Integrated Collaborative Building Design Using Internet Technology Solutions

Dilan Roshani

School of Civil Engineering
The University of Nottingham, Nottingham, UK
Email: DRoshani@gmail.com

Abstract: The issue of communication between the disciplines involved in a construction project has been the subject of research activities for many years. The construction industry has a long tradition of collaborative working between the members of the construction team. At the design stage, this has traditionally been based on physical meetings. This paper considers the shortcomings of the current state of the Architecture, Engineering and Construction (AEC) design process and gives an overview of the design, implementation and evaluation of a COllaborative Design Environment (CODE) system. The system's facilities allowed for active human-centred collaboration, and confirmed that the Internet is a feasible and accessible technology for hosting multi-disciplinary and distributed collaborative design work.

Keywords: *Collaboration, Integrated, IT, AEC, Building Design, Product Model, Internet, IFC*

1. Introduction

In today's global market, the Architecture, Engineering and Construction (AEC) operations are spreading around the world. Not only marketing and production, but also design and development are influenced by globalisation. New challenges are emerging constantly, which require global solutions for geographically distributed building design and construction. Due to the global nature of the current building construction and the evolving impact of IT on the design process, the AEC design project activities are continuously changing. These activities require support with communication, interaction, integration, design facilities, and shared up-to-date design specification. IT facilities such as the Internet can potentially provide integrated collaboration environments satisfying these requirements.

The AEC design processes considered as highly fragmented so the responsibility for design and construction is often broken into many small parts handled by different actors. At the same time, the quantity of data to be dealt with is constantly growing. Building design often requires collaborative working between members of geographically distributions design teams. Therefore, the need for effective information and communication technologies becomes necessary [1].

Current AEC industry depends increasingly on communication and cooperation. More and more, AEC actors need to share resources through distributed computing and databases, shared data information and global collaborations. Independent survey results have been released to support the feasibility and the impact of IT on the AEC industry, which in some cases show more than 90% improvement in the quality of work, as well as 77% increases in team productivity. These included lower cost management of building projects, deliveries and travels [2].

The future of the construction industry is evolving towards using an integrated product model where sharing all project data at any phase is possible. A product model is a formal description of ideas and facts, with rules, which together form a "simplified yet complete, accurate, logical and computer sensible representation of physically realisable object" [3]. The design process of a product might be explained as a systematic, "intelligent generation and evaluation of specification for components whose form and function achieve stated objectives and satisfy specified constraints" [4].

An electronic, integrated, shared, and Web-based building product model enhances data exchange and facilitates the development of a collaborative process model. A solution as such through platform independency and remote real-time access to data models, will offer advantages to all parties involved in the design process. Implementing such a solution based on Internet technology will provide the opportunity for a radically changed and much more efficient communication process. Considering the highly complex data and communication needs, the collaborative engineering world has developed some facilities that are mostly its own. This primarily involves rendering and analysis tools and rather complex intelligent virtual reality devices, as well as similar tools that use the Virtual Reality Modelling Language (VRML), Java 3D Modelling or Vector Mark-up Language (VML) to interact across the Internet and Intranets.

Since the advent of global World Wide Web (WWW), a number of frameworks have been proposed for Web-based AEC collaborative design systems, but most of them remained as proof of concept prototypes or never made it to commercial market [5]. Research into AEC collaborative work using Internet technology has mainly been focused on sharing documentation data. The potential of the Internet providing an integrated shared environment has not been fully explored. This study has evaluated over 25 commercial and non-commercial systems, such as Web-EX, InetCall, AutoTask, Columbus, SHARE, OSCONCAD, [5] etc to identify whether integration, shared product data model, and real-time interaction, have been considered. Some research work such as [6], [7], [8], [9] and [10] have tackled the issue of collaboration but they have not considered the integrated process of a standard shared product model structure such as Industrial Foundation Classes (IFC). The IFC platform specification provides data structures for the AEC industry shared project model [11].

The research work presented in this paper implemented a Web-based COllaborative Design Environment (CODE) for coordination, collaboration and effective management of conceptual design. The work is aimed at investigating the feasibility and likely impact of such a Web-based integrated system on AEC collaborative design activities. The following sections give an overview of the design, implementation and evaluation of the CODE system.

2. The CODE system

2.1 The CODE system requirements

The study reported here has highlighted that any integrated collaborative system needs to address a number of requirements that are necessary to ensure a functional and feasible system for real-time collaborative work. The requirements that are established to support such an implementation are:

- provision of user friendly, intuitive and interactive interface.
- provision of electronic communication facilities, such as e-mail and messaging system.
- provision of real-time interaction facilities that provide different interaction modes such as synchronous and/or asynchronous and presence awareness of other actors in the environment
- provision of tools for the project management of the design process allowing for the interrogation of the state of the design and controlling the input of the multi-disciplinary team.
- use of knowledge management techniques to capture and reuse shared knowledge, previous comments, and solutions
- use of standard data format for a shared and real-time product model
- conduct of design tasks through platform independent Web-based design tools, such as visualisation, design calculation and simulation
- handling of security and privacy issues

2.2 The CODE system architecture

The CODE implementation uses client/server architecture. A client is defined as a requester of services and a server is defined as the provider of services. In order to support collaboration, Web based design servers need to communicate with the product data of the design representation in a way that user can put queries about formal design concepts and data information. To facilitate a viable design environment, Web based design servers must also engage users in an active interaction regarding shared product modelling, design representation, design solutions, and project information.

Figure 1 gives an overview of the CODE system architecture. The system includes a relational database and a series of portals on the server side. The user interfaces bridge the interaction with the system through a standard Web-browser.

