# Collaborative Design with NetDraw

Dongqiu Qian and Mark D. Gross
*Sundance Lab for Computing in Design & Planning, University of Colorado, Boulder, CO, USA*

**Key words**:   synchronous collaboration, shared drawing,

**Abstract**:   The paper describes NetDraw, a Java-based object oriented drawing program that employs a server-client architecture to provide a shared drawing environment for collaborative design.  NetDraw goes beyond conventional shared whiteboard applications in its support for concurrency control, groups and constraints, and ephemeral gesture objects.  Small and simple enough that users can learn it quickly, NetDraw is designed to run on small platforms such as handheld computers.  We describe NetDraw's features and an early evaluation of its use.

## 1.      INTRODUCTION

Design is team work.  Architects and engineers work together to produce a design, and at times the design team sits face-to-face, usually around a table, drawing on a piece of paper or a whiteboard with a variety of colored pens. During those meetings designers draw concurrently or serially to show their ideas to others and they point to elements in the drawing as they discuss the design.   One important tool for synchronous distance collaboration in design is a shared drawing surface where designers can discuss and mark up proposed designs. Although the Internet makes it possible to carry on these graphical conversations at a distance, computer support for this activity remains limited and primitive.

The idea is hardly new: Along with the development of personal computer and internet technologies in the 1980's and 1990's, a community of  researchers  has emerged that is developing tools and approaches for Computer Supported Cooperative Work, or CSCW.  Using technology to support collaboration among distributed participants was the focus of the early Media Spaces experiment at Xerox PARC [Harrison, 1993; Bly 1988], which used video  cameras  in  collaborators'

offices to create a shared virtual workplace. Since then numerous efforts have sought to develop computer based drawing environments that designers working from different locations can share. Commercial Web browsers (e.g. Netscape Communicator) now include an option for a virtual whiteboard along with text and video conferencing. These products are primarily intended for general purpose meeting support, not for architectural design.

We describe NetDraw, a synchronous collaborative drawing program built in Java—initially as a learning project—that responds to this need. We begin with a review of collaborative drawing software, and locate NetDraw with respect to similar systems. We then outline NetDraw's basic functions, focusing on those that go beyond off-the-shelf solutions. We discuss the evaluation of NetDraw in which several designers used the program to carry out a simple design task. We conclude with lessons learned from building and evaluating NetDraw and our plans to modify NetDraw to take in to account these lessons.

## 1.1 Scenario

Here is an example of how a shared drawing program might be used in a design teaching context. Steve, (a student) working at home has encountered a problem with his site plan: He doesn't know how to deal with the circulation system. Steve decides to discuss the problem with his teacher (Theresa) using NetDraw. (See figures 1 and 2). Steve and Theresa connect their computers to the Internet. Theresa starts a NetDraw server and both NetDraw clients connect to the server. Steve loads his design draft, which he has saved or scanned in an image file, into NetDraw. The image appears on Theresa's screen so that both Steve and Theresa are looking at the same design.

In the chat window, Theresa types: "What's the problem?" Steve replies, "I need help on the entry and circulation system." Theresa asks, "Show me what you have in mind." Steve uses the rectangle tool to draw two rectangles representing two buildings. The two rectangles appear on Theresa's screen as well. Theresa says, "Your entry is in a difficult place. Look at the flow of pedestrian traffic." Theresa draws two circles to indicate the entries. Then she draws an arrow on the screen to indicate the pedestrian traffic flow. The gesture appears on Steve's screen, but over a short period of time, it disappears. Steve selects the two rectangles, groups them, and moves them to a better position. "Like this?" he asks.

## 2. COMPUTER SUPPORTED COOPERATIVE DRAWING

We examined commercial products for interactive conferencing where users draw on a real or screen whiteboard, as well as research developments in CSCW drawing tools. All these systems support synchronous design drawing and discussion. Some are bitmap based and do not allow users to manipulate the objects of the design. Others, though object-based, provide inadequate mechanisms to deal with concurrent object manipulation. Some products, like Netscape's NetConference and

Microsoft's NetMeeting include simple whiteboard programs, but they provide only simple diagramming tools. On the other hand, more sophisticated solutions, such as Bentley Systems' Microstation/J, offer an object oriented model but require users to buy into an entire CAD package that is inappropriate for early or casual design discussions.

