Available online at www.sciencedirect.com

ScienceDirect

Computers in Industry 57 (2006) 740–751

COMPUTERS IN INDUSTRY

www.elsevier.com/locate/compind

A method and tool for human–human interaction and instant collaboration in CSCW-based CAD

Fazhi He [a,*], Soonhung Han [b]

[a] School of Computer Science and Technology, Wuhan University, Wuhan 430072, China

[b] Department of Mechanical Engineering, Korea Advanced Institute of Science & Technology, Daejeon 305-701, Republic of Korea

Accepted 27 April 2006

Available online 3 July 2006

Abstract

From human–computer interaction to human–human interaction and from information sharing to instant collaboration are major advances from traditional standalone and distributed CAD to collaborative CAD. This paper explores model, method and tool to achieve this goal. A human-centered collaborative design model to support human–human interaction is presented. A method to transparently add aware capabilities to single-user CAD applications is outlined. Based on the method, an instant collaboration tool is implemented to support human–human interaction in collaborative design process. The collaboration tool involved some core collaborative issues such as agent structure, communication, group awareness, consistency maintenance and collaborative tasks. We also test the method in a 3D collaborative tool for heterogeneous CAD applications.

Keywords: Human–human interaction; Instant collaboration; CSCW; CAD; Agent

1. Introduction

CAD systems can be classified as standalone, distributed, or collaborative [1,2]. Traditional commercial CAD systems are standalone, such as I-DEAS, ProEngineer, Solidworks, Unigraphics and so on. The individual designers work with the localized computer systems independently in design process. Afterwards, the design results can be exchanged, viewed and checked by other designers. Therefore, an indirect and time-consuming collaboration among the individual designers can be possible by exchanging, viewing and checking design data of standalone CAD systems.

CAD systems are considered as distributed when different CAD modules and resources are geographically distributed over network. In distributed CAD, there can be only one individual designer to call remote CAD modules and to use distributed resources. The remote modules and resources are transparently deployed by other individual designers who can be offline at the time when the modules and resources are used. The word ‘‘transparently’’ used in this situation means that individual designers do not need to interact with each other in design process. Although the distributed computer systems give the individual designers a lot of convenience to exchange, view and check design data, the collaboration is still indirect and time-consuming.

Furthermore, when distributed CAD systems enable a group of designers to work interactively, concurrently and synchronously, the distributed CAD systems can be considered as collaborative. The atomic user unit of collaborative CAD systems must be a group of designers, at least two designers. The designers in same group will have a good awareness of each other and will intensely interact with each other in collaborative design process. Therefore, the human to human interaction and instant collaboration among the group designers are the key features of collaborative CAD.

Compared with current standalone CAD, the collaborative CAD is ‘‘not generally accepted’’ because of both technical and non-technical problems [3]. While the cultures, educational backgrounds and design habits are the non-technical problems, the weakness in interactive capabilities and

* Corresponding author. Tel.: +86 27 68778495; fax: +86 27 68778582.
E-mail addresses: fzhe@whu.edu.cn (F. He), shhan@kaist.ac.kr (S. Han).

0166-3615/$ – see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.compind.2006.04.019

real-time/convenient collaboration are identified as the major technical problems.

The intention of this paper is to explore human–human interaction and instant collaboration in collaborative CAD. The initial motive behind this research is that Computer-Supported Cooperative Work (CSCW) and Computer-Supported Cooperative Design (CSCD) reflect a trend from using computer to solve computing problems (such as geometric computing) to using computer to support interaction and collaboration among people. While traditional standalone and distributed CAD are system-centered, the collaborative CAD is human-centered.

The rest of the paper is organized as follows. Section 2 lists some related work. Section 3 presents a human-centered collaborative design model to support human–human interaction. Section 4 discusses an instant collaborative tool to construct collaborative CAD systems. Section 5 explores a 3D collaborative tool to support heterogeneous3D CAD systems. Conclusion and future work are summarized in Section 6.

2. Related work

The works related to this paper can be divided into three categories: (1) design theory and method in product design, (2) agent and multi-agent techniques, and (3) collaborative systems and tools.

2.1. Design theory and method in product design

Since the beginning of CAD in 1963, CAD systems have been single-user driven stand-alone systems only providing human–computer interaction (HCI). From HCI to human–human interaction (HHI) and from information sharing to instant collaboration are the major developments from traditional standalone and distributed CAD systems to collaborative CAD systems as shown in Table 1.

However traditional design theory and method, such as General Design Theory [4], Axiomatic Design Theory [5], Systematic Design Model [6], concern how the design decisions are made. Although the traditional design theory and method is useful in guiding a conscious design decision, they are generally oriented to single-user design process.

Some research extends the traditional design theory and method to distributed and collaborative situations, which consider collaborative design as individuals accessing product data and sharing design knowledge [7]. In these views, the collaborative design process becomes the conflict management of individual design decisions. However, it is always difficult to establish and coordinate the interdisciplinary dependencies of data and knowledge in computer systems. Therefore socio-technical (human-based) methods arise [8]. Among various socio-technical models, the direct communication, human to human interaction and instant collaboration among group designers in collaborative design process are emphasized in this paper.

Finger and Dixon classify design theory and method into four levels [9]. At descriptive level, the descriptive models describe how design is done. At prescriptive level, the prescriptive models prescribe how design should be done. At computer level, computer-based models are concerned with how computers (tools, systems and environments) can assistant or automate the design process or parts of the design process. This terminology and ontology are also useful to explore collaborative design method and models in developing distributed and collaborative CAD systems.

2.2. Agent and multi-agent techniques

The concepts of agent and multi-agent have been widely used in distributed and collaborative CAD/CAM/CIMS/CE environment [10–14]. This paper concerns related work in fields of software agents as techniques and tools to integrate legacy CAx (or DFx) systems into collaborative design environment.

There are two kinds of integration techniques in software engineering. The first one is based on distributed object technology or component technology or middleware (such as CORBA or DCOM). For example, Component Framework for Advanced CAD/CAM (CFACA) is presented as a component framework for feature-based design and process planning in CAD/CAM integration [15]. The key implementation tactics of CFACA is to wrapper ACIS geometry library [16] and to develop new ACIS-based application components which have a common core interface. CFACA does not discuss how to wrapper and integrate the existing CAD applications. Similar to CFACA, the PRE-RMI framework also does not study the integration of existing CAD applications [17]. It is claimed that DOME framework uses CORBA as an infrastructure to integrate DOME-based design problem models with existing application packages [18]. While it is clear that the DOME models are based on a previously developed OME library, it is not clear how to integrate existing application. Because no example of existing application is demonstrated, we cannot figure out what the existing applications or systems are.

Table 1
From HCI to HHI: collaborative CAD system

HCI: human–computer interaction	HHI: human–human interaction
Example: single-user CAD system	Example: collaborative CAD system
Sketchpad: a man-machine graphical communication system [40]	Computer-supported cooperative work: a research area to support group work
WYSIWYG: What You See Is What You Get	WYSIWIS: What You See Is What I See
Awareness and feedback from computer dynamic navigation, automatic snap	Group awareness and feed-through from other participants: tele-pointer, remote-cursors, tele-caret

In an agent- and CORBA-based application integration platform, Chan clearly states that it is necessary to integrate a legacy system (existing application) into the platform [19]. Chan presents general wrapper architecture for legacy application. This architecture uses both outside encapsulation and inside encapsulation. Chan gives a clear explanation to outside encapsulation, which is near ORB and accesses data and requests operations through inside encapsulation. Outside encapsulation is responsible for translating the operation request of client into the method invocations. After giving explanation to outside encapsulation, no explanation to inside encapsulation is presented in following parts of Chan's paper. Even the word ''inside encapsulation'' does not occur again.

