Sharing CAD Applications

M. Re P. Cremonese
Finsiel S.p.A.- Tecsiel Business Unit
Via Santa Maria, 19 - Pisa, ITALY 56126

Abstract

The work presented in this paper has been partially funded by the EC under the TEN-IBC projects B1013 and B2013 "IDEA" and partially funded by Finsiel. The technology of shared window systems is presented and its application to the special requirements of design centers in the car industry is addressed. The work focuses on the issues of market availability, network access, and bandwidth requirements providing also an evaluation of overhead introduced by the protocol stack adopted to perform the end-to-end communication.

The technologies presented in this paper have later been esperimented and evaluated not only over Ethernet LANs, but also over national and international PEAN links. The observed values show that the experimented CAD packages can be effectively used alone over 2 Mbs links in a point-to-point configuration also considering the typical big size of CAD models used in the car industry.

1. Introduction

The European car industry is currently looking for appropriate evolutionary changes in order to be competitive with the rest of the world. The main directions where improvements are key to success are: costs reductions, quality improvements, and time reductions.

Referring to the time aspect, a study carried out by the Harvard Business School at the end of the eighties, showed that the production cycle of European car industries was almost completely sequential, in opposition to what was going on in Japan where much parallelism was present: as a consequence, the average cycle time for a car was 42.6 months in Japan versus 62.8 months in West Europe and USA. During recent years the European car industry has modified its approach in order to be more competitive, mainly by means of an increased parallelism and by assigning concept and development services to specialised firms; however, the cycle time gap with respect to Japan is still present (less than what reported above) and constitutes a major point of competition in the car industry.

Another crucial issue is communication. Since the design of a car is now often assigned to specialised external firms, this involves in turn an high amount of communications, especially in the preliminary phase of a project, where the style and design of a car has to be drafted. Communications include travels for meeting, extensive data exchange of huge CAX (i.e. CAD/CAM/CAS/...) files, and its costs can be approximated to 10% of the total cost of a complete design process.

Having this in mind it is clear how CSCW (Computer Supported Co-operative Working) tools may be of great interest provided that they are available today and their usage is economically convenient. Video conferences may be useful in reducing travels and meetings, with a consequent reduction of costs and time. But especially the possibility for remote designers to jointly view and edit CAD data by means of current applications such as CATIA or CADDS5 is of particular interest to the design centers. These tools may dramatically reduce costs by requiring less travelling among designers; they can also improve quality by providing better access to experienced designers and by improving communication and access to information; they can also reduce time and facilitate parallelism and inter-relationships between different phases of a project by managing them in real-time independently of user location.

Additionally our users were interested in having satisfied the following requirements:

1. Network Transparency
2. Integration among different services
3. Safeguard of existing Hw/Sw
4. Integration with existing CAD applications
5. Products implementing standards for CSCW
6. Future application portability

Of the above list, only items 1, 3, and 4 were possible to achieve with the current technology.

This paper concentrates on the technology of *shared window systems* applied to the world of car industries and especially design centers, where the issue of bandwidth becomes extremely important due to the demanding applications involved. Finsiel has been participating to this work initially as task leader for the specification of the application environment, and now as task leader for the evaluation of the monitored data.

Chapter 2 introduces the basic concepts of shared window systems as opposed to collaborative-aware systems. Chapter 3 describes the basic functionality of some shared window systems existing on the market. Chapter 4 provides some indications on the payload requirements of such systems in the context of the car industry. Chapter 5 describes the network configuration of IDEA, a TEN-IBC project aimed at experimenting the technologies described in this paper; additionally chapter 5 makes an estimate of the bandwidth required by such systems taking into account all the protocol overheads. Chapter 6 summarises the main issues and criticality

2. Shared window systems

The technology focused in this paper, i.e. shared window systems, provides the ability to synchronously share in real-time an application among two or more users interconnected by means of some LAN/MAN/WAN network [1].

Basically, there are two different approaches for obtaining this result, i.e.:

- "*collaborative aware*" applications, i.e. applications such as video conference or teleconference applications which have been explicitly designed for a collaborative environment;
- "*shared window systems*", using collaborative-naive applications which have been designed for single users: collaboration among different users is provided by means of an extended window system.

Currently available shared window systems are based on the X11 standard window system. The common idea at the base of such systems is to intercept the traffic between an X client and an X server by means of an "extended server" distributing the traffic from a client to multiple servers as well as from multiple servers to a single client. This process has to take into account the different characteristics of the X servers involved in the conference, requiring the shared window system to perform some adaptation of the device-dependent part of the X11 primitives.

