*Proc. of ImechE, Part B, Journal of Engineering Manufacture*, Vol 218, pp. 1009-1015, 2004.

# Incremental Simulation Modelling for Internet Collaborative Design

S.F. Qin and D. K. Wright
Sheng.feng.qin@brunel.ac.uk; David.wright@brunel.ac.uk

Department of Design, Brunel University, Runnymede Campus, Surrey TW20 0JZ, UK

## Abstract

In order to support Web-based collaborative design in terms of transferring or updating models dynamically and efficiently, new incremental modelling and local updating strategies have been developed for simulation modelling application since simulation is more focused on visualisation effects than on geometry details. Based on an assembly connection concept, a drag-and-drop assembly method has also been proposed in simulation assembly. An assembly connection is defined as a group of assembly constraints and it makes assembly easier. A case study example is given to show the content of the proposed research.

**Key words:** incremental modelling, local updating, simulation, collaborative design.

## 1. Introduction

Collaborative design requires the cooperation of different design teams (or individuals) at geographically different sites using commercial and non-commercial engineering tools such as CAD and modelling tools. They may work in parallel and separately with various engineering tools. They may also work together with a shared tool in a synchronous or a mixed synchronous and asynchronous way. At any moment, individual designers may be working on different versions of a design or viewing the design from various perspectives, at different levels of detail [1].

In order to support various collaborative design, it is necessary to develop collaborative design environments and supporting tools. Internet/Web-enabled technologies have rapidly evolved and been adopted into collaborative design, because a Web-based system has a universal interface, uses open standards and is globally supported. However, development of a Web-based CAD modelling tool is challenging, because current CAD systems are designed for standalone applications. A 3D CAD file is usually tens or hundreds of megabytes in size posing much difficulty in communicating a CAD model in a Web-based environment dynamically and efficiently so as to keep the information consistent in the server and client's real time.

This paper reports a new distributed simulation modelling tool for supporting Internet collaborative design. A new representation scheme has been developed to support the dynamical and efficient updating of models in a network environment by means of incremental local updating. In the remaining part of this paper, related research is reviewed in Section 2. The incremental simulation modelling and its local updating strategy are described in Section 3. Section 4 presents a drag-and-drop assembly method for collaborative assembly. The system implementation and a case study example are introduced in Section 5 and finally conclusions are drawn in Section 6.

## 2. Related research

For collaborative design, some groupware based geometric design systems have been developed [2,3]. However, Rezayat [4] pointed out that distributed design and manufacturing environment must be Web-based because of its universal interface, open standards, ease of use, and ubiquity.

A number of frameworks have been proposed for Web-based collaborative design systems. The Distributed Object Modelling Environment (DOME) system at the MIT CAD laboratory provides a flexible environment for modelling and evaluating design problems using modules and distributed modules [5]. In this environment, Java remote method invocation (RMI) and java native interface (JNI) are used to interact with various engineering tools such as CAD modellers. The DOME system focused on sharing product development information and data exchanges, not on geometric modelling. The cPAD system [6] supports Internet-based collaborative product design with assembly features. Polygonised representations of assembly models have been utilized with the Parasolid Kernel for rapid updating visualization in Java3D at client sides. The system provides designers with the ability to perform real-time geometric modification, assembly constraints specification and concurrent design of different components. However, designers cannot design a component synchronously. An Internet-enabled interactive fixture design system [7] has been developed using Java and XML. Designers can create solid models on the server through a RMI interface and JNI calls to Parasolid Kernel and view design models as facets in XML format files. These two systems used a thin-client-and-fat-server architecture. Updating geometric model process might be slow because the geometric form and CSG constructive constraint descriptions need to travel from the client side to the server side through the RMI mechanism and get the corresponding solid models and facet representations by JNI calls to the Parasolid kernel. The resulting facet data has to be sent back to a Java3D canvas at the client side for display. Therefore, after any modifications, display updating needs to go through the round transfer. A multi-user CAD prototype system Co-CAD [8] has been developed. The Co-CAD was largely from a mechanical engineer's perspective and limited to collaboration between two people. Coordination strategies to avoid conflicts between design participants have been proposed by Klein [9].

