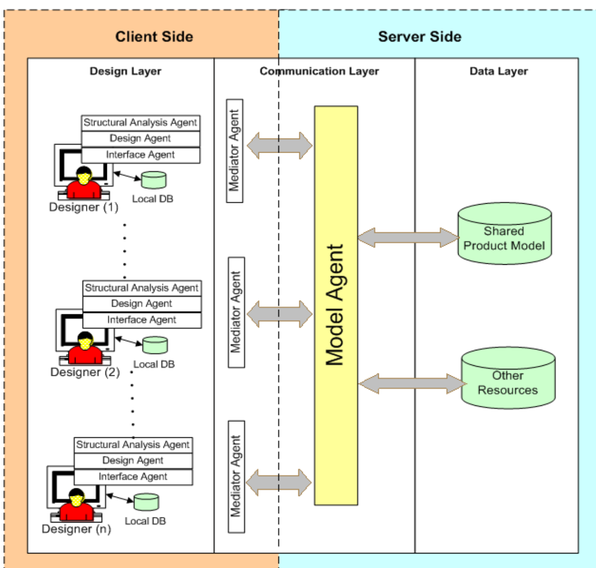

Figure 1: System architecture

The design layer includes the design system which composes a Structural Analysis Agent, a Design Agent and an Interface Agent. These agents assist the designer during the design process and they are located on the client side.

The communication layer includes a Mediator Agent and the Model Agent. The Mediator Agent is located on the client side and the Model Agent located on the server side. Data exchange is achieved through the communication between those two agents.

The data layer consists of a shared product model and all relevant resources. It is located on the server side and maintained by the Model Agent. However, a copy of the product model database will exist on each client side to reduce the number of calls between the client and the server. This improves the system performance. Consistency between the models is maintained by a locking mechanism.

2.1 Communication Model

Working with distributed applications involves handling network calls within the limitation of a given network speed. This is an important issue if to avoid networking becoming the bottleneck of the design process. The adopted approach was to get more done on the client side with fewer calls across the network. Reducing the number of calls is critical when creating high-performance distributed applications.

Figure 2 shows a typical communication scenario between the Mediator Agent and the Model agent. It can be seen that most actions are carried out on the client side with fewer calls with the server. This ensures a better performance. In addition to reducing the communication calls with the server, each message is packed to reduce the size of the data transferred over the network.

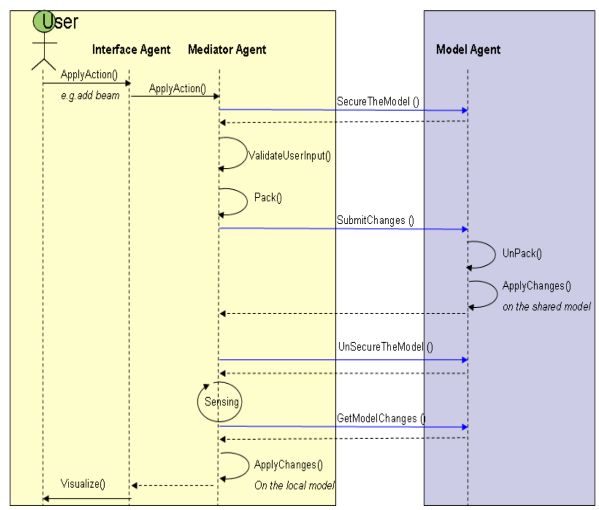

Figure 2: Communication model

One challenge in managing the shared model integrity by ensuring that no two users can attempt to update the same information simultaneously, leading to corruption of the data being stored. The system provides Lock and UnLock methods to ensure that only a single user can update the shared model at any given time. The system automatically locks the model database prior to modifying any data stored in the database, and unlocks it once the modification is complete.

The user, however, can optionally own the model lock and prevent all other users from modifying the model until it is released again. It is recognised that such model locking might hinder collaboration. However, it is provided so that large one-off modification can be made efficiently.

2.2 Collaboration Model

The collaboration model does not suggest how the design team members should deal with design conflicts. However, the design team will be able to place their status of acceptance on the integrated proposal. They have three options: complete agreement, no agreement and no comment. Under 'no agreement' the design members have to communicate electronically in asynchronous or synchronous manner through the system tools to sort out the disagreements and reach consensus. It is possible that after several rounds of iteration, the 'no agreement' situation could remain unchanged. Under such circumstances, it may be necessary to prioritise specific design requirements at different levels and communicate synchronously (e.g. e-chat, face-to-face).

