DEVELOPMENT OF A GRAPHICAL USER INTERFACE FOR DISTRIBUTED COLLABORATIVE WRITING AID

Technical Report 95-06

Kai H. Chang and Shefali H. Gajiwala

Department of Computer Science and Engineering
Auburn University
Auburn University, Alabama 36849

May 16, 1995
DEVELOPMENT OF A GRAPHICAL USER INTERFACE
FOR
DISTRIBUTED COLLABORATIVE WRITING AID

Technical Report 95-06

Kai H. Chang and Shefali H. Gajiwala

Department of Computer Science & Engineering,
Auburn University,
Auburn, Alabama 36849-5347

May 16, 1995
ABSTRACT

Computer supported collaborative writing is an interdisciplinary research area that addresses the issue of the design, implementation and use of technical systems that support people working cooperatively. Despite the technical advances over the last few years there are relatively few applications available that provides effective functionality for collaborative writing. The distributed collaborative writing aid (DCWA) is a project developed for networked workstations. In this report, DCWA is described with the emphasis on the graphical user interface (GUI).

The GUI is implemented in X-window system using Xt intrinsics and Motif Widgets. It provides an user friendly interface and supports a text editor with all the standard editing and search facilities. This GUI also provides shared and private workspaces, space for hierarchical tree like representation of the structural organization of the file and the space for the feedback from the system. This GUI also provides support for the semantic network. Above all these, the GUI is designed on the "just-in-time" philosophy. This means that the information is kept hidden from the user until requested or immediate attention is needed. This concept makes the system more user friendly by reducing the mental burden on the user.

1
## Table of Content

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Background</td>
<td>3</td>
</tr>
<tr>
<td>2. General Classification and Literature</td>
<td></td>
</tr>
<tr>
<td>Review of CSCW Systems</td>
<td>5</td>
</tr>
<tr>
<td>3. Introduction to DCWA</td>
<td>11</td>
</tr>
<tr>
<td>4. DCWA Software Architecture</td>
<td>14</td>
</tr>
<tr>
<td>5. Graphical User Interface</td>
<td>16</td>
</tr>
<tr>
<td>5.1. User’s Procedure</td>
<td>17</td>
</tr>
<tr>
<td>5.2. Features of GUI</td>
<td>22</td>
</tr>
<tr>
<td>6. Conclusion</td>
<td>32</td>
</tr>
<tr>
<td>7. References</td>
<td>33</td>
</tr>
<tr>
<td>8. Appendix</td>
<td>37</td>
</tr>
</tbody>
</table>
Chapter 1
Background

Most of the computers today are designed for access by one user like personal computers or personal accounts on shared systems. They are considered in isolation from other tools and from other people. This is different from the scenario of real life. That is, most human activities require some degree of coordination with others. They exchange ideas, information, and thoughts among other things. The recognition of this fact can contribute to the evolution of computer technologies that support the cooperative aspect of the human endeavors.

This recognition resulted in the emergence of a new class of application software for group activities known as "Groupware" or, more formally, "Computer Supported Cooperative Work (CSCW)". It is a class of software designed to take group work into consideration in an integral way. CSCW focuses on the role of the computer in group work.

Two general approaches exist in the development of CSCW systems [1]. One approach deals with the development of systems that support the exchange of information between the users, whereas the second approach exploits the sharing of information to allow cooperation.

Many tools, such as text editors and graphic editors, are suitable for sequential information sharing. Some tools, like electronic mail, are available for information exchange. But, they are diverse in appearance and behavior so they all can not be used on the same platform. There is a need for tools that address the issue of
information exchange and information sharing in a single system.

