

Web-based Product Design Review: Implementation Perspective

GQ Huang and KL Mak

*Department of Industrial and Manufacturing Systems Engineering,
The University of Hong Kong, Pokfulam Road, Hong Kong, PR China
gqhuang@hku.hk*

JY Shen and JQ Yan

CIMS Institute, Shanghai Jiao Tong University, Shanghai, PR China

Abstract

Product design review is one of the typical scenarios of collaborative product development. We developed a web-based prototype framework for supporting the collaborative tele-product design review on the Internet. It provides a suite of tools to establishing and management the new product design review process. This paper discusses the issues related to the development and implementation of web application in prototype implication. The typical 3-tier architecture is explained to show how the CyberReview components work together to achieve the intended functionality. The VRML EAI and Java Applet-Servlet pair technology was included to support the implication.

1. Introduction

Product design review is one of the typical scenarios of collaborative product development [4, 8, 11]. A team is tasked with the design and development of a new product. The team consists of members from multiple disciplines. Some members are lead users (key customers), some are core (key) suppliers, and others may come from various functions and units of the organisation. The members are typically geographically dispersed.

Traditionally, the design review is conducted by circulating the documents related to the product design to be reviewed from one member to another [2]. After all members have investigated into the subject, a meeting is then arranged to resolve the different opinions. This process is very inefficient, especially when some external members such as key customers and suppliers are involved from other regions.

On the one hand, design review is a mandatory requirement by ISO9001 [7, 10]. On the other hand, Information Technology (IT), especially Internet/web Technology, has provided opportunities for effective design review in the extended enterprise [3].

Despite the fact that Information Technology (IT) has been employed to facilitate this process, the progress is far from satisfactory. This observation has been verified from our contacts with industrialists involved in both IT and manufacturing sectors. The roles of word processors and email messages are limited. Instead, a new paradigm is needed to completely reengineer this vital process of design review.

The aim of this research is to develop an overall methodology for enabling more efficient and effective design review system in new product development process, and to demonstrate the framework through a prototype web-based platform on the Internet/intranets using the web technology.

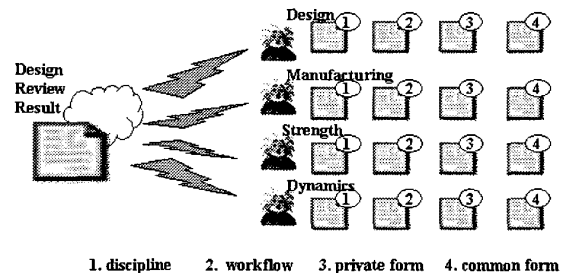


Figure 1. Web based product design review.

2. CyberReview Framework and Components

After the initial round of investigation, a prototype system called CyberReview has been developed. CyberReview is a web site where design documents are submitted and subsequently reviewed, and the resulting review documents are archived. A rich set of facilities are provided to facilitate the entire design review process. The system includes the following main components:

- *CyberReview Explorer*. This is the main component to which all the other components and facilities are attached.
- *Project Explorer*. This lists all the CDR projects.
- *Review Coordinator*. This component provides a set of facilities for the project manager to plan and manage the activities and resources involved in the design review process, in particular, for establishing the review committee, defining design documents, and preparing review documents.
- *Design Explorer*. This component is provided for the product development and design team to upload the product design onto the CyberReview database for future access. The review team uses the similar facilities to download design documents for review.
- *Comments Explorer*. This is an electronic discussion forum for review team members to submit their comments freely concerning the design documents submitted by the design team.
- *Review Explorer*. This collects all the online and offline forms and documents that are prepared during the review process.
- *Meeting Explorer*. This is a sophisticated module providing a variety of facilities to support holding review meetings for both the chairperson (project manager) and the team members before, during and after the meeting.
- *BOM Explorer*. This is a module associated with the design explorer. BOM is treated as a special type of design document required for design review when it becomes available. VRML files, comments and reviews may be directly related to BOM items.
- *VRML Whiteboard*. This is a whiteboard based on the VRML display of product features. This module would be necessary during the review process.
- *CyberReview Utilities*. In addition to the above main modules, WeBid provides a set of general utility facilities such as user registration, searching for necessary information, etc.

The review procedure using the CyberReview system in this case study generally includes the following main activities:

- With the help of the Review Coordinator, the project manager establishes a review team or committee and prepares review documents (proforma and procedure).

- The design team uses the Design Explorer to upload the design in the VRML format onto the database.
- Individual members in the review committee use the Comments Explorer to carry out their reviews by submitting comments and suggestions to the CyberReview database. This is generally asynchronous.
- With the help of the Meeting Explorer, a review meeting is called upon to resolve the comments from individual members.