Error! Reference source not found. represents collaboration in an asynchronous manner. The design decisions are disjointed and the design members input their designs separately. Time is usually not a driver and each group has sufficient time to document their input in the improvement proposal. However, since each member's presentation is only weakly connected to the other design participants a longer time period is normally required for obtaining a consensus. In this instance the asynchronous approach is less effective in resolving the 'no agreement' case.

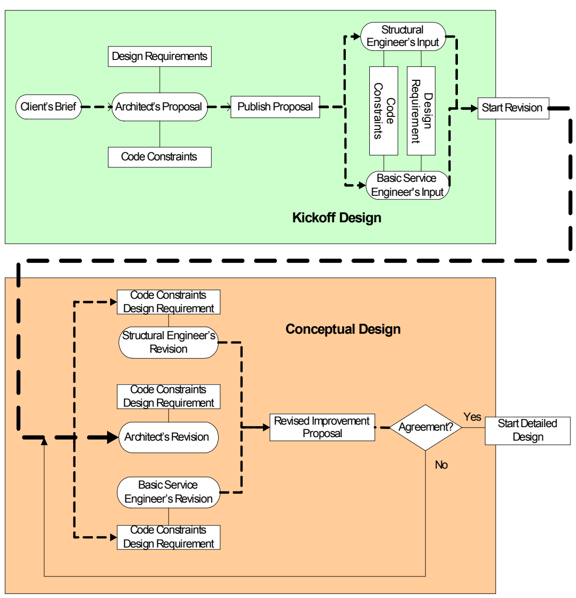

Figure 3, Collaboration model

2.3 Product Model

Supporting the use of existing neutral data formats should be highly encouraged in the development of modern IT systems, so that information can be easily transferred between compatible software products.

Therefore, preliminary strategies for the development of the proposed system considered the adoption of existing product models (IFC, CIS) to be used as part of the system shared product model.

The CIS/2 was initially considered, due to its comprehensive support for steel structures and detailing of the building, and its structural analysis model. However, it did not represent the architectural aspects of the building design, and the non-steel elements of the building could not easily be modelled. The IFC was then considered, as it provides a more comprehensive model. Experimenting with the model identified the following problems when working with IFC classes:

- The model is very generic, and does not constrain the design process. The research is concerned with tying the building design process closely to the information being used, and so this would not be appropriate.
- The model was not designed to support concurrency. In fact, the model developers' main goal was to provide a neutral data format to exchange data between different software and it was not developed to be used as an internal model.
- The incompleteness of the model. Although the model is relatively comprehensive, but the required domain is not yet fully covered (i.e. steel building domain).

It was therefore decided that the best strategy, to allow the greatest freedom within the research, would be to develop a suitable product model that is not based on either of these models but makes use of the IFC and CIS schemas, and structuring the model in a way to support collaboration.

The main focus that was taking into account during the development of the product model was to facilitate concurrent access to the data model. The model is developed as constrained-based model. Its structure allows many designers to work concurrently. The designers are able to provide a range of constraints on their own data. For instance, the architect can set the constraints against the columns position while leaving their cross sections unconstrained. Hence, allowing the structural engineer to modify relevant structural properties without affecting the architectural constraints.

The model was based on previous work [7]. Figure 4 shows the product model structure. It comprises three tiers: Design Intent, Manufacturing Model, and Analysis Model.

The design intent is input, formalised and stored in the first of the product model tiers (Design Intent Model) constituting the "IDEA" (tier 1). The IDEA is composed of the decisions and choices made by each of the building designers. As such, this tier contains the most valuable information in the building's design, and the other two tiers are ultimately a logical development of the ideas expressed within.

A manufacturing model constituting the "PROTOTYPE" (tier 2) is then generated from the "IDEA" using the second process group: Prototype Generation. The PROTOTYPE is the outcome of the total design process and principally includes the physical product such as the steel frame components, floor system and cladding system.

Having generated a workable PROTOTYPE, various processes are applied to test the conformance of the building to the set constraints and the general engineering principles. This is done by generating transient "TEST" analysis models (tier 3). And then by carrying out suitable checks on these models in order to report on the conformity with the set IDEA.

