

# The World-Wide Web As A Platform For Supporting Interactive Concurrent Engineering

Martin Hanneghan<sup>†</sup>

Madjid Merabti<sup>‡</sup>

Gary Colquhoun<sup>†</sup>

<sup>†</sup> School of Engineering and Technology Management

<sup>‡</sup> School of Computing and Mathematical Sciences

Liverpool John Moores University, Byrom Street, Liverpool, L3 3AF, UK.

E-mail: {m.b.hanneghan, m.merabti, g.j.colquhoun}@livjm.ac.uk

## Abstract

This paper describes the use of the Internet and World-Wide Web (WWW) as a means for disseminating information and enabling collaborative working in a Concurrent Engineering (CE) environment. It discusses the authors architecture for a concurrent engineering support environment, the CONCERT environment, and describes the implementation of a WWW-based interface to this environment.

## 1. Introduction

Concurrent engineering (CE) has been described in the literature under a number of synonyms including: simultaneous engineering, integrated product development, concurrent product and process design, life-cycle engineering, design for production, design for assembly, integrated and co-operative design, design fusion and producibility engineering. It now has the almost universally accepted definition of: "... a systematic approach to the integrated concurrent design of products and their related processes, including manufacture and support. This approach is intended to cause the developers, from the outset to consider all elements of the product life-cycle from conception through disposal, including quality, cost, scheduling and user requirements" (Winner et al. 1988).

Changes are inevitable during the design and development of any product. In a traditional sequential product development process, up-stream problems are fed back to the previous development phase to be corrected in an iterative fashion. The cost of this reworking can amount to a substantial proportion of the total development cost — (Sheldon and Perks 1990) report that around 80% of the total cost of a product is committed during the design and planning stages. Therefore, consideration of potential up-stream problems at this stage can help reduce costs. Concurrent engineering aims to eliminate the possibility of costly design errors by utilising multi-

discipline or cross-functional teams, i.e. teams with members from each functional area of the product's life-cycle — from conception to disposal. Improved communications and dramatic changes in working practices (Nickerson 1990; Trapp et al. 1992) have led to large reductions in product development lead time and overall cost (Coupland 1992; Dowlatshahi 1994).

Recent developments have led to a number of computer-based support environments to enable Computer-Aided Concurrent Engineering (CACE). These environments attempt to provide shared access to a common product information model over a computer network and so enable collaborative working.

We believe that concurrent engineering support environments should provide at least the following functionality:

- Allow transparent access to all the necessary tools and applications needed throughout the development process, e.g. CAD, CAM and project administration
- The ability to treat geographically dispersed resources as though they were local
- A communication sub-system for effective participation between all people involved in the project
- Facilities for controlling and tracking the project tasks and resources including people
- A consistent user interface even across multiple computer platforms and architectures
- The ability for the environment to be extended to include further tools and technologies as and when they become available

With these issues in mind we have proposed an architecture for a CE support environment which we have called the CONCERT (CONCurrent Engineering support) environment (Hanneghan et al. 1995b).

Organisations employing a CE strategy usually have team members and resources who are geographically dispersed throughout the company (or even the World) yet these people need to collaborate on projects and communicate on a regular basis. These people have a clear need to share information and making this process as efficient as possible can lead to significant cost savings.

The CONCERT environment attempts to address these problems by providing three fundamental components or *support services*. These are:

1. *A distribution support service* — this is to accommodate problems of geography and allow team members to co-operate in a distributed, client-server fashion

2. *A repository support service* — this is to enable team members to share a common repository (database) for any items (objects) created during the life-cycle of the product (this involves both storage and retrieval of objects within the repository)
3. *A Computer Supported Co-operative Working (CSCW) support service* — this allows collaboration and communication between team members.

The remainder of this paper describes the use of the World-Wide Web as a means of implementing the CONCERT environment and discusses some of the issues raised as a result of using the WWW for distributed, interactive client-server applications. Section 2 discusses our rationale for using the WWW while section 3 details the role of communication within a CE environment. A description of our implementation is given in section 4 and we give some conclusions about WWW-based development in section 5. Finally we describe some areas for future research in section 6.

## 2. Rationale for Using the World-Wide Web

The WWW is a relatively mature networked hypertext system. It is a distributed architecture on which to base simple client-server applications and a number of novel WWW-based applications are currently being used and developed to exploit this. Examples include on-line insurance quotations, credit ratings, stocks and share prices, Internet home shopping malls and even the purchasing of a new car.