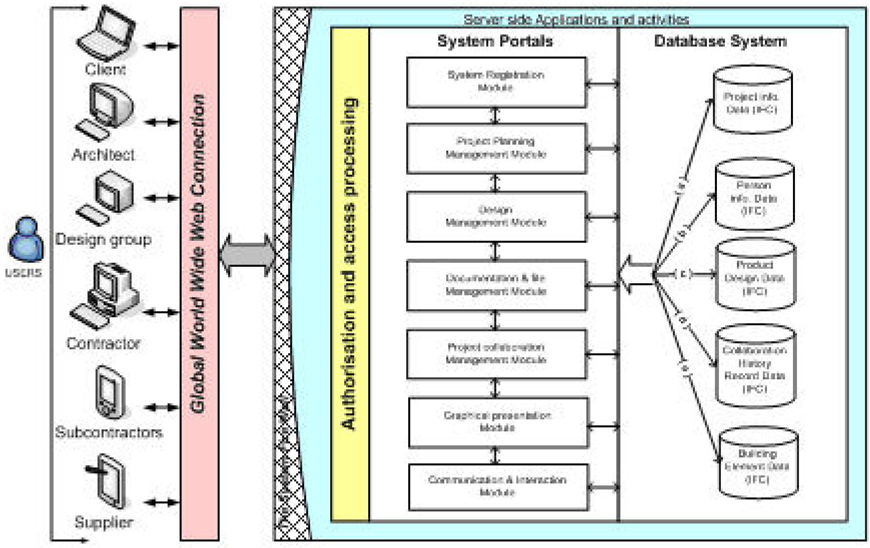

Figure 1: CODE system architecture

This Web-based system architecture promotes better solutions when compared with other applications for a variety of reasons, such as:

- Centrally hosted application eases software-maintenance and better control version requirements.
- Initial and maintenance expenses for software applications are significantly reduced with browser-accessed systems.
- Browser-based access significantly reduces the need for specialist end-user training.

The realisation of the CODE system architecture followed three major design and implementation operations: definition of use case scenarios, design and implementation of the system database, and the design and implementation of the portal system. The implementations involve resolving a number of issues related to:

1. Establishing authorisation of access control/access rights
2. Specifying design administrator's control
3. Ensuring the inter-independency rules of each design discipline by creating logical constraints.
4. Uploading documentation into the project
5. Managing an object library components
6. Controlling history changes, who does what and when!
7. Facilitating interactive communication
8. Applying, sending, discussing, and approving changes

9. Facilitating response to changes
10. Providing automated notices and communication

The authorisation and access process of the CODE system is based on the concept of five hierarchy layers, Figure 2. Each layer provides predefined constraints based on the Project Code, User ID, Password, and project owner's group allocation. These will guide the actors to the specific project space.

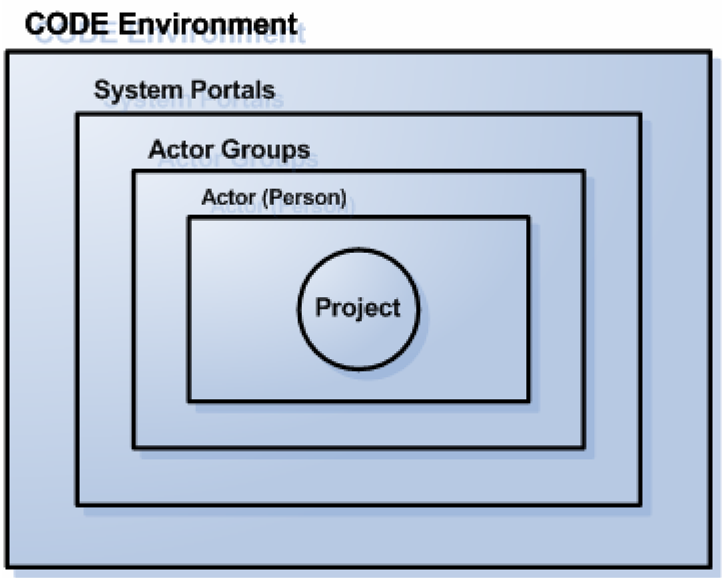

Figure 2: CODE building project in the centre of a multilayer system [1]

The idea is to organise the access rights of a project based on the design responsibilities of the designers. A project owner will be assigned, normally the client, who will in term forms design groups (Actor Groups) with appropriate access rights matching these responsibilities. The owner can then assign individual designers (Actors) to the appropriate groups. Actors can be members of more than one project and possibly having differing access rights for each. Such approach should allow the designers to follow a process that commensurate with their intended roles and interact appropriately.

The interaction between a user and the system is controlled through the layers. The management of the interaction in the CODE system is illustrated further in Figure 3. The system distinguishes between different access levels through a predefined identification process of Actor Groups' layers. The system authorises access through design, non-design, and read only group activities.

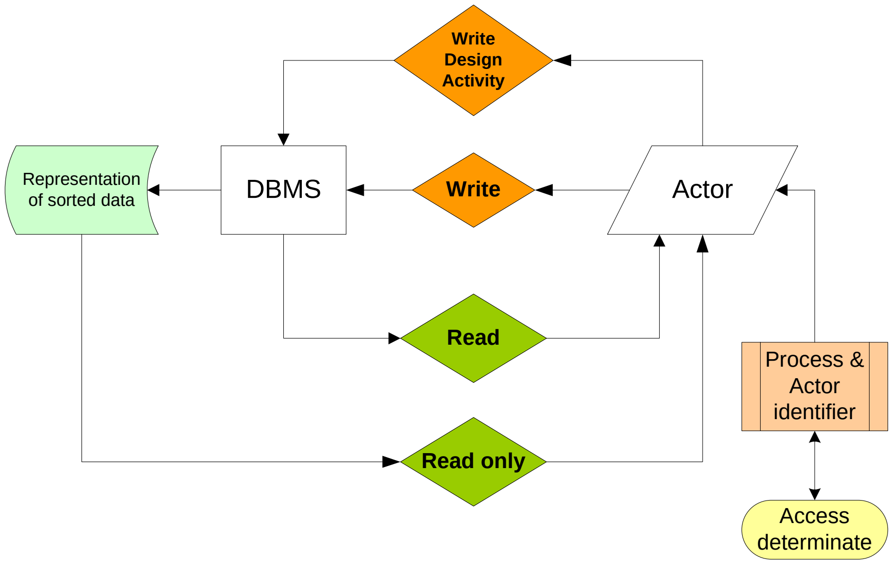

Figure 3: Database Management System DBMS relation schema in CODE [1]

2.3 The CODE system analysis

The schema of the CODE database management system (DBMS) is based on the global user cases defined by the Actor Groups constraints for a particular project. Figure 4 illustrates a UML use case scenario which defines users' activities of the CODE's DBMS. Each use case is made up of a set of scenarios. Each scenario is a series of interactions between one user and the CODE system. The use case of the CODE system brings various scenarios together that carry out a specific goal of the user in the system. A project set-up scenario is described next [1].