3. System Design Considerations

Numerous factors must be addressed during the implementation of CyberReview facilities. Some of them are:

The industrial practices in design review vary widely from one company to another. The web-based collaborative product design review framework must have sufficient flexibility to incorporate and reflect good practices across the industrial spectrum.

Design review naturally involves 3D geometries of the product features in addition to other textual descriptions. It has been decided to employ VRML format in CyberReview for this purpose.

Comments made during the design review process are usually specific to the perspectives or positions of the 3D product drawings from which the comments have been made.

Design review is usually interactive and collaborative. Discussions and arguments are usually conducted based on topics or threads. The electronic forum is adopted in CyberReview for this purpose.

Design review can be either asynchronous or synchronised or both. Synchronization poses a great challenge in implementation.

Majority of the CyberReview facilities can be implemented based on various standard web and Internet constructs in relatively straightforward ways. The following considerations have been taken when implementing CyberReview:

- The standard 3-tiered architecture of web application has been used for building CyberReview framework.
- The CyberReview application servers and database have been deployed in the MS Windows 2000 environment.
- HTML and ASP (Active Server Pages) have been used as the basic language for implementing the application servers and clients, supported by Java and VB scripts when necessary.

- Java Applets and Servlets have been extensively used in the implementation of application servers and clients.
- A relational database is used. The connection to application servers is through ODBC and ADO (ActiveX Data Object). No direct connection is allowed from application clients to the database.
- CosmoPlayer is used for the VRML browser although other alternatives have also been experimented.

However, some complication has been encountered when implementing the synchronization mechanisms for the VRML Whiteboard and Meeting Explorer. The synchronization of the Meeting Explorer has been achieved by periodically refreshing the web pages. The time can be set at 1-10 seconds.

4. System Implementation

4.1 System structure realization

WWW-based Java solution is one of very important styles of the Internet computing model. With the help of Java programming, here, client-side computing can be implemented with the combinations of Applets and HTML files under the control of browsers and server-side computing with ones of Servlet, URL calls, or extended HTML files under the control of Java-enabled web server. It will also be available to integrate a background database, to develop drivers for networked devices, to connect with client-side local standalone applications, etc. So, it is the very important technology solution for the Design Review in External Enterprises.

Our system used a 3-tier database application that used Java Servlet and the Java Database Connection (JDBC). The application is partitioned into three tiers: user interface layer, the web server layer and the data store layer.

- **First Tier:** The first tier is a web browser, which serves as our universal client. In the first phase of the application, an HTML front-end was used for user-input and displaying the review comments, which submitted by the participator and stored in the database.
- **Second Tier:** The second tier of the application is implemented with a Web server capable of executing Java Servlet. The Java Servlet harnesses the power of JDBC to access the database to store/retrieve information as needed. A dynamic HTML page is generated by the Servlet based on the database results.
- **Third Tier:** The third tier is composed of our back-end database server. The database server

stores the information that is used by the application. Thanks to the JDBC API, the Servlet can access the database in a portable fashion by using the SQL call-level interface.

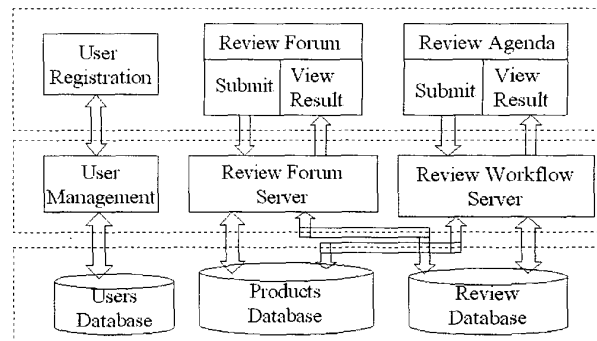


Figure 2. The framework for collaborative design review.

4.2 EAI Realization

4.2.1 Applet initializes VRML EAI

EAI (External Authoring Interface) is an API for connecting a VRML plug-in and external program [1, 5, 6, 9]. EAI allows for an external program to control VRML object in VRML plug-in, in our system, the review workspace is a VRML file, and the review Applet is an external program. The use of VRML EAI requires that several class be initialized when the connected Applets. Therefore the execution of collaboration Applet begins with the initialization. First of all, A browser class in initialized, which makes it possible for Applet to access the VRML plug-in used, then node calls in the VRML file, such as transform and touchsensor nodes are obtained from the browser class. There by, the VRML nodes attribute values can be referenced or change in the Applet while controls the VRML object. Change of node's attribute is called an event, There are two event: EventIn and Event Out. EventIn is an event that change and attribute from an external program. EventOut sends out a change attribute to an external program. Both events are derived form the node class. With the class initialization, is enable the transmit user input on the Applet to control VRML object to VRML plug-in.