The Internet is an exponentially growing global network which can be accessed relatively easily and cheaply by both large and small companies. There is no discrimination against who can join the Internet — it is not owned by any single organisation. This may be of special importance to companies in third world and developing countries who wish to use the Internet to apply a Concurrent Engineering strategy.

(Ellsworth and Ellsworth 1994) lists ten reasons why businesses are using the Internet and it is clear that these reasons have a lot in common with the goals of a CE strategy:

- Communication (internal and external)
- Corporate logistics
- Levelling the playing field — globalisation
- Gaining and maintaining competitive advantage
- Cost containment
- Collaboration and development
- Information retrieval and utilisation

- Marketing and sales
- Transmission of data
- Creating a corporate presence

The WWW can be considered as a medium for delivering *requested* information. In a CE environment, multiple members of a design team (also described as a virtual team) *produce* and *consume* documents: designers and engineers create new *designs* or proposals and submit them into a repository where they can be viewed, commented upon and reviewed or modified by other team members. These team members also need to communicate with each other to share and discuss ideas and problems (Siemieniuch and Sinclair 1994).

By enabling the Internet to be used as a collaborative design and working media we believe there is an opportunity to:

- Increase the level of co-operation through the greater awareness of the information that is available
- Increase the level of understanding of data through the use of hyperlinks to related documents
- Increase the ease with which support applications can be developed and then subsequently distributed and made available to team members within the given environment.

We have therefore produced the following set of design requirements for any application designed to support a concurrent engineering strategy:

1. The application should be distributed over the Internet using the World-Wide Web or other such mechanism
2. Documents and displays should be formatted, where possible using a dynamic hypertext-based interface such as that provided by HyperText Mark-up Language (HTML)
3. Documents should be automatically generated from source objects in an Object Database Management System (ODBMS) repository
4. Documents should be content-rich (i.e. they should accommodate the acts of navigation, searching and browsing) to enhance the understanding and reduce the complexity of the data in the repository
5. The application should have an appealing, intuitive user interface.

By fulfilling these requirements, we hope to provide a better environment for companies employing a Concurrent Engineering strategy.

## 2.1 Client-Server Development over the WWW

Traditional WWW-based applications operate in a client-server fashion. A client (also known as the browser software) will request information from a server machine using HyperText Transfer Protocol (HTTP). The server will then deliver a specific document to that client (see Figure 1).

In more detail, the process is as follows:

1. The client software requests a specific item of information (the *document*).
2. The server then processes the request and either delivers the document or executes a Common Gateway Interface (CGI) compliant script program that will generate the document. In this way, CGI scripts can be used to provide pseudo-dynamic documents on-the-fly. These documents are pseudo-dynamic because even though they are created dynamically their content is strictly of a static nature i.e. the document does not have the ability to change its own content as time progresses<sup>1</sup>.
3. Documents are delivered in either one of a number of predefined Multipurpose Internet Mail Extension (MIME) types (Borenstein and Freed 1993) or some *undefined* type. If the browser software is capable of displaying the document MIME type (e.g. it is HTML or ASCII format) it will do so. Otherwise it may be able to launch an auxiliary application to view the document<sup>2</sup>.

The link between the server and the user is terminated when the document has been delivered. This means that a user cannot modify a document and return the modified document to the server for other users to view.

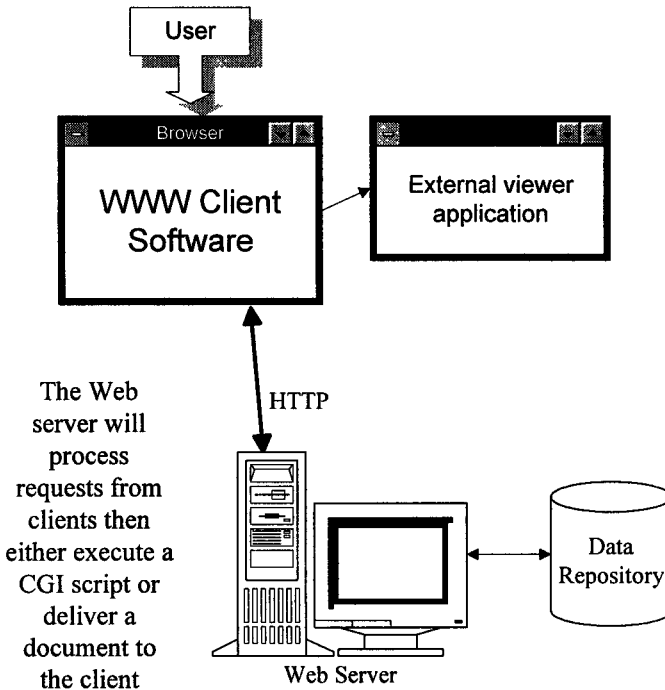
In this type of WWW-based application, CGI server scripts need to be optimised so that they run either as fast as possible or use as few resources as possible depending on whether priority should be given to the user or the server load. If the server machine is also serving other requests such as mail and printing, it may be more desirable to have the server scripts use as few system resources as possible therefore reducing the response times to user requests.