In order to support Web-based solid modelling, a Java-based solid modelling library [10] has been developed for easily creating, manipulating and viewing solid models on the World Wide Web. This library makes it possible that the modelling engine can work on the client or the server side. Some Web-based CAD tools such as "CyberCut" [11] used a solid interchangeable format for the transmission of feature-based descriptions of mechanical parts so as to make design tools shareable and interoperable.

For 3D simulation modelling, a model is represented as a CAD assembly with some motion definitions. Creation and manipulation of a simulation model on the Web might be very slow if a 'fat' server is adopted in the system design. In order to support collaborative design (modelling) of a simulation model, incremental modelling and local updating strategies are taken as they were used in [12].

## 3 Incremental component modelling

In a collaborative design environment, geometric models are shared and viewed. Any model changes need to be updated at the server and broadcast to the client for updating display. Local updating is a strategy for making incremental changes to a geometric model. Only those parts of a model that are affected by creations and modifications are recomputed and re-exchanged between the server and the client, which considerably increase efficiency compared to updating the complete model and its display. Methods for local updating in a simulation modelling system is based on an incremental simulation modelling tree and Java3D visualization functionality. The strategy can be described in a typical three tier distributed system [13] (Fig. 1).

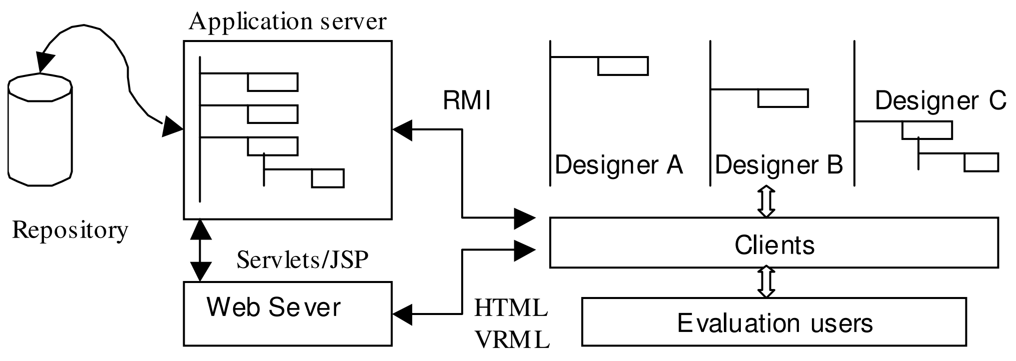


Fig. 1 Architecture of the system

A simulation model is represented in a model tree. The model tree at the server side is shared by collaborative designers at the client. For example, designer A may work on the first node, designer B may work on the second node, and meanwhile designer C may work on the third node. Creations and modifications of a model need only to update leaf nodes between the shared modelling tree at the server and the corresponding nodes from different clients. Each three node is a Java object which contains 3D geometric information of 3D Java primitives or a pre-defined design feature, positioning information of the geometry related to its parent, and motion-related definitions. Java3D is regarded as a good means of developing visualisation-based design system such as simulation, because it supports visual realism of a 3D scene constructed by union operations of primitives without needing a real modelling engine, e.g., ParaSolid Kernel. For subtractions, the result can be achieved by using a Java-modelling library [10]. Thus, with support from Java3D and Java modelling library, a little 'fat' client application (Applet) for simulation can be developed to create a model and display it directly at the client side. Only contributions from other designers need to be broadcast through the server and downloaded for model updating and display at the client.

This incremental modelling can support application scenarios as follows:

(1) A single model designer with a group of evaluators

The designer can create, modify, and display a simulation model within the client-side application. If there is no model evaluator at the modelling time, the model can be created at the client side until the designer save the model through RMI calls to the server. If there are some model evaluators working concurrently, once a new modelling node is created, it will be uploaded to the server and the server will then update the corresponding Web contents through the Java Servlets and Java Server Pages (JSP) mechanisms. The Java3D modelling contents will be transferred into the VRML format contents.