As CSCW is still at the primitive stage of research, not many systems or tools have been developed so far that satisfy all the needs of collaborative work. In this report, a CSCW prototype, Distributed Collaborative Writing Aid (DCWA), that has addressed some issues of collaborative work is discussed. This report first gives the general discussion and classification of CSCW systems. Following this, an introduction to DCWA is provided along with its major features and elements. The Graphical User Interface (GUI), a key element of the tool that was designed and implemented by the author, is also described in this report.
Chapter 2

General Classification and Literature Review of CSCW Systems

CSCW users, in general, have complex and conflicting requirements and responsibilities. They are from different backgrounds and work in an environment that may not provide a conventional sense of participation. That is, they work at distance and sometimes at different times. These constraints lead to the following classification of CSCW systems:

1. Based on the geographical distribution,
2. Based on the closeness of collaboration, and
3. Based on the conflict resolution.

1. Classification based on the geographical distribution

This classification is based on how the users are distributed geographically [2].

(a) Co-located System: This type of systems requires the presence of all users at one location, such as a meeting room.

(b) Virtually Co-located System: This type of system does not require all users to be present at one location, like a meeting room, but must be within the reach of a LAN.

(c) Locally Remote System: This type of system provides high bandwidth real-time accessibility between participants, using a high speed WAN.

(d) Remote System: This type of system provides only minimal accessibility between participants such as e-mail.
2. Classification based on the closeness of collaboration

    This category includes concepts of "division of labor" and "shared mind" [3].

   (a) Parallel Working: In this class, a task is divided into subtasks and all the
       subtasks can be accomplished in parallel. It supports the concept of a "division of
       labor".

   (b) Sequential Working: In this class, a task is divided into subtasks in such
       a way that the output from one stage is passed to the next stage sequence.

   (c) Reciprocal Working: In this class, the partners are allowed to work
       together to create the common product by mutually adjusting their activities as shown
       in Figure 1. This class promotes the concept of a "shared mind".

3. Classification based on the conflict resolution

    Participants of a collaborative work team have different requirements,
    expectations, values, beliefs, and roles. In addition, they are working at geographically
    distributed locations. All these result in the conflict of some sort. This type of
    conflict can be sorted out to some extent at the design level. This leads to the
    following subclasses:

   (a) User's authority for access: In this type of CSCW systems, the users are
       given roles to play when a document is created. The user can function as co-author,
       leader, commenter, etc. Thus, the participants are responsible for specific actions in
       the specific segment of the document, thereby reduce the conflict to some extent.

   (b) Segment level locking: It is interesting to note that collaborative work is
       based on the mutual confidence and trust among the participants. As a result, in some
       systems, all the participants are allowed to edit and comment all segments of the
       document. This results in "bumping into each other" while trying to select the same
       segment to work on. This requires some type of segment level locking.
Figure 1 Strategies for coordinating collaborative work.
(c) **Private and Public Workspaces**: Collaborative work becomes more effective and fruitful if users can also see the others work in progress. Some systems provide private and public work spaces. In a public work space, users can read others’ work in progress, and can modify his/her own segment in the private work space.

**Literature Review of CSCW related systems**

As collaborative writing is one of the most popular domains of CSCW, a brief discussion on collaborative writing will be given first followed by a review on multimedia applications.

One experimental collaborative writing system using electronic mail was developed by Diaper[4]. This system is effective only for a small group of users. It allows users to work in both sequential and parallel collaborative styles. The system can be used by a variety of hardware and software platforms that are geographically scattered. The main limitation of the system is that the text should be of standard ASCII. In addition, giving comments is a difficult and time consuming process in this system. The HICOM system[5] developed in UK, provides members with information services that are collaboratively maintained. The advantage of HICOM is that it uses VAXNotes. This gives some structural organization to messages.

Cognotes[6], a tool developed at Xerox PARC’s helps members of a group to discuss their ideas in a highly interactive environment. The tool permits the users to develop individual ideas in a private workspace and then post them to a shared "liveboard". The main restriction is that users can not see the generation of the private notes. The new version, known as "Cnoter"[6], provides shared editing and the "What You See Is What I See (WYSIWIS)" paradigm.
LIZA[7] provides a high level collection of tools for rapid groupware prototyping. These high level tools provide support for sending messages, editing collaboratively and monitoring the group[4]. ShrEdit and Grove[8] are shared editors designed for parallel editing. Participants can watch others’ work as it is created. This continuous updating gives user the most updated version of the shared document. The deficiency is that both tools support only the synchronous mode. ShrEdit has segment level locking which is not provided in Grove.

