RESEARCH REPORT ISIS-RR-94-1E

Support of Human Thinking Processes with D-ABDUCTOR

Kazuo Misue and Kozo Sugiyama

January, 1994

Institute for Social Information Science,
FUJITSU LABORATORIES LTD.

Numazu office
140 Miyamoto, Numazu-shi, Shizuoka 410-03, Japan
Telephone: +81-559-23-2222 Fax: +81-559-24-6180

Tokyo office
9-3 Nakase 1-chome, Mihama-ku, Chiba-shi, Chiba 261, Japan
Telephone: +81-43-299-3211 Fax: +81-43-299-3075

Support of Human Thinking Processes with D-ABDUCTOR

Kazuo Misue and Kozo Sugiyama
FUJITSU LABORATORIES, ISIS
{misue,sugi}@iias.flab.fujitsu.co.jp

ABSTRACT

D-ABDUCTOR is an interactive system to support personal and group thinking processes by using diagrams as media of interaction between a person and a computer or among persons. This system has the following features: (1) supporting diagrammatic thinking processes interactively, (2) providing an advanced graphical user interface, (3) providing direct manipulation and animation environment, (4) exploiting advanced capabilities of computers. This paper gives the outline of D-ABDUCTOR with its features and explains each of its facilities. Then it shows how to use D-ABDUCTOR by showing a practical example on "future directions for D-ABDUCTOR," and reports applications of D-ABDUCTOR to practical tasks. Finally the result of the practical example is described as future directions in the concluding remarks.

1. INTRODUCTION

We often exploit diagrams when we think. One reason of using diagrams is that they are good media to represent and organize personal thought, and to share ideas among persons on group works. However, editing, reforming, and re-drawing diagrams, which are necessary on thinking processes, are troublesome. Moreover they may not be essential for thought. In this paper, we describe an idea organizer that exploits diagrams effectively as media for human-computer interaction and for interaction among persons.

We developed a diagram based idea organizer D-ABDUCTOR[1,2] based on the KJ-method[3,4] which is well known in Japan as an effective card and diagram based method for organizing ideas and solving problems without computer support. So far we have developed elementary techniques for diagram handling[5]: automatic drawing of diagrams[6], generating freehand-like shapes[7], multi-viewpoint perspective display method[8] offering the whole and detail view of a diagram, and diagrammatic dressing[9] by using importance of diagrammatic elements. Lately we have integrated these elementary techniques with a direct-manipulation and animation environment into a system D-ABDUCTOR, and have supplemented communication facilities[10] for group works, and multi-media facilities to deal with pictures to make D-ABDUCTOR more practical.

The name of the system, "D-ABDUCTOR" originated from that the originator of the KJ-method called the essential part of the method, in which fragments of ideas were re-arranged and organized, "abduction" in his book[4, p.33]. The initial letter "D" means "diagrammatic."

As related works and systems[11], we should mention the KJ editor[12], which was developed to aim at simulating manual works of the KJ-method on computers and CONSIST[13], which supports knowledge acquisition by using the KJ-method. We should also mention some software tools called "outline processors" (for example, Acta 7, MORE and Inspiration of Macintosh) because they can be regarded as systems to support thinking processes even if they are not related with the KJ-method directly. D-ABDUCTOR can be distinguished from these systems and tools by that it provides automatic graph (or diagram, not only trees) drawing facilities and use them to support dynamic thinking processes effectively.

The reminder of this paper is organized as: section 2 presents an outline of D-ABDUCTOR. Section 3 explains each facility of D-ABDUCTOR in more detail. Section 4 explains how to use D-ABDUCTOR by using a practical example. Section 5 reports experience with applying D-ABDUCTOR to practical tasks. Section 6 describes concluding remarks and planned future research.

2. FEATURES OF D-ABDUCTOR

D-ABDUCTOR runs on Sun SPARCstations and uses the X Window System. D-ABDUCTOR is available either for individual use or for group cooperative-work when worksta-

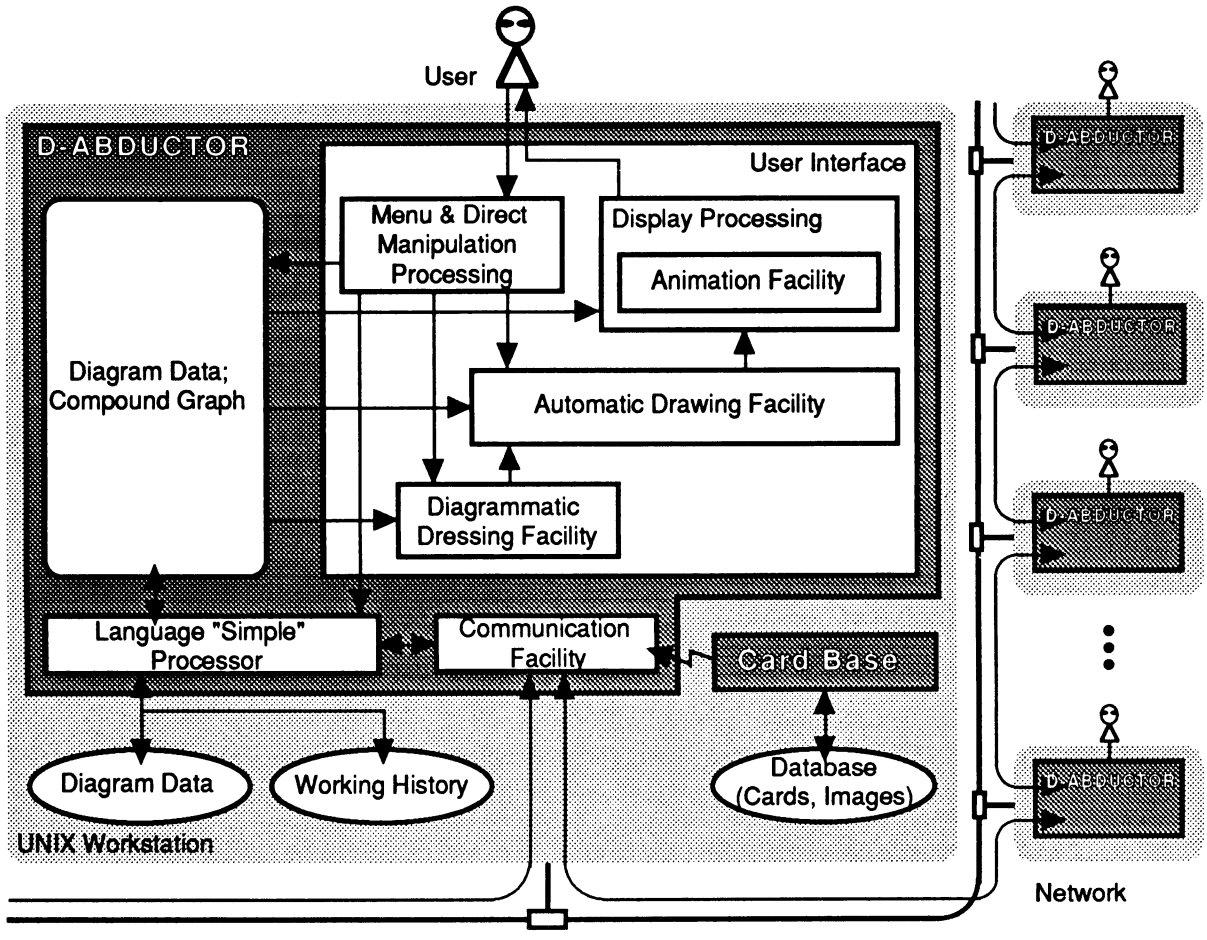

Figure 1: The system architecture of D-ABDUCTOR

tions are in a network environment. Figure 1 represents the system architecture of D-ABDUCTOR.