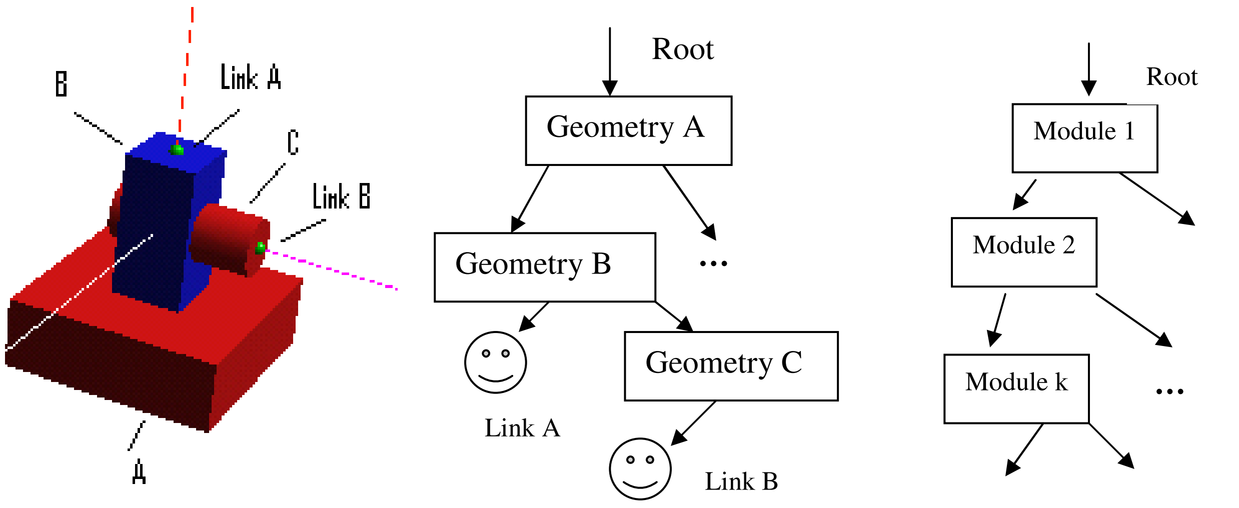
(2) Multiple model -designers with a group of evaluators

In this application scenario, multiple model designers at different sites create a model. Each designer can work on different parts of the model concurrently and separately. Once a new leaf node is completed, it will be uploaded to the server and require the server to broadcast it to the others and then the application at the client will check if there are any new leaf nodes from the others waiting for download and updating. If that happens, the client application will download the waiting nodes and add them on the modelling tree at the client site and update display locally. If there are some evaluators working at the same time, once receiving a new node, the server will update the corresponding Web contents for them.

With the topological information provided from modelling trees, a model can be efficiently updated when it is modified. Once a node is modified, only the node needs to be updated, not the complete existing model. The topological information also facilitates the user conflict management.

## 4. Drag-and-drop assembling

In our system, we divide simulation models into two types: module models and machine models. A module simulation model is a virtual representation of a physical module to demonstrate a modular design's motions, functions, control logics, and geometric appearances. In addition, a module model is associated with possible connections (defined by unified interface assembly features) with other modules. A machine model may be a sub-system model or a system model assembled from a module model.



(a) An instance of a module (b) Structure of a module (a) Structure of a machine

Fig. 2. Hierarchy structures of the simulation models

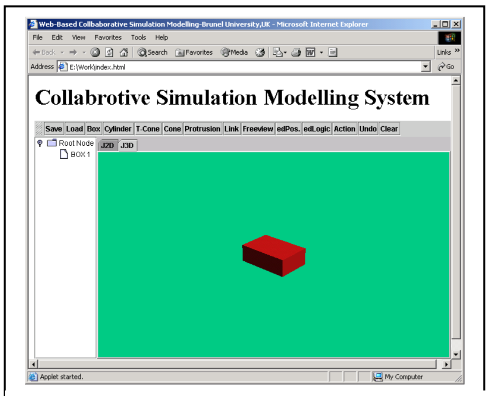
A module model is visualised as a 3D simulation model and is represented as a tree structure. It is built from elementary geometry such as cylinders and boxes in a hierarchy way. An elementary geometry is defined in a local co-ordinate system

(LCS). Its position and orientation is defined with respect to its parent's geometry. Motion with an elementary geometry can be defined in its own LCS. An elementary geometry can not only have other elementary geometry as child nodes, but also some pre-defined connections as children. A connection is represented as an attachment point associated with assembly constraints with respect to another module's model. For instance, a module (Fig 2(a)) can be modelled from the Base geometry A. The Base has a child of the geometry B (Slide), which can move on the top face of the Base. The Slide has a child of the geometry C (Shaft), which can rotate about its axis. The Slide also has a connection "Link A" at the centre of the top face, which links to a module "A". The Shaft has a connection "Link B" as well, which links to a module "B". Connections are geometrically represented as small spheres. The corresponding hierarchy structure is shown in Fig. 2 (b).