4.2.2 Sending events

Once an instance of EventIn class is obtained, an event can be sent to it. EventIn is abstract class so the cast to appropriate EventIn subclass is needed. This subclass contains a method for sending events of given type. If a VRML scene contains the folowing node:

```
DEF SCALER Transform { ... }
```

the scale can be set to values (1.0, 2.5, 1.0) from Applet like this:

```
Node scaler = browser.getNode("SCALER");
EventInSFVec3f scale =
    (EventInSFVec3f) scaler.getEventIn("set_scale");
float sc[3] = { 1.0, 2.5, 1.0 };
scale.setValue(sc);
```

4.2.3 Reading events

Once an instance of EventOut is obtained, a value of event or exposedField can be read. Since EventOut is abstract class it needs a cast to appropriate subclass which contains method for getting events of given type. Current value of scale field can be read like this:

```
EventOutSFVec3f scale =
    (EventOutSFVec3f)
scaler.getEventOut("scale_changed");
float current_sc[3] = scale.getValue();
```

4.3 Applet-Servlet Communication

In our application, all the user's registration information, the position information got from the VRML EvenOut and EventIn message was encapsulated in a Java object. For example, in the meeting room, when a participator changes the position of the VRML prototype, the new position message was sent to the Servlet. Also, it is our desire for the Servlet to send the Applet an updated message list as a vector of Msg objects. This will allow the Applet to quickly and easily display the changing of the VRML. And the Servlet could add a choice position message to the database. Figure 10 displays the object interaction between the Servlet and the Applet.

4.3.1 Sending object from an Applet to a Servlet

The Applet sends a position object to the Servlet when a participator change the VRML in the COSMO player. In order to send a Java object from the Applet to a Servlet. We should make a Java object to be serializable, its class must implement the java.io.Serializable interface which allows an object to be flattened and saved as a binary file.

The Applet is actually sending a POST method to the Servlet. This client-side code fragment opens a URL connection to the Servlet URL. We inform the Servlet connection that we are sending output data over the connection and receiving input. Methods are also called such that the connection will not use cached versions of the URL. An important call in this code fragment is setRequestProperty(...). This method sets the content-type in the HTTP request header to the MIME-type application/octet-stream. The application/octet-stream

MIME-type allows us to send binary data. In our case, the binary data is our serialized Student object.

4.3.2 Sending object from a Servlet to an Applet

The Servlet can return an updated list of changed positions. The updated positions list is returned as a vector of position objects. When the Servlet returns the vector of position objects, there is no need to iterate through the vector and serialize each position object individually. The Servlet can simply serialize the entire vector in one step, since the class java.util.Vector also implements the java.io.Serializable interface. And the Servlet is now capable of storing the position message into the database.

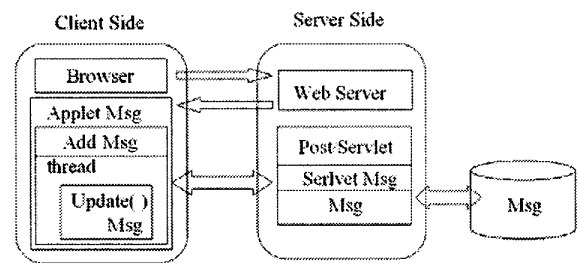


Figure 3 Applet-Servlet communication.

5. Case Studies

This section presents two typical scenarios of collaborative design review.

5.1 Scenario 1: Review Forum

The first typical scenario of collaborative design review is that team members are free to make their comments on a product design in asynchronous mode. Typical steps are as follows:

1. Participators access the URL of the product design review Web site individually.
2. The VRML file of the new product downloads to the different participators' browser. The participator chooses the part from the BOM of the product
3. The review list of this part was displayed, which include the author and the date. The user clicks on the hyper-link of the topic.
4. The contents of the review topic will display. The user can check the VRML by click on the "View VRML" hyperlink on this part.
5. The user can submit response to this topic by click on the "response to the topic" hyperlink.
6. In the response interface the user can define the view point of the part VRML file, input the

response topic and the content, then click on "submit". The review result will store in the review database.

7. The user can also create a new topic on the part by click on the "new topic" hyperlink.

In the new topic interface, the user can define the view point of the part VRML file, input the response topic and the content, then click on "submit". The review result will store in the review database also.

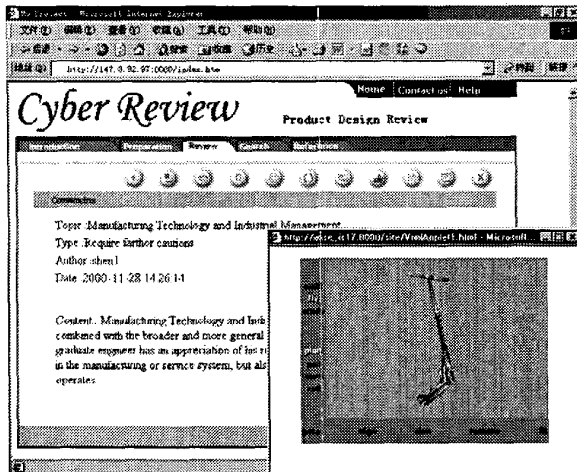


Figure 4. Review forum display.

