SUPPORTING INTERACTION WITHIN VIRTUAL STUDIOS

Y.Z. Chen,
Imperial College,
Dept. of Electrical & Electronic Eng.,
Exhibition Road,
London SW7 2BT, UK

T.W. Maver,
University of Strathclyde,
ABACUS,
131 Rottenrow,
Glasgow G4 0NG, UK

ABSTRACT *In this paper the author describes the development of a virtual studio environment, which is intended for supporting communications for both dispersed human designers and distributed CAD applications. By applying the metaphor of the real world design studio, a virtual studio model has been defined as an electronic locale in the computer networks, which contains distributed resources and is inhabited by dispersed designers. Virtual studio environment (VSE) has then been proposed to refer to such a multi-user environment which supports the creation, operation and management of virtual studios. A distributed implementation architecture, which loosely couples the domain resources with the VSE base system through resource agents, has been designed. Conceptual building design has been chosen as the application domain for prototyping. Several typical scenarios of interaction with VSE will be discussed. One of the prominent features of this system is that the supported interaction takes place within, instead of through or external to, the design systems.*

KEYWORDS:*building design, virtual design studio, human-human interaction*

1. INTRODUCTION

Large construction projects rely on integrating contributions of multiple disciplines from different participants in a coherent and smooth manner. Communication between project participants and inter-operation among CAD applications are of paramount importance. There are thus two forms of human-human interaction to be supported: direct interaction by means of dedicated communication facilities or indirtect interaction via the sharing of domain design artifacts (e.g. design tools or data).

Many past efforts have addressed design integration issue in terms of design tool inter-operation. While many insights have been gained on the multi-disciplinary nature of design, the human's role has been largely left undefined in such system-oriented integration appoaches. Recognising that in the building industry the project participants constitute a *virtual organisation* which exists throughout the life of the project then disbanding, virtual design studio (VDS) has been coined in MIT to refer to "networked facilities that provide participants with access to the virtual organisation's databases and computational resources, messaging and data exchange, and vedio conferencing, in a highly integrated fashion" [1]. From this defintion, a VDS would support not only the inter-operation of design artifacts (data and tools), but human level interaction as well.

The VDS concept actually conveys an ideal for design studio of the future. Yet the question is: *how to actually realise such an ideal?* To this end, there have been active VDS experiments in the last couple of years [2, 3]. In these experiments existing CMC (Computer-Mediated Communication) facilities, including e-mail, WWW and vedio conferencing systems, have been employed to support design collaboration across universities and countries. The experiments have been reported as quite successful with real design work carried out. Overall, as the experimenters also recognise, the present VDS activities have not touched high degree of collaboration yet, as the enabling technologies used are not sufficient or dedicated enough. While presentation level sharing via CMC has been involved, semantic

information sharing at CAD applications level has not been exploited. Furthermore, the VDS practice could have been better supported, if dedicated virtual studio frameworks were available to integrate the separated CMC facilities which were involved.

In this paper the authors reports a dedicated effort on VDS development, which attempts for supporting communications for both dispersed human designers and distributed CAD applications. By applying the metaphor of the real world design studio, a virtual studio model has been defined as an electronic locale in the computer networks, which contains distributed resources and is inhabited by dispersed designers. Virtual studio environment (VSE) has then been proposed to refer to such a multi-user environment which supports the creation, operation and management of virtual studios. A distributed implementation architecture, which loosely couples the domain resources with the VSE base system through tool agents, has been designed. With this architecture, the general CMC facilities are embedded within the VSE base system, whilst the domain resources can be dynamically plugged in during use. Conceptual building design has been chosen as the application domain for prototyping. One of the prominent feature of this system is that the supported interaction takes place *within,* instead of *through* or *external to,* the computer systems.

2. VIRTUAL STUDIO AND VIRTUAL STUDIO ENVIRONMENT

While the phrase "virtual design studio" naturally implies a metaphor of the real world design studio, this has not been fully explored yet in the present VDS practice. The use of metaphors in the design of computer artifacts is very common. Metaphors function as suggestions to the user that work with computer artifacts is similar to the work the user is familiar with in ordinary work situation. Properly applied metaphors make the systems look natural and easy to use. By purposefully applying sound metaphors the developers can more easily design and implement the systems functions and structures. Applying metaphor would normally involve extracting the relevant properties from the original images and mapping them into the computerised mirrors.