Please note that the shared application is actually running on a single site, i.e. the site which started the application on the local client; analogously the data is not distributed and remains physically read or updated from its original location. Moreover shared window systems actually share windows rather than applications; this implies that your application may run on machine A (see **figure** 1), its output may be displayed on machine B (by setting the DISPLAY environment variable), where a shared window system provides shared access to the application. This characteristic is particularly interesting since it allows to share different applications running on different hardware platforms by means of a single shared window system.

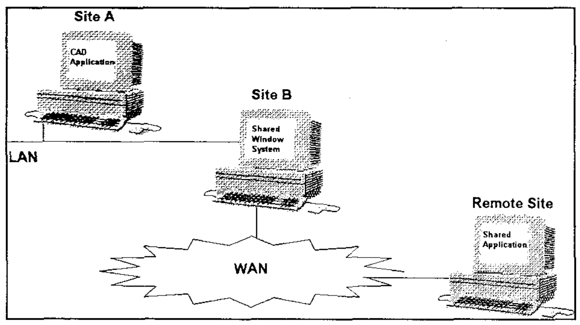

Fig. 1 Remote sharing of a CAD Application

In addition to the above functionality, shared window systems provide tools for controlling the conference, such as starting a conference, joining a conference, starting a tool, give/take the token, etc.

Even though both application-aware and shared window systems provide collaboration tools, they usually differ for different aspects, including technology, efficiency, and functionality. Some of the most important differences are:

- Shared window systems have the great advantage to be based on a common support tool: any existing single-user application is transformed into a multi-user application (provided that it satisfies certain requirements that are listed herein after) without any additional expense or effort, by using the shared window system;
- Collaborative-aware applications have built-in user interfaces providing access to the collaborative tools (i.e. release the token, add a user in the conference,

etc.); shared window system have a separate window with a generic user interface providing collaborative tools;
- Certain tools, such as video conference, are non-sense for a single user, hence they will only be available as collaborative-aware applications;
- Collaborative-aware applications can optimise their efficiency, especially in terms of bandwidth requirements, i.e. the sites involved in a conference can exchange messages containing commands instead of exchanging data with the result of the command.

In the case of CAX applications the only viable commercial solution today is, especially for industries, to use shared window systems. In fact, CAX applications are usually complex and demanding applications, requiring expert users in order to fully exploit the potentialities of the tool: hence it is not possible to use prototype or public domain collaborative-aware solutions. On the other hand there is not any product available on the market providing collaborative aware applications that could be compared with the graphics capabilities provided by existing CAX applications, such as for example CATIA or CADDS5.

The following aspects must be taken into consideration if you want to share CAD applications:
- first of all the application to be shared must be a pure X11 application
- using X11 requires a reliable transport protocol beneath, typically TCP
- the available network bandwidth must be appropriate, according to the size of CAD models and to the number of simultaneous users (see paragraph 4 on payload requirements)
- our experiences in the IDEA project have successfully demonstrated the sharing of CATIA and ICEM; some experiments with other CAD packages claiming full support of X11 have been perfomed without success
- some input interfaces are not allowed when sharing applications, e.g. DIALS, LPFKs, and TABLET.

A number of projects is currently being performed in Europe about shared window systems: examples are the TEN-IBC project "IDEA", especially dealing with the usage of these technologies in the co-operative environment of industrial design for the car industry, the RACE project "CIO", and the DeTeBerkom project "Teleservices".

3. Available tools

Many products are available, both as public domain and commercial packages, providing shared window systems. Apart from the Xy public domain software providing multicast transport connections (which in turn needs a modified version of the X11 protocol), the products do not differ very much in their functionality. However, some apparently minor differences may take a major role in deciding which is the most suitable and usable solution in a co-operative environment.

XTV (X Teleconferencing and Viewing) is a public domain (ftp.cs.odu.edu) distributed system developed by the Old Dominion University and the University of North Carolina at Chapel Hill. A process called the "packet switch process" (PSP) runs on the site where the client application is running; the PSP is in charge of distributing the output of the client application to all the remote servers. On each server site, a "packet translator process" (PTP) is in charge of adapting the messages received by the PSP by correcting the resource IDs; afterwards the PTP sends the modified messages to the X server for local display.