DistEdit[9] has the ability to support different editors simultaneously. It allows users to take turns at making changes while other users observe the changes as they occur. It allows one to build interactive group editors for a distributed environment. The inadequacy is that the user can not edit and work on the text simultaneously. Conus system[10] has a common workspace area where interactions take place. Outside this area, applications and data are private. Opening the common workspace indicates the user’s readiness for interaction with others. But, for common workspace, it has only the WYSIWIS mode.

Some of the systems described above have rigid structures for textual collaborative writing. The term, hypertext, describes any system employing nonlinear structuring of text, graphics and other media[11]. These systems contain a network structure of nodes and links. Hypertext documents are made up of connected nodes. Connection between nodes is a link that represents an association between the information held in the nodes. Windows are used to provide the display and editing services. It also allows the user to traverse the links. The drawbacks of the hypertext are disorientation and cognitive overhead. This can be partially resolved by improving the user interface design.

The full hypertext developed at Xerox PARC is NoteCards[12]. It is basically designed to gather information about a topic and then produce an analytical report. A
valid document, known as Notefile, could consist of a single node or many interrelated nodes existing in a user defined hierarchical structure. The Notefile consists of any number of individual nodes. Each node occupies its own space on screen and there is no limit on the number of nodes visible at any one time. NoteCards allow several users to work on the same Notefile at the same time. Creation of new types of nodes including text, video, animation and graphics is also allowed. QUILT[13] is a hypertext system that provides co-authoring services. In this system, all users can read publicly available documents but only certain privileged users, authors, or co-authors can modify the document.

The AQUANET system[14] is a hypertext tool that provides collaborative knowledge structuring in a WYSIWIS mode. But, it does not provide synchronous shared view on any communication facilities. The SEPIA[14] and rIBIS[15] are synchronous hypertext systems providing various collaboration modes. The modes they provide include independent, loosely coupled, and tightly coupled. This allows users to work independently. It also allows them to work by sharing information and resources with different degrees of constraints. Major improvements of SEPIA over rIBIS include automatic mode switching and the use of composite nodes.

CAVECAT[16] is intended for use in specially designed meeting rooms. It supports distributed meetings where participants use desktop video for the conference. The drawback is the quality of the network audio connection. Otherwise, it is a good system for synchronous meetings. MERMAID[17] and SPIN[18] systems are prototypes that provide real-time conferencing environments for geographically distributed participants by using synchronous textual, audio and video communications. Rapport[19] and Cruiser[20] also support interactive real time distributed conferencing among two or more users. But, they support only locally remote groups. They also support the conventional telephone service within its conferencing framework. MILAN[21] is another powerful system that allows video conferencing along with collaborative writing in the private and the shared workspace.
Chapter 3
Introduction to DCWA

Till today, very few systems have been developed for collaborative writing. Distributed Collaborative Writing Aid (DCWA) is a tool that provides services for graphical and textual editing, viewing, and searching for segments of document. The software is being developed using 4.3 BSD UNIX, the TCP/IP protocols, OSF's Motif toolkit, ANSI C and C++. The DCWA has most of the characteristics required for the collaborative writing. In addition, this software has many other features that are not included in other software packages of this type.