2.1 Real world design studio

Real world design studio's properties may be analysed in terms of its *identity*, its *structural form* and its *behavioural functions*. A design studio has its identity. A studio is used by specialists to engage in collaborative design work. This property of *use* makes studio differentiated from other types of offices, and further determines its structures and functions.

A design studio defines a specific region through physical boundaries, which differentiate itself from other spaces. Such a bounded region serves to include some elements while excluding others. The normally included elements are the inhabitants and the facilities, such as drawing boards, desks, file cabinets, instruments, telecommunication facilities, notice boards, and so on. A real world studio is an open system in the sense that its elements -- both inhabitants and facilities -- can be dynamically included/excluded.

Design studio accommodates both individualistic and collective activities. More personal workspace may be established around the desks allocated to individuals. More importantly, studio provides a collaborative environment for collective activities, which are constituted by the coordinated individual activities. This coordination is facilitated by a variety of resources and mechanisms provided by the studio. These may be broadly classified as two categories: the physical co-presence of colleagues makes it possible for each other to become instantly aware of *who is doing what* through the natural vocal and visual communication channels and other interactional cues. Other sharable resources such as public file cabinets and notice boards support semi-synchronous and asynchronous, formal and informal coordination. Apart

from "job-related" activities, design studio also supports socialising. In fact, in real-world workplaces formal and informal activities are always co-existent.

Overall, a studio provides a natural environment for human design communication. It is open, interactive and integrated with its particular identity. It is expected that all these features of the real world studios can be transferred into computer-based virtual studio systems. It should be noted that, choosing real world design studio as the source of metaphor is only because of its familiarity to human designers. This should not imply that design studio is the best metaphor for developing collaborative design systems. In fact, as Professor Lansdown of Middlesex University pointed out, real world design studio may not be the best approach to designing in the first place. Still, the authors believe that such a metaphor is a natural choice.

2.2 Virtual studio model

A virtual studio model emerges when we transfer the properties of the real world design studio into computer systems.

Virtual studio identity A virtual studio provides a bounded virtual region in the form of integrated computer environment for a design project. It is "bounded" only in the sense that the memberships and resource inclusion are highly controllable. There could be no physical boundary; this mostly makes *virtual* studio different from *real* one. Instead, the physical boundary of a virtual studio can extend to the limit of the effective communications, which the computer networks can deliver.

Virtual studio's structures Virtual studio resources include both CMC utilities and CAD applications. The formal assist human level interaction, while the later are the fundamental means for carrying out design tasks. Virtual studio is structured by imitating the real design studio. The interfacing between human and computer should be designed by applying the spatial metaphor of the physical studio. Both human participants and studio resources are visually represented as icons in a graphical user interface, which is organised to resemble the looks and feel of real studio. Such a virtual studio can be visually presented as a 2D or 3D space, which is further divided into areas. Each area presents a particular studio resource or function. For example, a virtual note board can be designed as a studio area for users to post and read notes; a virtual co-present area may be designed to display the current inhabitants of the studio such that the participants can instantly "see" each other; and so on.

Virtual studio's functions Virtual studio supports both private and collective activities by acting as both private and shared virtual workspaces. Ownership and accessibility are implemented with virtual studios. Interaction in virtual studio takes place through CMC and/or CAD applications. Mutual awareness is both facilitated and inhibited by virtual studios due to studio's spatial properties. For example, users would be more aware of each other when they participate in the same virtual studio than in different studios. Like its real world counterpart, virtual studio encourages informal socialising. This implies that virtual studio activities can be loosely defined. Such flexibility is important to support self-organisation and design process variety.

Therefore, a virtual studio is an electronic locale in the computer networks, which contains as elements both distributed resources and dispersed human participants. Like its real world counterpart, virtual studio is an open and integrated system. Most importantly, virtual studio supports a level of user embodiment in the sense that human participants are somehow immersed within the computer system through mutual awareness mechanisms. The users thus form an integral part of the environment and the interaction takes place *within*, instead of *through* or *external to*, the systems.