2.1. Interactive Supporting Diagrammatic Thinking Processes

To investigate the possibilities for computer support of human thinking, we have aimed to automate the main part of the KJ method, which is the phase of organizing idea fragments. This phase consists of four steps: (1) label making, (2) group organizing, (3) chart making, and (4) description. Several rounds of these four steps are usually repeated. We explored the diagrammatic features of the KJ-method and considered the procedures in the method from the viewpoint of diagram languages[14]. Table 1 shows the procedures and the diagrammatic operations needed for each step of the KJ-method.

2.2. Advanced Facilities to Handle Diagrams

In building D-ABDUCTOR we put emphasis on developing an advanced facilities for a graphic user interface (GUI) to handle diagrams. The diagrams used in the KJ-method have

Table 1: The procedures and the diagrammatic operations in each step of the KJ-method.

STEP		PROCEDURES	OPERATIONS
(1) Label Making		Collect segments of ideas and information, and write each of them on a card.	Create cards. Delete cards. Input text, illustrations and pictures.
(2) Group Organizing	(2-1) Label Spreading	Spread cards (at random) so that all of them can be read.	Move cards.
	(2-2) Label Grouping	Move cards with related ideas together to make groups.	Move cards.
	(2-3) Title Making	Write a title for each group on another new card, then bundle a group together, and treat it as a card.	Create cards. Input text. Collapse groups (make each group into a card).
(3) Chart Making	(3-1) Spatial Arrangement	Arrange cards or groups suitably on a planar surface and unbundle cards.	Move cards. Move groups. Expand cards (into original groups).
	(3-2) Chart Drawing	Draw a boundary line around each group and adjacency lines between cards and groups.	Draw boundary lines. Draw adjacency lines.
(4) Description		Make the contents of the chart into a document, or present it by using the chart.	Convert diagrams into documents.

features of Venn diagrams that represent sets and subset relationships by geometric inclusion relationships between areas, as well as network diagrams that represent semantic relationships between ideas with lines connecting card images. The logical structures for such diagrams can be modelled as "compound graphs"[6] which extend the notion of a graph. Roughly speaking, compound graphs are graphs in which each node can include a graph, the notion is closely related to that of higraph[?]. Advanced facilities for handling such diagrams include automatic layout for compound graphs, diagram dressing, and animation. These facilities are described in detail in section 4.

2.3. Communication Facilities for Group Works

We are trying to find new styles of supporting thinking processes by exploiting advanced capabilities of computers such as communication facilities The communication facilities enable the support of not only personal thinking processes but also group processes. Two or more users

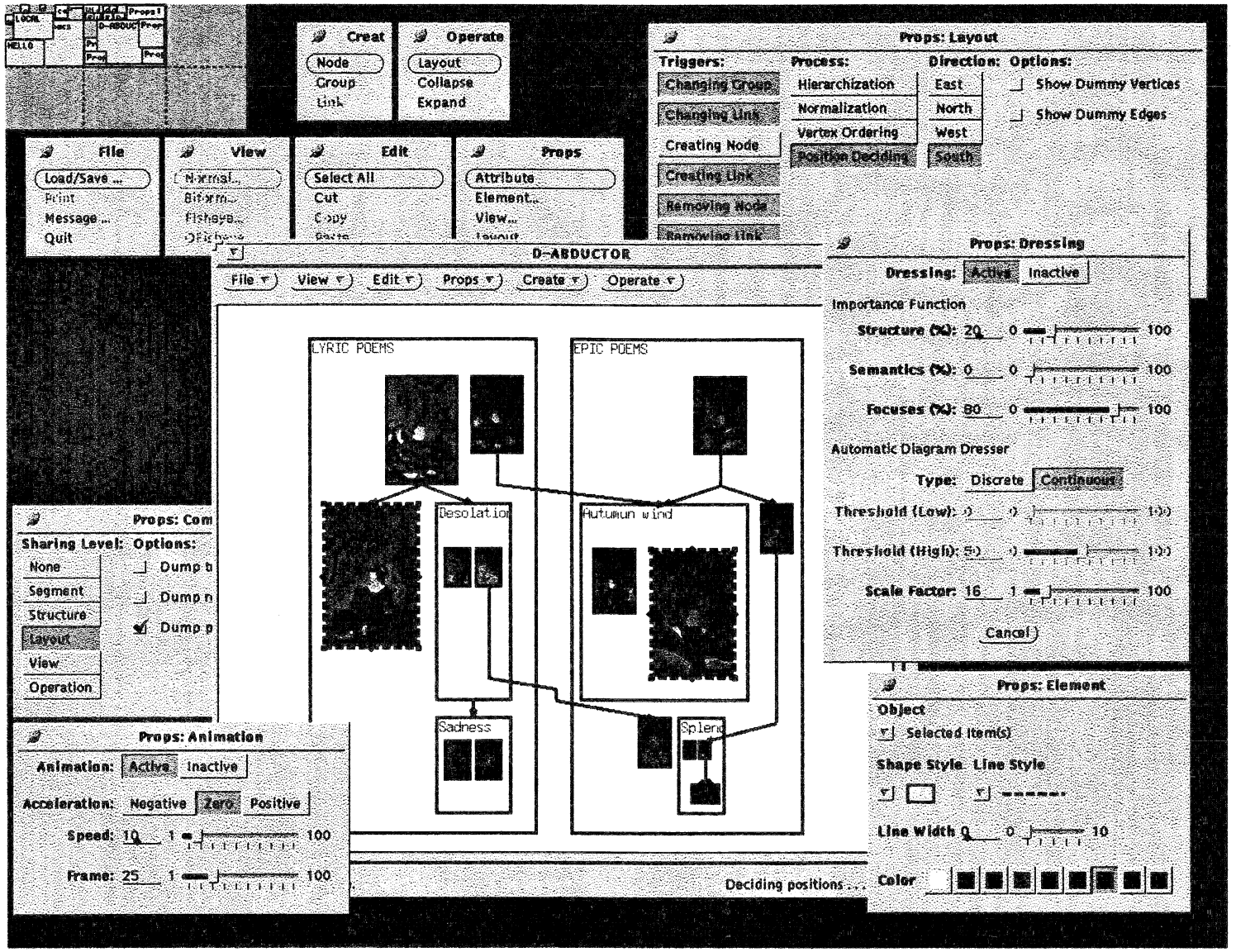

Figure 2: A workstation screen with D-ABDUCTOR

who are working with D-ABDUCTOR on different workstations can share diagrams in real time for cooperative work. D-ABDUCTOR operates in an environment in which users see the face and hear the voice of remote colleagues, using cameras and microphones. A snapshot of a co-operative work session is in Figure 2.

3. FACILITIES OF D-ABDUCTOR

In the previous section, several facilities of D-ABDUCTOR have been introduced in the context of a certain task. Now we explain each of the facilities in more detail.

The facilities are classified according to their purpose, to make it easier to understand why D-ABDUCTOR provides them. There are four classes:

(1) Facilities to simulate manual operations of the KJ-method (shown in Table 1) on computers.
(2) Facilities to relieve users of tedious tasks that are not essential in thinking processes.
(3) Facilities to overcome disadvantages of using a computer as a working environment for the KJ-method.
(4) Facilities to exploit advanced capabilities of computers to find new styles of supporting thinking processes.