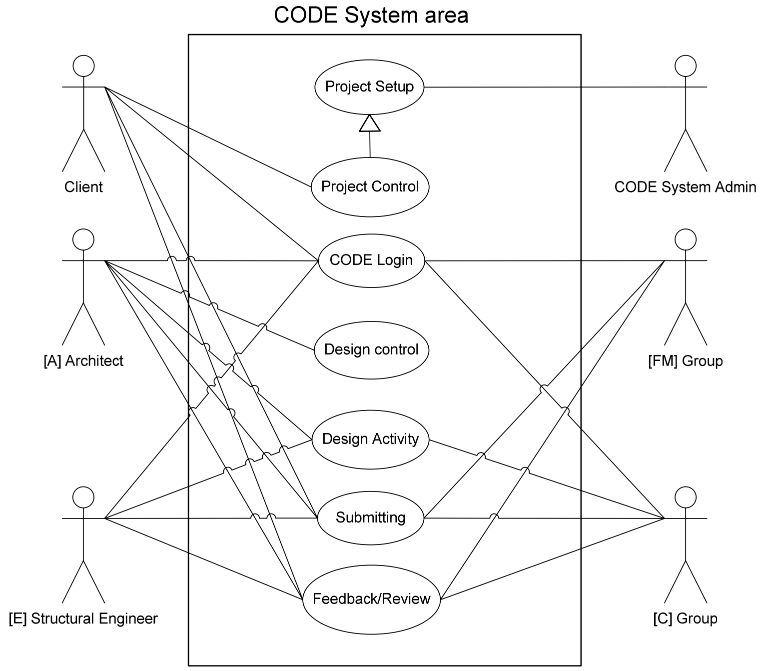

Figure 4: CODE use case diagram for Web server activities

The scenario starts with the CODE system administrator approving an application for a project space from the project owner (shown above as the Client). The process ends if no approval is given. The owner will setup the project properties, which will provide facilities to accommodate the project activities at all levels. The role of the owner might also be played by the Architect.

The CODE system administrator will provide the owner with the necessary tools to invite his design teams. The team members access the CODE by registering with the project space. Each approved registration involves group allocation, client note to the actor, and password setup. The actor will receive an e-mail with access details to activate her/his account. The owner will have absolute control over all activities at the project level [1].

The interaction between the CODE system's users and their specific project space starts immediately after the activation process. Figure 5 illustrates a UML sequence diagram providing an overview of a typical interaction between a user with design access and the CODE system. The sequence process shows how the System Interface facilitates the major connection between the user and the product data through system modules. The interactions take place simultaneously with no delay, unless the actions are dependent on the response of the team members.

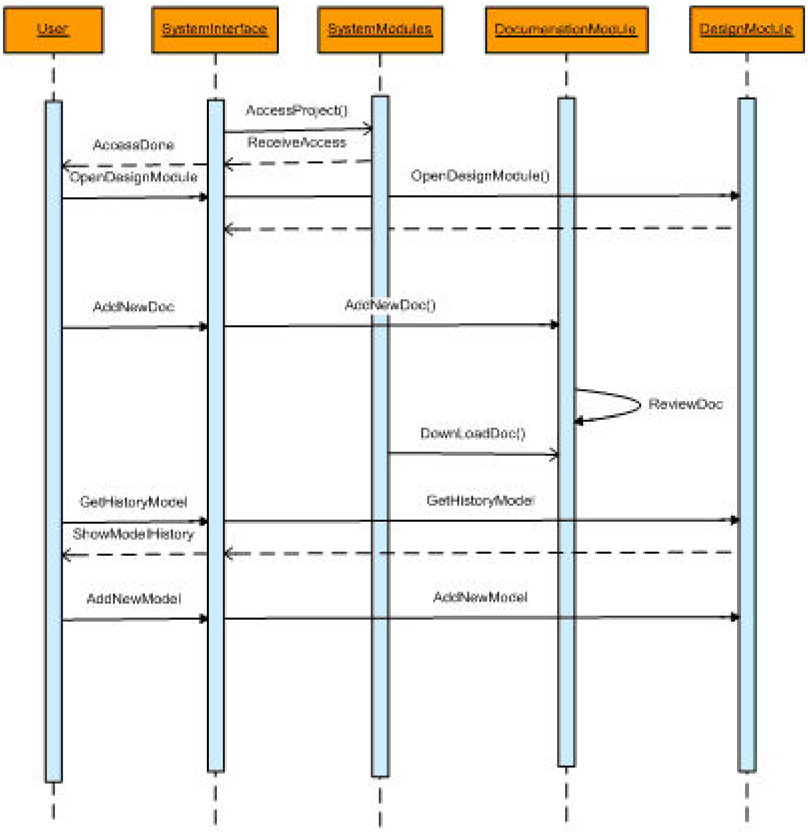

Figure 5: The interaction between the user with design access and the system

There are at least six different high level operational activities in the system, such as login to CODE system, entering project phase activities, conducting collaborative actions, interacting with other members, and reviewing materials and data as well as modelling and sharing. Figure 6 shows an overview of such an activity process.

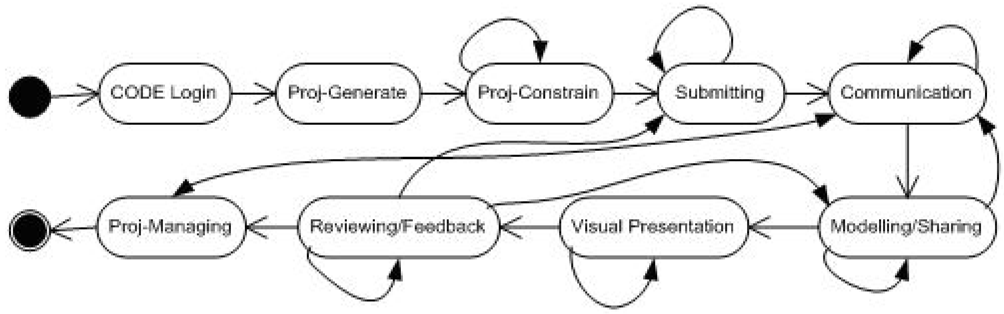

Figure 6: The CODE project activities in an UML Activity diagram

Such an activity process in the CODE system will identify the actor (person) as an object related to a Project, Group discipline, Building Elements and, connected to any related activities in the project space. This will provide a detailed design chain activity in the system, which in turn puts the project model in the centre of a human connected to a project activity and a project connected to human activity. This is defined as human-project-human working activity.