2.3 Virtual studio environment

With computer's automatic power, it is expected that different virtual studios can be instantiated and configured with ease to form different settings for different design tasks. Big design project may use several virtual studios, and one person may participate in several virtual studios. Therefore, virtual studios may be related to one another. Mechanisms are needed for the users to construct and navigate across different virtual studios. To meet such kind of demands, *virtual studio environment* (VSE) has been proposed by the author to refer to such *a multiuser environment which supports the creation, operation and management of the virtual studios*. VSE is thus a virtual studio management system, which can provide persistency and manage the life-cycles for the virtual studios.

Extending from the concept of singleton virtual studio into VSE results in a big leap from the perspective of technology exploitation. Instead of serving as an one-off, special-purposed design system, a VSE possesses the kind of abstraction and generality, which allow different kinds of virtual studios (each of which serves as an integrated design environment) to be spawned with ease. Naturally, different virtual studios within a VSE can easily share resources such as CAD applications

3. SYSTEM DESIGN

The key requirements for a VSE have been analysed from two inter-related perspectives. As workplace, a virtual studio must be task-aware. This requires the modelling of design domains, in terms of design databases and design tools. Here the sharing and interaction of the distributed domain resources is the focal point. As facilitating medium for collaborative use of the domain resources, a virtual studio system must provide rich CMC facilities for collaborators to engage in collective activities in distributed settings. The support for participation, interaction, and coordination is the key issue. While VSE integrates the two perspectives, there is still a need for distinguishing between the CMC facilities and domain resources. While the formal may be pre-defined and structured for all VSE activities in general, the later could vary according to the specific need of the task domain. Together, these requirements call for the support for the definition and evolution of the design process and the virtual studios, and an open extendible implementation architecture.

Distributed design models

A distributed view has been adopted, in which design is modelled as constructing a common distributed design database and as the interaction of the distributed design tools. Design process is defined in terms of design tools. Each tool is an autonomous module, which is equipped with mechanisms for local databases and for communicating with other tools. The studio design database is formed as collections of pointers, which point to the local databases associated with individual aspect design tools. Specifically, studio design production is organised in terms of design sessions. A new session begins when an aspect design tool is invoked from within a studio. During the invocation, an aspect design point may be made via a checkpoint mechanism to record the design data and/or simulation results in multi-media. This aspect design point, together with the descriptions of the intent and the information originator, may be signed off as a pointer to the studio. A studio design point is thus constructed as a set of pointers to the aspect design points. The studio design space is then constructed via studio design points, and studio design database is distributed along with the design tools across the networks

Studio definitions

As unit of resource management and medium for group formation, a virtual studio is proposed to be characterised by three major components.

The first component is the **Particulars**, providing the general information about the purpose and the activity for which the studio is created and used. This component further includes elements such as *studio name*, *studio ownership*, *studio accessibility*, and *studio description*.

The second component is the **Resources** that populate the workspace of the virtual studio. The resources are classified into three categories: *CMC utilities*, *CAD tools* and *studio databases*. *CMC utilities* include (i) text-based commands for VSE information access, conversation, and message; (ii) an image communication kit for users to exchange on a real-time basis part or whole of the screen displays of each other's workstation; (iii) a note board for users and VSE server to announce events or for similar purposes. The *CAD tools* in studio model are actually pointers, pointing to the distributed tools. Two default *studio databases* are "NoteArch" and "Production". The formal is for archiving old notes which have been removed from note board, while the later for recording domain-specific design production. Since each aspect tool maintains its local database, studio design database is actually formed as collections of pointers, which point to the distributed multi-media databases.

The third component is **Persons**, who are participating in and interacting with the studio. Each VSE user is associated with a *Person* record/object, which is attributed with elements including *user ID* and *password*, *email address*, *visual image*, *info*, *working locale*, *state* and *activity*.

VSE system architecture

A distributed system architecture has been designed for VSE realisation. As illustrated in **Figure 1**, it consists of two subsystems: *VSE base system* and *Domain Resources*. The VSE base system, which includes *VSE server* and *VSE clients*, is a general, multi-user interaction environment. While VSE server provides persistency for VSE objects and maintains an operational context, VSE clients act as interfaces for human users to participate in VSE activity. CMC facilities for HHI are designed as built-in resources, which are bundled to VSE base system. Such resources are automatically available to each studio created.