In distributed software environment, outside encapsulation can be generalized according to specification of distributed object technology or component technology (such as CORBA or DCOM), which can be regarded as communication interoperation in architectural level among the heterogeneous operation system software and hardware platform. Furthermore, outside encapsulation can be also generalized to enable function interoperation in application level among heterogeneous application software by standardization of API, such as AIS [20], SDAI [21], Djinn API [22] and OMG CAD Services [23,24]. But inside encapsulation of legacy application is case by case. That is the reason why inside encapsulation is not explained in Chan's paper.

Actually, what we do in this paper is fit well into the catalogs of inside encapsulation. We hope that, in near future, we can combine the generalized outside encapsulation with our specialized inside encapsulation. This direction is beyond the range of this paper.

The second kind of software integration techniques is not based on distributed object technology or component technology or middleware. In other words, the second kind of software integration techniques can be regarded as inside encapsulation according to Chan's wrapper architecture.

A very famous work belonged to this catalog is PACT agent (Palo Alto Collaborative Testbed) [25]. PACT uses interaction agents (that is, programs that encapsulate engineering CAx/DFx systems) to integrate four sub-systems (e.g. NVisage, DMD, Next-Cut and Designworld). A PACT agent includes a CAD system and a wrapper. However, no details were presented in publications so that we can re-implement the PACT agent. The reasons may be: (a) the major goal of PACT is for knowledge sharing and data-exchange standards; (b) the subsystems are not commercial systems, therefore the PACT researchers can share the source code of the legacy systems.

Since then, many researches just copy the idea of PACT agent without serious consideration on how to construct a software agent to integrate legacy system without source code. This paper discusses in detail a reactive agent to integrate commercial legacy CAD systems in collaborative design.

2.3. Collaborative systems and tools

There are many different classifications for collaborative CAD systems. However, what this paper concerns is to consider the instantly collaborative CAD as one kind of simultaneous CSCW systems. Therefore, from the view of CSCW, there are two basic issues in developing an instantly collaborative system.

The first issue is collaboration framework, which means how to develop a collaborative system. Typical collaboration framework is either aware (collaboration awareness) or transparent (collaboration transparency) [26,27]. In typical aware frameworks, the collaborative systems are specifically designed to code both applicative functions and collaborative capabilities from scratch. Many research-oriented systems, such as collaborative editor, are aware frameworks. On the contrary, the typical transparent frameworks construct collaborative systems by adding collaborative capabilities to legacy single-user applications without access to source code of the applications. In this situation, the existing single-user applications are regarded as transparent applications. The application sharing systems, such as Microsoft Netmeeting or XTV [28,29], are typical transparent frameworks.

The main drawback of aware framework is the extra development work (required for both application and collaboration) which limits the usability and function of collaborative system compared with corresponding single-user commercial applications. On the other hand, the transparent framework lacks support for key collaboration features, such as flexibility, concurrency and group awareness. Since both of the typical frameworks have advantages and disadvantages, some atypical (also named hybrid) frameworks are explored in recent years.

Java Collaborative Environment (JCE) [30] and Habanero [31] require a little modification to source code of single-user application. The problem is that commercial applications do not provide source code to end users. The other problem is that they are limited to Java applications.

Flexible Collaboration Transparency in Java Applets Made Multiuser (JAMM) [32,33] achieves flexibility by replacing interactive components of single-user applet with aware multiuser components at runtime. However JAMM is limited to swing-based Java applets and requires additional platform capability (such as process migration, runtime object replacement, dynamic binding) which is not available in most platforms.

Intelligent Collaboration Transparency (ICT) in Collaborative Objects Coordination Architecture (COCA) [34,35] achieves intelligence by capturing and replaying user inputs. ICT begins to deal with heterogeneous situations using semantic translation among different kinds of text editors. But ICT and COCA are mainly about sharing text editors. Similar to COCA and ICT, Application Cooperating [36] achieves better feature than application sharing in sharing a drawing tool (Microsoft PBrush in Windows system) by interception, distribution and replay of user events.

Different from typical transparent frameworks (such as application sharing) which distribute rigid display pixel data over network, the atypical (hybrid) frameworks communicate the flexible event messages. By this way, the atypical frameworks try to add the aware communication (and eventually collaboration) ability to transparent applications. However the event messages in above frameworks are generally low-level I/O messages (keyboard and mouse events). Therefore it needs a lot of effort to filtrate, segment, reduce and semantically translate the low-level streaming events in large-scale and complicated engineering applications, such as engineering CAD system. This is the reason why only a simple text editor or drawing tool is showed as examples in above atypical (hybrid) frameworks.

What we do in this paper also falls into the same category of atypical framework. However, in our prototype system, not only the generally low-level I/O events but also the semantic messages (such as CAD commands, API and model data) can be communicated among the transparent CAD applications.

The second issue is system architecture: typically either centralized or replicated. We will discuss the architectural issues for transparent systems in this paper. In centralized architecture, several client sites send message to one server site which runs one copy of shared application and get display back from the sever. In replicated architecture, the shared applications are installed and run on every site.

Since to implement replicated transparent systems is more difficult than to implement centralized, the centralized transparent systems become the representative of transparent systems. Ref. [32] made some comparisons of transparent systems versus aware systems in following collaborative effects such as WYSIWIS, group awareness, network bandwidth and collaborative tasks.

This paper will show how a transparent system can approximate the collaborative functions of aware systems in case of CAD application. Actually, we use a replicated architecture for transparent systems. The matrix of framework and architecture for collaborative system is shown in Fig. 1.

This paper is the extension and revision of international conference papers [37–39].

3. An human-centered collaborative design method

3.1. Descriptive model of human–human interaction in design

This section presents a collaborative design model to describe human–human interaction in collaborative design process as shown in Fig. 2.

Product design processes are generally described by a mapping from function to attribute. Collaborative mapping model consists of a group of mappings from function to attributes. While the function requirements represent the design intention of group designers, the design parameters represent the physical attributes of a product. The mappings are conducted interactively, concurrently and synchronously by a group of designers.

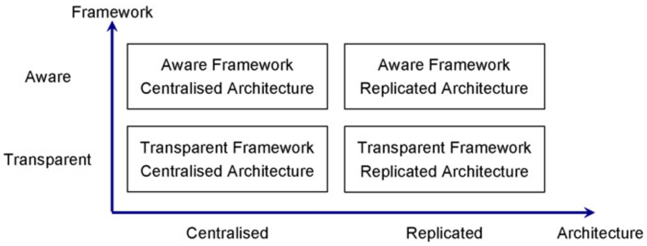

Framework
Aware
Transparent
Aware Framework Centralised Architecture
Aware Framework Replicated Architecture
Transparent Framework Centralised Architecture
Transparent Framework Replicated Architecture
Centralised
Replicated
Architecture

Fig. 1. Matrix of framework and architecture for collaborative system.

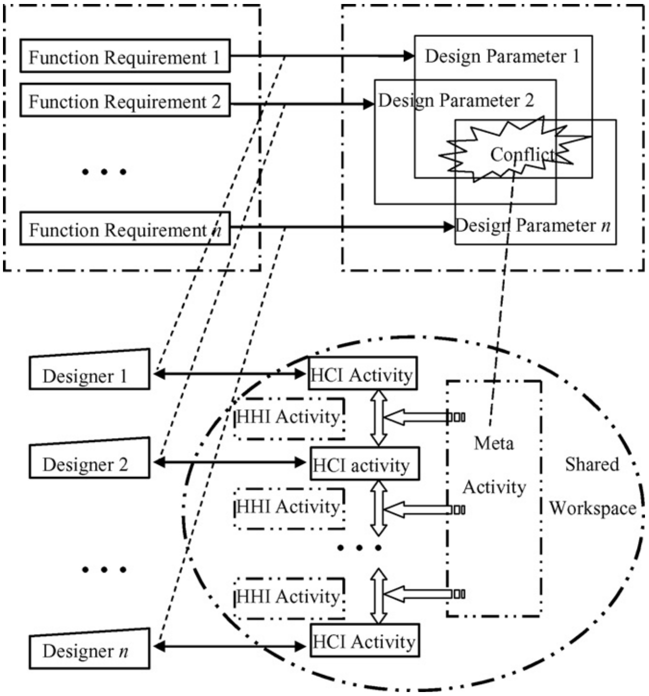


Fig. 2. Collaborative design model.