NetDraw tries to strike the middle ground between simple whiteboard programs and complex CAD package functionality. It provides basic 2D drawing program features including grouping and simple constraints, solid concurrency control in a server-client architecture. In addition, it provides an export function to write DXF files for AutoCAD. It also can communicate with our freehand sketching program (the Electronic Cocktail Napkin) over a network. NetDraw is a thin client, suitable for running on handheld computers that support Java, and we intend to use it also in mobile CAD applications [Gross et al., 1997].

We consider related research work with respect to several design issues: structured graphics versus bitmap drawing, Web based architecture versus point-to-point 'standalone' systems, the handling of concurrent operations, and how the systems support (or don't) multi-user awareness.

## 2.1     Structured vs. bitmap graphics

Two kinds of graphics structures are often used in computer based drawing tools: structured or object-based graphics and bitmap graphics. Many early collaborative drawing applications were based on bitmap graphics. For example, WE-MET [Wolf, 1992] is a pen-based sketching tool designed for small, informal work groups. It records all the meeting events, so during or after the meeting these events can be reviewed. GroupSketch [Greenberg, 1992] is a multi-user sketchpad: designers make and erase marks on a bitmap surface. It uses a pen-shaped cursor to indicate freestyle drawing mode. BoardNoter [Stefik, 1987] imitates the informal functionality of a chalkboard. It provides four tools: a piece of chalk, an eraser, a keyboard and a pointer. To draw a rectangle, the designer must first select the chalk to pick it up, then draw the four strokes of the rectangle.

Conversation Board [Brinck, 1992] is a shared drawing tool based on structured graphics. It provides multi-user sketching with various colored markers, editable multi-line text, circles, rectangles, lines, and connectors (a connector is a line or arrow between two other objects and it remains connected to them even when the objects are moved.) Tivoli [Moran, 1995] is an object-based system. As it is designed primarily for meeting support, it uses a set of built in objects for managing whiteboard text, for example, lists and tables of items. GroupDraw [Greenberg, 1992] is also object-based, users can create, move, resize and delete drawing objects. It also provides a scrollable work surface.

## 2.2     Web-based vs. 'standalone' application

Today the Web offers an 'enabling technology' for CSCW applications [Bentley, 1997]. BSCW (Basic Support For Cooperative Work) [Bentley, 1997] is

an extension of a standard Web server using the CGI (Common Gateway Interface) programming interface. It provides shared workspaces for document management. Every shared workspace is a shared repository that can contain information such as documents, images, links to other pages or FTP sites, member contact information etc. Every authorized user has the same right to access this shared workspace. Users employ applications they are familiar with to create documents (e.g. Microsoft Word for text editing); then they can upload the documents to the shared workspace with the web browser. Other users then download the document from the shared workspace and make modifications.

Alliance [Salcedo, 1997] is a cooperative authoring application on the Web that allows geographically dispersed collaborators to produce and maintain documents together, asynchronously. Every user site runs an HTTP server and an instance of the Alliance software. Unlike BSCW, Alliance provides an editor for users to work on the shared document. In a collaborative session, every user works with a local copy of the shared document, but the modification of the local copy doesn't appear on others' screens automatically. The modifier must perform a 'save' operation to explicitly notify other users about the changes; and other users have the right to decide whether or not to have their screen updated. Alliance doesn't use HTML and a Web browser; it uses HTTP (Hyper Text Transfer Protocol) to communicate directly with CGI scripts running on the servers. These systems supported by CGI and HTTP technology are asynchronous, and they use HTTP, which is poorly suited to real-time conversations.[Dix, 1997]

## 2.3      Concurrent Object Manipulation

GroupDraw [Greenberg, 1992] uses 'coupling Status' to indicate an object is a 'private', 'public' or 'sharable' one. But this 'coupling Status' only indicates whether an object can only be viewed and manipulated by the creator (private), can only be viewed by others (public), or can be manipulated by all others (sharable). When two or more users want to manipulate an object at the same time, the system assumes optimistically that there will be no conflict. Conversation Board [Brinck, 1992] uses action verification to solve the concurrence problem. Before an action is initiated, the system will check whether it can be completed. For example, the resizing tool will verify that an object still exists before it tries to change the size of the object.

SimpleDraw [Brinck, 1993] uses a token to indicate who has the right to modify objects on the drawing surface. When one user holds the token, others can only watch. In BSCW, a user can set access rights to control the visibility of the information and the operations (e.g., rename, delete, edit, and replace) that others can perform. BSCW has no locking mechanism; it adds "a note" to an object (e.g. a file) to indicate the object's status. For example, it places a note on a shared workspace, "Fred is editing this document." Other users can see that, although they may have the right to edit the document, now is not the best time.