The major attributes that make the DCWA unique are as follows:

(a) Synchronous and asynchronous modes

Synchronous working mode allows geographically distributed group members to work concurrently on the same file. This saves time and money for bringing the group together for a face-to-face meeting. Thus, this facility bridges the place gap. On the contrary, in an asynchronous working mode, users can join, quit and rejoin at any time. This gives user the required time to think and reflect on the given problem before reaching a decision without using the time of the other users. This indicates that asynchronous mode bridges the time gap. Unlike most of the available collaborative working tools, the DCWA is designed to work under both the synchronous as well as the asynchronous modes, thereby enhancing its efficiency and effectiveness.
(b) Organizational structure of the document

In DCWA, a tree-like node structure is used to represent the document in which the leaf nodes contain the non-overlapping "chunks" of a document. The organizational structure of a document such as main topics, subtopics, and sections is represented in a hierarchical tree fashion. The label of a node gives an idea about the contents of the node. Moreover, DCWA allows the user group to define their own structure rather than having to work with an application defined structure.

This effectiveness is greatly enhanced by integrating a semantic network with the node structure. Each node in the node structure has attributes that describe the nature of node contents. Many of the attributes are static like creation of node and time of creation, but others may change over time. For example, the attribute about the last time the node was modified is dynamic in nature. In addition, semantics about the node, like keywords and topics, are also included. Users can also defined their own attributes and values.

(c) Version Control

When multiple authors and commenters are making changes to a document, there has to be a reliable mechanism that ensures everyone is working on the latest version of the document. The approach taken in DCWA is somewhat related to the approach taken by Miles[8]. That is, segment level node locking is used. A detailed description is given later in the report.

(d) Communication Channel

When users work on a shared object or document, it is necessary that they have an effective means of communication. It is interesting to note that other editing tools such as Shredit[4] and Grove[8] do not provide any support for direct
communication within the tool. The users have to rely on other means of communication, like telephone or e-mail, to coordinate their activities. In DCWA, the general solution to the problem is provided by means of an easily accessible and effective method of information exchange -- the "talk" channel. Just by selecting a recipient's name from the group list, the user gets the window to "talk" to the other group member. If the other user is not ready to "talk", the person sending the "talk" message would get appropriate feedback.
Chapter 4
DCWA Software Architecture

DCWA can be divided into four subsections -- Front-End, the Group Server, the Database, and the Graphical User interface[22]. They are briefly described below:

The Front-End

In DCWA, the user can have two types of interactions. One is a text based interaction and the other is a graphical interaction. The creation of a group, the addition and the omission of a group member are available through the text based Front-End. User can use any UNIX command on the prompt of the Front-End, including invocation of a text-editor, such as vi. The full scale graphical interface can also be invoked from here.

The Group Server

The communication among all processes is an important design aspect of DCWA. The major challenge is the multicasting of messages to all members in a group. The approach taken is to arrange all group members in a circular ring. This specifies the precise position for each user. Each member thus has a unique predecessor and a unique successor. A message is sent over a ring, so there is no duplication of the message for multiple destinations. This reduces the total number of messages sent over a network. The message passes through all members when it completes the circle. The original host, then receives the message again and removes it. This approach leads to the reduction in the delay of message delivery.
The Database

For efficient and logical functionality of the tool, the database is a very important component. Many issues are addressed by the database such as backup, recovery, defining views, view sharing, conflict resolution, and run time modification of the logical structure. In addition to the above common issues, the overlay of a semantic network over the document organizational structure is unique to DCWA.

The Graphical-User Interface (GUI)

The user interface is an important element for any tool. In a complicated system like DCWA, the Graphical User Interface is provided to make the tool user friendly. The GUI at each local machine provides a user with the necessary functionality to participate in the cooperative writing. The design of this GUI is based on the "just-in-time" philosophy. This means that the information is kept hidden from the user until requested or immediate attention is needed. This concept makes the system more user friendly by reducing the mental burden on the user. When the designed GUI is initiated, only a basic text editing window with various pulldown menus is provided. A user can choose to work alone or work with other group members by tailoring the interface layout to meet his/her own needs. Detailed discussion of the GUI is given in the next section.
Chapter 5
The Graphical-User Interface (GUI)

The DCWA GUI is implemented in the X-window system using the Xt intrinsics and the Motif Widgets. The reason for selecting X-windows are as follows:

- It has many standard graphical user interface objects such as menus buttons and dialogue boxes that are widely used in other editing tools.
- It has extensive libraries of high level user interface objects that can be combined into a powerful user interface.
- The use of X-Windows is becoming more widespread and is available on most graphic workstations.
- It uses the Client-Server architecture and any applications developed under it will be device-independent.
5.1 User’s Procedure

In the following paragraph, the procedure for using this tool is described along with some edit commands that differ from the conventionally available tools.