2.4 The CODE database design

The CODE provides a dynamic Web-enable data model. The data model used is independent of hardware or software limitations. The CODE concept will not represent the data as a database would see it, but representing the data as it is appearing in the "real world". The idea will bridge the concepts that make up real-world activities and the physical representation of those concepts in the form of a database and Web-portals [1].

The goal of the data model is to provide a reliable data information for real-time accessibility and to make sure that all data objects required by the database are represented fully and accurately in a widely accepted standard structure, such as IFCs [12].

The CODE database system uses the data specification known as the IFC's Object Model, which is defined using a top-down approach, Figure 7. The concept starts with a very general view of the AEC design approach, where an overall model of a building can be defined in the system and successively worked into a detailed model. The system uses the IfcObject definition of a model to provide its shared product model. The IFC concept describes the behaviour, relationship, and identity of a component object within a model. Figure 7 shows seven elementary entity types in the IFC model definition, which are all subtypes of IfcObject.

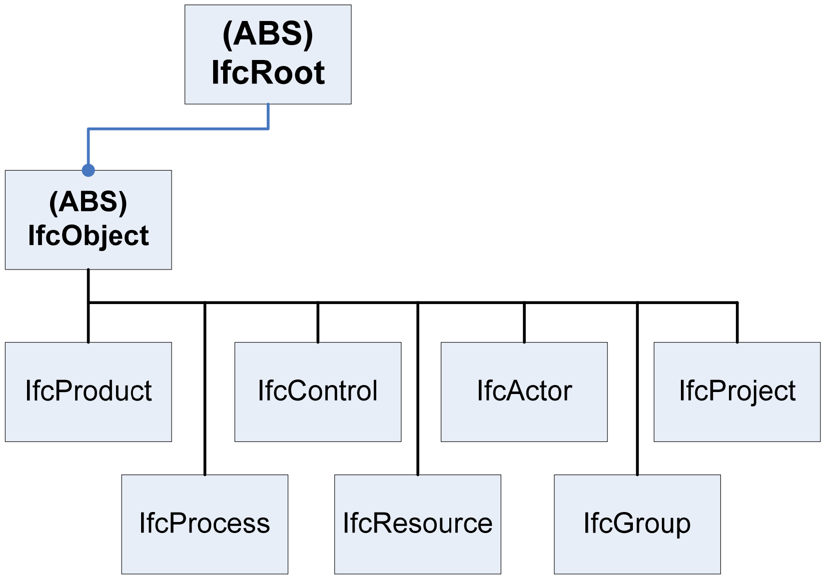

Figure 7: Definition of the IfcObject subtype of IFC classes (ABS: Abstract) [13]

The IFC object model is the core of the CODE system's Database model. There are elements of every subtype entities represented in the CODE's database model. These entities are carefully selected to provide a functional prototype of IFC's structure for the CODE Web-based concept [1]. The model is an Entity Relational Database (ERD) that is developed using Microsoft Access application. The ERD model provides an added data structure thought necessary to manage design collaborative activities in the CODE's Web-enabled system.

There is no inheritance mechanism in Microsoft Access database application, which makes it difficult to take into account the considerable amount of inheritance hierarchy in the IFC model standard. The documentation for any of the IFC entities show that they are inherited from other entities, therefore some of these entities appear only as attribute in the main entity of the CODE database solution.

The CODE database solution adds to the IFC model additional attributes in various entities to overcome the limitations of the IFC data structure for Web-based activities. The CODE proposed entities are prefixed with "CODE-". This will help distinguish between the CODE and the standard IFC attributes.