3.1. Simulate Manual Operations

Interaction Mechanisms

D-ABDUCTOR provides simple and easy to understand operations such as direct manipulations by mouse, choosing facilities from menus, and interaction with the system through dialog boxes. Editing operations on diagrams such as moving nodes, creating edges, changing members of group nodes (nodes which include other nodes) are purely mouse driven.

Visual Feedback

D-ABDUCTOR provides a great deal of visual feedback for its operations. Modification to a diagram cause immediate visual feedback. The objects (nodes and edges) of an operation are visual indicated by handles which dynamically appear and disappear on the objects. For example, a candidate for a new group node is indicated by eight handles appearing as soon as the cursor enters the new group when nodes are being moved from one group to another.

3.2. Substitute for the Users

Automatic Layout Facility

Diagrams used in the KJ-method need good layout to be effective - a good layout can

enhance the thinking process, a poor layout can confuse and inhibit idea organize.

Manually re-considering a layout and re-drawing diagrams are both time-consuming chores. To release users from these kinds of chore, D-ABDUCTOR provides an automatic diagram layout facility based on a compound-graph drawing algorithm [6], which draws compound graphs as readable and aesthetic pleasing diagrams. This facility is always available as a menu item. Furthermore, the facility is the foundation of several other facilities, such as the automatic incremental layout facility and the diagrammatic dressing facility mentioned below.

The automatic layout facility is fast enough. For instance, it took about 0.1 seconds to compute the layout of the diagram shown in Figure 4(f) on a SPARCstation 2. The speed is important for the following reason. When we have two or more proposals to organize ideas, and when we have opposing suggestions in group work, it is helpful to make tentative charts to consider each proposal and each opinion. The speed of this facility means that tentative modification of diagrams is very easy and can be considered quickly.

Automatic Incremental Layout

If we handle diagrams manually, we need many extra operations even for a simple purpose. For example, putting some nodes together needs some space to be opened. Creating a new edge needs some nodes avoid its route.

For the proper purpose of the users, the computers should substitute the users as performing extra operations, which are regarded as unessential in thinking processes. With D-ABDUCTOR users can exploit the automatic incremental layout that is some editing operations trigger the automatic layout facility immediately. D-ABDUCTOR allows users to set some of the editing operations as triggers.

3.3. Overcome Disadvantages of Computers

Diagrammatic Dressing

Ordinary workstation screens are considerably smaller than desks or tables, on which we usually perform the KJ-method. A scrolling facility is the major existing technique to cope with this disadvantage. However if a scrolling facility, we loses the critical aspect of diagrams: we can grasp whole structures at a glance.

D-ABDUCTOR dresses diagrams by adjusting visual attributes such as visibility and size of each element according to its importance. The importance of each element is calculated by using logical structure of the diagram, semantics of elements and focuses of the users[9]. Users activate "diagrammatic dressing" to obtain various appearances of a diagram according to their viewpoint of the diagram. The total diagram is displayed, while highly important nodes are magnified and detailed and less important nodes are demagnified and abridged. Diagrammatic

dressing is useful in tackling the disadvantage of small screens. D-ABDUCTOR can effectively display large diagrams even on small screens. Examples using up to 377 idea fragments have been organized with D-ABDUCTOR.

Animation Facility

The change in appearance caused by the automatic layout facilities can be instantaneous and drastic. Thus the change of view might destroy the users "*mental maps*"[15] of the diagrams. D-ABDUCTOR provides animation facilities to smoothen the visual changes. Using animation has proven to be an effective way to prevent the destruction of mental maps. The animation facility has been particularly helpful in grasping changes in diagrams, and users seldom lose track of the nodes.

3.4. Exploit Advanced Computer Capabilities

Communication with Other Workstations

Providing facilities to share information among two or more workstations enables cooperative idea organization activity by groups even if they are separated in remote sites. D-ABDUCTOR can send statements describing the users activities in the language Simple[2] to other processes of D-ABDUCTOR on other workstations connected in a network. The D-ABDUCTOR processes that have received the statements immediately interpret and execute them. With this facility two or more users of different workstations can share operations on the same diagrams.

Communication with Other Tools

D-ABDUCTOR also fills the role of supporting "convergent" thinking processes as a sub-system of a group idea processing system *GrIPS*[16], which aims to totally support idea processing activity by groups. For the sake of integration with GrIPS, D-ABDUCTOR provides the facility to receive information from other sub-systems by an easy direct manipulation called "drag-and-drop" method.

We have also developed a card database *Card Base*[2], which works with D-ABDUCTOR. This is a kind of database management system which allows us to retrieve text and pictures from logical expression of keywords, and sends statements in the language Simple to a D-ABDUCTOR process to create nodes with the retrieved data as labels.

Multi Media Facilities

Humans use various media for thought and thus D-ABDUCTOR is expected to provide various kinds of media. By exploiting multi-media technologies, not only text but pictures are available as labels. Moreover, movies and sounds, which are not available on paper cards, will

be also available. As a primitive multi-media facility, we have developed Card Base, which can deal with pictures, and thus D-ABDUCTOR can display not only text but also pictures in nodes.

It is technically possible to use movies and sounds too. In our application, that is, diagram based thinking support, however, some problems need to be settled to combine these media with diagrams effectively. Conflicts occur between the critical advantage of diagrams, of which we can view totally of them at once, and the properties of movies and sounds which take time to see or hear.

4. HOW TO USE D-ABDUCTOR

This section explains the use of D-ABDUCTOR through an example. The task was to plan the future directions of D-ABDUCTOR and put our ideas in shape by the KJ-method. In the description of each step of the task, "U:" is prefixed to the operations performed by the users and "C:" is prefixed to the operations done by the computer (i.e., D-ABDUCTOR). The result of this task is explained in section 6.

Note that D-ABDUCTOR does not control the organization of ideas, but merely assists. The example below shows only one use of D-ABDUCTOR to accomplish the specified task.

4.1. Label Making

In order to plan the future directions of D-ABDUCTOR, we first made the primary list of problems from our own notes, the comments of colleagues on its demonstrations, and the request of its users. Then we distributed the list to the persons concerned by e-mail to get them to add items or to modify them. As a result, we obtained 76 problems. Next we omitted the problems clearly derived from bugs in the development period and chose the problems related to handling of diagrams (to make example compact). At that time the list was a text file of 39 problems (see Figure 3; this is a translation from the original Japanese).

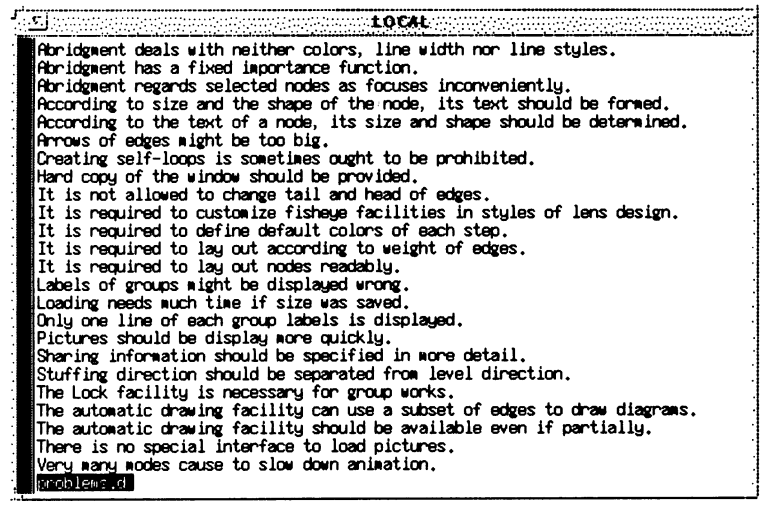

Figure 3: A list of problems (a portion)

Here we began to use D-ABDUCTOR. We started D-ABDUCTOR, then created nodes each with problems.