As a result of this, it has become apparent that there is a need for more control and processing responsibility to be on the client side thus reducing the load on the server.

---

<sup>1</sup> This shortcoming has been addressed by Netscape Communications Corporation who have implemented a non-standard way of making HTML documents automatically refresh their contents (i.e. request that the server re-send the document) by using a meta-command such as `<META HTTP-EQUIV="Refresh" CONTENT=5>`. When used in conjunction with a CGI script, this technique can deliver documents with changing content.

<sup>2</sup> This is dependent on the browser software and each user's local configuration of that browser — two separate users can safely and legally use different applications to view a given document type.



**Figure 1.** Client-server applications using the World-Wide Web

Clients typically use powerful machines anyway, so we can harness that power to improve productivity and increase resource utilisation.

The WWW therefore provides us with support for distribution in its present form but has no in-built support for co-operative working or the submitting of documents from the client to the server. It is therefore unsuitable for developing interactive and dynamic environments without first extending the means by which clients interact with the server. This is a recognised problem and a number of solutions have been proposed to address this.

## 2.2 Alternative Methods for Applications on the WWW

A number of new technologies have recently emerged to enable WWW clients to accept processing responsibility and provide interactive WWW-based applications. These technologies typically achieve this by extending standard HTML with new tags. If an existing (or new) WWW browser does not understand the new tag it will ignore it by default and hence the new tag will have no effect. This type of transparency is essential for open systems but it also means that WWW-based application developers who use these techniques have to ensure that the client

software accessing their WWW-based system must be capable of supporting the new tag. Examples of these technologies include Java applets, Weblets, JavaScript and Hush Tcl / Tk.

- *Sun Microsystems Java™ applets*

Java (Sun Microsystems 1995) is a dynamic, object-oriented language and environment. At the time of writing, a number of WWW browser applications including Sun's HotJava and Netscape's Navigator provide support for Java applications via a non-standard extension to HTML: the <APPLET> tag. This allows small applications (or applets) to be delivered over the WWW to be executed on the client machine.

Java applets are sent to the client by the server at run-time ensuring that each user has the same revision of the software at all times thereby creating an environment that has truly pervasive applications (see Figure 2). This improves the level of standardisation and allows a quicker route for application developers to distribute new versions of software or new prototypes.

- *Eòlas Technologies Weblets™*

This technology uses a technique similar to Java applets which the developers call Weblets (Eòlas Technologies 1995). Weblets are self-contained applications like applets but can be run either entirely on the client machine (i.e. local

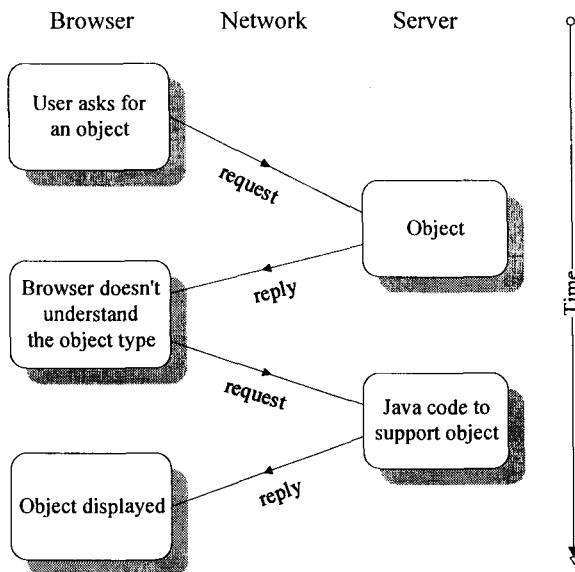


Figure 2. Java™ applet communication

processing) or on the server machine (remote processing) in client-server fashion. This technology utilises a non-standard `<EMBED>` HTML tag.