Alliance defines four roles for every document fragment (a fragment is a part of a document): Reader, Writer, Manager, and Null. The 'reader' role only allows a user to read the fragment; the 'writer' role allows a user to modify the fragment; the 'manager' role allows the assignment or change of roles and modification of a

fragment; the 'null' role forbids the viewing of a confidential fragment. At the outset, the 'manager' role is assigned to the author who creates the document. Then the 'manager' can assign all these four roles to other users. At any time at most one author can edit each fragment. For example, if users A and B share a document with two fragments, both can view the whole document, but they cannot have the 'writer' role for the same fragment at the same time.

## 2.4      Multiuser Awareness

One feature that distinguishes a multi-user from a single-user application is that in groupware, users must be aware of who is around and what activity is happening. Several approaches have been explored. Some systems implement multiple cursors to identify different participants. In GroupSketch and GroupDraw, every participant's cursor is labeled with their name and the cursor changes shape according to different tasks. Every user in the session can see who is doing what on the shared drawing surface. Some systems, such as BoardNoter, implement a telepointer that is seen by all the users in the same collaborative session, while the individual cursors are not. To draw attention to a specific area of the shared drawing surface a user can pick up the telepointer and move it around the area.

BSCW uses different icons to support event and activity awareness. It uses five different icons to represent five event types: New, Read, Change, Move, and Touch. For example, when a document has been modified, the icon associated with this document changes to indicate that a 'Change event' has happened. The 'note' linked with a document also makes other users aware of a document's status.

Alliance also uses different icons to indicate different roles. Inside every user's local copy of the document, an icon included with every fragment indicates the user's role with respect to this fragment. When a user acts as 'reader' for a fragment, each time when a new version of that fragment becomes available, the icon before the fragment changes to notify the reader.

We agree with Stefik that when a collaborative group reaches a certain number of users, "The WYSIWIS display of cursors from multiple users is unacceptably distracting"[Stefik, 1987].
Unlike NetDraw's gesture object all these icons and telepointers are static ; designers may find it difficult to use them to show their ideas to others.


## 3.      THE NETDRAW SYSTEM

NetDraw was implemented in Java 1.1 on a Windows PC; it runs as both an application and an applet on any Java-enabled browser. It uses a client-server architecture: Any number of users can log in to the NetDraw server and use their Web browsers to observe and participate in a graphical conversation. The server echoes drawing and editing actions to all clients. NetDraw provides a simple suite of drawing commands: As with standard Draw programs a designer can create and

edit drawing objects, choose color, line weight, etc.  NetDraw goes beyond standard Draw features in several ways:

   • A designer can group and ungroup objects and apply geometric constraints to groups of objects.
   • A 'gesture' command allows the designer to make a non-permanent graphical mark on the drawing to indicate parts of the drawing during a discussion (deictic reference).
   • NetDraw also records snapshots of the drawing, which a designer can play back to review the history of a design process.  It can be set to record snapshots at regular intervals, or only to record them on command.
   • Users can provide different methods  for objects, allowing objects to have different appearances and behaviors.
   • A text based chat facility augments the graphical conversation.
   • In addition, NetDraw can communicate with AutoCAD via file transfer, and with Cocktail Napkin network clients.

   NetDraw provides two  collaborative functions:  Draw and Chat.    NetDraw's drawing features include: object based graphical editing, gestures that fade away over time, graphical underlay, grouping and graphical constraints, and recording design history snapshots.  NetDraw's text-based chat  offers  a  substitute  for  face-to-face conversation or video links. Figure 1 shows the chat window. At the top is a one-line type-in area; on the bottom a multi-line text field displays the conversation transcript. Whenever one user types a sentence, it appears on the screens of all other users. NetDraw prefaces each sentence with the author's name.
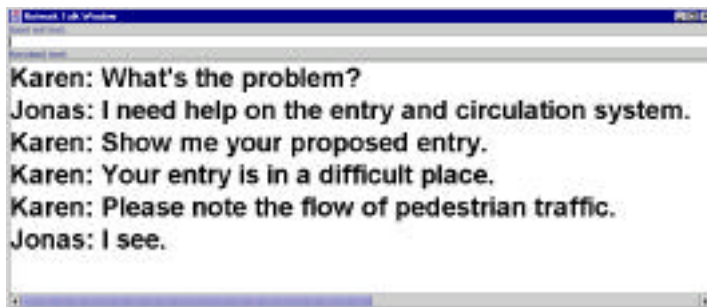


Figure 1.  The text chat window displays the author's name with each comment.