By typing "dcwa" at the system prompt, a user can start working with the tool. From the Front-End, a user can also start the vi-editor or the text editor. As a result of this, the user can work with other available editors without going out of DCWA. Facilities for creating a group and joining the group are also available in the front-end. The full-scale Graphical User Interface (GUI) can be invoked from here.

Once the user invokes the GUI, he/she has two options - either to select the text editor or to use the graphic editor. Once the user makes the selection, the other option will be "dimmed". In this report, emphasis is made on the Text editor, as it is developed by the author of this report.

As shown in Figure 2, the screen of the text editor has been divided into three major parts as described below:

- Private space for the user to edit his/her segment of the shared document,
- space for the tree like structural representation of the shared document,
- space for the feedback of the user’s actions.

All others will be displayed only at the user’s request. This type of display reduces the complexity of the screen.

At this point in the GUI, the user will have two options for editing,
1. Open an existing file with a logical view, or
2. To create a new logical view.
Figure 2  Snapshot of texteditor showing some edit options.
After opening a file, the user can read any leaf node or edit any unlocked leaf node. Once the user selects any node for editing, that particular node is "locked" for all other users. He/she then has the following options:

1. OPEN: Opens file selection box. By selecting the filename from this box, GUI requests the Database for the permission to open the requested existing file.

2. NEW: Requests the Database for permission to create a new file.

3. OPEN_FILESELECTIONBOX:
   Opens file selection box. Selected existing file will be open from file server.

4. SAVE: Saves the current version in the temporary memory of the database.

5. COMMIT: Saves current version of the node permanently to the group file.

6. EXIT_WIN:
   Allows the user to quit texteditor. If data is not saved user gets appropriate message.

7. QUIT: Allows the user to quit the GUI without saving current node version. If the user tries to quit without saving, the GUI gives a warning message to the user and allows the user to take appropriate actions.

In addition to the above mentioned options, there are several other options available for the user. These include the display of the group members who are currently active, the segment(s) of the shared document that is being edited. For example, if the user A wants to see the portion of the document that is being edited by another group member B, A just needs to open the menu for the list of group members and clicks at the name of user B. A new window will be popped up (with "Read Only" permission) showing the contents of the node on which user B is working. These windows are resizeable and repositionable, and the user can close them as icons.
and reopen them as a full size windows. This also reduces the complexity of the screen.

In addition, conventional functions like FIND, REPLACE, FIND AND REPLACE, CUT, and PASTE are also available at this point for editing purpose as shown in Figure 3. Labels and actions of these options are similar to what is currently available in other tools like WordPerfect and IslandWrite. This also makes it easy for the user to use the tool with relative tranquility.

Besides these, other node manipulation functions are also available: divide node and merge node, redo, undo. Each node of the file structure has some attribute-value pairs. A user can assign values to these attributes any time after creating the node. Values for some attributes may serve as the "keywords" that define the contents of the node. Nodes with the related content can be determined by assigning common values to these attributes. Search facilities to find related nodes is also part of the user interface functionality. All of the above functions are only for node operations, so they are not available from the main menu. They are available only from the graphical representation of the node structure. This again reduces the complexity of the main menu and makes menu more user friendly.
Figure 3  Snapshot of texteditor showing find-replace option.
5.2 Features of the GUI

The DCWA prototype provides some unique features. The support provided by GUI is described in the following section:

Private and Shared Views

A fundamental requirement for collaborative working tools is to provide a coordinated interface for all participants. Such a multi-user interface allows the participants to interact with each other easily through a computer medium. To support and encourage cooperation, a CSCW tool must allow users to be aware of the activities of others. The purpose of a cooperative multi-user interface is to establish and maintain a common context, allowing the activities of one user to be reflected on the other user’s screen. This common context is achieved by application information sharing, real-time presentation and shared information manipulation. The extent to which a multi-user interface supports the information sharing depends on the interface coupling.