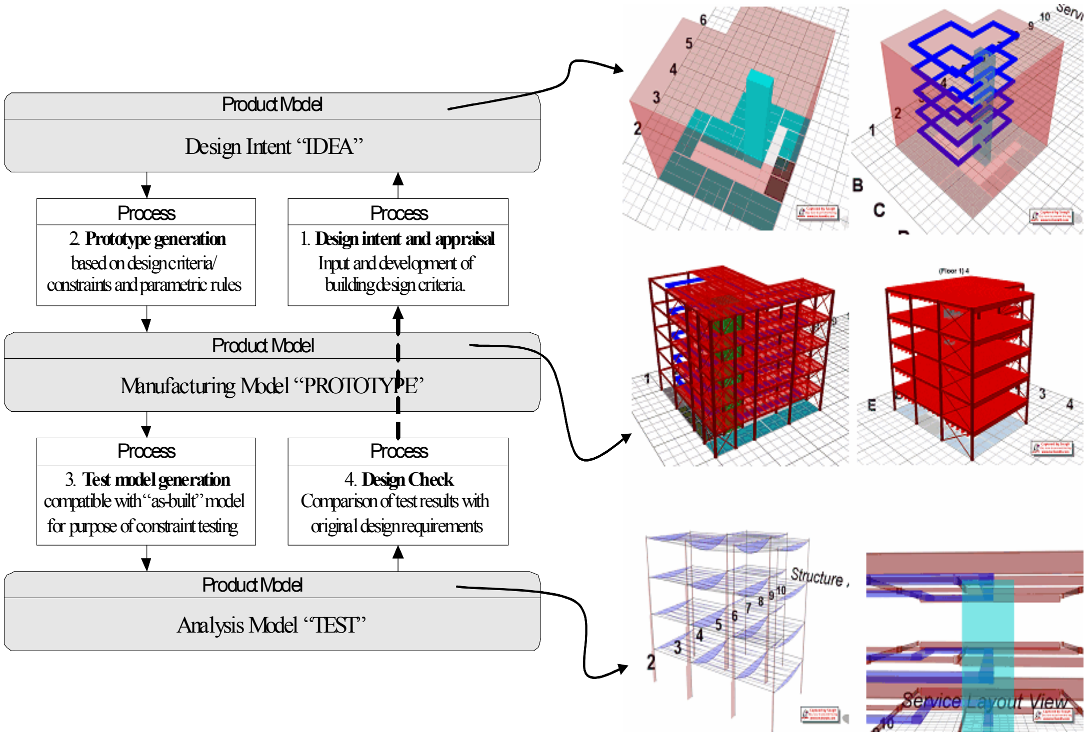

Figure 4: The product model tiers and the VR representations

2.4 3D VR Model

The graphical representation of the design data (i.e. the product model) is vital for engineering design systems. It was necessary that the 3D model to be well-linked to the product model so that the designers are able to handle the product model through the interaction with the graphical model. Figure 4 shows the links between the two models.

The practicality and performance are taken into account when developing the virtual 3D model. It was considered that the model should be able to show the whole model details and to be easily navigated and manipulated.

Alongside developing the VR model, a support mathematical model has been built. The objective of the model is to facilitate vector operations and matrix transformations such as translations, rotation and scaling.

3. IMPLEMENTATION OF THE PROTOTYPE SOFTWARE

The software has been developed using the object-oriented methodology and written mainly in The C# programming language [8]. All processes and product models are implemented in terms of objects logically interconnected. A modular approach is taken to the addition of agents used to carry out the processes of structural analysis, checking and design. The virtual reality interface was developed in C++ and OpenGL (Open Graphic Library)[9] to create a real time dynamic system where all actions upon the model can be carried out through the 3D interface. The communication infrastructure was implemented using ".NET Remoting". .NET Remoting enables different applications to communicate with one another, whether those applications reside on the same computer, or on different computers. .NET Remoting allows agents to communicate either by message exchange or direct invoke method [10].

The software incorporates a real time interface with the ability to simultaneously visualise all aspects of the building's design. It includes the ability to have specialised views for each discipline that combine several different visualisation options alongside the tools appropriate for that discipline. Default views are provided for each of the traditional roles including the architect, structural designer and services designer. Model manipulation is specialised for each of the views so that no information over-load will occur. However, each view can be customised to visualise any of the available options by overriding the default set. So, the Structural engineering view might be customised to super-impose any of the architect detail over the structural detail.

All of the design ideas are visualised in terms of their corresponding manufactured model (which is at all times synchronised with the design intent), so that the interactions between the building components can be better understood. The results from the analysis tier are also visualised within the same 3D interface, which makes manual design checking possible, and provides the building's designers with an intuitive way of visualising the effect that changes have upon the model.