Unlike CMC resources which are tightly embedded within VSE base system, the domain resources are *loosely coupled* with VSE base system. In another word, domain resources are dynamically plugged in/out during the use of VSE and studios. This has been achieved through the use of resource agents (RAs).

A RA is a piece of autonomous software, which abstracts/encapsulates the functionalities of one or more domain resources (e.g. design tools) as a set of well-defined services, and provides the services upon request. A RA interacts with VSE community on behalf of the domain resources it represents. During domain resource registration, the resource description and the address of a RA form a domain resource entry to be added into VSE resource directory. This entry can then be added into one or more studios. When a domain resource is invoked by a user from within a studio, VSE passes the request to the corresponding RA. RA processes the request and invoke the real resource (this would usually result in a remote design tool to be displayed on the user's screen). The tool-tool inter-operation in VSE is through the serving/requesting of design objects. This has been supported by domain information models, which define the common terminology and structures

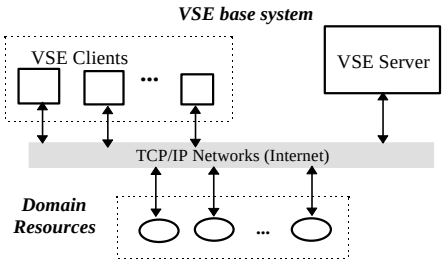

FIG 1: VSE system architecture

The loosely coupling of the domain resources with VSE base system at a system architecture level has several advantages. First of all, it results a generic and extendible system architecture, in the sense that different domains or projects can be accommodated by swapping or changing the domain resources. Such change is independent of the base system. Also, it encourages participation. For example, the individual participants may contribute their favourite tools to the studio, which can be either solely used by themselves or shared by other people as well. All these advantages are largely indebted to the use

VSE operations

A state diagram is used, as shown in *Figure 2*, to organise the major VSE operations. It has been decided that the VSE system will be operated as other multi-user *Internet* applications (e.g. MUD and IRC) via user IDs (and passwords), based on which higher level of user embodiment is implemented.

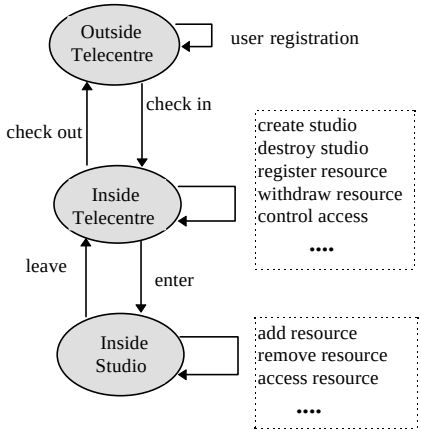

FIG 2: VSE operations

Three system states have been defined: *Outside Telecentre*, *Inside Telecentre*, and *Inside Studio*. Here *Telecentre* is used to denote the state immediately after a user has successfully logged in on VSE server. Therefore, Telecentre actually stands for the hall of a virtual building. The user needs to register with the VSE server to obtain a user ID and password. After check-in, the user may create/destroy studios, register/withdraw resources, change the access state of the studios, or enter an existing studio. When *Inside (a) Studio*, the user may add resources from the global resource directories into the studio's domain resource list, access or remove the domain resources which are already added in.

Most of the functions (including both CMC and domain resource invocation) are *studio sensitive* in the spirit of the spatial metaphor. For example, when someone "say[1]" something, only inhabitants in the same studio can hear; a design tool can only communicate with other tools which are added in the same studio; and so on.

4. VSE PROTOTYPE DESCRIPTION

A VSE prototype has been developed [4] in the context of conceptual building design. A simplified scenario (**Figure 3**) has been designed to test the design framework. The FGC (Function, Geometry & Construction) refers to a core design aspect which is intended to centralise the heavily shared data items. Three building performance aspects -- lighting, thermal and costing -- are considered. Interfaces are required between FGC and each of building performance aspects. The thermal and costing costing require direct communication links to implement the sequential dependency on the energy consumption data. The core module "FGC" has been implemented by customising and further developing AutoCAD (R12). A

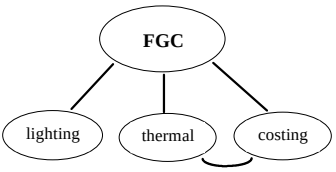

FIG 3: Conceptual building

[1] If someone wants the people in other studios to hear, she can "shout" or "page".

thermal tool called “AngliaDom” is developed by implementing a BREDEM-8 model. A costing tool called “PSCost” and a lighting tool called “NATLIT” are based on the algorithms and sources codes of two existing programs from ABACUS of Strathclyde University. Each of these design aspects has been implemented as a design tool, which is associated with a resource agent.