One type of interface coupling is strict WYSIWIS (What You See Is What I See) which gives all group members the same image on their displays. This type of interface has limited usage. For DCWA, a relaxed version of WYSIWIS is used. It gives each user a private working space as well as a public space. In addition, the relaxed version permits the user to change the size and position of all windows anywhere on the screen, thus allowing personalized screen layout.

The DCWA also provides an environment that permits parallel working on shared documents. At any given time, different users can select different segments of the shared document to work on. The DCWA tool permits this type of simultaneous interaction by storing non-overlapping segments of the shared document in leaf nodes.
By selecting one of the nodes, the user can edit the content of that particular node. The content of that node appears in the private working space of that user’s display and that particular segment of the document is locked by that user. The color of that particular node changes on the screens of all other users at the same time. This gives other users indication that some editing is taking place in that particular segment of the shared document. All other group users will have the READ ONLY access for that node while editing is going on and their corresponding screens (if open) will be refreshed every 15 seconds (Figure 4). In addition, any user of the group can see which part of the shared document each participant is editing by clicking at the name of that particular user. This allows group members to have an up-to-date information on the shared document at any given time. The user performing editing can "unlock" that particular node by selecting another node. Thus, different users can work on different segments of the same document at any given time. The only restriction of this system is that different users can not work in parallel on the same segment.

Logical View

The node structure of a document, also known as logical view, can be represented as a tree-like hierarchical structure. Leaf nodes contain linear, non-overlapping contents of a document. Once the logical structure is created, the user can access the contents via structural representation. This allows the user to edit a small segment at a time without manipulating a large portion of the text. Users also have options to create and modify the logical structure.

1. Logical view for an empty document:

In DCWA, the user can create the logical view for an empty document. When the user selects "Create logical view" option from the options, the default "Head" node appears. The user can divide that node into subnodes by choosing the "Divide Node".
Figure 4  Snapshot showing other users’ working area.
option. The user has to provide the name of the node. However, at this point, the content of that node will be NULL. The user may also provide node attributes to a NULL node.

2. Logical view for an existing document.

A logical view can also be created for an existing document. In order to create the logical structure for an the existing document, the user needs to first open a file from the NFS Server, by choosing the option "OpenFileSelectionBox". After that, the user needs to choose the "Create logical structure" option. This will create a default node labeled "Head". The entire document will then be the content of the Head node. The user can then highlight the required segment of that document using the pointing device like mouse. When the user chooses the "Divide Node" option from the menu, a new node is created and the system asks for the node name. When the user provides the node label, the highlighted portion of the document becomes the content of the new node and at the same time the highlighted portion is removed from the head node.

There is an additional feature provided by the GUI for error prevention. That is, if there is a content in the text editor, the user has to make a selection from the beginning. This prevents losing any data by error.

3. Division and merging of nodes:

A user can alter the existing structure by dividing or merging nodes. The procedure for dividing the node is same as above. For merging, the user has to select nodes that do not have any child and have the same parent. Then by selecting "Merge Node" option from the menu, the contents of leaf nodes will be placed in the parent node. The process can be explained as follow.

As shown in Figure 4, the two leaf nodes "create_menu" and "make_node" have the same parent node "menu_create". Thus, these two leaf nodes can be merged
into parent node by "merge_node" option. This process will transfer the contents of two leaf nodes into the parent node, and simultaneously will eliminate leaf nodes from the node structure. The nodes "edit_node" and "make_node" can not be merged, as they have different parents. Similarly, the nodes "definition", "node_create", and "add_node" can not be merged as the node "node_create" has children.