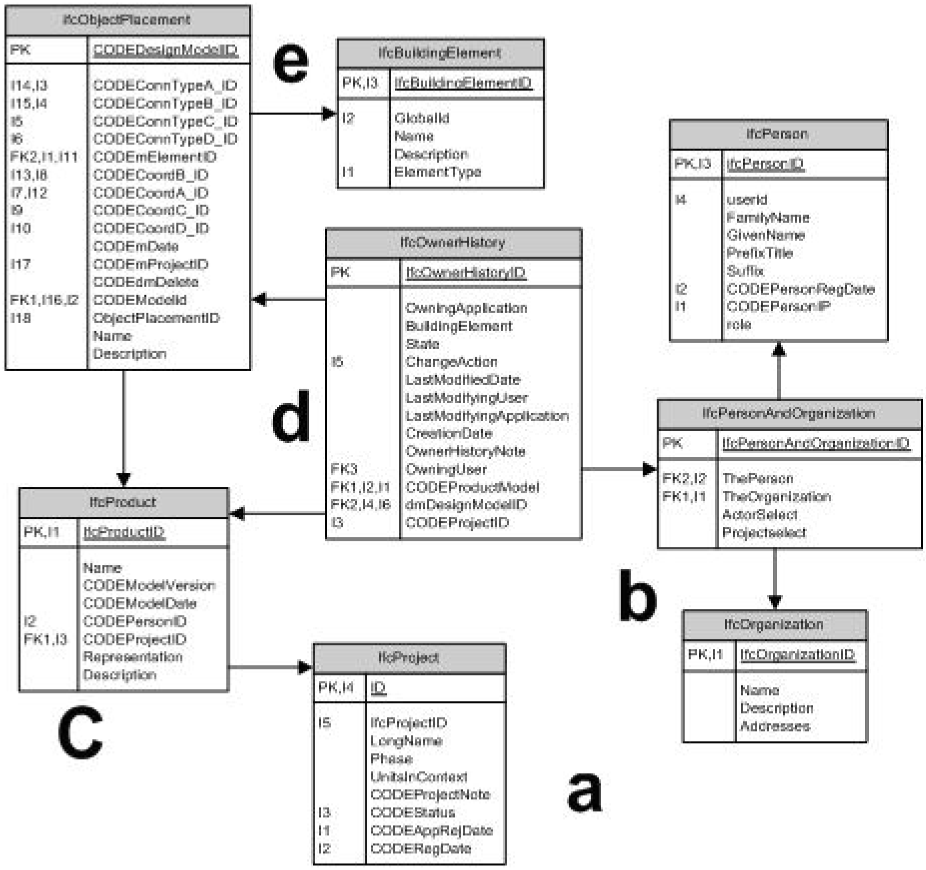

Figure 8: Overview of five major sections of the CODE database system

The database structure is arranged to cover the main data information which represents a workable shared product data model. The aim of the database is to fulfil the criteria of the preset requirements by this study. Figure 8 shows an overview of the structure of the database. It is divided into five major areas:

a. Project information data represented by IfcProject entity.
b. Person/User information data provided by IfcActor, IfcPerson, IfcOrganization, IfcGroup, and IfcControl entities.
c. Product model design data structured as IfcProduct entity.
d. Collaboration work record data retrieved by IfcOwnerHistory, IfcProcess, IfcResources, and IfcConstraint entities.
e. Building elements data collected in IfcBuildingElement, and linked to design activities through IfcProduct and IfcObjectPlacement entities.

Each of the major section in ERD model (Figure 8) has more detailed data structure with related sub-entities. For example, the building components are defined as a global components library based on IfcBuildingElement concept, as shown in Figure 9. This library functions as a central databank of all building components needed for the creation of a building model. The library stores active components from any disciplines, along with their attributes. The idea of implementing the global component library reflects the reusability of the components by other projects.

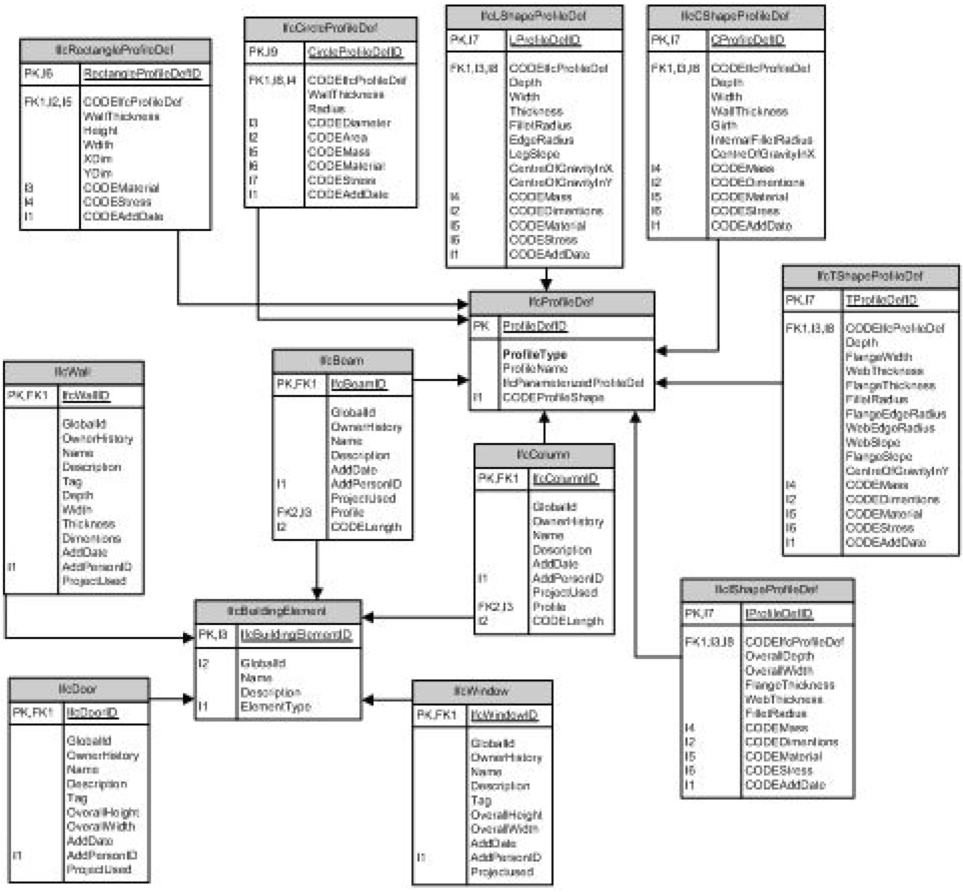

Figure 9: Object Components database model in CODE system

The implementation of the building components also provides the possibility of manipulating the product data model in the form of individual components. The project model data are stored in the database, which contains data information for all related AEC disciplines' activities. This is made possible based on the concept that any activity related to a building component is connected to the graphical representation of that particular component in the project space.

The database implementation allowed for:

- The linking of the components' data to graphical representations
- The recording of designer actions
- The recording of attributes of uploaded and downloaded external resources (mainly documents) and linking them to actors and related project activities

2.5 The CODE portal systems

The portals are represented in the form of individual ASP scripted pages with client-side Web interfaces. The ASP is a Web-oriented technology that is designed to enable server-side scripting environment. It can be used to create dynamic and browser independent Web pages or build Web applications. VBScript, which is based on the MS Visual Basic, is embedded as a small program on the server side ASP pages that are interpreted and executed by the Web client. The MS Access functions well with ASP and VB Script and is easily uploaded to the server.

Each portal has been designed to provide a set of functions to cover the preset system requirements. The portals are represented as a set of modules with specified functions, Figure 10. These modules perform sets of functions, which enables actors to process a collaborative work, see Figure 11.