There are three types of activities in collaborative design model: HCI activity, HHI activity and meta activity. The HCI activities in collaborative CAD are similar to the HCI activities in traditional standalone CAD. The designers interact with the computer and get WYSIWYG (What You See Is What Your Get) fed-back from computer.

HHI activities are instant communication and interaction among group designers in one shared-workspace. The group designers communicate and interact with each other and get WYSIWIS (What You See Is What I See) fed-through from other designers. Therefore, the HCI activities become the one part of HHI activities in collaborative design process.

Meat activities are global activities which coordinate the HCI activities and HHC activities in collaborative design process.

3.2. Prescriptive model of human–human interaction in design

Since the group designers map function requirements to the design parameters concurrently, the combined result after mapping maybe overlap. The conflict is unavoidable and must be resolved in collaborative design process. An interactive method is used to deal with the conflict.

This method is divided into two levels. Firstly, at systematic level, one kind of concurrency control method (such as floor control) is used to coordinate the HCI/HHC activities and maintain a consistent result in all of the sites.

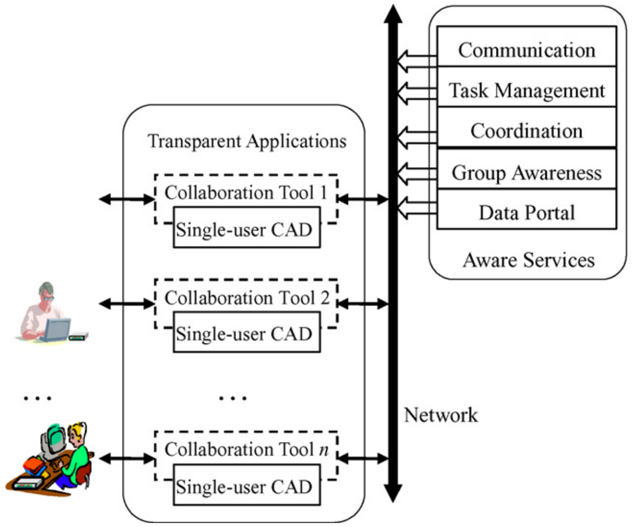


Fig. 3. A computer model for collaborative design.

However, in some cases, the consistent result maybe neither senseful nor valid in context of design backgrounds. Therefore, the interaction in application level is needed. Supported by group awareness, group undo/redo and HHI interface, the group designers interact with each other just like face to face negotiation and discussion. A senseful and valid result will be reached finally in this way.

3.3. A computer-based model to support human–human interaction in design

An instant collaborative tool is introduced into computer-based model to support human–human interaction in collaborative design as shown in Fig. 3.

The core technical issues involved in the instant collaborative tool are shown in Fig. 4. We will discuss the tool in next section in details.

4. Instant collaborative tool to construct collaborative CAD systems

4.1. Collaborative tool agent

The collaboration tool is built on reactive agent structure as shown in Fig. 5.

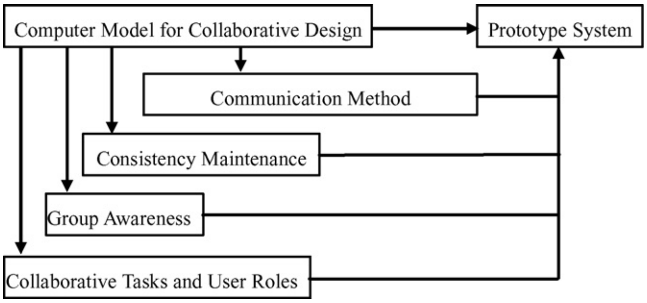


Fig. 4. Core technical issues.

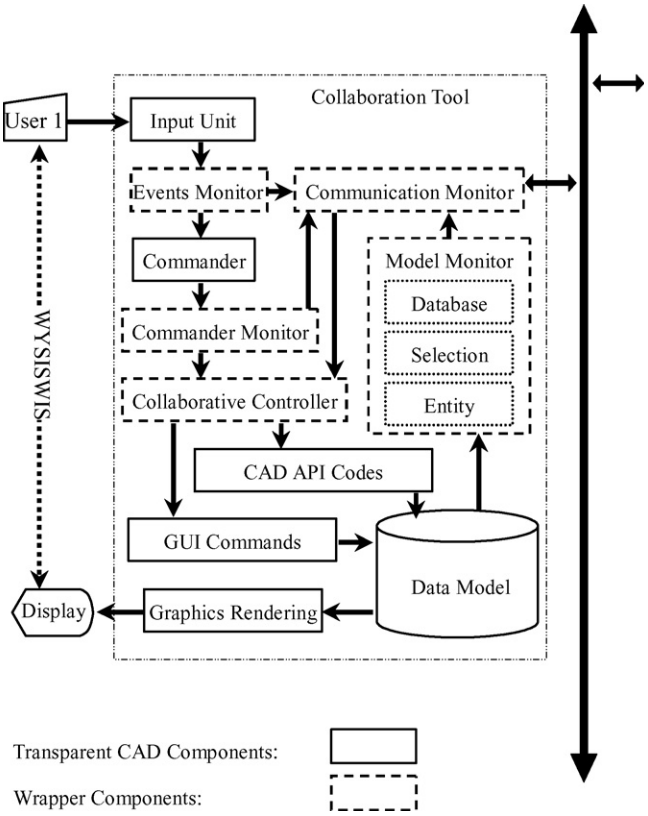


Fig. 5. Reactive structure of collaboration tool agent.

Event monitor checks the input unit to catch keyboard and mouse events input from designers. Then the messages go ahead to commander and are translated into semantic commands by the commander.

The commander monitor checks the commander to capture the information related to semantic commands. Typical information includes command name formatted by commander and command state to indicate that the command is about to begin execution or has completed or has been canceled or failed to complete successfully.

Collaborative controller coordinates the local messages from commander and remote message from network and decides how to call the GUI commands, or API codes or both.

Communication monitor collects local message from other monitor and send them to the network. In the same time, communication monitor listens and receives remote message from network and forward them to collaborative controller.

Model monitor checks the data model in different object granularity. In the coarsest granularity, database monitor can look out any change of the whole database in CAD system. In the finest granularity, only the change of entity attached to an entity monitor can be looked out. In the middle granularity, selection monitor looks out the change of entities in the selection.

4.2. Communication method

The communication model is organized into three levels and approximately corresponds with OSI model as shown in Table 2.

Table 2
Communication model

Communication models			OSI reference models
Applications	Semantic messages	General messages	Application Presentation
Cooperative support	Group communications		Session
	Socket		
Network	TCP or UDP		Transport
	IP		Network
	Communications network		Data link Physical

In the application level, the transparent applications use socket to exchange semantic messages (such as CAD commands, remote API calls, CAD data and so on) as well as general messages (such as mouse and keyboard events).

In cooperative support level, the group communication is one of the most fundamental communication services. In network level, TCP/IP is used.

We design and implement an Internet Control Message Protocol (ICMP) echo/response software to verify the direct IP connection of transparent CAD applications. As shown in windows task manager in Fig. 6, a 2D CAD application sends ICMP data to a remote site and receives echo from it.