- *The JavaScript scripting language*

Netscape Communications Corp. in conjunction with Sun Microsystems has developed JavaScript to support the Netscape Navigator WWW browser (Netscape Communications Corporation 1996). This is an application programming interface (API) that allows scripting of events, objects and actions. It allows the WWW application developer to detect and act upon events such as start-ups, exits and user mouse clicks. The scripting language is based on Sun's Java language and can be used to glue HTML, special auxiliary plug-in applications and Java applets to each other. The syntax for this language extends HTML to include a new `<SCRIPT LANGUAGE="JavaScript">` tag.

- *Hush Tcl / Tk*

In common with the other techniques described here, this system (van Doorn and Eliëns 1995) defines a new `<HUSH>` HTML tag which allows the application developer to embed Tcl / Tk program code (Ousterhout 1994) into a WWW page. It provides the application developer with a simple WWW browser window in as little as four lines of Tcl code.

### 3. Communication in a CE Environment

We have already mentioned the need for support for communication and collaborative working within virtual teams but to fully describe the types of communication methods that are needed between team members in a CE environment we need to think in terms of life-cycle time scales. The time scale we are concerned with refers to the length of time that it takes to produce a finished product from its initial conception. The time taken will, to a large extent, dictate the level and speed of communication needed within the environment.

This can be clarified with some real life examples:

- In the production of a *daily newspaper* there is a very short life-cycle (typically less than 24 hours)
- In the production of *high street fashion clothing* there is a short life-cycle (typically measured in weeks — from *catwalk* to *shop*)
- In the development of *software* there is a medium to long life-cycle (typically in the region of months to years)
- In the development and production of a new *space shuttle* there is usually a long to very long life-cycle (typically many man-years).



In very short production life-cycles, there is an obvious need for immediate (i.e. synchronous) communication between team members. In longer production cycles, communication can be less immediate (i.e. asynchronous).

### 3.1 Computer-Mediated Communication Methods

Computer-mediated communication methods can be sub-divided into two classes: synchronous and asynchronous. Synchronous methods provide *immediate* feedback between two or more people. Examples include:

- Direct video link (videoconferencing)
- Direct voice link (audioconferencing)
- Telephony
- Direct text-based conferencing (terminal / teletype text link)

Note that these communications can be of a *one-to-one* or *one-to-many* nature.

Asynchronous methods provide *delayed* feedback between two or more people. Examples are:

- Electronic mail
- Bulletin boards (organisation wide)
- USENET News group (organisation wide or global)

Again, these communications are of a *one-to-one* or *one-to-many* nature.

The importance of computer-mediated communication should not be underestimated: results of recent studies have shown that electronic brainstorming is superior to traditional verbal brainstorming for large groups involved in exchanging ideas (Aiken et al. 1994).

## 4. The Concurrent Engineering Web Workbench

We have created a WWW interface to our concurrent engineering support architecture (the CONCERT environment highlighted earlier) which we have called the Concurrent Engineering Web Workbench (CEWW) (Hanneghan et al. 1995a). The CEWW uses NCSA's WWW server software running on a Sun Microsystems SparcStation 4 with Solaris 2.4. Storage within the environment is provided by the Postgres95 Object-Relational Database Management System (Yu and Chen 1995). We use the Java language to implement special applications within the environment and these will be discussed in this section. The environment therefore requires access to WWW browser software that supports Java applets.

## 4.1 Facilities Available in the Environment

This section describes the types of WWW-based application that are currently available within the CEWW. Java applets are very dynamic in nature unlike the largely static format of HTML. In practice this means that we can use Java to produce applications that are capable of responding to requests and actions in real time, thus giving the user immediate feedback using dynamic data.

The facilities that are currently available or targeted for implementation include:

- group decision support
- user-definable bulletin-board systems (information notice boards)
- user tracking (information-rich equivalents of the UNIX `finger` and `who` commands)
- multi-user synchronous text conferencing
- one-to-one user text communication
- electronic mail

These activities fall mainly into the area of CSCW tools. In addition to this we have support for WWW-based document editing. This allows a user to download a document and edit the contents on their local workstation. They then have the ability to re-submit the document to the server machine for subsequent re-editing<sup>3</sup> by themselves or other users.

### 4.1.1 Group Decision Support

In concurrent engineering teams group consensus is often used to solve problems. A facility is available the CEWW for team members to express a problem as a statement and have other team members electronically vote (i.e. agree, disagree or abstain) on the statement. The system can automatically collate results and display the outcome in the form of a graph. This information is retained within the repository for the lifetime of the project.