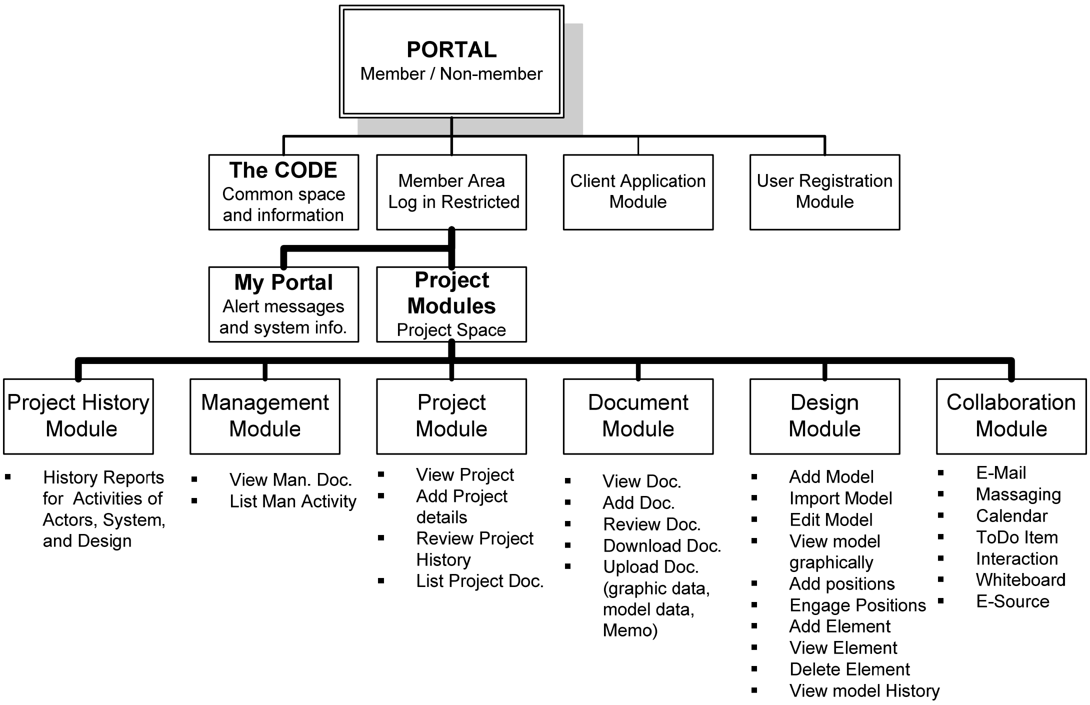

Figure 10: The structure of the portals in the CODE system architecture

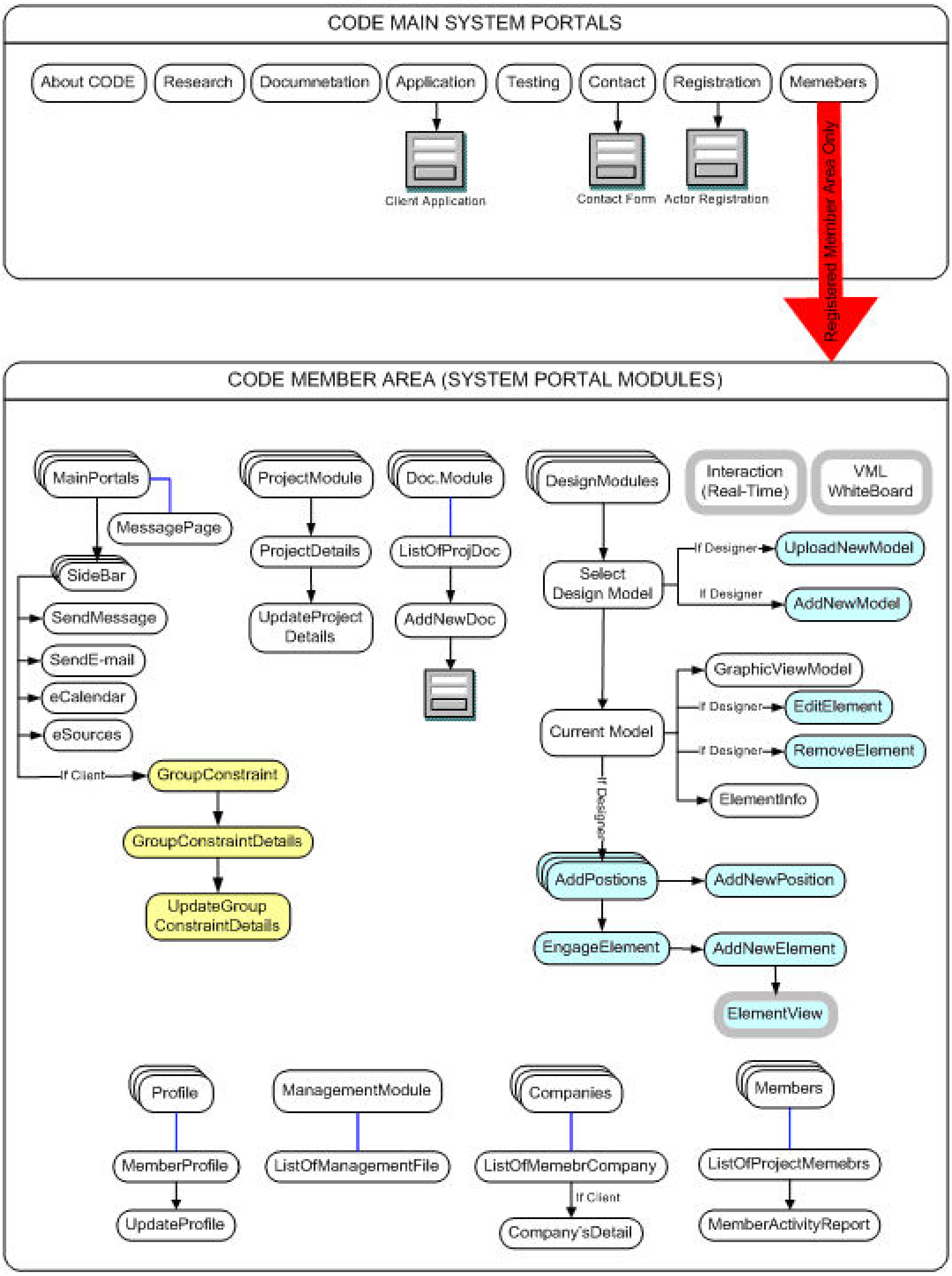

Figure 11: The map of the CODE main Web interface system portals

Figure 12 provides a screenshot of the main portal where the users informed about name, role, project and the group level constraint. The access point is providing an exclusive personal portal to every member based on their disciplines. It also shows how the portal's Web interfaces are organised for an actor with design rights.