U: Choose Load/Save from the File menu to open the Load/Save dialog box, then through it issue a

command to load the file.

C: Load the specified file, make a node for each line of the file, and then initially lay them out on a grid (see Figure 4(a)).

Also, we could have created a node and input it as text or a picture whenever an idea fragment comes to mind. We could have also add ideas represented by pictures.

4.2. Group Organization

The group organization step consists of the following three sub-steps, and these sub-steps are usually repeated until there are only a small number of groups. In our task of planning D-ABDUCTOR, we repeated these sub-steps three times to make four groups.

(1) Label Spreading (1st)

We do not need to perform this step for the initial iteration because nodes were initially laid out on a grid by D-ABDUCTOR.

(2) Label Grouping (1st)

We move nodes with related ideas together to make groups.

U: Peruse all nodes and find some related nodes, and then put the related nodes together by moving them (by dragging the mouse) (see Figure 4(b)).

(3) Title Making (1st)

U: Select nodes, and then choose **Group** from the Create menu.

C: Create a group node as a rectangle includ-

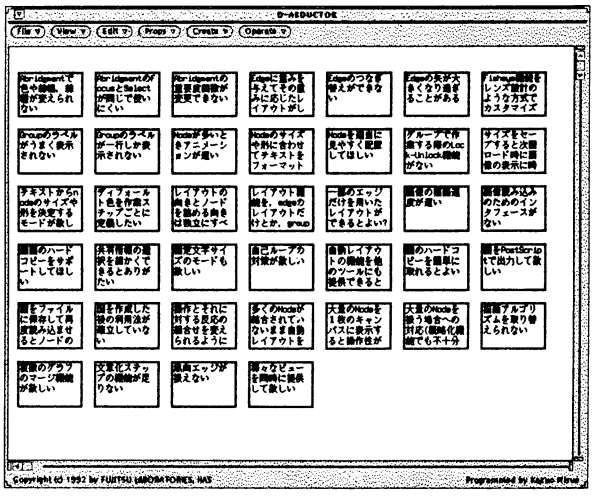

Figure 4(a): A file were loaded. (1st Label Spreading)

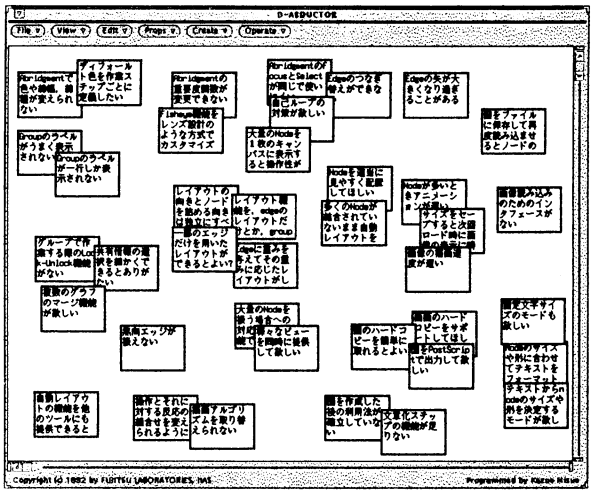

Figure 4(b): Related nodes were put together. (1st Label Grouping)

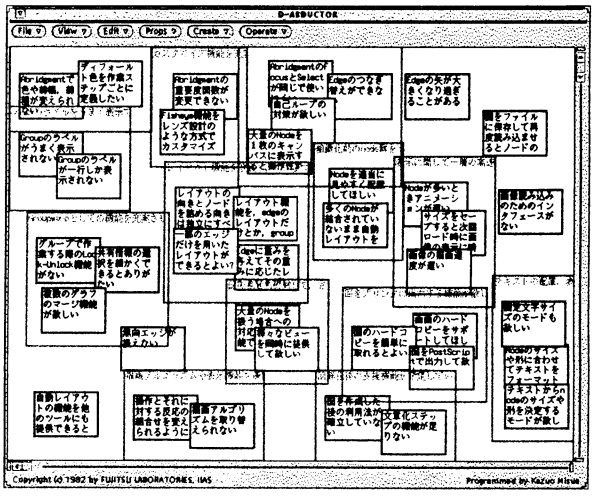

Figure 4(c): Title was given for each group. (1st Title Making)

ing the all selected nodes, and then open a text editor on the group node.

U: Think of a suitable title representing the contents of the group, input the title, and then close the text editor.

C: Display the text (i.e., the group title) in the group node (see Figure 4(c)).

(4) Label Spreading (2nd)

Before we perform the next iteration of the three sub-steps, we collapsed each group into a node.

U: Select all newly created group nodes, then choose **Collapse** from the Operate menu.

C: Hide the all nodes included by the group nodes, and then reduce the group nodes to normal size (using animation).

We spread nodes so that all of them can be read.

U: Move nodes (by dragging the mouse) so that the text and pictures of all nodes are visible (see figure 4(d)).

(5) Label Grouping (2nd)

Repeat in the same way as the initial iteration (see Figure 4(e)).

(6) Title Making (2nd)

Repeat in the same way as the initial iteration (see Figure 4(f)).

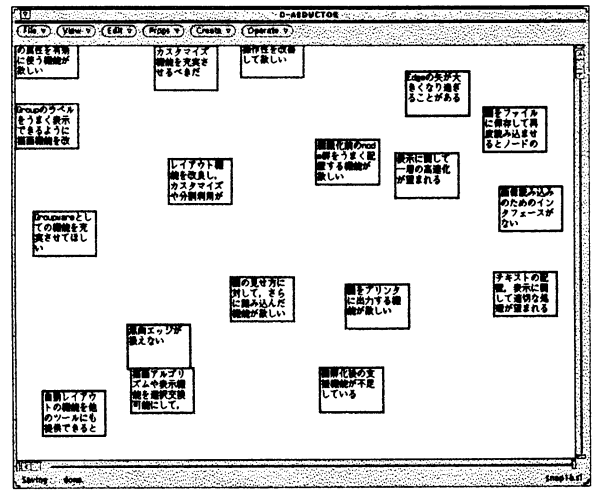

Figure 4(d): Every group were collapsed, and nodes were spread. (2nd Label Spreading)

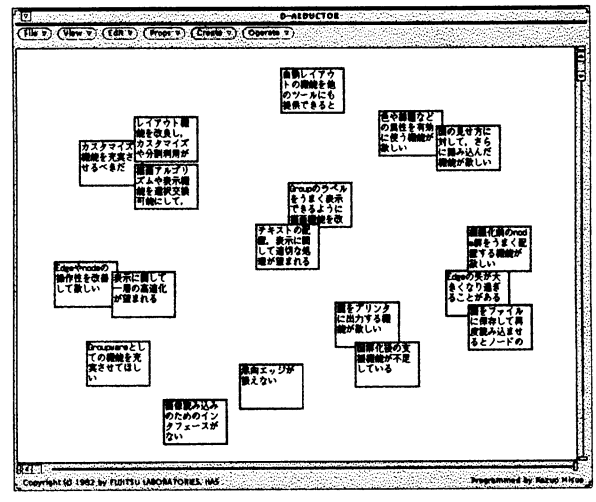

Figure 4(e): Related nodes were put together. (2nd Label Grouping)