## 3.1      Drawing Features

   NetDraw  is  an  object-oriented  drawing  program.  Designers  can  draw  lines, circles, rectangles and freehand sketches in different colors, and they  can  select, move, resize, and delete those objects.  Designers can access those drawing and modifying tools quickly from a tool palette. Figure 2 displays NetDraw's drawing window. On the right is the shared surface where graphical objects are drawn. Every change made to the drawing area is immediately echoed on all designers' displays. On the left is a snapshot area where designers can capture an image of the current

state of the session. By clicking on the saved snapshots, designers can quickly review a previous state of the design history.

NetDraw can also use a graphical image as the background of the drawing surface. The image may reside on the local machine or on any Web server. This image could be a topographic base map or a sketched design drawing imported from another program. The designer can link objects with descriptive text. The text can simply indicate the object's name, e.g. "table", "kitchen", or it can provide information about the object. Each user also controls whether to display the text associated with each object. Besides the normal graphical objects offered by other drawing packages, NetDraw also provides a 'gesture' object. A designer can draw a gesture on the drawing surface; like other drawing objects, it will appear on every other designer's screen. However, unlike ordinary drawing objects, gesture objects are temporary; they fade away over a short period of time. Designers can draw attention to a specific area of the drawing surface with a gesture. A designer can use a gesture object not only to draw attention, but also to discuss their design concepts with others.



Figure 2. The NetDraw graphics screen with a scanned image underlay, geometric objects, and an ephemeral gesture (red arrow). Two snapshots from the design history are on the left; the command menu bar is on top.

7

## 3.2     Groups and Graphical Constraints

NetDraw provides a "Group" command with which designers can combine two or more graphical objects into a group. Instead of modifying every graphical object in a group one by one, designers can apply modifying actions to the group. NetDraw also provides geometric constraints that can be applied to groups. When a constraint is applied to a group, the NetDraw assumes that the first object in the group is the dominant object. For example, if a designer applies "Top =Top" (top alignment) to a group of objects, NetDraw will change the Top coordinates of all other objects to equal the value of the first (dominant) object's Top.  Later, when a designer modifies an object in the group, NetDraw will make this object the dominant object and re-apply the constraints. For example, consider that objects A, C, B, and D are grouped in that order.  When a designer applies "Top=Bottom" to make a vertical stack of the objects in the group, the system uses the following assignments to satisfy the constraints:

> C's Top ← A's Bottom
> B's Top ← C's Bottom
> D's Top ← B's Bottom

Later, when you modify object B, moving or stretching it, the system uses these assignments to satisfy the constraints:

> C's Bottom ← B's Top
> A's Bottom ← C's Top
> D's Top ← B's Bottom

Designers can add one or more constraints to a group. However, NetDraw doesn't check for conflicts among the constraints. For example, if a designer first applies "Top=Top", and later adds "Bottom=Bottom" to the same group, constraint satisfaction proceeds as follows:

1) satisfy "Top=Top", to make all the objects in the group top-aligned.

2) satisfy "Bottom=Bottom" to make all the objects in the group bottom-aligned.

If all objects are not the same height, the "Top=Top" constraint will not be maintained. In the geometric constraints satisfaction process, NetDraw only moves objects; it will not resize them.

## 3.3     Concurrency Control

The NetDraw Server mediates and coordinates designers' actions.  Before collaboration begins, one designer (the host) starts a NetDraw Server. Then all designers, including the host, can connect to the server. Once these connections have been made, the designers can begin their collaborative work. During the session, any participant can join and leave at any time as long as the server is running. When a new participant joins the session, the NetDraw server sends a message notifying all other participants.

It's likely that two or more designers may try to manipulate the same object simultaneously. NetDraw uses a locking synchronization mechanism to ensure that no two designers can modify one object at the same time. Every object NetDraw object has an unique identifying number.  A central server distributes those unique

numbers and conducts the object lock management. The NetDraw Server adopts a 'first come, first served' mechanism. When an instance of a NetDraw client starts, it connects to the NetDraw server to join the current collaborative session. At that time, the server assigns a number range to the NetDraw client. Then every time the client creates a new object, NetDraw picks a number from the number range and assigns it to that object. The NetDraw server maintains a list of currently locked objects. Before a client can change an object, the object must be locked: the client must ask the server for permission to modify the object. If another client has already locked the object then the server will queue the request and send a  notification message back to the client. Once the object is unlocked, the server grants permission to the first request in the queue.  The designer who made the request is notified and can start modifying the object. If the designer deletes an object, the server removes all pending requests to access this object.