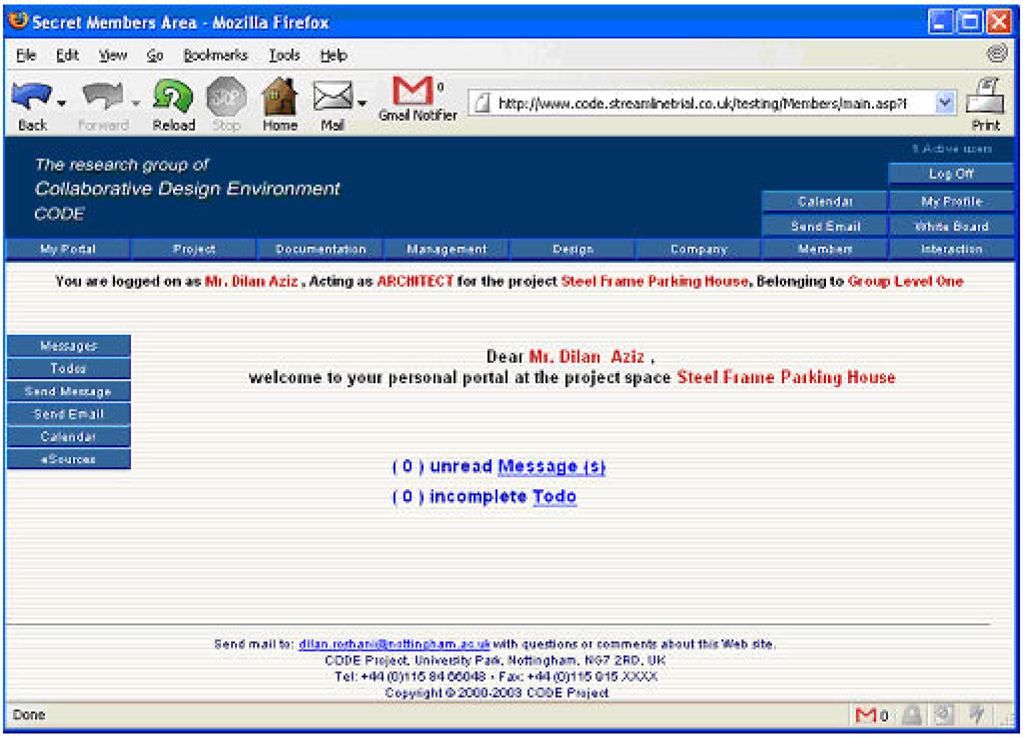

Figure 12: A screenshot of initial personal portal page

2.6 The CODE system Communication and Interaction

The CODE system supports both real-time and delayed interactions. The portals are felicitated with messages, to-dos, interactive chat, calendar, and whiteboard tools to support interaction. The system records the actor's interaction activities in the project space. The owner will be able to monitor the activities of the actors, control the project management, and prevent project delays. This will also enhance knowledge management through sharing solutions and past experiences. In addition, the CODE system engages users in a dialog like interaction that covers a range of activities, such as product design modelling, design versioning representation, and user interaction. Such a scenario is shown by Figure 13 where members of design team are discussing changes to an element in the model. The Interaction through CODE environment is part of activities of virtual collaborative work as in traditional working process.

Real-time interaction is supported with electronic meeting and conversational facilities allowing the actors to communicate and exchange views. This tool lists all the online actors in the project space. Each actor is able to participate in the interaction at any time. In addition, the messaging system with e-mail support enables the actors to send notes in the project space.

The graphical presentation concept of the CODE system provides another tool to assist in visualisation and interaction. The graphical presentation is based on a VML (Vector Markup Language) solution that is generated from the data model values through an isometric projection. This allows the viewing of design solution within the internet browser and without the need for an additional plug-in. The graphical representation of any design element is also made into a hyperlink to its design data. This will allow for inspecting and editing of design data of any component by clicking on its graphical representation.

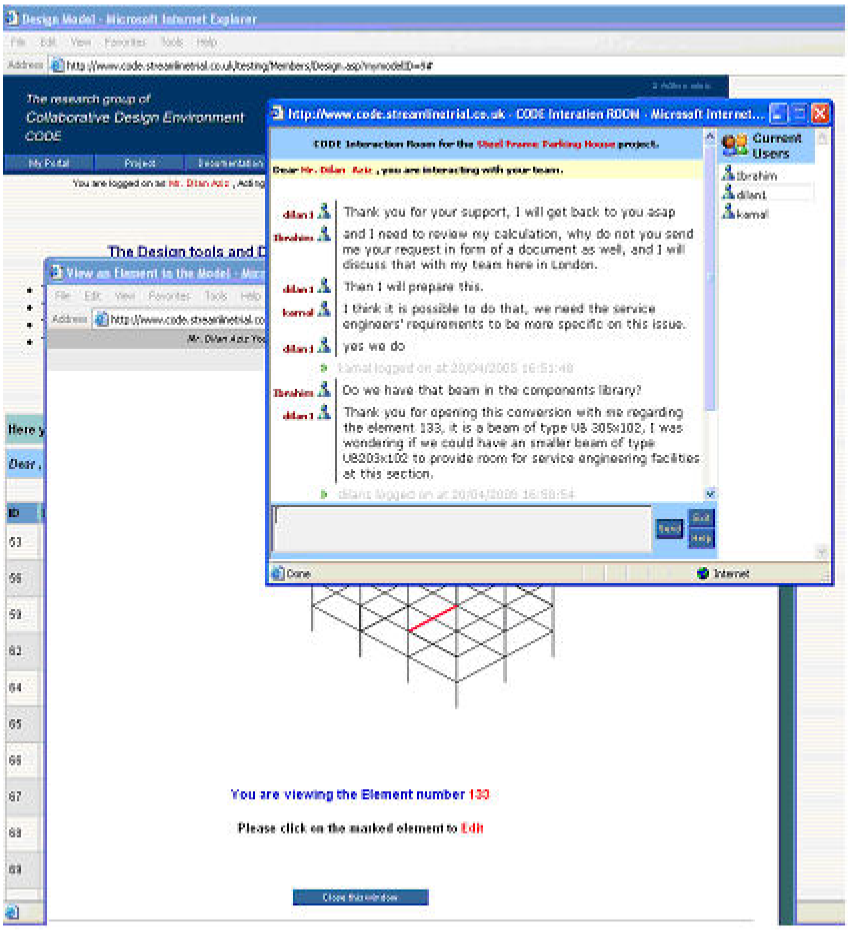

Figure 13: Screenshot of real-time interaction between actors of the project

2.7 Evaluation of the CODE System

A testing design project was used as a case study in order to evaluate the CODE system. The case study was the conceptual design of a multi-storey steel car-parking structure. The participants were required to navigate, communicate, interact, provide information in the form of electronic documents, documents revisions, textual communiqué, drawings, sketches, and carry out design activities in their virtual project space.