The CMC facilities developed include on-line conversation and messaging commands, a note board, and an image-based communication kit. Together these facilities support a wide range of communication modes. In addition to these CMC resources, the human users themselves are also explicitly integrated into the environment through a dedicated *co-present* mechanism.

As an Internet-based operating environment the VSE base system consists of a VSE server and VSE clients. VSE server provides persistency for VSE objects and manages all operations in regards to the use of VSE. The client GUI is designed by applying the spatial metaphor. As depicted in **Figure 4**, the main window of the client GUI displays a virtual studio in a 2D space.

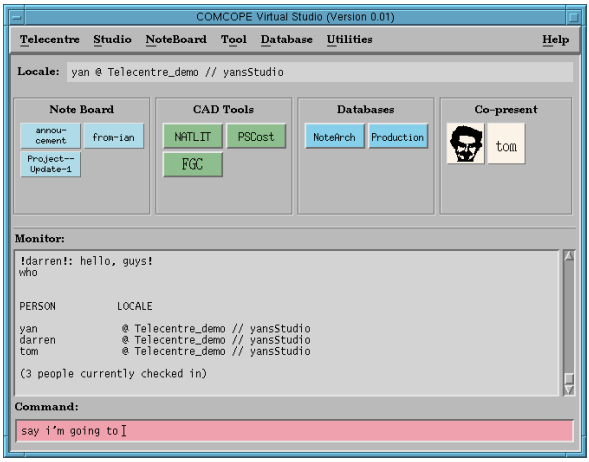

FIG 4. VSE client in operation

- **Locale** The Locale panel displays the user name and location in VSE for the client user.
- **Studio areas** Studio is visually presented as four areas: *Note Board*, *CAD Tools*, *Databases*, and *Co-present*. These areas display the resources/personnel which are currently available/present in this studio. Each area can display up to six items in the form of icons. These icons are active or invokable. By clicking an icon with the activation mouse button (the left button), the intended function associated with the icon will be invoked. This could be displaying a note in a pop-up note browser, activating a possibly remotely hosted CAD tool but displaying it locally, browsing a studio database in a database browser, or showing a person’s status in the monitor window. By clicking an icon with the selection mouse button (the right button), a small pup-up menu will be popped up to allow the user to remove the note from NoteBoard, to view the information about a tool, and so on. Note that while the capacity of each areas is designed as holding six items, the actual number of items for each category of the resource (and personnel) can be indefinite (up to computer’s capacity). This is because internally the resources of the studio are represented as lists. While the latest six items are displayed in GUI areas visually as icons, the whole lists can be browsed and/or invoked from the pull-down menus under the top menu bar. In fact, all manipulation functions, most of which are not accessible through the area icons, are organised through the pull-down menus. Therefore, the capacity-constrained areas actually function as a buffer or front end for the resource lists to achieve “studio-alike” impact visually, and the operation convenience through direct manipulation.

Note that the *Co-present* area does not display the icon of the client user herself. This is because that in the real studio one can only see others, but not herself. The GUI is supposed to be an electronic equivalent to the real studio, so the user cannot see herself either.

- **Monitor** The Monitor displays texts for user operation feedback (including user's typing) and external events. This is implemented as a scrolled window with large buffering capacity. Therefore, the user can examine the history of previous VSE operations simply by scrolling this text window back.
- **Command**The Command text field in the bottom of the GUI allows text-based commands to be entered. These commands may relate to conversation, messaging or VSE information interrogation.
- **Menubar** The menubar on the top of the GUI provides through pull-down menus rich facilities for users to undertake all VSE activities. This menubar is organised as seven functional categories: "Telecentre", "Studio", "NoteBoard", "Tool", "Databases", "Utilities" and "Help". Specific functions are implemented with each category. For example, in the *NoteBoard* pulldown menu there are functions "Create Note", "Browse Note" and "Remove Note". The "Utilities" pulldown is for organising those facilities which are not visually presented in studio areas. At the moment, the image-based communication kit, which includes a Grab&Post tool and an image receiver, is organised in there.