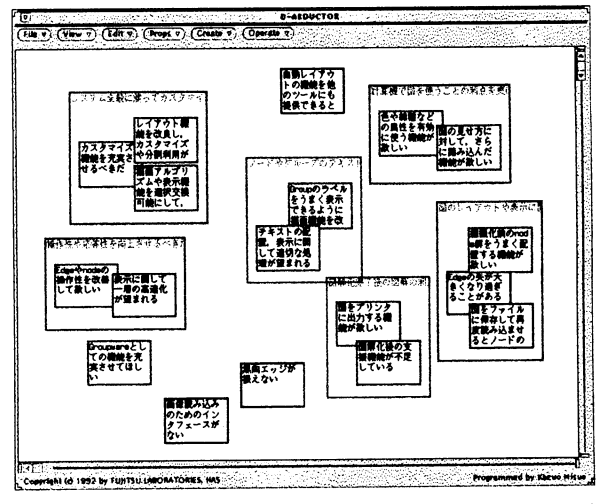

Figure 4(f): Title was given for each group. (2nd Title Making)

(7) Label Spreading (3rd)

Repeat in the same way as the initial iteration (see Figure 4(g)).

(8) Label Grouping (3rd)

Repeat in the same way as the initial iteration (see Figure 4(h)).

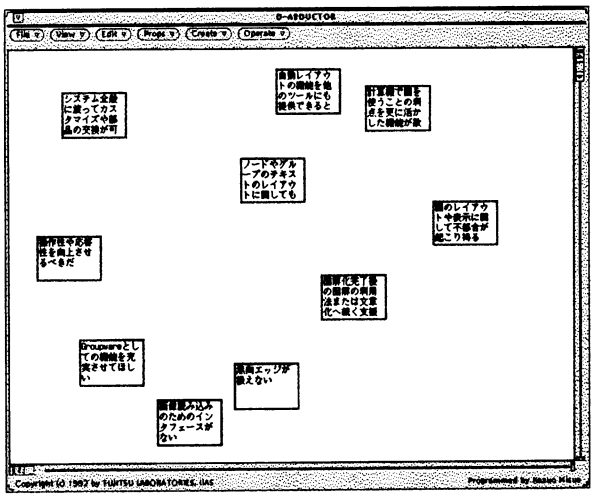

Figure 4(g): Every group were collapsed, and nodes were spread. (3rd Label Spreading)

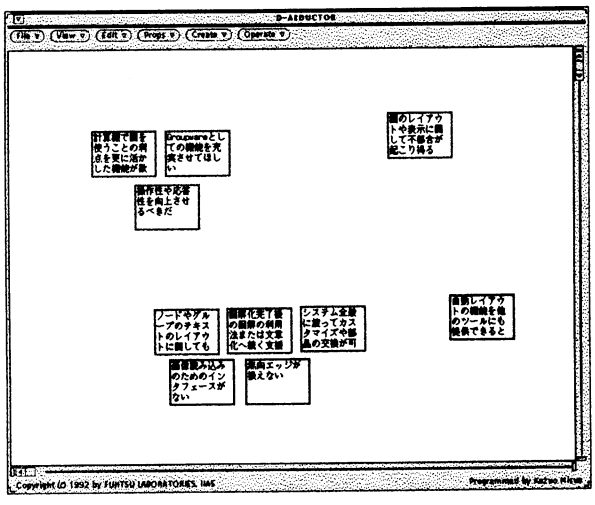

Figure 4(h): Related nodes were put together. (3rd Label Grouping)

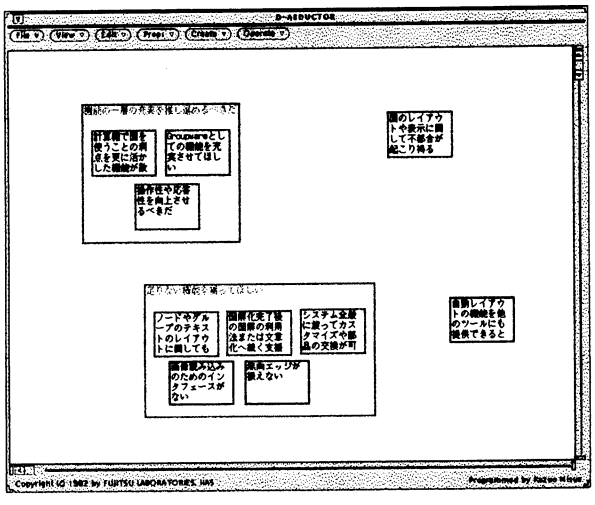

Figure 4(i): Title was given for each group. (3rd Title Making)

(5) Title Making (3rd)

Repeat in the same way as the initial iteration (see Figure 4(i)).

We collapsed each group into a node to finish the group organization step.

U: Select all group nodes, then choose **Collapse** from the Operate menu.

C: Hide the all nodes included by the group nodes, and then reduce the group nodes to normal size (see figure 4(j)).

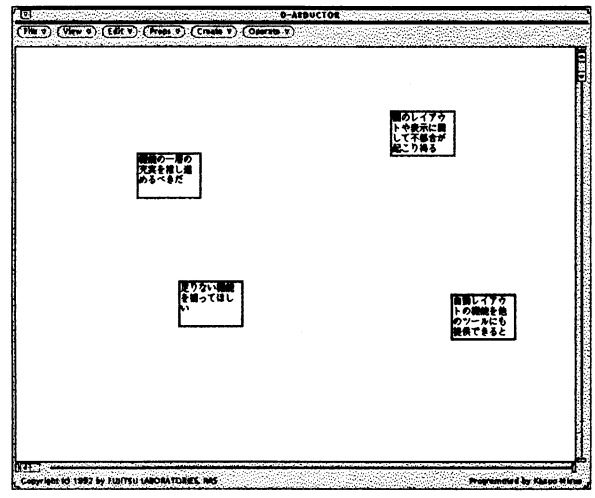

Figure 4(j): Every group were collapsed.

4.3. Chart Making

After the group organization step, we moved on to the chart making step. We created edges between two related nodes (or group nodes), while expanding group nodes.

We first laid out nodes manually.

U: Move nodes (by dragging the mouse) (see figure 4(k)).

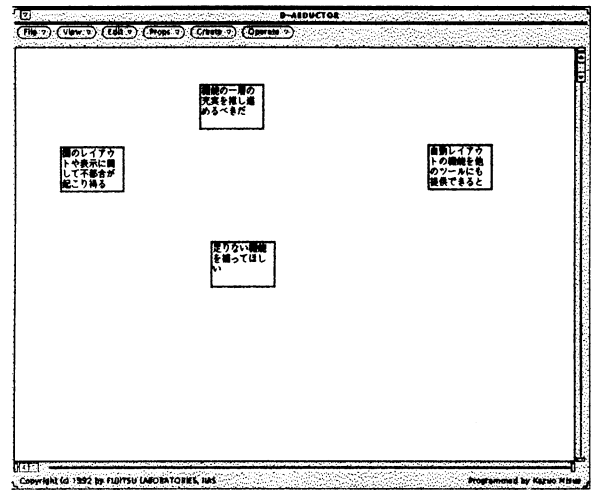

Figure 4(k): Nodes were laid out manually.

We expanded group nodes.