### 4.1.2 Project-wide Publishing via Bulletin boards

The CEWW has facilities for team members to post information to a project bulletin board. Team members are notified of any new messages on the bulletin board when they login to a project. Information in the bulletin board has a finite lifetime and will

---

<sup>3</sup> To accommodate multiple users, we enforce a version control system on documents so that the user is always editing a new version of an existing document. When editing is complete, the new version is checked into the object repository and its availability is publicised to all users.

expire after a given date<sup>4</sup>. It can therefore be used to provide notification of forthcoming events and other temporal information.

#### 4.1.3 Identifying Users on the WWW

It is possible using a suitable WWW server to force users to authenticate themselves using a username and password combination before delivering a requested document. This provides the server with the following details (amongst others):

- the designated username of the user (this is not the person's login username for the system from which they are accessing the WWW page but an agreed and known username used solely for accessing the WWW service.)
- the name and Internet Protocol (IP) address of the client machine which is attempting to access the WWW page, e.g. `ace.cms.livjm.ac.uk` and `150.204.51.2`. These two values can be used interchangeably since the machine name can be determined from the IP address using a Domain Name Service (Hunt 1993).
- the client software<sup>5</sup> (agent) that is attempting to access the WWW server.

When we force CONCERT users to login, we can retrieve the IP address and machine name, software agent name and their username and password. We also need to ascertain which project the user wishes to work on since the CONCERT environment supports multiple projects running concurrently so we request this information from the user by means of a drop-down list of all available projects. Using all these details, the database relation in Table 1 is populated.

**Table 1.** Database relation for login information

<i>Username</i>	<i>IP address</i>	<i>Login Date</i>	<i>Project title</i>	<i>Client software agent</i>
martin	150.204.51.2	31/3/96	Computer	BrowserX

<sup>4</sup> The information is not deleted from the repository. It will simply not be shown when the user requests a list of current notices. Notices can easily be made visible for long periods by setting a long expiry date for the message.

<sup>5</sup> It is important to know what this is because there are applications within the environment which run as Java applets and therefore require the client software to be a Java-capable browser. If the client software accessing the CEWW system is not Java-capable then we need to take some other appropriate action (at present we inform the user that they should upgrade their browser software before continuing and provide a list of prescribed browsers which are suitable for use.)

This information is logged in the database for the entire session, i.e. until the user logs out of the workbench.

#### 4.1.4 User tracking

One of the aims of the CONCERT environment is to increase awareness and improve the level of knowledge about other users in the environment. Finding out where a particular user is at any given time is extremely important in a multi-user collaborative environment. Each user accesses the system from a unique terminal. Each terminal has characteristics such as:

- the building / city / country that the terminal is situated in
- the organisation or company which owns the hardware
- the number of the nearest telephone to the terminal

The CONCERT system collects details of all the terminals that access the system in order to populate a database of this information. When a user logs in to the system from a terminal that the system doesn't know about, the CONCERT system will interrogate the user to determine information about the terminal from which they are accessing the system.

Within the system database is a relation holding the personal details of all team members including such information as name, job title and organisation, office address, electronic mail name, telephone and fax numbers and a graphic image. There is also a record of which project and task a given team member is currently working on. In addition to this we determine information about the terminals connecting to the system can therefore locate a particular team member logged onto the system. This information is available to all team members and, of course, the project leader in order to track, co-ordinate and exercise control amongst the team.

#### 4.1.5 Multi-User Synchronous Text Conferencing

This applet allows multiple users to broadcast text messages among a group of their co-workers. The following points summarise its functionality:

1. The applet starts up a separate conference application window (in addition to the normal browser window, to allow participants to access other data during the conference). This conference application then repeatedly interrogates the repository looking for new messages and displays them in the window.
2. Each conference has a unique title since there may be multiple conferences taking place at any one time.
3. Users specify which conference they wish to contribute to before submitting any messages. When a person joins a conference, details of *who* and *when*

are logged for future reference e.g. “*John Doe* has joined the conference at 10:21am.”

4. Any person can initiate a new conference but only the person who started a conference has the ability to terminate it, i.e. a given user will own the conference. When a conference is started, a message is broadcast to all users of the environment informing them that the conference is about to begin. The date and time that the conference began is logged for future reference.
5. The applet allows users to view a log of all messages sent by all users in the form of a scrollable text window.
6. Any messages submitted to a conference automatically includes the name of the sender and the data and time of the message.
7. Every message submitted is stored in a temporary file space on the server until the conference has concluded. All the messages are then collated together, formatted and logged into the repository as the minutes for the on-line meeting. This can then be retrieved by users within the environment at any time for consulting.