XTV provides the following functionality:
- whiteboard (point, type, hand write, copy, paste, ...)
- chat panel
- join a conference (latecomers are allowed in the conference)
- generate control output messages for packet statistics
- add a tool (either running on a local or on a remote machine)
- request/release control of a tool
- terminate/join a tool
- drop participant

Another public domain tool is the **Xy** tool developed by Gero Hoffmann at the Technical University of Berlin (gh@prz.tu-berlin.de). Xy differs from the other shared window systems in its implementation of the Xmc protocol, a multicast version of the X11 protocol. Xmc defines the protocol between an Xy Server (which is running at the location where is the application client), and the Xy Agents (running over the multiple remote sites where the users are located). The Xy server is mainly a standard X11 server, with the device-depend part replaced by new code generating the Xmc protocol; on the other side, an Xy Agent converts the Xmc protocol in the standard X11 protocol required by the X server (it mainly converts the resource IDs used in the multicast protocol in the equivalent ones of the local server). At the transport level, Xmc uses MTP (Multicast Transport Protocol), with the exception of some phases of the connection when a point to point connection is required: in these phases Xmc uses TCP since MTP is not able to use an established multicast connection in order to establish a point to point connection; neither it is able to address a sub-group of the multicast connection.

X/Telescreen is a commercial product available since 1993 by VisualTek Solution, running on Sun, SGI, and DEC workstations (availability on IBM RS6000 is announced). It may work in a "Polite" mode or in "Anarchy" mode. In Anarchy mode any user who wants to get control of the token may do it, whereas in Polite mode, you have to request the token. Anyway, when the user has control of the token, he gains control over both the application and the mouse (only one mouse is allowed). Please note that multiple mouse can be very useful especially when dealing with complex objects such as in the case of co-operative design in the car industry.

According to VisualTek documentation, the major difference of X/Telescreen with respect to other similar packages is its reliability, due to the fact that the product has been developed from the "ground up" as a commercial application instead of a research application that evolved.

ShowMe 2.0 SharedApp[1] is a SUN software that allows users to run any X-based application simultaneously on multiple displays and over local or wide-area networks. Features include shared input control, multipoint sharing of applications, Motif GUI, shared pointers, audio support (ShowMe Audio is included in Showme SharedApp); additionally it may be integrated with the rest of ShowMe 2.0 tools including white board support, video conference, and whabi support. The base technology for ShowMe SharedApp was acquired through a Technology Transfer between SunSolutions and Sandia National Laboratories.

SharedApp requires a SPARCstation(TM) system to initiate the session and Solaris(R) 1.x or 2.2 or later operating environment. It has been targeted for applications that use colour and hence it does not support monochrome systems. Ports of ShowMe have been publicly announced and are underway for HP-UX, IBM AIX, Microsoft Windows, and Solaris X86.

HP SharedX[2] is the proprietary easy-to-use HP product providing sharing of X11 release 3 or 4 based applications. Among its characteristics are: multiple windows can be shared to several machines at one time, tele-pointers (i.e. "rigid" pointers) can be placed on a shared window to focus on a certain part of the window, OSF/Motif based user interface. HP claims that HP SharedX is a fully tested and supported product from HP, in opposition to the public-domain software packages. Please note again that SharedX only allow "tele-pointers" instead of full-purpose multiple pointers which can be useful in complex co-operative environments.

[1](c) 1993 Sun Microsystems, Inc.
[2](c) 1993 Hewlett-Packard Company

4. Payload requirements

Some preliminary experiments have been performed during the first phase of project IDEA - B1013 in order to understand the expected payload required by a co-operative shared window system in industrial design environments. Obviously, bandwidth requirements mainly depend on the kind and size of application and data to be shared as well as on the commands issued by the user. The preliminary tests done so far have been performed by means of a monitoring program called Xscope (X Window System Protocol Monitor), a program that monitors connections between an X server and an X client, printing the contents of each request, reply, error, or event that is communicated between the server and client.

The tests have been performed by activating the CATIA software with a sample 3D object consisting of a mathematical model of a car component composed of 3.564 elements on a couple of IBM RS6000 workstations acting as client and server, respectively. **Figure 2** shows some of the results obtained with some sample user command, such as starting the application, activate all the layers of the current model and generate their graphics, perform a 3D Zoom or Rotation, and Exit the CATIA program. In the example, the bandwidth required for a typical operation such as a rotation, a zoom, or a pan, required approximately 3 Mbits. If you consider that such operations may be activated by the user by means of continuous dial inputs, you could see the traffic growing very fast, as an example to 15 Mbits/sec for a rotation of the dial input involving 5 different rotation values.