4.3. Consistency maintenance

Every entity of CAD data is given an ID. CAD applications have one set of mechanism for naming, recording and retrieving CAD entities. Since the CAD applications are transparently shared, the naming mechanisms are also transparently inherited by collaboration tools. For example, the transparent AutoCAD application names the entities in an undo way even after the entities are deleted by user. Therefore, the consistency problem in this situation is the problem of name consistency.

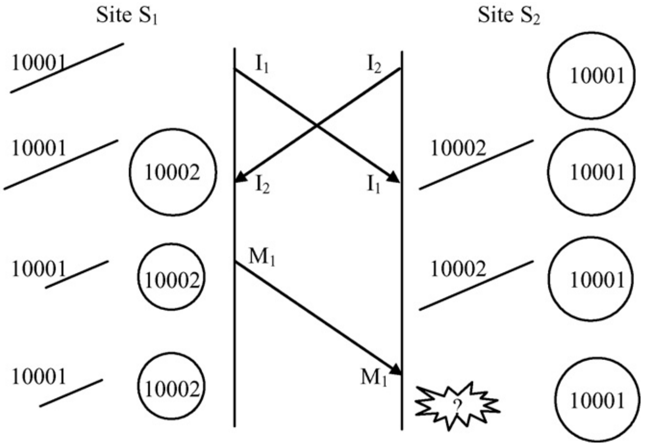
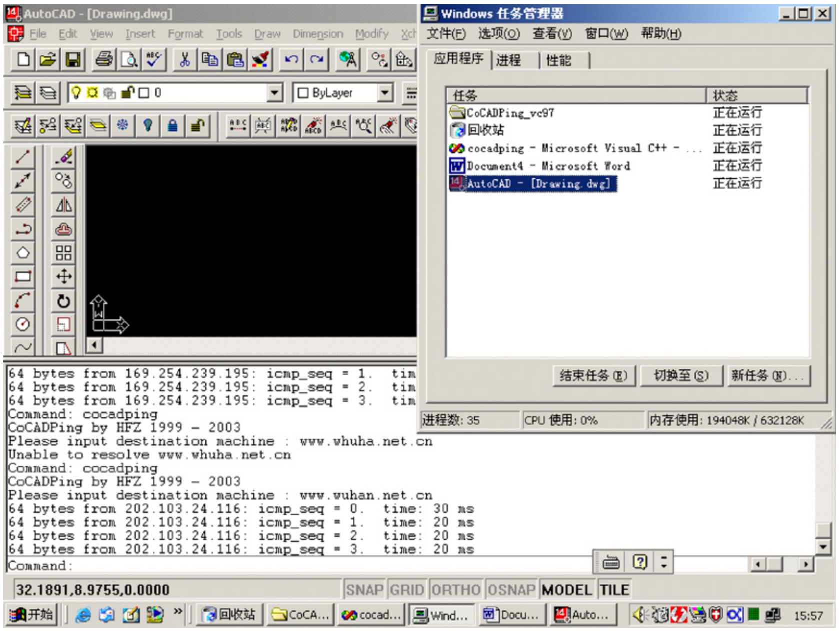


Fig. 7. Type mismatch.

There are two kinds of naming problems in this architecture. The first one is type mismatch as shown in Fig. 7. The operation M1 enlarges a circle of ID 10002 in S1. But the retrieving result of ID 10002 in S2 is a line. Therefore M1 cannot be executed in S2.

The second naming problem is entity mismatch as shown in Fig. 8. The operation M1 in S1 shortens a line of ID 10001 which is different from the line of same ID in S2. Therefore the result of modification in site 2 is not what we expect in site 1.

In order to keep the name consistency, the collaboration tool should guarantee that the insertion operations will be executed in same order to all of sites. The prototype system in following section of this paper uses a typical method – floor control to maintain a global consistency.



任务	状态
CoCADPing_vc97	正在运行
回收站	正在运行
cocadping - Microsoft Visual C++ - ...	正在运行
Document4 - Microsoft Word	正在运行
AutoCAD - [Drawing.dwg]	正在运行

进程数: 35   CPU 使用: 0%   内存使用: 194048K / 632128K

```
64 bytes from 169.254.239.195: icmp_seq = 1.  tim
64 bytes from 169.254.239.195: icmp_seq = 2.  tim
64 bytes from 169.254.239.195: icmp_seq = 3.  tim
Command: cocadping
CoCADPing by HFZ 1999 - 2003
Please input destination machine : www.whuha.net.cn
Unable to resolve www.whuha.net.cn
Command: cocadping
CoCADPing by HFZ 1999 - 2003
Please input destination machine : www.wuhan.net.cn
64 bytes from 202.103.24.116: icmp_seq = 0.  time: 30 ms
64 bytes from 202.103.24.116: icmp_seq = 1.  time: 20 ms
64 bytes from 202.103.24.116: icmp_seq = 2.  time: 20 ms
64 bytes from 202.103.24.116: icmp_seq = 3.  time: 20 ms
Command:
```

32.1891,8.9755,0.0000

Fig. 6. ICMP echo response for transparent CAD application.

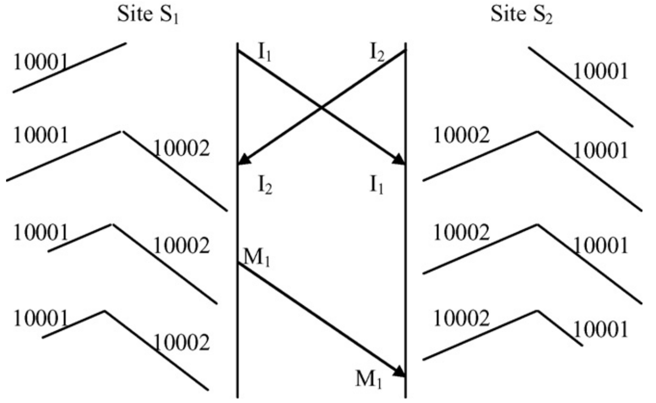


Fig. 8. Entity mismatch.

4.4. Group awareness

The collaboration tool uses remote-cursors as basic group awareness to enhance human–human interaction in process of collaborative design.

In strict WYSIWIS interface of centralized transparent systems, the graphics displays of windows of shared application are the same. It is OK to send the position of cursors to remote sites in strict WYSIWIS. Different from strict WYSIWIS interface, the interface of the collaboration tool is relaxed WYSIWIS. Simply sending the position of cursors will result in error pointing of remote-cursors.

In order to keep the logical position of remote-cursors, the collaboration tool uses a mapping method with following steps:

- Transform the window coordinates of local cursors into CAD model coordinates at local site.
- Send the model coordinates to network.
- Transform CAD model coordinates into the window coordinates in remote site.
- Render and execute the remote-cursor.

In some situations, if a cursor is specially associated with a CAD entity, the entity coordinate is mapped.

The rendering of remote-cursors is simulated with device context by the collaboration tool. One of the key issues in the rendering process is how to actively update the position of the cursors which is moving. The active updating method is shown in Fig. 9.

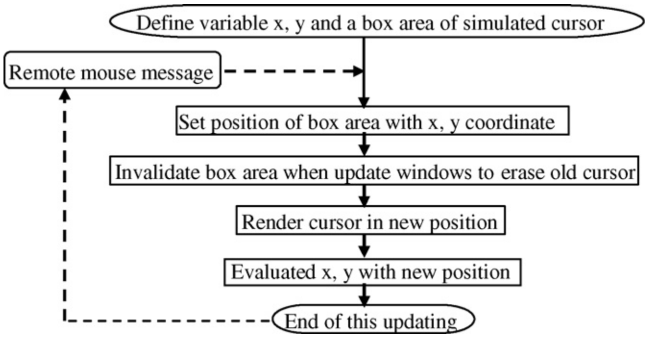

Define variable x, y and a box area of simulated cursor
Remote mouse message
Set position of box area with x, y coordinate
Invalidate box area when update windows to erase old cursor
Render cursor in new position
Evaluated x, y with new position
End of this updating

Fig. 9. Active updating method for simulated cursor.