4. SYSTEM SUPPORT UTILITIES

Complementary tools were developed to enhance the collaboration among the design team. These tools are: the system restore tool, the design history viewer and the communication tool.

4.1 System Restore Tool

There are occasions where it would be necessary to revert the model back to an earlier stage of design. Examples of this include cases of unsolved conflicts or the need to follow what-if scenarios and experimentation or simply to back up the model. For that purpose, a restore tool was developed. It allows for the restoration of the design model data to a prior state for whatever reason. The system administrator can specify the restore time of the product model or choose from a predefined restore points. The System Restore Utility automatically rebuilds the product model up the restored point.

4.2 Design History Viewer

The product model may be changed by any of the design members, and each design element may be reached by more than one designer. So from design point of view, it will be helpful to check the history of changes occurred upon any element. Moreover, it is sometimes important to make inquiry to determine the responsibility for any changes. The product model was implemented in away that each design element stores all the changes it has since its creation.

A utility to read and display all history changes of a design element in the shared product model was implemented in the system.

The Design History Viewer allows designers to inquiry about the changes by element name, or type, or date interval. Figure 5 shows the history of a column. Three actions can be viewed a creation and two modifications.

Because the messages exchanged in the system are formatted using XML, which is difficult for human to read, the messages are mapped automatically to human readable text before they are displayed.

4.3 Communication Tool

During the design process, the design team may need to communicate synchronously to sort out any raised conflicts or just for general discussion.

The system offers a virtual meeting facility. It allows a real-time interaction among those who are on-line. Designers can directly contact each other to ask questions or discuss various topics.

Besides the ability to exchange instance text messages, the designers are able to remote their design view screens which make the discussion more feasible.

Figure 6 shows a scenario of text message exchanging between two designers with a remoted view.

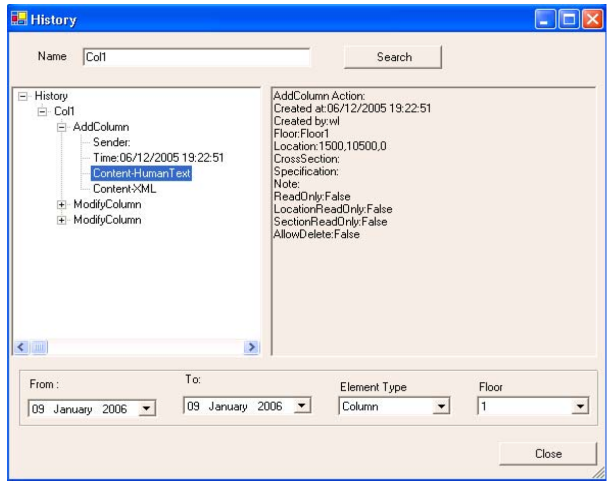

Figure 5: History changes of a column

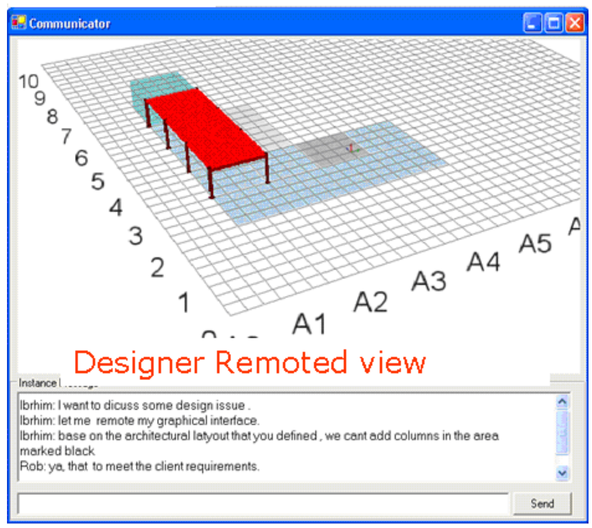

Figure 6: Virtual communication

5. OVERVIEW OF A TYPICAL DESIGN PROCESS

The design process makes no assumptions concerning the roles of actors, though it is helpful to consider the design activities in terms of the actors traditionally involved in those roles. As such, the activities within the design process developed in this model can be divided between the client, architect, structural engineer and services engineer. A typical collaborative design scenario, that may be followed using the software, is described below.

The client starts the design process by defining the building's overall requirements. This includes the building's target cost that may be subdivided between the structure, services, flooring and cladding (Figure 7). The client is also responsible for setting the total floor space required within the building as a whole. At all times during the building's design, the costs and areas are automatically calculated and compared with the client's requirements. All building design participants will be notified in case the actual figures fail to meet the targets.