U: Select collapsed group nodes, then choose **Expand** from the Operate menu.

C: Restore the selected group nodes to their original size (using animation) and show the all nodes included by the group nodes with original layout.

We also created edges between nodes (or group nodes).

U: Find a relationship between two nodes or group nodes, then select one of them.

C: Show eight handles on the selected node.

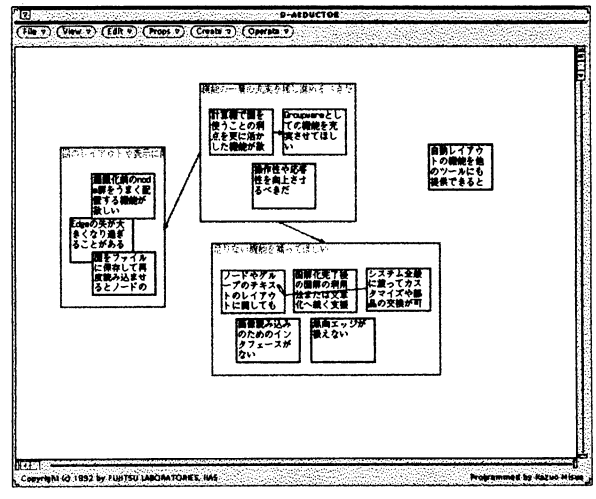

Figure 4(l): Groups were expanded while edges were created.

U: Drag the mouse from a handle to the other node.

C: Create a new edge between these two nodes (see Figure 4(l)).

At this time we had the diagram re-drawn by the automatic drawing facility.

U: Feel that the diagram is difficult to understand, and thus choose **Layout** from the Operate menu.

C: Re-draw the diagram using automatic layout and animation (see Figure 4(m)).

After that we activated automatic incremental layout.

U: Choose **Layout** from the Props menu to open the Layout dialog box, and then through it select editing operations to be triggers of the automatic drawing facility.

We continued to create edges while expanding group nodes. Each editing operation caused re-drawing of the diagram.

U: Expand a group node.

C: Re-draw the diagram.

U: Create an edge.

C: Re-draw the diagram.

After a while, the diagram became larger than the window, and was automatically reduced to fit the window (see figure 4(n) and 4(o)). Text was also reduced, using gray scale anti-aliasing. When the diagram became even larger, characters eventually

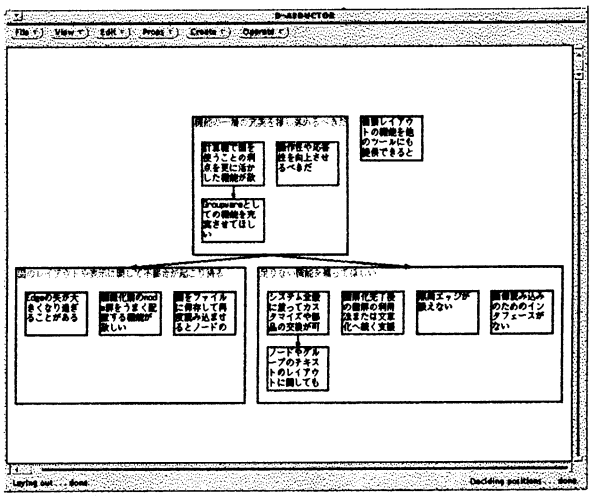

Figure 4(m): The diagram was re-drawn automatically.

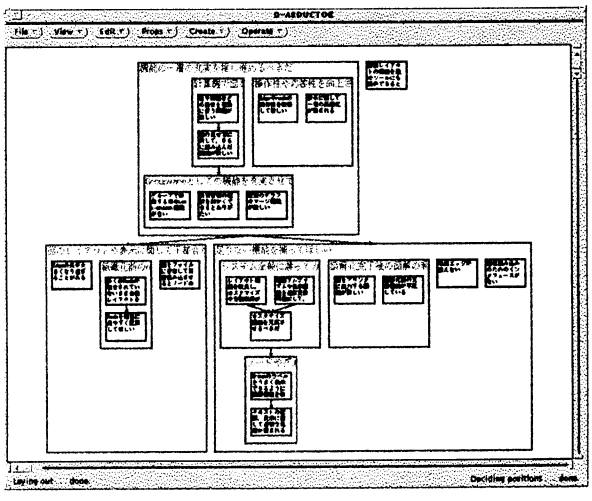

Figure 4(n): Groups were expanded while edges were created (automatically re-drawn and reduced).

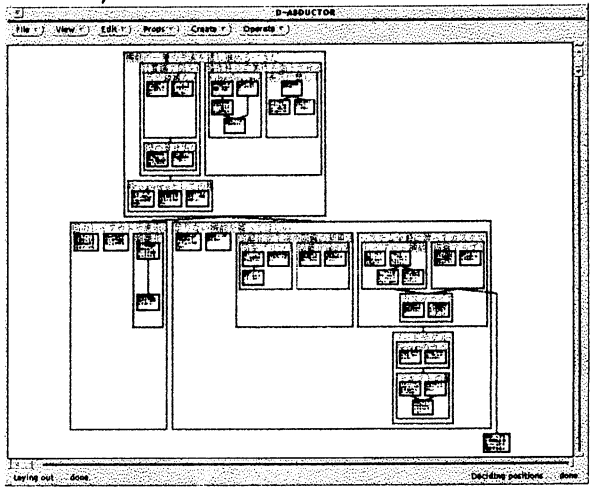

Figure 4(o): Groups were expanded while edges were created (automatically re-drawn and further reduced).