Table 3
Awareness fields and overloaded semantics

Dynamic information	Awareness fields	Overloaded semantics
Remote-cursor	Present	Remote user
User name	Identity	Who
Coordinates	Position	Where in the shared space
Shape, colour	Activities	What to do
	Status	How about idle or busy
	Intention	Highlight the object/entity

The remote-cursors are finally executed in a dynamic way. The combination of dynamic information can be used to perceive who, where, what and how in collaborative design. The dynamic information, awareness fields and overloaded semantics are shown in Table 3.

4.5. Management of tasks and users

The collaboration tool uses task wizard to help participants to create a new collaborative task or join an existing task. When users log in the information server, they inquire concurrent tasks at first and then decide either to create a new task or to join an existing task as shown in Fig. 10.

The overall flow chart of management of task and users include system works like this way:

- One information server, which is statically deployed. The IP address and port of information server are open to teamwork of the design group.
- At every site, AutoCAD is installed and wrapped with the collaborative tool. The tool is coded with ARX program, which is a DLL program specialized for AutoCAD.

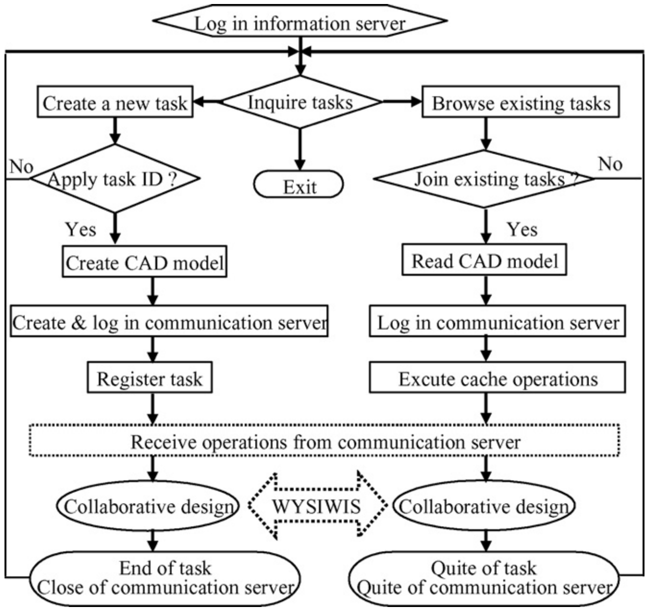


Fig. 10. Task wizard.

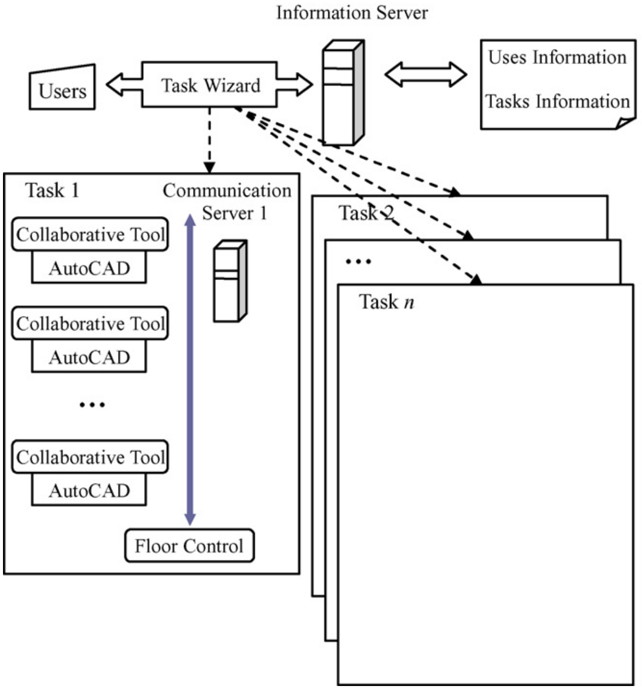


Fig. 11. 2D prototype system.

- When a user creates a new task, a dynamical communication server is created. The IP address and port of the communication server are registered to the information server. Other users can browse the task information by inquiring the information server.
- When creating a new task, a floor control client (coordinate client) connected to the communication server is also created. There is status tag in the client which at any time is only associated with one user of same task. Only this user can make operation.
- There can be several tasks in the same time. Every task has one independent set of CAD model, communication server and floor control client.

The user who creates a new task is automatically endowed with role of chairman in a collaborative task, while others to join the task are participants. The role of the chairman can end the task.

4.6. 2D prototype system

The version 1.0 of the collaborative tool adds above collaborative capabilities to a 2D transparent CAD application (commercial system AutoCAD) as shown in Fig. 11.

The version 1.0 of the collaborative tool has been used in engineering design of crane. A snapshot of human to human interface is shown in Fig. 12. Although the size and layout of windows of transparent applications are different, the logical position of remote-cursor keeps identical.

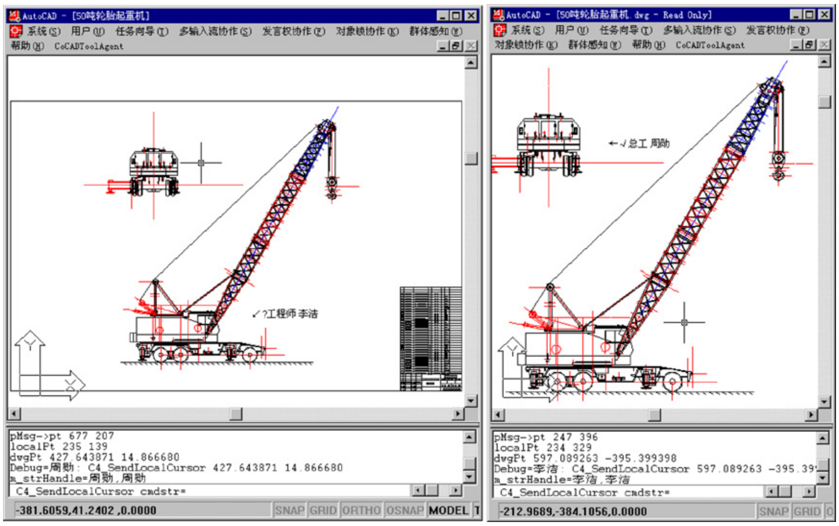
5. Exploratory study on 3D collaborative tool to support heterogeneous CAD

In order to develop 3D collaborative tool to support heterogeneous CAD, this section explore the Client/Server communication, semantic communication and group communication for transparent 3D CAD applications.

5.1. Client/Server communications for transparent 3D CAD applications

We organize three kinds of Client/Server communication for transparent 3D CAD applications. In the first experiment (see Fig. 13a), the client is a transparent CAD application while the server is a standard console program as shown in Fig. 13a. A console server is started to establish a listener waiting for a connection from client at first. Then a 3D CAD application is started and sends a sample message – Co3D_Hello to console server.