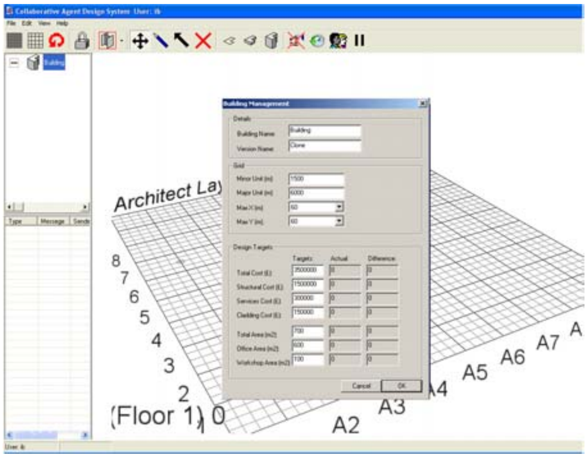

Figure 7 Building management dialog box

The architect then specifies the layout of the building's perimeter, the number of floors, and the internal height requirements for each floor (Figure 8). The height of any given storey of the building can be given in terms of the non-structural floor depth, clear depth and the ceiling depth, which includes the services and structural depths. This defines a set of requirements for the structural and services design to conform to, and sufficient information to specify a general model of the building in terms of size and spacing.

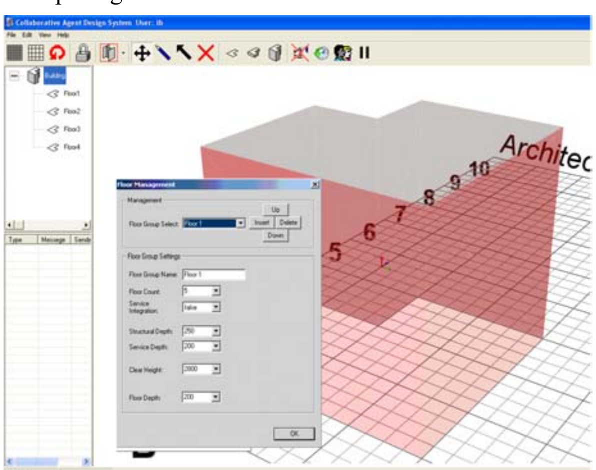

Figure 8: Definition of building boundary

The architect is responsible for specifying the floor usage for each part of the building (Figure 9) and the cladding types assigned to each part of the building's perimeter. The cladding input provides the data to calculate the claddings cost, which will then be compared with the clients brief. It also provides an early indication as to the external look of the building, which may be useful for assessment of early prototypes by the client. In terms of the building's structure, the cladding information also provides data regarding its loading requirements that will then be considered into the structural analysis.

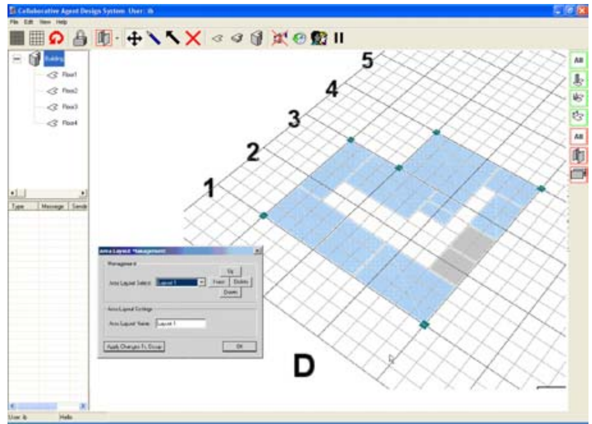

Figure 9: Definition of floor plan and usage

The architect may also provide information regarding extra loading areas within the building, such as around building cores so that the appropriate loads will be added to the building's structural analysis model. The architect may also designate column spacing areas within the building, in order to restrict the structural engineer from positioning columns inappropriately; or positioning the columns himself and constrain them to be repositioned and optionally free their cross sections to be changed by the structural engineer. The architect is also responsible for specifying cores in terms of their location and designated purpose (Figure 10). This information will be important for use by the structural engineer and the services engineer.