NetDraw uses colors to indicate an object's lock status: yellow, pink and green. After the designer picks up a tool and selects an object to operate on, the object turns yellow indicating that it has requested permission from the server to lock the object. The designer cannot start the action until permission is granted. After the server replies, the object will turn either green or pink. If the object turns green, the designer was granted permission to modify, the object is locked, and the designer can begin to operate on the object. If the object turns pink, it indicates that another designer is editing the object right now, and the designer must wait. As the request is queued, once the object is unlocked, its color will turn to green indicating that modification can begin.


## 4.       NETDRAW EVALUATION

In order to obtain relevant feedback for the continuing design of NetDraw, we conducted a pilot experiment evaluating the use of NetDraw.

## 4.1     Setting

We used two PCs running Windows NT with 17" color monitors in a laboratory setting. We used keyboard and mouse as the input devices and the monitors as output devices. Both PCs are connected to the local area network, and then to the internet.  Two participants sat back to back at a distance of 8 feet, so we asked them not to talk during  the  session. This  setting  allowed us  to  observe both participants at the same time. The two participants in the pilot experiment were recruited from the students of the  College  of  Architecture  and  Planning  at the University of Colorado who had experience with design and computer use.

## 4.2     Procedure

The study participants were given a 20 minute training session on the system functionality. Then after a 10 minute practice session, we gave the participants the

design task and asked them to begin the collaborative session with NetDraw. Here is what we told them:

"The task is to design the layout for a digital design studio. The studio must accommodate 12 students, each with a drawing table and a computer workstation. In addition, the studio should provide a small lecture area for the students to gather for presentations, and a shared scanning and printing station. The room for the design studio is 34'x37', with a corner removed by an angled wall. The plan of the room has been scanned into a image file, and the file can be downloaded from a Web server."

## 4.3     Findings

The goal of our experiment was to learn how NetDraw performs in supporting a real design task. During the session we observed participants and made notes.

Although the system appeared to work well technically, no group completed the design task. All the drawings participants made are only conceptual designs. This is consistent with the experience reported in a larger experiment with collaborative CAD [Vera, Kvan, et al 1998].   There, designers responded to limitations in bandwidth by reducing low-level design exchanges and focusing on higher level discussions of design issues.

We found that the system worked well when participants were brainstorming general ideas about the design, but when it came to generate the details, one problem emerged: the participants had no idea of the size of the desk they had drawn. Here is the recorded discussion between the two participants, Paul and Mike:

| | |
|---|---|
| (after working for a while) | |
| Paul: I like the configuration. | *agree with the configuration* |
| Mike: ok so how long is the desk? | *get to the detail* |
| Paul: 4 feet? | *confused* |
| Mike: we really need a ruler function. | *identified the problem.* |

NetDraw uses one window for graphical drawing and another window for text chatting, but the screen size is not big enough to keep two windows clearly visible at the same time. During the session, the two windows partially overlapped. This caused a problem: When Participant A made a change in the drawing, or sent a text message to Participant B, if B was working in the other window as A then B did not immediately notice the new text or change to the drawing.

We also found that "associating text with graphical objects" was useful for the text discussion. During the experiments, participants often added "associated text" to a graphical object to indicate its identity. For example, after one user associated "printer station" with a rectangle object, he referred to it by name:

| | |
|---|---|
| Mike: how is the printer station? | *referred to "printer station".* |
| Paul:  it fits nicely in the corner. | *understood which object referenced..* |

During the experiment, participants often used the gesture object to show their ideas about a specific design task. For example, during a discussion about locating the 'lecture area', Mike used a gesture to show his idea of how to set up the video projector (Figure 3).



Figure 3: Using a gesture to indicate desired location of projector.

We conducted a post-experiment interview with the participants to obtain their general reactions and comments. The lack of ruler and scale was their largest concern: the participants had no idea whether the sizes of desks they drew were reasonable. Other issues were as follows:

Text scroll and wrap: Currently the text field used to display the incoming chatting scrolls, and each new sentence is added to the bottom of the window. Participants found it inconvenient to scroll to the bottom to read: the new sentence should be on the top of the window. When the sentence was too long, participants had to scroll horizontally to read it: Text wrapping should be implemented. Some participants found that changing between the text chatting window and the graphical window was tedious: integrating them on the same window would be helpful.