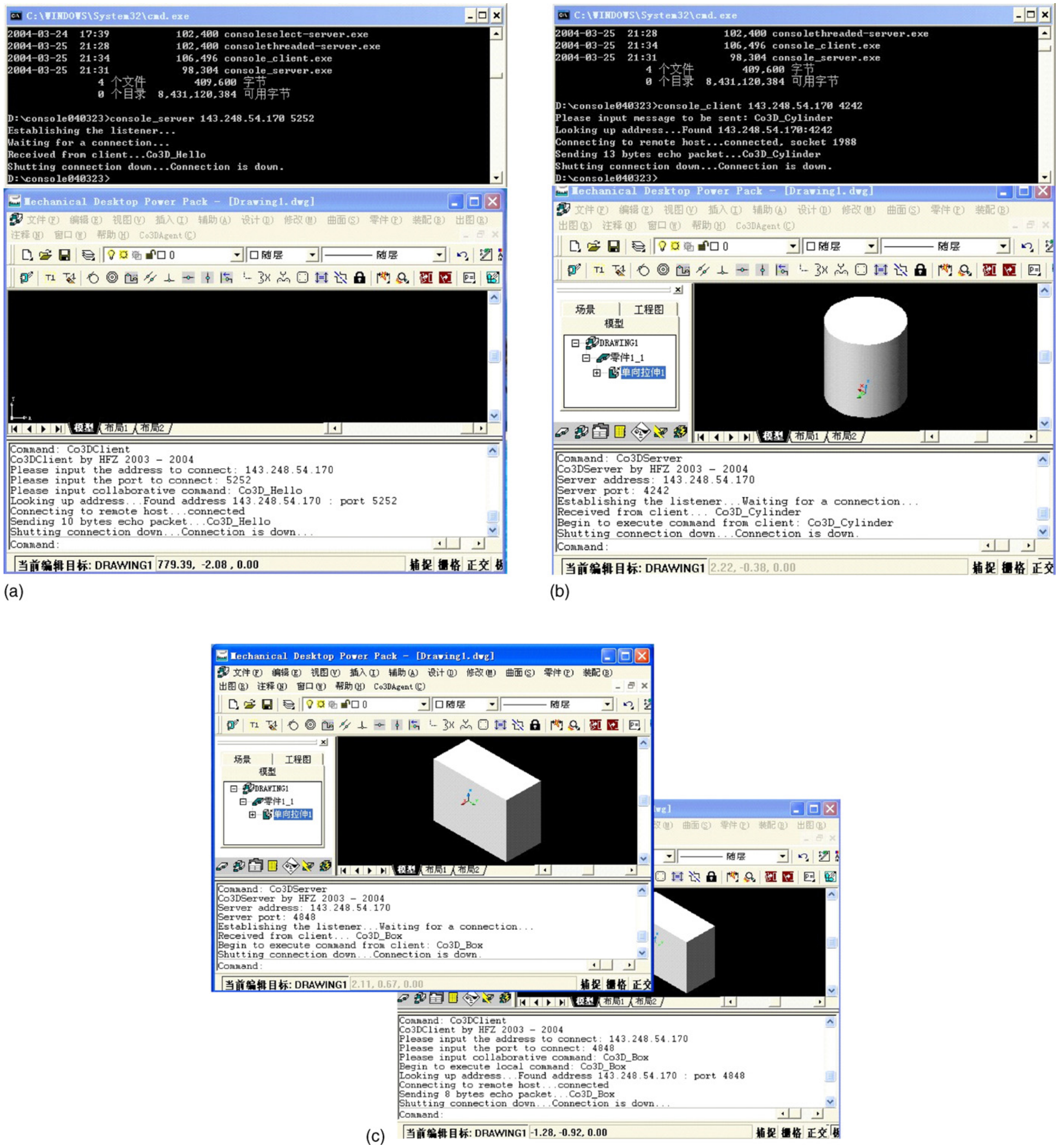
In the second experiment (Fig. 13b), the client is a standard console program while the server is a transparent CAD


AutoCAD - [50吨轮胎起重机]
系统(S) 用户(U) 任务向导(T) 多输入流协作(S) 发言权协作(F) 对象锁协作(K) 群体感知(W)
帮助(H) CoCADToolAgent
✓?工程师 李洁
SNAP GRID ORTHO OSNAP MODEL T
AutoCAD - [50吨轮胎起重机.dwg - Read Only]
系统(S) 用户(U) 任务向导(T) 多输入流协作(S) 发言权协作(F)
对象锁协作(K) 群体感知(W) 帮助(H) CoCADToolAgent
←√总工 周勋
SNAP GRID O

```
pMsg->pt 677 207
localPt 235 139
dwgPt 427.643871 14.866680
Debug=周勋: C4_SendLocalCursor 427.643871 14.866680
m_strHandle=周勋,周勋
 C4_SendLocalCursor cmdstr=
-381.6059,41.2402 ,0.0000
```

```
pMsg->pt 247 396
localPt 234 329
dwgPt 597.089263 -395.399398
Debug=李洁: C4_SendLocalCursor 597.089263 -395.39
m_strHandle=李洁,李洁
 C4_SendLocalCursor cmdstr=
-212.9689,-384.1056,0.0000
```

Fig. 12. A snapshot of the prototype system (version 1.0).


(a)
C:\WINDOWS\System32\cmd.exe
Mechanical Desktop Power Pack - [Drawing1.dwg]
文件(F) 编辑(E) 视图(V) 插入(I) 辅助(A) 设计(D) 修改(M) 曲面(S) 零件(P) 装配(B) 出图(R) 注释(N) 窗口(W) 帮助(H) Co3DAgent(C)
随层
模型 布局1 布局2
当前编辑目标: DRAWING1 779.39, -2.08 , 0.00
捕捉 栅格 正交
(b)
场景 工程图 模型
DRAWING1
零件1_1
单向拉伸1
当前编辑目标: DRAWING1 2.22, -0.38, 0.00
(c)
当前编辑目标: DRAWING1 2.11, 0.67, 0.00
当前编辑目标: DRAWING1 -1.28, -0.92, 0.00

(a)

```
2004-03-24  17:39           102,400 consoleselect-server.exe
2004-03-25  21:28           102,400 consolethreaded-server.exe
2004-03-25  21:34           106,496 console_client.exe
2004-03-25  21:31            98,304 console_server.exe
               4 个文件        409,600 字节
               0 个目录  8,431,120,384 可用字节

D:\console040323>console_server 143.248.54.170 5252
Establishing the listener...
Waiting for a connection...
Received from client...Co3D_Hello
Shutting connection down...Connection is down.
D:\console040323>
```

```
Command: Co3DClient
Co3DClient by HFZ 2003 - 2004
Please input the address to connect: 143.248.54.170
Please input the port to connect: 5252
Please input collaborative command: Co3D_Hello
Looking up address...Found address 143.248.54.170 : port 5252
Connecting to remote host...connected
Sending 10 bytes echo packet...Co3D_Hello
Shutting connection down...Connection is down...
Command:
```

(b)

```
2004-03-25  21:28           102,400 consolethreaded-server.exe
2004-03-25  21:34           106,496 console_client.exe
2004-03-25  21:31            98,304 console_server.exe
               4 个文件        409,600 字节
               0 个目录  8,431,120,384 可用字节

D:\console040323>console_client 143.248.54.170 4242
Please input message to be sent: Co3D_Cylinder
Looking up address...Found 143.248.54.170:4242
Connecting to remote host...connected, socket 1988
Sending 13 bytes echo packet...Co3D_Cylinder
Shutting connection down...Connection is down.
D:\console040323>
```

```
Command: Co3DServer
Co3DServer by HFZ 2003 - 2004
Server address: 143.248.54.170
Server port: 4242
Establishing the listener...Waiting for a connection...
Received from client... Co3D_Cylinder
Begin to execute command from client: Co3D_Cylinder
Shutting connection down...Connection is down.
Command:
```

(c)

```
Command: Co3DServer
Co3DServer by HFZ 2003 - 2004
Server address: 143.248.54.170
Server port: 4848
Establishing the listener...Waiting for a connection...
Received from client... Co3D_Box
Begin to execute command from client: Co3D_Box
Shutting connection down...Connection is down.
Command:
```

```
Command: Co3DClient
Co3DClient by HFZ 2003 - 2004
Please input the address to connect: 143.248.54.170
Please input the port to connect: 4848
Please input collaborative command: Co3D_Box
Begin to execute local command: Co3D_Box
Looking up address...Found address 143.248.54.170 : port 4848
Connecting to remote host...connected
Sending 8 bytes echo packet...Co3D_Box
Shutting connection down...Connection is down...
Command:
```

Fig. 13. Client server communication for transparent 3D CAD: (a) experiment 1, (b) experiment 2, (c) experiment 3.

application as shown in Fig. 13b. A 3D CAD application is started to establish a listener and wait for a connection. A console client is started to send a 3D command – Co3D_Cylinder to Co3DServer. After receiving the command message, the 3D CAD application creates a cylinder.