5. INTERACTION WITHIN VIRTUAL STUDIOS

Given the resources (both domain specifc design artifacts and generic CMC facilities) provided within VSE prototype, the potential opportunities for HHI-based design integration activity is limited only to one's imagination. The following three scenarios, as illustrated in **Figure 5**, are used to illustrate the possible interactions in VSE, and the relationship among tool inter-operation, HHI, and design integration.

Directly sharing an artifact In case (a), two persons interact by directly sharing an artifact. Each person can manipulate the artifact and receive the feedback for his/her action. At the same time, each person can also feel the other person's action by observing (such observation may be called "feedthrough"). This can take place either synchronously or asynchronously. Some examples include multi-user drawing (e.g. white board) for the former and shared databases/files for the latter. Direct communication could also be used for additional coordination. For example, on-line conversation and message notification can be very useful coordination support for multi-user drawing system and data sharing system respectively.

Indirectly sharing an artifact In case (b), only one person(P1) has direct access to the artifact, and the other person(P2) obtains the information about the artifact from the first person through direct communication. This can be both synchronous or asynchronous too. For example, one person is editing the layout of a building with AutoCAD-based FGC tool, which is communicated to the second person through real-time image-based communication facilities. This is in synchronous mode. Asynchronously, the first person may send the drawing to the second person through on-line messaging facilities.

Accessing different but inter-operable artifacts In case (c), two persons interact by accessing two different but inter-operable (by whatever means) artifacts. A typical scenario may be: one of the them have direct manipulation on one artifact, but the desired feedback can only be obtained from the other person by manipulating another artifact. For example, the architect, who is doing the layout design using FGC, wants to know the thermal implications

of a specific geometrical decision. She might ask a thermal specialist, who would operate a thermal tool AngliaDom, to obtain this thermal feedback and pass it to her via direct communication. This could be either synchronous or asynchronous. For the former, on-line inter-operation connection needs to be established between the thermal tool and the geometrical tool, such that the thermal specialist could receive the geometrical change instantly after the geometry change. In the latter, the geometrical changes could be archived in a database, from where the thermal specialist will access the geometry data.

Depending on the specific situations, some of them may be more useful than others. Together, they speak for one fact: both domain specific artifacts and general communication facilities can equally contribute to design integration. The boundary of these two superficially different types of resources becomes blurred during the action of design work. However, this can be true only when these resources are integrated in the same environment such as virtual studio, such that both domain and CMC resources are readily available at users' finger-tip.