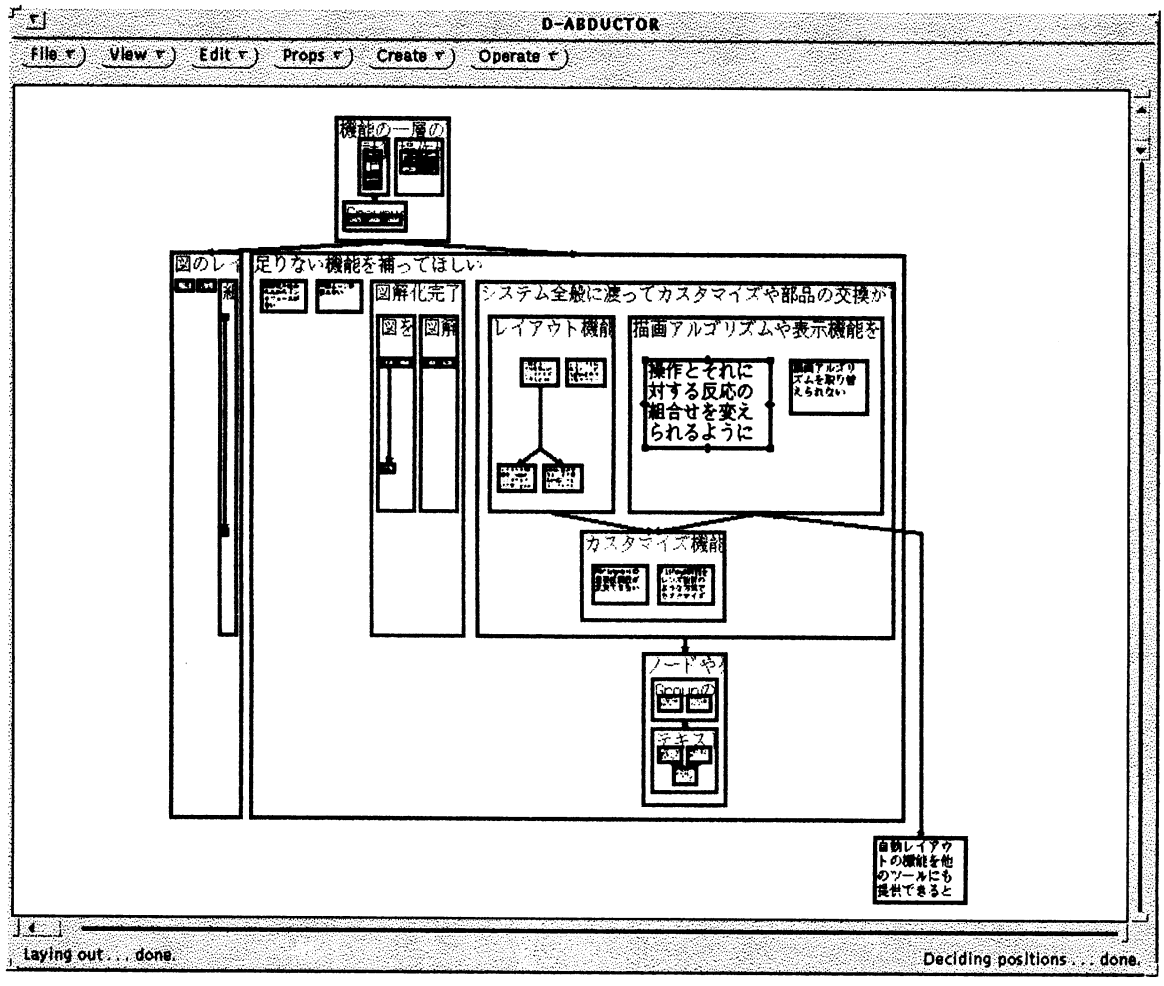

Figure 4(p): Chart making was finished (displayed by using diagrammatic dressing).

became illegible. Thus we exploited the *diagrammatic dressing* facility.

U: Choose **Dressing** from the Props menu to open the Dressing dialog box, and then through it, make the diagrammatic dressing facility active and set the parameters for abridged representation.

U: Select nodes to be displayed larger and in detail.

C: While magnifying the focus nodes (i.e., selected nodes) so that their text is readable, demagnify the remainder so that the whole diagram fits the window (see Figure 4(p)).

We obtained the diagram, which had 60 nodes including the node representing groups, shown in Figure 4(p).

4.4. Description

We made a text file from the finished diagram.

U: Choose **Load/Save** from the File menu to open the Load/ Save dialog box, and then through it issue a command to save the diagram in outline format.

C: Save the text of all nodes into a file as outline format.

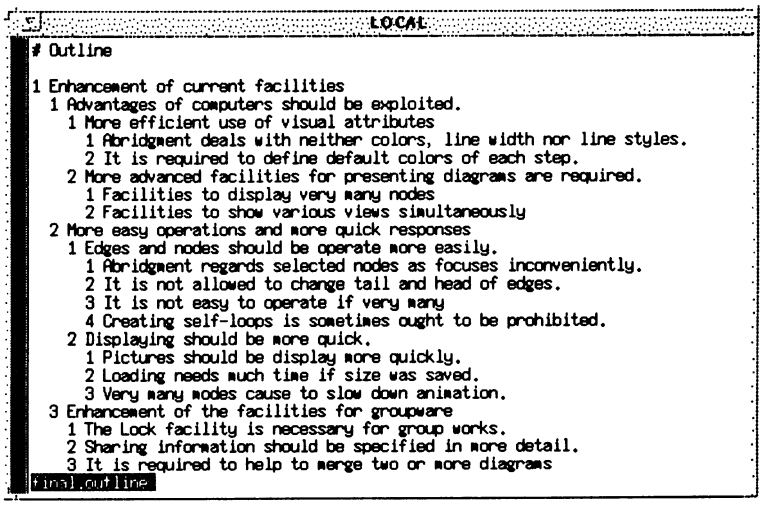

Figure 5: Outline text generated from the final diagram (a portion)

We obtained the text shown in Figure 5. Each item in the text has a nesting number that reflects layout of nodes. We can consider these numbers as numbers of chapters, sections, subsections, and so on. After that we transformed the file with an ordinary word processor to section 6 of this paper.

5. REPORTS OF TRIAL USE

We are distributing D-ABDUCTOR to different departments in FUJITSU LIMITED and universities according to requests, and collecting reports to evaluate the system. This section presents an early report form a staff of a department.

5.1. Practical Tasks

She used D-ABDUCTOR for three kinds of practical tasks. The followings explain these tasks.

(1) Consideration of Document Structures

She was at the final phase of an on-the-job-training and writing a final report for it. The first task was to determine the main theme of the report and think the document structure of it.

With D-ABDUCTOR, first she picked up candidates of topics, and created a node for each candidate while sorting out the candidates and constructing a structure by making groups and edges. Then she obtained outline structure of the report represented by a diagram. Finally based on the diagram, she accomplished writing a report and made materials (i.e., slides) for reviewing presentation.

(2) Design of Program Structures

She designed a software system. The second task was to find a good diagrammatic representation of relations among modules.

She picked up callback functions and low-level modules, and created a node for each of

them while expressing relations among them by edges. Then she modify the diagram representing the structure of modules while finding a readable diagrammatic representation. In this task D-ABDUCTOR helped her draw many tentative diagrams.

(3) Design of Database Schema

She considered to manage customer information of products by using a database. The third task was to design a schema of the customer database.

She picked up all data should be included a record. With D-ABDUCTOR, she created a node for each data to express the schema diagrammatically. She modified a diagram representing the schema while finding the most efficient schema.

5.2. Results and Comments

She described her impression of working with D-ABDUCTOR and comments as follows.

I believe that D-ABDUCTOR is useful to think structures of something. In terms of time efficiency, I should mention that by using D-ABDUCTOR I could raise working efficiency, especially for the first task. I also wrote this kind of report last year. In this time, however, I could obtain an outline structure of the report in about a half time of last year. For the second and the third tasks, D-ABDUCTOR was useful to be able to represent structures as diagrams and to observe the diagrammatic structures from the new points of view by changing layouts of the diagrams. When I consider this kind of problem by myself, it is hard to change viewpoints. With D-ABDUCTOR, however, I could easily change my viewpoints.

I think D-ABDUCTOR is efficient in the following two points. The first point is that it is easy to organize data tentatively because the unit of operations is a node. I am not good at thinking logically in beginning stage of this kind of design work. Thus the working style in which first I list all related data, then I observe these data while thinking is very efficient for me. Moreover, it is useful for proceeding thought to be able to confirm moving nodes with animation. The second point is that the automatic drawing facility of D-ABDUCTOR gave me good stimuli for creating ideas. I often reach a deadlock when I perform these kinds of tasks mentioned above alone, for my thought easily gets fixed. At that time, I have asked advice of other persons, for instance, my boss so far. However, D-ABDUCTOR sometimes leads me to unexpected good directions by changing layout of diagrams, and it was effective for my works.

Though above reports are not many enough at all to evaluate D-ABDUCTOR, we may summarize the reports as follows: with D-ABDUCTOR, (1) they could accomplish above three kinds of tasks, (2) they could raise working efficiency for certain tasks, and (3) changing layout

of diagrams may give them good stimuli for thought.

6. CONCLUDING REMARKS

A diagram based idea organizer D-ABDUCTOR, which supports human thinking processes by exploiting diagrams as media of interaction, was explained.