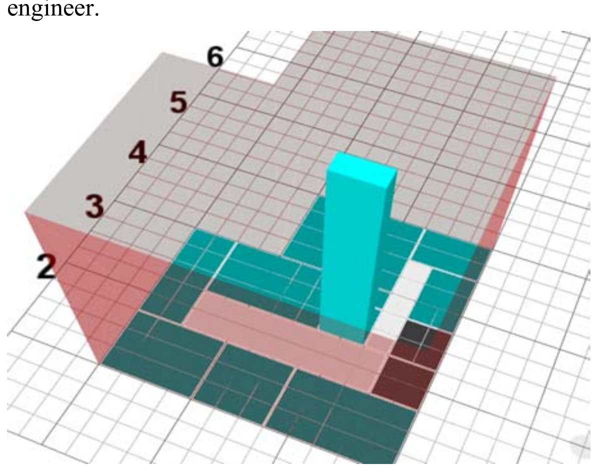

Figure 10: Definition of building cores

The architect's design input, therefore, has a far-reaching effect on the building's overall design. It must meet the client's requirements in terms of the spacing requirements, while setting forth the requirements of the structural building services engineers.

The structural engineer's first task is to specify the column positions within the building if not specified by the architect (Figure 11). The height is not specified, as this is automatically calculated by the Design Agent based on the architect's settings. It is also not necessary to specify the dimensions of the columns at this stage, as they may be calculated automatically from parametric rules by the Design Agent and then later refined based on the columns structural response. The next step in the structural design is to locate the primary beams and either specify their dimensions or, as with the columns, allow the Design Agent to automate this process (Figure 12). The connection detailing between the primary beams and columns is optionally generated automatically, and may be redesigned later when structural analysis information is available. Bracing is similarly specified using simple connections (Figure 13). 'Floor areas' are then specified, which allow for the designation of the secondary beams and structural floor in one action (Figure 14). As with previous structural design stages, any or all aspects of the floor may be specified by the engineer including the secondary beam dimensions, spacing, orientation, and the floor type and floor dimensions. The detailing for the remaining design aspects, if any, will then be calculated automatically.

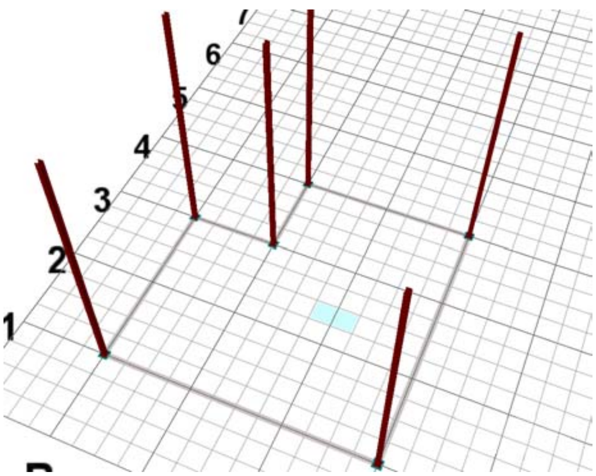

Figure 11: Structural: positioning of column grids to suit architectural constraints

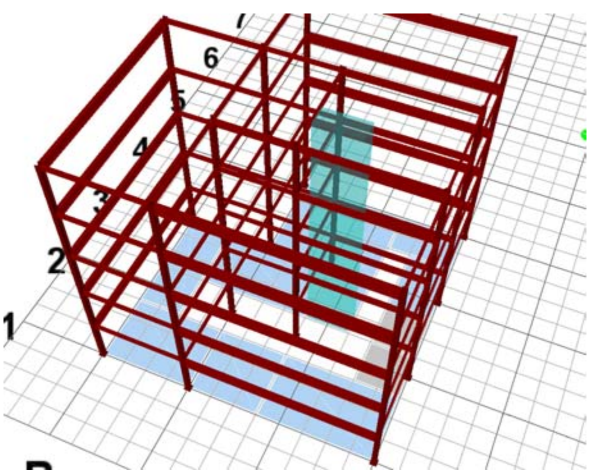

Figure 12: Structural: propose framing system

The structural engineer is therefore able to commit to any level of detail that he feels necessary, and this allows him to leave the less globally influential design issues to the software argents that are better suited to this task. As a result of these actions, data is generated concerning the self-weight of the structure and the cost of the building elements for use later on in the design process.

The services engineer is able to specify the location of service entry points to the building, the location of service cores and the main service routing in plan around each floor (Figure 15 & 16). The height of the actual service ducts is based on the service strategy chosen, and will either be positioned beneath the structural layer or within the structural layer. In the case of the latter, the beams will be appropriately notched, and this will be taken into account in the building's structural analysis and costing.