#### **4.1.6 One-to-one User Communication**

This is a similar applet to the one described above except that communication is between two team members only and the messages that are sent are not logged within the repository. It allows private discussion to take place in a synchronous manner. In addition to this there is access to electronic mail to allow asynchronous communication with team members who may be temporarily unavailable.

## **4.2 Working on Objects**

Objects (or items) within the repository can be anything that can be created during the entire life-cycle of the product. In the CONCERT environment, users need to submit new items and work on existing items already placed into the repository. Since the WWW does not provide us with a method of submitting objects from a client to a server, we have devised the following methods to accommodate this service.

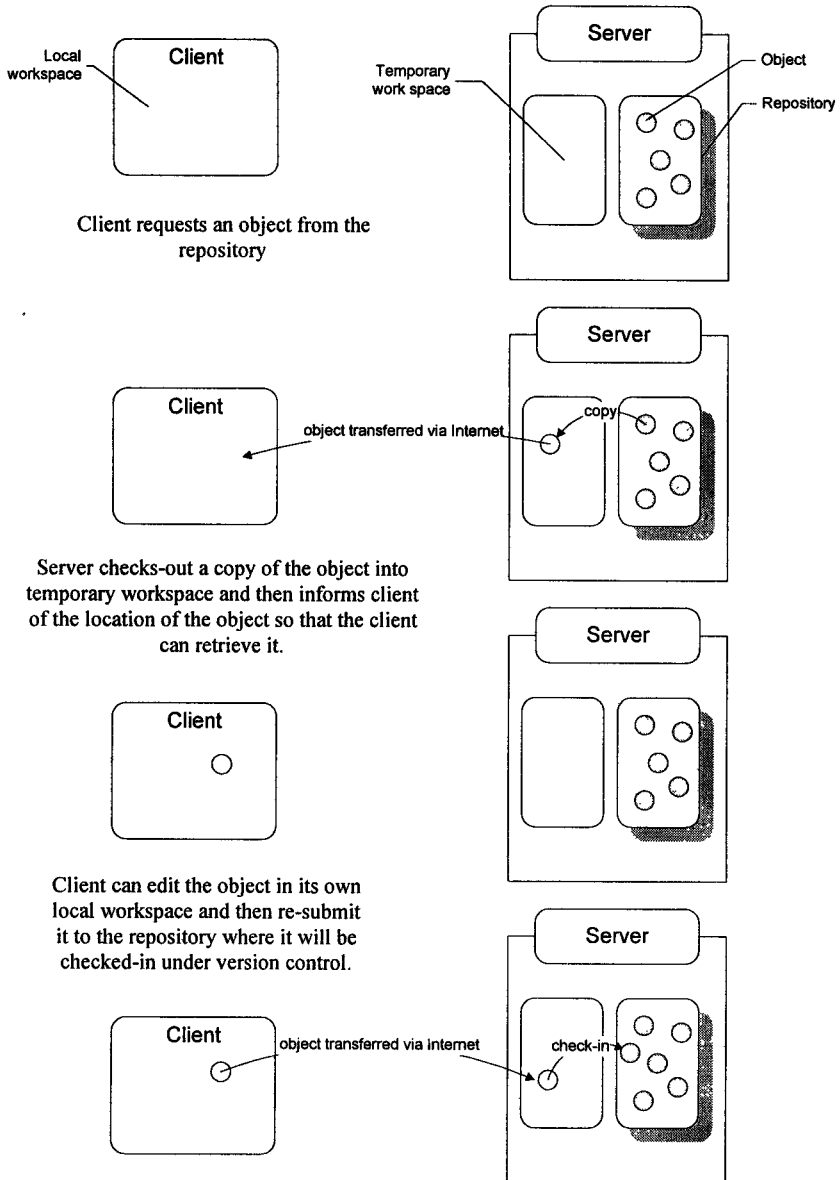
### *Working with existing objects*

The technique we use for allowing objects to be transferred from the server to the client and back to the server again is shown in Figure 3. The process is as follows:

- a team member requests an existing object
- the server checks out a copy (new version) of the object from the repository and places it into a known file space where the client can access it
- the client then retrieves the object into its local file space where the team member can edit it

- when the team member has finished editing the object it is transferred back to the server and checked back into the repository under version control

In Figure 3 we have shown the repository residing on the server machine. In practice, the repository may be distributed on another remote machine (as is the case in our implementation) but the same principal applies.