In the third experiment (Fig. 13c), both the client and server are transparent CAD applications as shown in Fig. 13c. A 3D CAD application in server side is started and waits for a connection. Another 3D CAD application in client side is started to execute a 3D command – Co3D_Box to create a box locally at first. And then the 3D CAD application in client side sends the command to the 3D CAD application in server side. After receiving the 3D command, the 3D CAD application in server side also create a box.

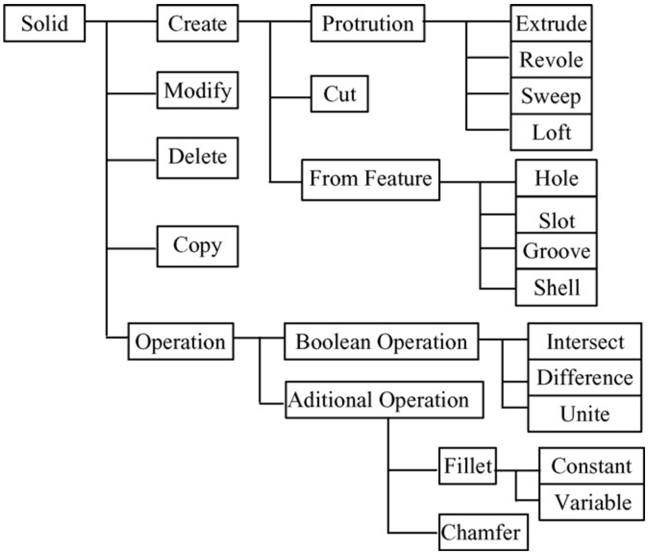


Fig. 14. Examples of application level communication protocol.

The third experiment can be regarded as one of the simplest situation for collaboration in peer to peer communication.

5.2. Semantic communications for transparent 3D CAD applications

The 3D collaborative tools work with each other by exchanging CAD semantic message. The application level communication protocol should support the semantic collaboration even among heterogeneous CAD applications.

We are developing a set of standard CAD modeling commands to deal with heterogeneity. The example of solid modeling commands is shown in Fig. 14.

5.3. Group communication for transparent 3D CAD applications

A collaborative design system involves a group of geographically distributed participants interacting with the system. Therefore, a group communication is needed.

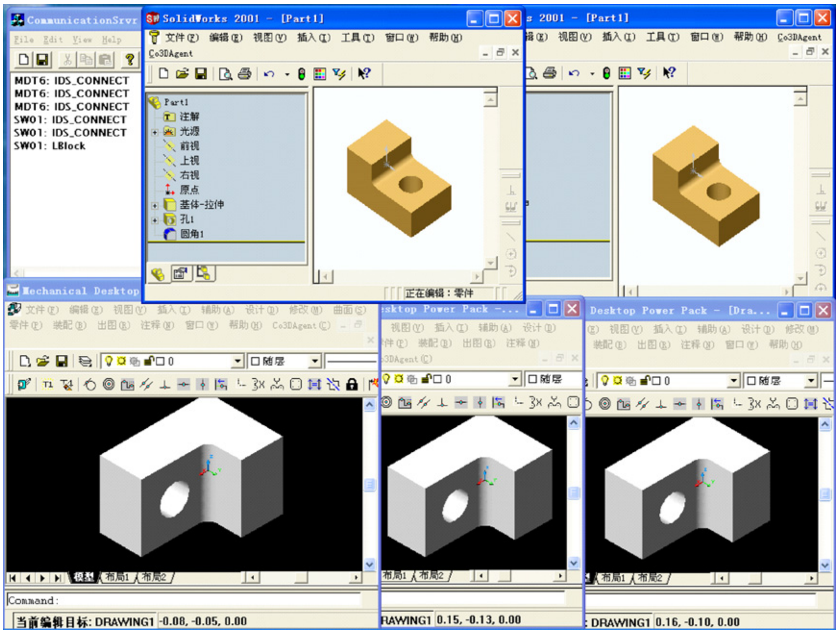
The group communication experiment is based on multiple Client/Server connections to support a limited number of participants. All of the messages are forwarded by a communication server.

In group communication experience, we deploy two heterogeneous CAD applications by exchanging of standard CAD commands. As demonstrated in Fig. 15, two Co3DAgents for SolidWorks and three Co3DAgents for MDT are supported by group communication to create an L-Block part.

6. Summary and future work

Human–human interaction and instant collaboration are the key features of collaborative CAD systems. Transparently adding collaborative capabilities to single-user CAD application is helpful but also difficult. Major contributions of this research are summarized as below:

(a) A human-centered collaborative design model to support human–human interaction. Descriptive model, prescriptive model and computer-based model of human–human interaction in collaborative design are presented.
(b) An instant collaborative tool to construct collaborative CAD systems. After resolving some technical problems such as agent structure, communication, group awareness, consistency maintenance and collaborative tasks, we demonstrate that a transparent system can also achieve the similar collaborative effect as aware systems.



```
MDT6: IDS_CONNECT
MDT6: IDS_CONNECT
MDT6: IDS_CONNECT
SW01: IDS_CONNECT
SW01: IDS_CONNECT
SW01: LBlock
```

```
当前编辑目标: DRAWING1 -0.08, -0.05, 0.00
RAWING1 0.15, -0.13, 0.00
DRAWING1 0.16, -0.10, 0.00
```

Fig. 15. Group communication for transparent 3D CAD.

(c) An initial 3D collaborative tool to support heterogeneous 3D CAD systems. The Client/Server communication, semantic communication and group communication for transparent 3D CAD applications are established.

We hope that this method can be considered as a general method to integrate proprietary CAD applications. The new challenges will be: interoperation of heterogeneous CAD applications, name consistency of 3D CAD model, group awareness in 3D workspace and more natural human to human interaction.

Acknowledgments

This research has been partially supported by the National Natural Science Foundation of China (NSFC Grant No. 60303029), NSFC/KOSEF international cooperation Project (NSFC Grant No. 60510627) and by the BK21 Project of Korea.

References

[1] Z. Lin, The appearance of CSCD making CAD technology to leap onto a new step, in: Proceedings of Second International Workshop on CSCW in Design, Bangkok, Thailand, 1997, pp. 162–167.

[2] D. Wu, R. Sarma, Dynamic segmentation and incremental editing of boundary representations in a collaborative design environment, in: Proceedings of ACM Solid Modeling '01, ACM Press, 2001, pp. 289–300.

[3] W.D. Li, J.Y.H. Fuh, Y.S. Wong, Collaborative computer-aided design—research and development trend, in: Proceedings of CAD'04 Conference, Pattaya, Thailand, May 24–29, 2004.

[4] H. Yoshikawa, General design theory and a CAD system, in: Proceedings of the IFIP Working Group Working 5.2 Conference, Man-Machine Communication in CAD/CAM, North-Holland Publishing Company, 1980, pp. 35–58.

[5] N.P. Suh, The Principles of Design, Oxford University Press, New York, 1990.

[6] G. Paul, W. Beitz, Engineering Design—A Systematic Approach, second ed., Springer, London, 1996.

[7] M. Saad, M.L. Maher, Shared understanding in computer-supported collaborative design, Computer-Aided Design 28 (3) (1996) 183–192.