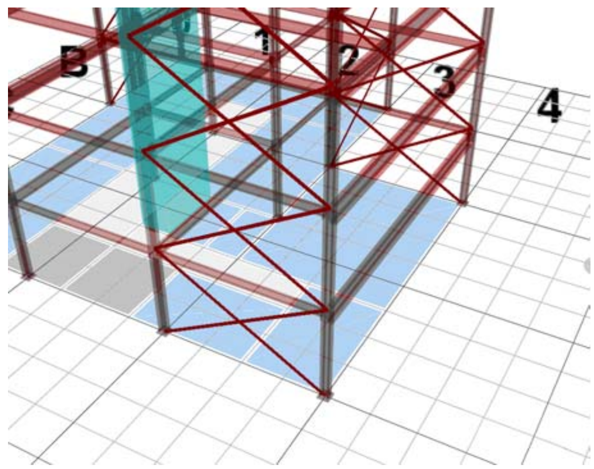

Figure 13: Structural: Devise structural bracing system

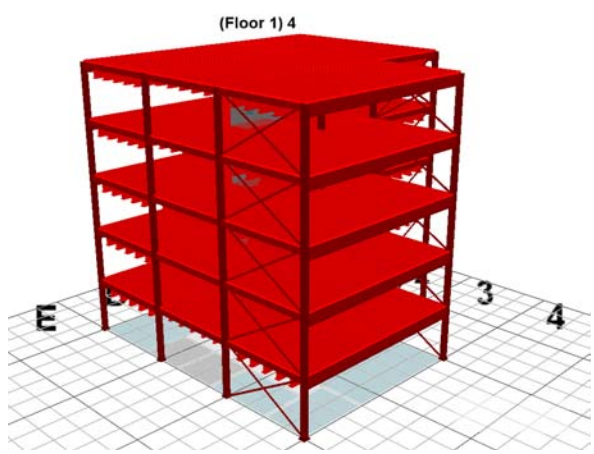

Figure 14: Structural: Complete structural system prototype

With the information provided through these processes, there is sufficient data to produce a 3D structural analysis model of the building (Figure 17). This is generated automatically by the Structural Analysis Agent to accurately model the members and connections with the appropriate use of analysis members and dummy members to model connection offsets, hence maintaining compatibility between the as-built structure and the analysis model. Loading information is also automatically generated from the data previously input by the architect, structural engineer and service engineer, in terms of factored dead and live loads. The analysis is carried out to produce the structural analysis response model (Figure 18). This information is used later to perform a design check on the structural members, which may then be used by the structural engineer or automated processes to iteratively improve the building's design.

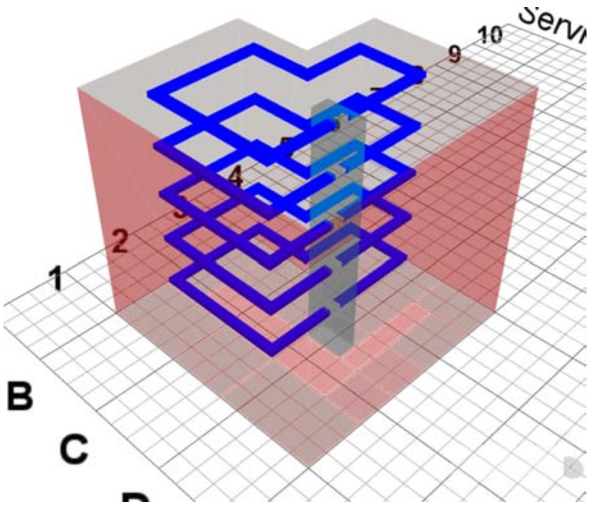

Figure 15: Services View: Adding services duct connected to services core

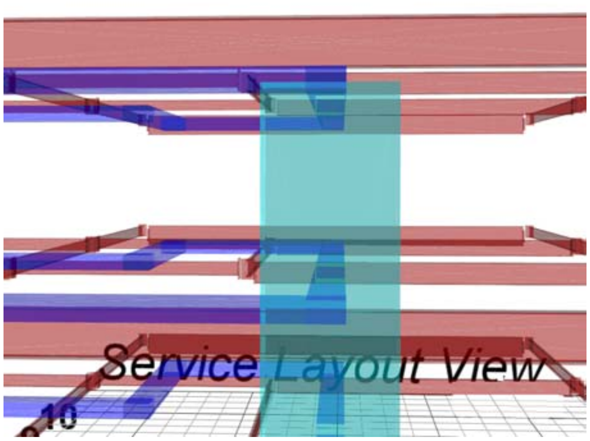

Figure 16: Services/Structural: Checking structural floors - services integration