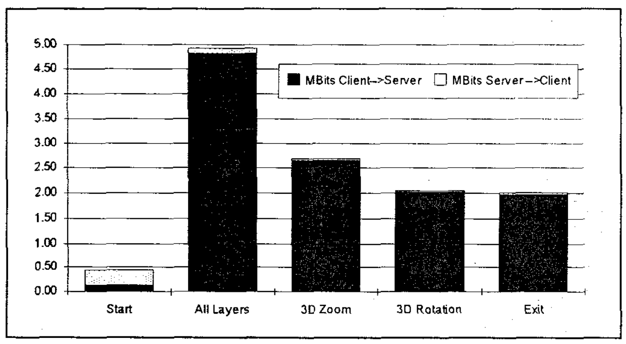

Figure 2 CAD Application Data Exchange

Additionally, the above numbers only represent the payload of the application, and do not take into account all the network protocols needed to use the shared window system in a wide area network, as well as the overhead introduced by the shared window system itself. The final figures will be obtained in the second phase of the IDEA project, where network traffic will be monitored in a wide area network based on the European ATM Pilot.

5. Network access and protocols overhead

The network selected by the IDEA project to experiment the sharing of CAX applications is the ATM Pilot between Italy (Turin) and Spain (Barcelona), and it is shown in **figure 3**. A user located in Turin is able to run CAD applications and to co-operate (i.e. share the same application) with another user located in Barcelona; the LAN interconnection is performed via ATM [5].

A workstation between the Turin Ethernet LAN and the local ATM performs routing functionality providing access via a UNI TAXI [2] (100Mbs) to the ATM local switch. The private ATM is then connected to the public ATM via UNI DS3/E3 (34Mbs). Additionally, since Turin is not a PEAN node, this last link is provided by a direct connection with Milan (the closest ATM node).

On the other side in Barcelona the workstations are connected to an Ethernet LAN connected via a WS router to the public ATM. The WS router is connected via an ATM board with UNI TAXI (100Mbs) while the Public service offered is a UNI-PDH DS3/E3 (34Mbs). Because of incompatibility of the interfaces, an adapter has to be added to perform the adaptation of the multifiber TAXI with the unimodal fiber provided by the public service.

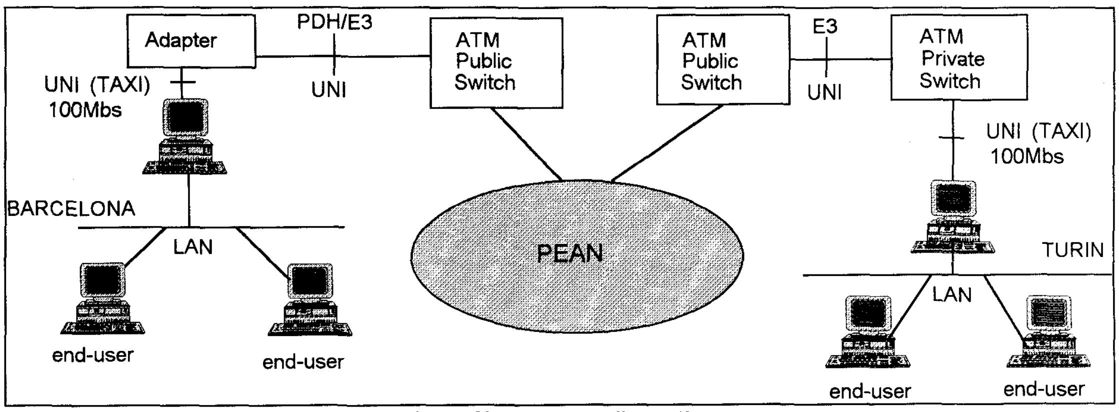

Fig. 3 Network configuration

The protocol stack that will be experimented during the trial mainly consists of TCP/IP on top of ATM. However, in order to allow interworking between IP and ATM, a convergence sublayer providing IP over ATM is needed. To this end, the solution adopted by most of the available products is the one proposed by IETF (RFC1483 [3] and RFC1577 [4]), encapsulating IP packets within the AAL-PDU. This is performed by providing an SNPA (Sub Network Point of Attachment) to IP by adding the LLC (IEEE802.2) header to the IP packets. With respect to the OSI model, LLC plays the role of SNDCP (Sub Network Dependent Convergence Protocol) and AAL plays the role of SNAcP (Sub Network Access Protocol). AAL5 (RFC1483) has been selected to access ATM: this provides a lighter SAR with respect to AAL3/4 providing bigger CPCS PDUs. The final protocol stack is shown in **figure 4**.