For the purpose of supporting diagrammatic thinking processes, we have also developed several elementary techniques besides ones mentioned earlier, for example, translating area diagrams into net diagrams, and vice versa[17], automatic drawing of graphs which have both directed edges and undirected edges[18].

The task shown in Section 3 classified future directions into four classes: (1) enhancement of current facilities, (2) supplement of shortage facilities, (3) settlement of odd behavior, and (4) separation of certain facilities. Remaining of this section describes them in more detail.

6.1. Enhancement of Current Facilities

(1a) More efficient use of visual attributes

The users can manually change visual attributes (colors, line width, line styles, and so on) of each of nodes through the Element dialog box according to their own rules. It is expected to use visual attributes more efficiently, for example, to assign different default attributes depend on a step of procedures, and to deal with them by diagrammatic dressing.

(1b) Separation focus from selection

If diagrammatic dressing is active, view of the diagrams changes whenever we change the selected elements to operate them since the selected nodes are also used by diagrammatic dressing as the focuses. The focuses should be separated from the selected nodes to increase operational efficiency.

(1c) Supporting new style of group works

Several levels of sharing information are prepared according to various styles of group works[10]. We expect that to make the level be changed dynamically conduct a new interesting style of group works as follows: first we start the lowest level, at which diagrams of all users are different to each other, then resolve the conflict among different diagrams while making the level higher and higher, and finally we obtain one shared diagram.

6.2. Supplement of Shortage Facilities

(2a) Drawing facilities to deal with larger class of compound graphs

In spite of that the diagrams used in the KJ-method has various edges including both

directed edges and undirected edges, the implemented automatic drawing facility deal with only compound directed graphs (i.e., adjacency edges are directed). We should implement more powerful automatic drawing facility can deal with compound graphs which have more various types of edges.

(2b) Facilities to support the description step of the KJ-method

D-ABDUCTOR supports just until making outline text. It does not seem to support the description step enough. We should develop some other facilities such as supporting presentations using the diagrams and a synchronized editing of outline text and the diagram (whenever we modify one of the text and the diagram, the other is immediately updated to keep consistency).

(2c) Customizable system configuration

Although some of individual facilities are each customizable, the whole system is not customizable. We should provide facilities to customize the system configuration such as replace the automatic drawing facility (algorithm) with another one.

6.3. Settlement of Odd Behavior

(3a) Saving geometric information

Geometric information such as positions and size of elements are not saved into files. First we estimated that this kind of information was not necessary to be saved because D-ABDUCTOR had an automatic drawing facility. When D-ABDUCTOR loads the files again, however, it sometimes puts nodes in reverse order, and throws the users into confusion. Geometric information should be savable as users hope.

(3b) Automatic layout for the group organization step of the KJ-method

The implemented drawing algorithm follows the convention of laying out nodes hierarchically. The automatic drawing facility based on the algorithm is useless at the initial stage of the group organization step of the KJ-method because it lays out all nodes on a horizontal or vertical line, for there are no edges (i.e., no hierarchy) then. Other algorithms adequate for this group organization step are required.

6.4. Separation of Certain Facilities

(4a) Separation of the automatic drawing facility

The automatic drawing facility of compound graphs is applicable to not only supporting the KJ-method but also other many applications in various fields. It is useful to separate these

kinds of facilities from the program of D-ABDUCTOR and to offer them as server programs.

ACKNOWLEDGMENT

The authors thank the GrIPS[16] development members for their useful comments about the implementation of D-ABDUCTOR, Mr. Kiyoshi Nitta for his cooperation in the works shown in Section 3, and Ms. Hiromi Ohkawara for her report on trail use. They are also grateful to Prof. Peter D. Eades of the University of Newcastle for suggestions to improve the paper.

REFERENCES

[1] Misue, K. and Sugiyama, K., An Overview of Diagram Based Idea Organizer: D-ABDUCTOR, Research Report IIAS-RR-93-3E, FUJITSU LABORATORIES, IIAS, 1993.

[2] Misue, K., D-ABDUCTOR 2.0 User Manual, Research Report IIAS-RR-93-9E, FUJITSU LABORATORIES, IIAS, 1993.

[3] Kawakita, J., *The KJ Method - A Scientific Approach to Problem Solving*, Kawakita Res. Inst., Tokyo, 1975.

[4] Kawakita, J., *The KJ Method*, Chuo-Koron-Sha, Tokyo, 1986 (in Japanese).

[5] Sugiyama, K. and Misue, K., Good Graphic Interfaces for Good Idea Organizers, Proc. of INTERACT '90, pp. 521-526 (1990).

[6] Sugiyama, K. and Misue, K., Visualization of Structural Information: Automatic Drawing of Compound Digraphs, IE*EE Trans. Syst., Man, Cybern.*, Vol. 21, No. 4, pp. 876-892, 1991.

[7] Misue, K. and Sugiyama, K., Freehand-like drawing of compound graphs: A fundamental technique for computer aided abduction, in *Proc. 37th Conf. Info. Proc. Soc. Japan*, pp. 1304-1305, 1988 (in Japanese).

[8] Misue, K. and Sugiyama, K., Multi-viewpoint Perspective Display Methods: Formulation and Application to Compound Graphs, in P*roc. of HCI '91*, pp. 834-838, 1991.

[9] Misue, K., On Abridgment of Figures and its Application to Abduction Support, Inform. Processing Soc. Japan, Research Report 90-HI-31-1, 1990 (in Japanese).

[10] Misue, K., On Adding Groupware Facilities to D-ABDUCTOR, Inform. Processing Soc. Japan, Research Report 93-GW-2-3, 1993 (in Japanese).

[11] Sugiyama, K., R & D Trends of Tools for Convergent Thinking: Computer Support Based on KJ Method, *Journal of Japanese Society for Artificial Intelligence*, Vol. 8, No. 5, pp. 32-38, 1993 (in Japanese).

[12] Ohiwa, H., Kawai, K. and Koyama, M., Idea Processor and the KJ Method, *Journal of Information Processing*, Vol. 13, No. 1, pp. 44-48, 1990.

[13] Shinohara, Y., Knowledge Formulation Support System CONSIST-II: Evaluation and Improvement of CONSIST, *Journal of Japanese Society for Artificial Intelligence*, Vol. 8, No. 5, pp. 57-64, 1993 (in Japanese).

[14] Sugiyama, K., Analysis of KJ Method from the Viewpoint of Diagrammatical Thinking Support, in *Proc. of 5th Symp. on Human Interface* (Kyoto, Oct, 1986), pp. 325-330, 1986 (in Japanese).

[15] Eades, P., Lai, W., Misue, K. and Sugiyama, K.: Preserving the Mental Map of a Diagram, in *Proc. of COMPUGRAPHICS '91*, pp. 34-43, 1991.

[16] Kohda, Y., Watanabe, I., Misue, K., Hiraiwa, S. and Masui, M., Group Idea Processing System: GrIPS, *Journal of Japanese Society for Artificial Intelligence*, Vol. 8, No. 5, pp. 65-74, 1993 (in Japanese).

[17] Misue, K., On Operation of Figures for Diagrammatical Abduction Support System: D-ABDUCTOR, in *Proc. 41st Conf. Inf. Proc. Soc. Japan*, 6G-4, pp. 237-238, 1990 (in Japanese).

[18] Misue, K., On a drawing of graphs which have directed edges and undirected edges, in *Proc. 39th Conf. Inf. Proc. Soc. Japan*, pp. 81-82, 1989 (in Japanese).