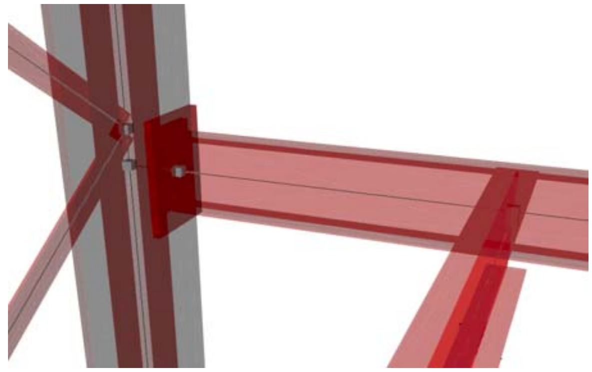

Figure 17: Structural: General structural analysis model compatible with as-built details

Figure 18: Structural: Structural analysis results (bending moment, shear force and deflected shape)

The information above has given an overview of a typical design process. Although, the design process may seem sequential, the design team can work concurrently all along the design process. They can view and inspect the model at any time. They also can exchange their views of the design and be notified by the actions of the others. For example, the architect may decide at a later stage to add an extra floor to the building by modifying the number of floors in the floor management dialog. Without any work on the part of the user, the columns, primary beams, and floor systems are re-generated in-line with this new requirement. The increase to the building cost can be immediately appreciated, and it can be seen that the column dimensions have adjusted to take into account the new loading. The other actors will be notified to either to agree on the changes and do further check upon the model or reject the changes and communicate to reach a consensus.

The information describing the building together with all relevant input is stored in the shared product model. Partial building designs may still be carried by concentrating the work on a single discipline. For example, a structural engineer can input a steel frame and carry out analysis and design without the availability of architectural constraints.

6. SUMMARY AND CONCLUSIONS

The complexity of the design process may seem to result from the complexity of the building itself but it is rather more due to complexity of design interrelation between the various disciplines involved in the design process.

Collaboration among the design team has a significant effect on the overall quality, cost and operation of a construction project over its life cycle. Communication since the early stage of the design helps the designers to have a comprehensive understanding of the design requirements, which reduces design conflicts in the later stages. Many researchers argue that collaborative and concurrent design environments are the future of design systems.

Technologies, particularly those for distributed applications, have widened the scope of collaboration. Today, collaboration includes virtual workgroups that bring people together virtually via telephone, email or videoconference, essentially reducing distances and the necessity of physical interaction.

The complexity of the design process can be generally reduced by providing 3D intuitive representations of the information.

This paper outlined the development of a collaborative design environment for multi-storey steel structures exploiting the latest information technologies for visualisation and communications. The 3D intuitive interface assisted the designers to view and manipulate the shared product model, and the design automation and communication were carried out using software agents.

The investigation outlined in this paper showed that the latest advances in information technology in terms of processing speed and supporting distributed application together with VR technology can deliver collaborative design systems that can handle the complication of both communication and engineering design.

7. REFERENCES

[1] Anumba,CJ, O O Ugwu, L Newnham, A Thorpe, 2002, Collaborative design of structures using intelligent agents: Automation in Construction, v. 11, p. 89-103.

[2] N.F.O. Evbuomwan, C.J. Anumba, Concurrent life cycle design and construction, in: Developments in Computer-Aided Design and Modelling for Civil Engineers (B.H.V. Topping). Civil-Comp Press, Edinburgh, 1995, pp. 93–102.

[3] www.iai-international.org. International Alliance for Interoperability (IFC).

[4] http://www.cis2.org/ CIMsteel Integration Standards.

[5] Parunak. What can Agents do in Industry, and Why? An Overview of Industrially-Oriented R&D at CEC. 1998.

[6]W. Shen, D. H. Norrie, and J. Barthes. Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing. Taylor & Francis Group, 2001.

[7] Smith, R. A., and W. Tizani, 2004, The role of VR in multi discipline building design: The 4th conference on Construction Applications of virtual reality, CONVR 2004, p. 173-178. Lisbon, Portugal.

[8] http://msdn.microsoft.com/vcsharp Microsoft Visual C# Centre.

[9] http://www.opengl.org. The Industry Standard for High Performance Graphics.

[10] http://msdn.microsoft.com. .NET Framework Developer's Guide. .NET Remoting Overview

[11] M.Wooldridge,NRJ. Intelligent Agents: Theory and Practice, 1995. 1995.