5.2 Scenario 2: Review Agenda

The second typical scenario of collaborative design review is that team members hold a meeting to make their comments on a product design in the synchronous mode. Typical steps are as follows:

1. The chair enters the system to prepare the relative works of the review, which including select a product, and establish review policies. Then send the message to the relative participators.
2. After get the message, the predicators access the website, and enter into the review system.
3. The relative participators submit the review comments according the agenda.
4. When they meet the problem during the review, they can discuss in the meeting room. Usually, The chair control the view of the product's VRML making it display on all the participator at the same view in the same time. All the participators can view the same content and submit they idea on the topic.
5. At this time, the participator can also apply to create a new view and a new topic. This applying was sent to the chair first. The chair decided if

this topic was submitted to discuss. If the administer agree on the new topic, he will control all the user's browser display the same view of the VRML file. And they can discuss on the topic now.

6. At final, all the participators fill in one form in the browser to make a conclusion of this review.

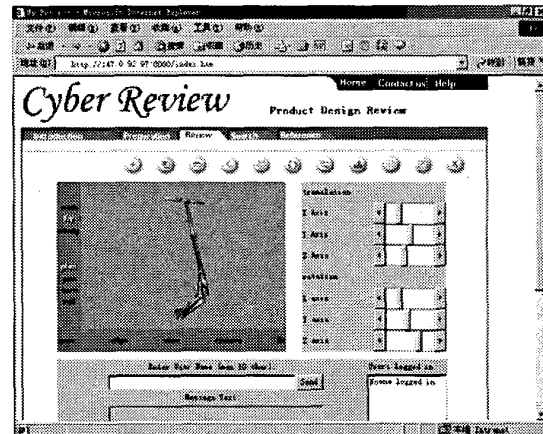


Figure 5. Meeting explorer.

6. Concluding discussion

The Web based collaborative review system is a prototype system whose ultimate goal is providing a platform for the product design review in the external enterprises. According to the design review procedure, the system allow the manager define the review agenda, and organize the review process easily. A VRML based synchronize meeting system was provided to support the product design review process. This PDR system is mainly distinguished by allowing it to run in the Web environment. This is important because the participators undertaking the product design review works can collaborate in a distributed and real-time environment without any additional client program.

The major contribution of this research is that it provides a framework of the design review on the Web. The framework may be used in such wide application areas as mechanical product design review, building construction design review, electronic components design review and software package design review. The usability of such a system would greatly increase as the network bandwidth improves.

Our further researches mainly focus on two aspects: the first is trying to integrate the review knowledge to the review system, making the review can be completed more convenience. The second is trying to integrated the

agent technology into the review system, to improve the system agility and intelligent.

7. Acknowledgments

Authors are most grateful to the Hong Kong Research Grant Council and Hong Kong University Committee on Research and Conference Grants for financial supports that made this research possible.

8. References

- [1] Ames, A.L., Nadeau, D.R., Moreland, J.L. (1996), *VRML 2.0 SourceBook*, Wiley, John & Sons, Incorporated.
- [2] Dhillon, B.S. (1996), *Engineering Design: A Modern Approach*, Richard D. Irwin, a Time Mirror Higher Education Group, Inc. Company.
- [3] Hardwich, M., Bolton, R. (1997), "The Industry Virtual Enterprise," *Communications of the ACM*, Vol.40, No.9, 59-60.
- [4] Hartley, J.R. (1998), *Concurrent Engineering: Shortening Lead Times, Raising Quality, and Lowering Cost*, Productivity Press.
- [5] Hartman, J. (1996), *The VRML 2.0 Handbook: Building Moving World on the Web*, Addison Wesley Longman, Inc.
- [6] Hunter, J., Crawford W. (1999), *Java Servlet Programming*, O'Reilly & Associates, Incorporated.
- [7] Hutchins, D.C. (1992), *Achieve total quality*, Cambridge: Director Books.
- [8] Ichida (1989), *Deizain Rebyu Jireishu*, Productivity Press, Portland, Oregon.
- [9] Lea, R., Matsuda, K., Miyashita K. (1996), *Java for 3D and VRML Worlds*, New Riders Publishing.
- [10] Schoonmaker, S.J. (1996), *ISO 9001 for Engineering and Designers*, The McGraw-Hill Companies, Inc.
- [11] Voigt, E.C. (1996), *Product design review, a method for error-free product development*, Productivity Press, Portland, Oregon.