Most of the participants had professional construction design experience. The multidisciplinary participants were expected to use their Internet connection to access the CODE testing session from their geographically distributed locations. The users participated in online cooperative and collaborative working sessions on demand and have completed the test scenario. All the activities, system performance and information data were backed up in the system's database. The opinions of the attendants were then requested through an online questionnaire. The questionnaires were filled in after the testing sessions. Although, the level of computer literacy among the participant was quite high, only 13% actually had some information about the concept of IFC and its impact on product modelling. This showed that the IFC knowledge is only common amongst the design actors with stronger IT background and field of interest in advanced construction engineering design work. The majority of the participants did not question any of the background technology used to facilitate the CODE system.

The participants emphasised the potential financial benefits and time-savings by using the CODE system. The users found the CODE system useful. The usefulness of CODE system, as a Web-based product modelling process system based on IFC standard, was measured according to the feedbacks from the testing sessions. Efficiency was measured in relation to time length of the project, effective communication between the design members, availability and accessibility of data, and ultimate project processing productivity. The participants believed that geographical-independent integrated system such as the CODE would provide greater flexibility to the design process.

3. Summary and Conclusions

The current design practice in the AEC industry is limited due to the lack of accessible modelling tools for presenting design processes and related dependencies throughout a building's life cycle. To address this limitation, the CODE system was developed in order to provide an IFC subset environment, which supports a rule-based approach to the product modelling process and determine dependencies between related activities over WWW portals. Since the actors requirements vary, the CODE system allows adding new inputs and feedbacks throughout the design process. The system acts as an active link between actors and creates a realistic image of real world communication in a collaborative workspace. The responsibilities are shared according to actor's discipline and group constraints, which are ruled by the CODE system and actor's awareness.

The CODE project demonstrates a possible strategy for the creation of IFC-based collaborative systems using Internet technology [1]. Although, these require further testing and software development before a fully functional system is well-established. The CODE environment is just a prototype, to demonstrate the concept of Web-based collaborative engineering work. The system is not designed to be the ultimate, fully developed and commercially marketable system; hence it has its own limitations. Due to the diversity and large scale of the research area, the implementation of the CODE was limited to a particular type of problem, which is testing the feasibility of the concept emphasised by this study.

The CODE system simply demonstrates that the product modelling is the basis for collaboration where actors share the information of the model in real-time interactive environment [1]. The CODE system facilitates a shared product data model and its supporting geometry within the same working space, where the product model is the centre of interaction and collaboration. The novel aspect and contribution of the CODE system is its strategy of targeting the project owner's needs, and its representation of multidisciplinary design data within a single project space. These allow actors to provide their services under supervision of the owner directly and preserve the administration hierarchy in the building design process as the traditional working approach [1].

The requirements outlined by the study were incorporated into the design of the CODE database system. The innovation of using product modelling through the WWW platform resulted in a database structure and use cases that provides an integrated working environment using most common IT solutions.

The uniqueness of adopting the IFC's Object Model was realised in the way the data entities were organised to describe the behaviour, relationship, and identity of a component object within a model. The study provided an implementation of the new attributes and created a set of new entities to bridge the IFC's hierarchy system for Web-based product data model activities. The impact and feasibility of the solution was put in a testing scenario by AEC/FM academic and professionals.

Reference:

1. Roshani, D. and W. Tizani. "Integrated IFC Based Collaborative Building Design Using Internet Technology" in The Tenth International Conference on Civil, Structural and Environmental Engineering Computing (CC 2005, 2005, Rome, Italy: Civil-Comp Press
2. Dakan, M., "What's the payoff for online AEC design?" CADalyst, 7(44), 2002
3. Crowley, A.J. and A.S. Watson, "CIMsteel Integration Standards Release 2", The Steel Construction Institute: UK 2000.
4. Dym, C.L. and P. Little, "Engineering Design: A project Based Introduction": John Wiley & Sons, Inc., 2000
5. Roshani, D., "State of the Art Collaborative Building Design Using Internet Technology", School of Civil Engineering: Nottingham, 1-50, 2002.
6. Han, S., et al., "Collaborative Engineering Design based on an Intelligent STEP Database". Concurrent Engineering, 10(3), 239-249, 2002
7. Jeng, T.-S. and C.M. Eastman, "A database architecture for design collaboration". Automation in Construction, 7(6), 475-483, 1998
8. Varma, A., et al., "Web-Base Tools for Engineering Design", in Workshop on Agents and Web Based Design at the Fourth International Conference on Artificial Intelligence in Design: Stanford 1996.
9. Marir, F., G. Aouad, and G. Cooper, "OSCONCAD: A Model-Based CAD System Integrated with Computer Applications". IT-com, 3, 1998
10. Lai, Y.-C., P. Christiansson, and K. Svidt, "IT in collaborative building design": Denmark 2002.
11. Betoni_IFC, "IFC data exchange in concrete construction", Confederation of Finish Construction Industries, 4, 2003.
12. IAI. "IFC Technical Guide", 2000 [http://www.iai-international.org/iai_international/Technical_Documents/documentation/IFC_2x_Technical_Guide.pdf.]
13. Liebich, T. "IFC 2x Edition 2; Model Implementation Guide", 2003 [http://www.iai-international.org/Model/files/20030630_Ifc2x_ModelImplGuide_V1-6.pdf.]