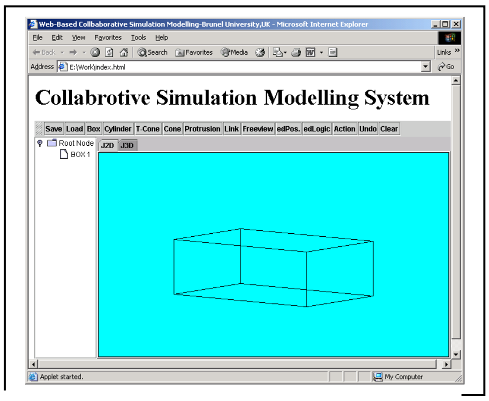
A machine model can be defined by assembling module models in a hierarchical tree structure (Fig. 2 (c)). A child module model can be assembled in a drag-and-drop manner. The system will automatically match connections between the child module model and its parent module model, and compute the assembly position of the child module with respect to its connected geometry in the parent module. Comparing with traditional CAD assembly [14,15] featuring low-level constraints such as mate and align, this assembling process is simple.

## 5. System implementation and an example

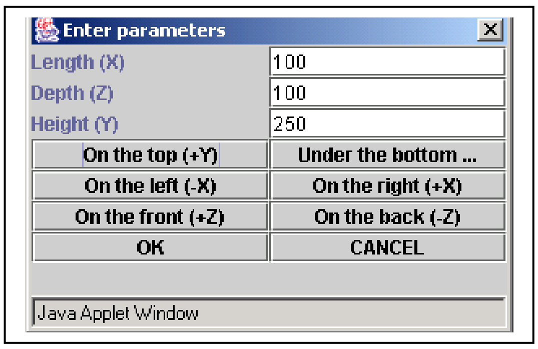
A prototype system has been implemented to support the proposed research. The client side application is a Java applet with Java2D and Java3D canvases. This is a little 'fat' application. It can create simulation modelling, and communicate with collaborative designers. The server side application is a Java application with its RMI interface with the clients. It can save and load a file for the design repository. It can generate VRML representations of simulation models. It will be able to create dynamic web-contents through Java Servlets and JSP although it is not implemented currently.


Web-Based Collbaborative Simulation Modelling-Brunel University,UK - Microsoft Internet Explorer
Collabrotive Simulation Modelling System
Save Load Box Cylinder T-Cone Cone Protrusion Link Freeview edPos. edLogic Action Undo Clear
Root Node
BOX 1
J2D J3D
Applet started.
My Computer

(a) A box from Designer A

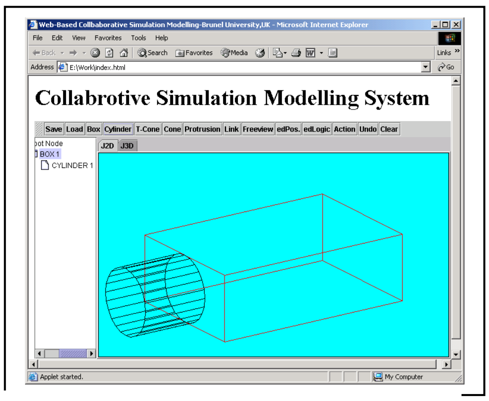

Web-Based Collbaborative Simulation Modelling-Brunel University,UK - Microsoft Internet Explorer
Collabrotive Simulation Modelling System
Save Load Box Cylinder T-Cone Cone Protrusion Link Freeview edPos. edLogic Action Undo Clear
Root Node
BOX 1
J2D J3D
Applet started.
My Computer