**Figure 3.** Working on existing objects within the repository

There is much debate over the security aspects of WWW-based transactions at present. Encryption and firewalling offer two approaches to security. In the highly competitive design and manufacturing world, organisations do not want their new designs being stolen by their competitors before they are released and so encrypting the data that is sent via the Internet is considered as a prerequisite for any environment. Firewalling sensitive data (i.e. the repository) is another technique that can be used to exclude access from outside organisations or only allow specific network sites to access the system. Since the CEWW controls how data is transferred between the user and the environment, it is relatively easy for us to incorporate a level of security into the data transfer process.

### *Creating new objects*

The creation of new objects uses a similar technique to that described above:

- a team member will create an object (either on their own local machine in private file space or on some remote machine) that they wish to store in the repository
- the team member notifies the system that the new object exists and transfers it to the server machine's temporary file space
- the server will then log the new object into the repository under version control and then publicise the fact to all team members working on the same project.

## **5. Conclusions**

This paper has described a WWW-based application to support organisations employing a concurrent engineering strategy. The Concurrent Engineering Web Workbench provides facilities for enabling virtual teams composed of all the people associated with the design of a product to collaborate and share information using the World-Wide Web.

Although the Internet is set to be important in all areas of industry, it is not without its faults. The biggest concern at the moment is that of Internet security. The British Computer Society Security Committee has summarised some of the key security issues as the fact that there is no strong central control, service availability guarantees do not exist, delivery guarantees are not available and there is no native confidentiality (BCS Security Committee 1995). Delivery guarantees require some method of hand-shaking so that each party acknowledges receipt of information. The CSCW service within the CONCERT environment provides a mechanism for this protocol. Confidentiality is catered for by an separate two-stage encoding-decoding process. Any data that is submitted to the central repository first has an encoding process applied and is transferred over the Internet in encrypted form. The CEWW

server then decodes and verifies the data upon arrival. If the Concurrent Engineering Web Workbench is applied to an *Intranet* or private internal network then the issue of central control can also be addressed.

Another problem we have found is with authentication. WWW servers currently authenticate users based on a username and password combination and a unique IP address. This brings with it the requirement that users should not leave their terminals unattended once they have logged in since the only way of validating a user is by determining their machine address. Once a valid link has been established, the terminal is open to abuse or misuse from third parties. One way to overcome this would be to periodically force users to re-authenticate themselves although this may prove to be a distraction to most users.

In this paper we have described our use of the Java language for delivering applications over the Internet. Since this is a relatively new language it is therefore subject to change and revision. The success of Java as a standard Internet programming language is not guaranteed and as we described, there are other languages available that are poised to take its place if it fails.

We believe that the Internet provides a medium on which to build portable, client-server applications on a global scale. By allowing controlled access from clients, who may be on opposite sides of the globe, we hope that an improved level of co-operation can be achieved. Engineering information is typically locked into the organisation's Local Area Network and is forced to undergo conversions and translations whenever data is needed outside the organisation for example by external consultants. By allowing the consultants to become clients to the CEWW, the data can be accessed remotely with relative ease.

## **6. Future Work**

The CEWW is an ongoing research project and the authors would welcome any comments about the system or requests for additional features within the environment. In particular, there are a number of areas which we would like to address in the near future.

We intend to conduct performance tests to evaluate the rate at which team members can retrieve objects from the repository. This is a function of the repository management service, the network traffic and the server / client load. We will also look into replicating the repository on a number of network sites to improve access times to items within the database. This will mean incorporating complex replication controls to ensure that data is consistent between users.

The CSCW tools available in the environment are only a small section of the range of CSCW tools currently available elsewhere. In particular we believe that screen sharing tools and non-destructive collaborative authoring tools would be beneficial to a virtual design team. We would also like in the future to be able to introduce a



facility to enable videoconferencing between team members using the environment. Developments in the field of WWW-based video transfer, such as the work undertaken as part of the *VideoMosaic* project (Campbell et al. 1995) and the wider availability of broadband networks could provide progress in this area.