Text gesture: After finding the graphical gesturing helpful, some participants proposed the idea of text gesturing. Similar to graphical gesturing, text gesturing would enable a designer to type directly on the drawing surface. The text would only stay on the surface temporarily. With text gesturing, participants can directly type something like this: "how about this printer station?" near the graphical objects that represent the printer station.

Gesture duration: Almost every participant who used the gesture object reported that the staying time of the gesture object is too short. They drew the same gesture object two or more times to make sure their partners understood their ideas. Some

11

participants asked us to implement keyboard short-cuts for the commands. They had used single user CAD applications and they found keyboard short-cuts helpful: when you use one hand with the mouse, the other hand triggers commands.

# 5. CONCLUSION AND FUTURE WORK

Our goal has been to support collaborative design, so we designed NetDraw to imitate the functionality of a physical design meeting. But the experiments suggested something that we missed: Different design tasks or even different phases of the same design task impose different requirements for the software system. NetDraw can support group brainstorming to generate general ideas for a design task, but it fails to support generating design details.

There seems to be a gap between designers' ability to make judgements in computer supported drawing environments versus drawing in the physical world. In physical design meetings, although the whiteboard or tracing paper have no rulers, designers can somehow still "feel" the size of a specific object. But working on the computer screen with the scanned image, it seems difficult for designers to determine dimensions. Simply mimicking the functionality of a real world whiteboard doesn't work well.

# 6. REFERENCES

Dix, A., 1997, "Challenges for Cooperative Work on the Web: An Analytical Approach", Computer Supported Cooperative Work: The Journal of Collaborative Computing, 135-156

Bentley R., Applelt W., Busbach V., Hinrichs E., et.al., 1997, "Basic Support for Cooperative Work on the World Wide Web", International Journal of Human-Computer Studies, 827-846

Bentley R., Horstmann T., Trevor J., 1997, "The World Wide Web as Enabling Technology for CSCW: The Case of BSCW", Computer Supported Cooperative Work: The Journal of Collaborative Computing, 1997, 111-134

Bly S.L., 1988, "A Use of Drawing Surfaces in Different Collaborative Settings", Proceedings of the Conference on Computer Supported Cooperative Work 1988, 250-256.

Brinck T., Gomez L.M., 1992,  "A collaborative medium for the support of conversational props",  Proceedings of the Conference on Computer-Supported Cooperative Work 1992,   171-178.

Brinck T., Hill R.D., 1993," Building Shared Graphical Editors Using the Abstraction-Link-View Architecture", Proceedings of the Third European Conference on Computer-Supported Cooperative Work 1993, 311-324

Greenberg S., Roseman M., Webster D., 1992, "Issues and Experiences Designing and Implementing Two Group Drawing Tools",  Readings in Groupware, 609-620

Gross M.D., Do E., McCall  R.J., 1997, "Collaboration and Coordination in Architectural Design: approaches to computer mediated team work", TeamCAD '97, 17-23

Harrison, S., 1993, "Computing and the Social Nature of Design." ACADIA Quarterly 12, no. 1, 10-18.

Ishii H., Kobayashi M., 1992, "ClearBoard: A Seamless Medium for Shared Drawing and Conversation with Eye Contact", Readings in Groupware, 829-836

Moran T.P.,  McCall K.  et al.,  1995, "Some design principles for sharing in TIVOLI, A whiteboard Meeting-Support Tool",  in Greenberg, Hayne, and Rada, Groupware for Real-Time Drawing -- a designer's guide,  McGraw-Hill, 24-36

Stefik M., Bobrow D., Foster G. et al., 1987, "WYSIWIS:Early Experiences with Multiuser Interfaces", Readings in Groupware, 585-595

Salcedo M.R.,  Decouchant D., 1997,  "Structured Cooperative Authoring for the World Wide Web", Computer Supported Cooperative Work: The Journal of Collaborative Computing, 1997, 157-174

Vera, A., Kvan, T., et al., 1998, "Expertise, Collaboration, and Bandwidth", Proc. ACM Human Factors in Computing (CHI '98), 503-510.

Wojtowicz J., 1995, Virtual Design Studio, Hong Kong University Press

Wolf C.G., Rhyne J.R., 1992, "Communication and Information Retrieval with a Pen-based Meeting Support Tool", Proceedings of the Conference on Computer-Supported Cooperative Work 1992,  322-329.