(b) Broadcast the box to Designer B

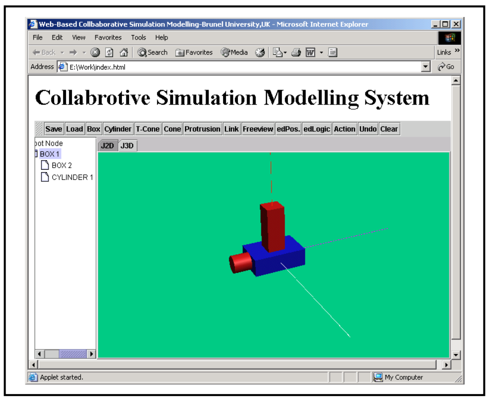

Enter parameters

| | |
|---|---|
| Length (X) | 100 |
| Depth (Z) | 100 |
| Height (Y) | 250 |

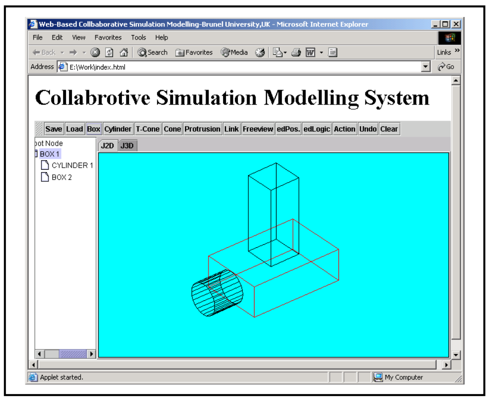
(c) Another box from Designer A


Web-Based Collbaborative Simulation Modelling-Brunel University,UK - Microsoft Internet Explorer
Collabrotive Simulation Modelling System
Save Load Box Cylinder T-Cone Cone Protrusion Link Freeview edPos. edLogic Action Undo Clear
BOX 1
CYLINDER 1
J2D J3D
Applet started.
My Computer

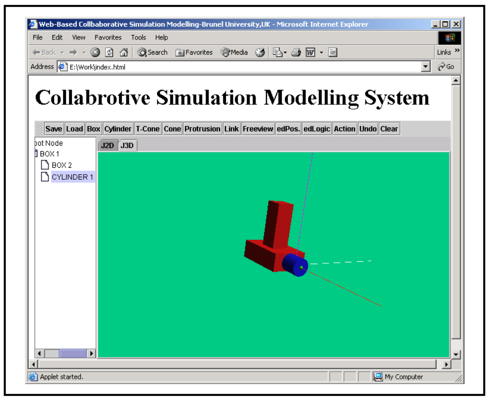
(d) A cylinder from Designer B


Web-Based Collbaborative Simulation Modelling-Brunel University,UK - Microsoft Internet Explorer
Collabrotive Simulation Modelling System
Save Load Box Cylinder T-Cone Cone Protrusion Link Freeview edPos. edLogic Action Undo Clear
BOX 1
BOX 2
CYLINDER 1
J2D J3D
Applet started.
My Computer

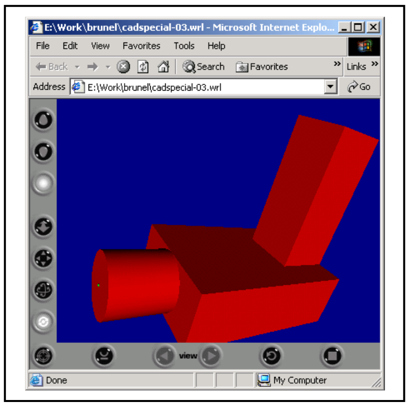
(e) Broadcast the cylinder to Designer A


Web-Based Collbaborative Simulation Modelling-Brunel University,UK - Microsoft Internet Explorer
Collabrotive Simulation Modelling System
Save Load Box Cylinder T-Cone Cone Protrusion Link Freeview edPos. edLogic Action Undo Clear
BOX 1
CYLINDER 1
BOX 2
J2D J3D
Applet started.
My Computer

(f) Broadcast the second box to Designer B


Web-Based Collbaborative Simulation Modelling-Brunel University,UK - Microsoft Internet Explorer
Collabrotive Simulation Modelling System
Save Load Box Cylinder T-Cone Cone Protrusion Link Freeview edPos. edLogic Action Undo Clear
BOX 1
BOX 2
CYLINDER 1
J2D J3D
Applet started.
My Computer

(g) Defining a connection


E:\Work\brunel\cadspecial-03.wrl - Microsoft Internet Explo...
Address E:\Work\brunel\cadspecial-03.wrl
view
Done
My Computer