Other areas of interest that could be incorporated into this research in the future includes the use of VRML (Virtual Reality Modelling Language). VRML (Bell et al. 1995) is a language designed to enable 3-dimensional worlds to be composed and displayed on-screen. Users can fly-through and navigate around these virtual worlds and point and click on hyperlinked objects. In its most basic form, this technology could be utilised by members of a design team to visualise and interactively review CAD data models. A more dynamic use would be the visualisation and navigation of complex data structures such as the data stored in the concurrent engineering repository that is described in this research. With the ever increasing processing power available to end-users nowadays, VRML is poised to become the standard way of navigating though objects on the Internet. However, the widespread adoption of VRML amongst the CAD community seems unlikely until there is a greater availability of CAD data model to VRML translation tools.

## 7. References

Some of the works referenced in this paper can be found on the World-Wide Web. Where this is the case, the Uniform Resource Locator (URL) for the document is given in the form: *http://website/document*.

- Aiken, M., Krosp, J., Shirani, A., and Martin, J. (1994). "Electronic Brainstorming in Small and Large Groups." *Information & Management*, 27(3), 141-149.
- BCS Security Committee. (1995). "Internet Security." *The BCS Computer Bulletin*, 7(6), 16-17.
- Bell, G., Parisi, A., and Pesce, M. (1995). "The Virtual Reality Modeling Language: Version 1.0 Specification." , <http://vrml.wired.com/vrml.tech/vrml10-3.html>.
- Borenstein, N., and Freed, N. (1993). "MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies." , <http://www.oac.uci.edu/indiv/ehood/MIME/1521/rfc1521ToC.html>.
- Campbell, R. H., Tan, S., Chen, Z., Li, Y., and Xie, D. (1995). "VOSAIC: Continuous Media Browser." , <http://choices.cs.uiuc.edu/Vosaic/Vosaic.html>.
- Coupland, J. W. (1992). *Concurrent Engineering: An Information Pack*, IEE Technical Information Unit, London.
- Dowlatshahi, S. (1994). "A Comparison Of Approaches to Concurrent Engineering." *International Journal Of Advanced Manufacturing Technology*, 9(2), 106-113.

- Ellsworth, J. H., and Ellsworth, M. V. (1994). *The Internet Business Book*, John Wiley & Sons, New York.
- Eòlas Technologies. (1995). "Eòlas Technologies Home Page." , <http://www.eolas.com/>.
- Hanneghan, M., Colquhoun, G., and Merabti, M. (1995a). "The Concurrent Engineering Web Workbench." , <http://ace.cms.livjm.ac.uk:1995/>.
- Hanneghan, M., Merabti, M., and Colquhoun, G. (1995b). "The Design Of An Object-Oriented Repository To Support Concurrent Engineering." OOIS'95 - Proceedings of the 1995 International Conference on Object-Oriented Information Systems, J. Murphy and B. Stone, eds., Springer-Verlag, London, 200-215.
- Hunt, C. (1993). *TCP/IP Network Administration*, O'Reilly & Associates, Sebastopol, CA.
- Netscape Communications Corporation. (1996). "JavaScript Authoring Guide." , <http://home.netscape.com/eng/mozilla/Gold/handbook/javascript/index.html>.
- Nickerson, D. P. "Introduction to Concurrent Engineering in Design." *Electro/90 Conference*, 174-179.
- Ousterhout, J. K. (1994). *Tcl and the Tk Toolkit*, Addison Wesley Publishing Company.
- Sheldon, D. F., and Perks, R. (1990). "Designing for Whole Life Cycle Costs." *Journal of Engineering Design*, 1(2).
- Siemieniuch, C. E., and Sinclair, M. A. (1994). "Concurrent Engineering and CSCW: The Human Factor." Computer Support for Co-operative Work, K. Spurr, P. Layzell, L. Jennison, and N. Richards, eds., John Wiley and Sons Ltd, Chichester, 111-125.
- Sun Microsystems. (1995). "Java (TM) - Programming for the Internet." , <http://java.sun.com/>.
- Trapp, G., Lawson, M., and Burkett, B. "Workshop on Concurrent Engineering." *CAD and Engineering Workstations '92 and Business Graphics '92 Conference and Exposition*, 766-792.
- van Doorn, M., and Eliëns, A. (1995). "Integrating Applications and the World-Wide Web." *Computer Networks and ISDN Systems*, 27(6), 1105-1110.
- Winner, K. E., Pennell, J. P., Bertrand, H. E., and Slusarzuk, M. M. G. (1988). "The Role of Concurrent Engineering in Weapons System Acquisition." R-338, Institute for Defense Analysis, Alexandria, Virginia.
- Yu, A., and Chen, J. (1995). "The POSTGRES95 User Manual Version 1.0." , Computer Science Division, Dept. of EECS, University of California at Berkeley.