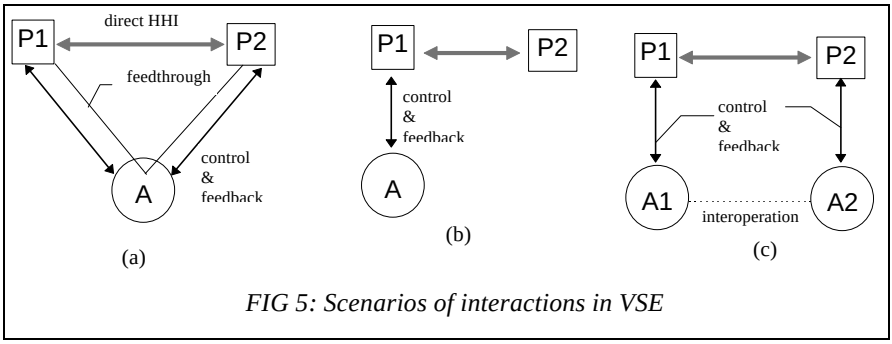

FIG 5: Scenarios of interactions in VSE

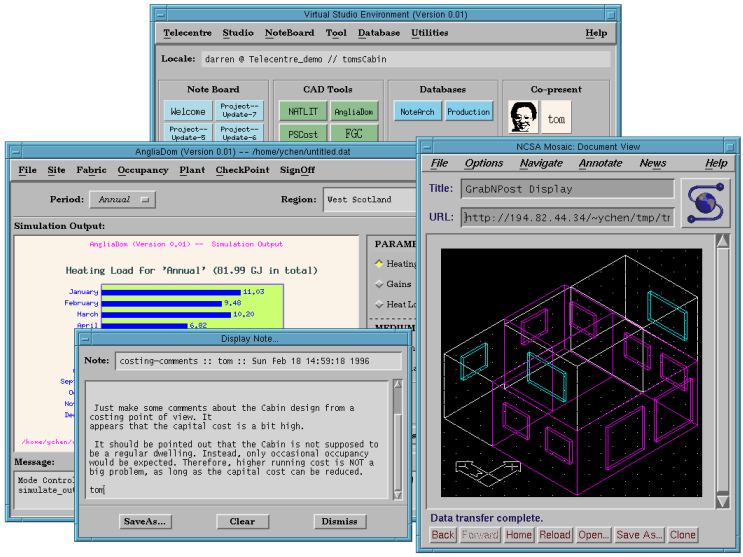

FIG 6: VSE in operation

Figure 6 is another snap shot of VSE in operation. The user "darren" is conducting an on-line collaborative simulation in a studio named "tomsCabin" with user "tom" and user "yan"(i.e., the guy who is visually represented with an image in the *Co-present* area). In previous

sessions “yan” designed a two storey cabin, which was found by “tom” as too costing. The user “tom” raised this issue by posting a note (named “costing-comments”), which “darren” is still reading in a note browser. The user “yan” thus decides to make some changes to the original design. Incidentally, “yan” found that “tom” and “darren” were in VSE (though in different studios). So “yan” decided to take advantage of this by setting up an on-line session. In this snap shot, “darren” is using AngliaDom to simulate the thermal performance of the re-design the user “yan” is currently working on. Apart from exporting the data through tool interfaces, the user “yan” also broadcasts the geometry image of the new design within the studio. Both “tom” and “darren” can receive this image via an image receiver. Therefore while “darren” is operating AngliaDom, he can “see” what the design actually looks like.

6. SUMMARY AND DISCUSSION

A virtual studio system has been developed, which supports with the same studio environment both the dispersed human interaction and distributed CAD resources integration. Since the human users are immersed within the system, human-human interaction (both direct and indirect) thus takes place within environment.

In comparison to the original VDS notion and present VDS activities, the VSE concepts proposed represent a significant expansion. Although the term “virtual design studio” itself implies a metaphor in the first place, but the present VDS efforts touches this metaphor only at a very conceptual level. In comparison, the virtual studio model refined in this paper stands for truly integrated environment with explicit studio identity. The studio metaphor has been applied to levels from presentation through functions to structures. Furthermore, VSE is a virtual studio management system, which helps create, configure and manage potentially large amount of studios for intra-organisation or inter-organisation collaboration.

The implemented prototype system only supports conceptual building design at present. To cover more design phases/aspects would be straightforward by adding more design tools. The system architectures, particularly the loose coupling between the VSE base system and the domain resources, and the resource agent integration method, has well prepared for such extensions.

References

1 *Design Studio of the Future*, in **PLAN 39** (Fall 1993), The Newsletter of the School of Architecture and Planning, MIT, Published by the Dean’s Office, Room 7-231, 77 Massachusetts Avenue, Cambridge, MA 02139.

2 Cheng, N, Kvan, T, Wojtowicz, J, van Bakergem, D, Casaus, T, Davidson, J, Fargas, J, Hubbel, K, Mitchell, W, Nagakura, T, and Papzian, P (1994), *Place, time and virtual design studio*, in Harmann, A C and Fraser, M (Eds) **Reconnecting: ACADIA’94**, Association for Computer Aided Design in Architecture, pp 115-131

3 Maher, M L, and Saad, M (1995), *The Experience of Virtual Design Studios at The University of Sydney*, in ANNZAScA Conference, University of Canberra, (Available at: **http://www.arch.su.edu.au/kcdc /design_studio/papers/anzasca95.html**)

4 Chen, Y.Z. (1996) The development of a virtual studio environment to support collaborative design, Unpublished PhD thesis, Anglia Polytechnic University, (Feb)