4. File Insertion:

In DCWA, a user can insert or append a file into another existing file. The process is as follows:

To append a file in existing file, user can open an existing file that has a defined logical view. In the Figure 5, the existing file is db.c. Then user can make another empty node which is a child of "Head" node. In Figure 5, that node is gui.c. Then by choosing Open_fileslectionbox option, user can get the file to append in the texteditor and can start dividing it as before.

To place the new file at any required intermediate place, the user needs to divide the required intermediate node of the existing file. The contents of the file to be appended should then be placed in the newly created NULL node. Thus, with the help of the functionality provided by the tool, the user may generate the logical structure as shown in Figure 5.

Node Attributes

As mentioned earlier, each node in the structure has a name or label. This label has limited utility, except that it can be used as the title or key word that describes the content. In DCWA, the attribute-value pair feature is augmented to enhance the functionality. Each node has some attributes. Some of them are permanent, e.g., creator of the node and creation time, whereas some change over
Figure 5  Snapshot showing appending of new the file (gui.c) to the existing file (db.c).
time, e.g., last time of modification. Some attributes get values from the system, like creation time and modification time, whereas some others are required to be filled by the user, like key words and topic. The list of attributes supplied to the Database by the user interface is shown below:

Name
Topic
Description
Keywords
User Defined Attributes
User Defined Attribute values
Read Access list
Edit Access list
Media type
Filename

When the user chooses "Add Node Attribute" option from the menu, the information box will pop up asking for the node label. When the user types the label of the node, new window pops up, allowing the user to add values for the given attributes (Figure 6). The values are temporarily stored by GUI. When the user COMMITS the structure, all the information of the node will be sent to the Database.

The flexibility is provided to the user by not forcing the user to provide values as soon as he/she creates a node. But, the user can fill the values for any attribute at any time during the process of creating the node structure.
Figure 6  Snapshot showing window for providing values of attributes.
Search and Queries

The ability to search and query nodes, is a unique characteristic provided by DCWA. A user can make a query for a node with some particular attributes. In return, the associated values are shown to the user (Figure 7). This helps users with different interests in locating particular type of nodes.

Semantic Network

As described above, each node has attributes and their associated values. This semantic information can be used for searching nodes. For example, the user can specify individual keys or combination of keys to locate nodes that have specific characteristics. Similarly, the user can keep track of his/her own work, by providing the attribute "Last Modification Time". The value of the attribute "Media Type" can also help the user to distinguish textual or graphical segments. Thus, automating the finding of document segments provides accuracy and efficiency to the users.

As mentioned in [23], the requirements of the user's awareness of the entire structure may hinder the "division of labor", but having the facility of semantic structure will greatly reduce the user's burden when they need to access nodes with specific criteria.
Figure 7  Snapshot showing window for node search and queries.
Chapter 6
Conclusion

In this report, the Distributed Cooperative Writing Aid (DCWA) is described with the emphasis on the design aspects of graphical user interface (GUI). The tool supports many functionalities for cooperative writing. It provides both textual and graphical editings, viewing, and coordination services to team members. The flexibility of the tool allows it to be used in either synchronous or asynchronous mode.

In order to make DCWA a successful collaborative writing tool, all group members are given complete access to the shared document. However, by incorporating the concept of segment level locking, complete accessibility of other group members is temporarily restricted when a particular segment of the document is edited by some member of the group. The GUI is designed in such a way that all members of the group will have up-to-date information on the shared document with a maximum delay of 15 seconds. Besides, the documents are organized in a hierarchical tree structure that defines the non-overlapping working spaces for the users. These features along with the designed "talk window" provide a conflict-free collaborative environment. The GUI also uses a semantic network that supports searching facility and allows a user to migrate from one node to another. Above all these, GUI is designed on the "just-in-time" philosophy. This means that information is kept hidden from the user until requested or immediate attention is needed. This concept makes the system more user friendly by reducing the mental burden on the user.
References


10.