[8] S.C.Y. Lu, J. Cai, W. Burkett, F. Udwadi, A methodology for collaborative design process and conflict analysis, CIRP Annals 49 (1) (2000) 69–74.

[9] S. Finger, J.R. Dixon, A review of research in mechanical engineering design. Part I: descriptive, prescriptive and computer based models of design processes, Research in Engineering Design 1 (2) (1989) 51–68.

[10] W. Shen, D.H. Norrie, J.P. Barthes, Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing, Taylor and Francis, London, UK, 2001.

[11] H.R. Frost, M.R. Cutkosky, Design for manufacturability via agent interaction, in: CD-ROM Proceedings of ASME DETC 1996/DFM-1302.

[12] D. Dornfeld, P.K. Wright, S. Roundy, A. Rangarajan, S.H. Ahn, AGENT interaction in CAD/CAM, Transaction of NAMRI/SME 2001, 29(5) 569–575.

[13] J.P. Barthes, C.A. Tacla, Agent-supported portals and knowledge management in complex R&D projects, Computers in Industry 48 (1) (2002) 3–16.

[14] K. Chao, P. Norman, R. Anane, A. James, An agent-based approach to engineering design, Computers in Industry 48 (1) (2002) 17–27.

[15] X. Liu, CFACA: component framework for feature based design and process planning, Computer-Aided Design 32 (7) (2000) 397–408.

[16] http://www.spatial.com/.

[17] J.F. Gerhard, D. Rosen, J.K. Allen, F. Mistree, A distributed product realization environment for design and manufacturing, ASME Journal of Computing and Information Science in Engineering 1 (3) (2001) 235–244.

[18] F. Pahng, N. Senin, D. Wallace, Distributed object-based modeling and evaluation of design problems, Computer-aided Design 30 (6) (1998) 411–423.

[19] F.T.S. Chan, J. Zhang, P. Li, Agent- and CORBA-based application integration platform for an agile manufacturing environment, International Journal of Advanced Manufacturing Technology 21 (6) (2003) 460–468.

[20] P. Ranyak, Application interface specification (AIS), volume I: functional specification, CAM-I Report R-94-PM-01 (Version 2. 1) (1994).

[21] ISO 10303-22, Industrial automation system and integration-Product data representation and exchange—Part 22, Implementation methods: Standard Data Access Interface specification, 1996.

[22] C. Armstrong, A. Bowyer, S. Cameron, J. Corney, G. Jared, R. Martin, A. Middleditch, M. Sabin, J. Salmon, Djinn—Specification & Report, Published by Information Geometers Ltd., ISBN 1-874728-13-5, 2000, 410 pp.

[23] OMG, CAD Services V1.0 Request for Proposal Object Management Group, Document: mfg/2000-06-07. Needham, MA, June 2000, 30 pp.

[24] OMG, Initial Submission CAD Services V1.0 Joint Proposal to the OMG, Document: mfg/2001-01-01. Needham, MA, January 2001, 12 pp.

[25] M. Cutkosky, R. Englemore, R. Fikes, M. Genesereth, T. Gruber, W. Mark, J. Tenenbaum, J. Weber, PACT: an experiment in integrating concurrent engineering systems, IEEE Computer 26 (1) (1993) 28–38.

[26] T.D. Palmer, N.A. Fields, Computer supported cooperative work. Special issue, IEEE Computer 27 (5) (1994) 15–17.

[27] J.C. Lauwers, K.A. Lantz, Collaboration awareness in support of collaboration transparency: requirements for the next generation of shared window systems, in: Proceedings of CHI '90, ACM Press, 1990, pp. 303–310.

[28] NetMeeting Home, http://www.microsoft.com/windows/netmeeting/default.asp.

[29] X. Abdel-Wahab, M. Feit, XTV: a framework for sharing X window clients in remote synchronous collaboration, in: Proceedings of IEEE Tricomm '91, IEEE Computer Society Press, Chapel-Hill NC, USA, April 1991, pp. 159–167.

[30] H. Abdel-Wahab, B. Kvande, S. Nanjangud, Using Java for multimedia collaborative applications, in: Proceedings of the Third International Workshop on Protocols for Multimedia Systems, PROM '96, pp. 49–62.

[31] A. Chabert, E. Grossman, L. Jackson, S. Pietrowicz, C. Seguin, Java object sharing in Habaneo, Communications of the ACM 41 (6) (1998) 69–76.

[32] J.M.A. Begole, M.B. Rosson, C.A. Shaffer, Flexible collaboration transparency: supporting worker independence in replicated application sharing systems, ACM Transactions on Computer–Human Interaction 6 (2) (1999) 95–132.

[33] J.M.A. Begole, Flexible collaboration transparency: supporting worker independence in replicated application-sharing systems, Ph.D. Dissertation, Virginia Polytechnic Institute and State University, Blacksburg, 1999.

[34] D. Li, COCA: a framework for modeling and supporting flexible and adaptable synchronous collaborations, Ph.D. Dissertation, University of California, Los Angeles, 2000.

[35] D. Li, R. Li, Transparent sharing and interoperation of heterogeneous single-user applications, in: Proceedings of CSCW '02, ACM Press, pp. 246–255.

[36] B. Xu, Q. Gao, W. Lian, A general framework for constructing application cooperating system in windows environment, ACM SIGSOFT Software Engineering Notes 28 (2) (2003) 15.

[37] F. He, S. Han, S. Wang, G. Sun, A road map on human–human interaction and fine-function collaboration in collaborative integrated design environments, in: The Eighth International Conference on CSCW in Design, Xiamen, China, May 2004, pp. 59–65.

[38] F. He, S. Han, Experiment research on aware communication method in support of transparent CAD applications, 2004 ACM CSCW Interactive Poster, Chicago, IL, USA, 2004.11.06-10.

[39] F. He, S. Han, Architectural issues on transparently adding collaborative capabilities to single-user CAD applications, ACM CSCW 2004 Workshop on "Making Application Sharing Easy: Architectural Issues for Collaboration Transparency", Chicago, IL, USA, 2004.11.06-10.

[40] I. Sutherland, Sketchpad: a man-machine graphical communication system, Proceedings of the 1963 Spring Joint Computer Conference, Spartan Books, Baltimore, MD, 1963, pp. 45–53.

Fazhi He is an associate professor at school of computer science and technology (http://cs.whu.edu.cn) of Wuhan University (WHU). His research interests include computer-aided design (CAD), computer graphics (CG), computer-supported cooperative work (CSCW). He has a BS, MS and PhD from Wuhan University of Technology (WHUT). He performed two post-doctor research experiments in State Key Laboratory of CAD and CG at Zhejiang University (http://cad.zju.edu.cn) and Intelligent CAD Laboratory (http://icad.kaist.ac.kr/) at Korea Advanced Institute of Science and Technology (KAIST (www.kaist.ac.kr)).

Soonhung Han is a professor at the Department of Mechanical Engineering (http://me.kaist.ac.kr/) of Korea Advanced Institute of Science and Technology (KAIST, www.kaist.ac.kr). He is leading the Intelligent CAD Laboratory (http://icad.kaist.ac.kr/) at KAIST, and the STEP community of Korea (www.kstep.or.kr). His research interests include STEP (ISO standard for the exchange of product model data), virtual reality (VR) for engineering design, and knowledge-based design system. His domain of interests include automotive and shipbuilding. He has a BS and a MS from the Seoul National University of Korea, another MS from the University of Newcastle upon Tyne of UK, and a PhD from the University of Michigan of USA. He is involved in the professional societies of CAD/CAM (www.cadcam.or.kr) and e-Business (www.calsec.or.kr). He is an editorial member of the web-based journal, International Journal of CAD/CAM (www.ijcc.org).