(h) a motion model

Fig. 3 A case study example

The Figure 3 shows a case study example. After starting the server application, two designers A and B registered and started the client-side application. Designer A started with creation of a base box (Fig 3. (a)), and broadcast the design to Designer B (Fig 3. (b)). The model was displayed in shaded in the Java3D panel for Designer A and in wire-frame in the Java2D panel for Designer B. Designer A continued to add a small box on the top of the base with input from a dialogue window (Fig 3. (c)). In order to specify an initial position of the geometry, a group of positional constraints, such as “On the top” of the parent geometry, can be selected and applied. Of course, the position can be modified later. While Designer A created the top box, Designer B created a cylinder in the front of the base (Fig 3. (d)). When they were finished, their modelling nodes were broadcast and communicated with each other (Fig 3. (e, f)). Afterwards, Designer A defined a connection feature for assembly on the top of the cylinder. The connection feature was geometrically displayed as a sphere (Fig 3. (g)). Finally, a motion was specified and associated with the top box, naming “fetch”. The corresponding VRML model of the design is shown in (Fig 3. (h)) after moving out from the centre position.

## 6. Conclusion

In order to support Web-based collaborative design in terms of transferring or updating models dynamically and efficiently, a new incremental modelling and local updating strategy has been developed for simulation modelling application since simulation is more focused on visualisation effects than on geometry details. Based on the assembly connection concept, a drag-and-drop assembly method has also been proposed in simulation assembly. An assembly connection is defined as a group of assembly constraints and it makes assembly easier. A case study example has been given to show the design and model transferring processes. Compared with updating complete models, local updating strategy can save time in transferring models between the server and the client.

The demonstrated system is a partially implemented system. It does not cover user conflict management. Currently, request for broadcasting models is done by menu selection. It should be done automatically.

**Acknowledgements**

This work is partially supported by the UK EPSRC (GR/M83070) research grant.

**Reference**

1. Shen W. Editorial of the special Issue on CSCW in design, J. Computers In Industry 2002; 48:1-2.
2. Greenberg S., Roseman M., Webster D., Bohnet R. Human and technical factors of distributed group drawing tools, J. Interacting with Computers (Special issue on CSCW), 1992: 4(3) 364-392.
3. Nam T.J., Wright D.K. The development and evaluation of Syco3D: a real-time collaborative 3D CAD system, J. Design Studies, 2001: 22, pp 557-581.

4. Rezayat M. The enterprise-Web portal for life-cycle support, J. Computer-Aided Design, 2000: 32 .85-96.

5. Abrahamson S, Wallace D, Senin N, Sferro P. Integrated design in a service market place, J. Computer-Aided Design, 2000; 32:97 -107.

6. Shyamsundar N., Gadh R. Internet-based collaborative product design with assembly features and virtual design spaces, J. Computer-Aided Design 2001; 33: 637-651.

7. Mervyn F, Senthil Kumar A., Bok S.H. , Nee A.Y.C. Development of an Internet-enabled interactive fixture design system, J. Computer-Aided Design 2003 (in press).

8. Gisi M.A., Sacchi C. Co-CAD: a multi-user collaborative mechanical CAD system, J. Presence, 1994: 3 341-350.

9. Klein M. Supporting conflict resolution in cooperative design, IEEE Transactions on Systems, J. Man and Cybernetics, Special issue on Distributed Artificial Intelligence, 1991: 21 (6).

10. Chan S.C.F, Ng V.T. Y., Au A.S.F. A solid modelling library for the World Wide Web, J. Computer Networks and ISDN System (1998); 30: 1853-1863.

11. Wang F.C, Wright P.K. Web-based Design Tools for a Networked Manufacuturing Service, 1998 ASME Design Engineering Technical Conferences and Computers in Engineering Conferences, Sept. 13-6,1998, Atlanta, GA.

12. Bronsvoort W. F., Garnaat H. Incremental display of CSG models using local updating, J. Computer-Aided Design, Volume 21, Issue 4 , May 1989 , Pages 221-229.

13. Qin S. F, Harrison R. West A.A, Jordanov I.N., Wright D.K. A famework of web-based conceptual design, J. Computers in Industry (2003); 50: 153-164.

14 Lee K, Gossard D.C. A hierarchical data structure for representing assemblies: part 1, Computer-Aided Design 1985, 17(1):15-19.

15 Lee K, Andrews G, Inference of the position of components in an assembly: part 2, Computer-Aided Design 1985, 17(1):20-24.