TCP/UDP	Transport
IP	Network
LLC	SubNetwork
AAL-CPCS	ATM Adaptation Layer
AAL - SAR	ATM Adaptation Layer
ATM	ATM

Fig. 4 Protocol stack

The MTU (Maximum Transmission Unit) for IP packets is 9180 octets according to RFC1483. This corresponds to TCP packets with a payload of 9140 octets. **Table 1** shows the overhead introduced by the involved protocol layers (from TCP to ATM), ending up with a total overhead of 1036 octets, i.e. more than 11.3% of the original payload of 9140 octets: please note that this is the minimum overhead introduced by the TCP/IP over ATM protocol stack. It is trivial that the large overhead comes mainly from the ATM layer.

If a typical CAD operation involves an exchange of 3Mb of data (see figure 2), we can suppose that the IP-PDUs sent across the network shall always have the length of the defined MTU. This is very similar to the reality, being TCP a stream-oriented protocol. Hence, according to table 1, the corresponding minimum overhead introduced by the protocol stack during a typical CAD operation is 356Kb.

Layer	Payload (in octets)	Overhead (in octets)
TCP	9140	20
IP	9160	20
LLC	9180	8
AAL-CPCS	9188	28
AAL-SAR	9216	0
ATM	9216	960
Total overhead: 1036 octets (11.3% of the original payload)		

Tab. 1 Min. Overhead introduced by TCP/IP on ATM

6. Conclusions

The paper shows the technologies available to share CAX applications in a local or wide area network environment.

Shared window systems are particularly well suited for the car industry where the complexity (and costs) of a CAX system is such that it is preferable to use existing collaborative-naive applications instead of new collaborative-aware applications.

A number of tools are available on the market, both as public domain and as commercial packages, but no standard is available, i.e. users using these tools still have to face different user interfaces, paradigms, functionality, management of the conference. This is a key issue for car industries, since it could decide to wait for an appropriate de-facto or de-jure standard before investing money in this new technology.

The greatest majority of existing products are based on unicast transport, and this corresponds to the most typical situation, but multicast transport looks promising when more than two designers have to co-operate.

The payload required by CAX applications in this environment is in the order of 3 to 5 Mbits for a single interactive user command; this payload approximately grows up at least to 3.4 Mb to 5.6 Mb as a consequence of protocols overhead.

The following table summarizes the peak aggregated throughput values observed on the LAN in the national and international case:

Service	International (2Mbs)	National (4Mbs)
Audio	0.4	0.4
Video	2.4	2.2
Whiteboard	1.3	3.3
CAD Sharing	1.2	3.5
PC Tools sharing	1.6	3.1
File Transfer	2.1	3.7
A/V+CAD Sharing	1.5	4.0
All Services Available	2.2	3.6

Tab. 2 Maximum Aggregated Throughput in Mbit/sec

Another main issue is the cost of communications: at the time of writing this report the PEAN tariffs are still not known, hopefully they will at the time of publishing this paper. The bandwidth estimates provided in this work can be one of the factors on the basis of which the user can make a cost/benefit analysis to evaluate the convenience of using these technologies.

The IDEA project is experimenting the use of these technologies over the PEAN, and the final result of this activity will be a quantitative and qualitative evaluation of the trial which will be performed under the leadership of Finsiel: we look forward to this next phase because it represents a major monitoring activity of shared CAD applications in a broadband WAN environment.

Acknowledgments

We would like to thank the members of the IDEA consortium for their support, and especially Giovanni Favilli from Finsiel, G. Savino and V. Vigna from Italdesign, M. Scamuzzi from Diseno Industrial, G. Venuti, A. Gaved, and M. Fossa from Cselt, R. Montero from Telefonica I+D, J. Casanova from Seat, S. Catellani from Bugatti, and F. Almerico from Tecnation.

References

[1] Karsenty, A. et al. "*GroupDesign: Shared Editing in a Heterogeneous Environment*", Computing Systems 6(2), pp. 167-192, Spring 1993

[2] ATM Forum "*ATM User Nework Interface 3.1*" 1994

[3] Heinanen, J. "*Multiprotocol Encapsulation over ATM Adaptation Layer 5*" rfc1483, July 1993

[4] Laubach, M. "*Classical IP and ARP over ATM.*" rfc1577, January 1994

[5] De Prycker, M. "*Asynchronous Transfer Mode*" Ellis Horwood 1993