Preliminary Experiments of A Computer System for Face-to-face Meetings

Haruhiko Kaiya *, Nobuyuki Miura †,
Go Anai ‡, Tsuyoshi Ebata‡, Hiroki Nagaoka‡, Motoshi Saeki §

E-mail: kaiya@jaist.ac.jp
URL: http://www.jaist.ac.jp/~kaiya/

abstract

We usually have several number of face-to-face meetings to construct a product, such as software specification document. And such product should reflect the contents of the meetings but normally does not. We consider that one of the aims of groupware is making the product reflect the contents. In this paper, we introduce preliminary experiments of a computer system for face-to-face meetings from this point of view. From the analysis of these experiments, in the meetings, workers should use the groupware for recording the contents of discussions and for referring the histories and/or minutes of the previous meetings. And they should use the groupware for editing the histories and for describing the products outside the meetings.

keywords

Cooperative Work, Groupware, HyperText, Human Interface, Software Engineering, Requirements Engineering.

1 Introduction

In recent years, computer systems for supporting group works, namely *groupware*, are developed. For example, many electronic meeting systems [1] are developed, that are tailored along with the variety of meeting environments, styles, objectives and so on. And, cooperative workers, who are geographically distributed, can be seamlessly work together by using computer communication systems such as TeamWorkstation[2] and Mermaid[3]. And also in the face-to-face communication, a computer system with large shared display would provide WYSIWIG features for the workers, hence cooperative workers can easily share their products with each other [4]. Groupware also supports the logical or conceptual aspects of groups. For example, tools[5], [6] for recording and arranging the argument structure, are based on IBIS. And KJ-method[7] is also used for groupware in idea generation [8]. And the model of human activities, especially conversation, is also utilized for supporting the activities [9], [10], [11].

In this paper, we focus on a task for developing a shared product, such as documents or figures, by the several kinds of workers, each of which specialties is different. Requirements elicitation during a software development process, is a typical example of such task. And we also assume that the task is accomplished in the several times of face-to-face meetings. By limiting the domain of tasks, we discuss how computer system can or should support such kind of task.

We consider that this kind of task has the following features;

- Though almost all the communication among the workers is achieved by the verbal conversation, the contents is volatile, and sometimes including topics which are not related to the task.

*Graduate School of Information Science, Japan Advanced Institute of Science and Technology, Hokuriku, Ishikawa, 923-12 Japan
†Department of Electrical and Electronic Engineering, Tokyo Institute of Technology, Tokyo, 152 Japan
‡Department of Computer Science, Tokyo Institute of Technology, Tokyo, 152 Japan
§Graduate School of Information Science and Engineering, Tokyo Institute of Technology, Tokyo, 152 Japan

- Normally, before each meeting, the agenda is prepared, and after each meeting, the minutes would be described. So the contents of a meeting is closely related to the tasks before and after the meeting.
- Of course, the documents, e.g. agenda, minutes and final products, should reflect the contents of discussions in the meetings.

From these features above, we assume that the computer system would provide the following functions for supporting the task.

- The function for recording the contents of discussions, e.g. volatile information in the verbal conversation.
- The function for arranging the records of discussions, providing the tags or annotations for each record and the structure of the record.
- The function for consulting or referring the contents of discussions easily. After the meeting, a scribe would easily describe the minutes, which completely reflects the contents of discussions.

For confirming the usefulness of these functions, we designed three types of experimental meetings, and observed the verbal and action protocol of subjects in the meetings. In each type of experiment, different kind of functions was provided for subjects. And by analyzing the protocols and products in each type of experiment, we discuss the advantages and disadvantages of these functions. In these experiments, we used the groupware for describing the minutes[12].

In the next section, we precisely introduce the objective and the method for our experiments. The computer system for our experiments is introduced in Section3. And in Section 4, we report and discuss the results of our experiments.

2 Method for Experiments

The main objective of our experiments, is to confirm what kind of support by computer systems is helpful to cooperative workers in the face-to-face meetings. In this paper, we held three experiments, where different kind of computer-support could be available, and compared the subjects' behaviors in each experiment. In each experiment, tasks with the following features would performed by the subjects, which were typical ones of face-to-face meetings.

- The objective of the task is to capture the requirements for a software system from customer(s). The requirements specification is written by the natural language.
- The following kinds of workers are participated in the task.
  - customer: Having the requirements for his/her application domain. Giving an approval of the final product, i.e. requirements specification.
  - software designer and analyst: Capturing and analyzing the requirements from the customer(s). Describing the products.
  - scribe: Scribe in the meeting. In addition, an operator for groupwre if it is used in the meeting.
- All meetings are held in a meeting room with a round-table.
- The agenda and minutes are described for each meeting.

For this kind of tasks, the following kinds of supports by computer systems would be helpful.

- Recording the verbal data in the meetings automatically. Volatile information in the meetings could be persistent, hence scribe would not miss the conclusions or reasons discussed in the meetings.
- Making the workers tag or annotate the labels on the records above. It is very difficult to arrange the records of verbal data because the records have few tags which can be recognized by the computer system automatically. So computer system should let the workers tag the labels easily. If workers can easily tag the labels and refer them, they may arrange the records along with the progress of the meeting, moreover, they could describe the products during the meeting.

  Another kind of support for this problem, is to provide the recording schema, e.g. issue, position and argument in IBIS, for the workers. But we do not take this way because workers should largely change their own way of working, after all such changing would hinder their way.

With respect to the functions above, we decide the following functions and features are available in each experiment.

Experiment#1 : A task with no computer system. Subjects hold the meetings without any computer system. They may use chalkboard and paper-documents during the meetings. Agenda and minutes are also prepared as a paper document.

Experiment#2 : A task only with recording function during the meetings. The computer system is used like video and audio recorders in the meeting, so subjects do not have to change their own way of working. After the meeting, the scribe can describe the minutes by referring and arranging the complete record of the previous meetings. Agenda and minutes are also prepared as a paper document.

Experiment#3 : A task both with recording function and referring/editing function during the meeting. In addition to the functions provided in Experiment#2, the subjects may use the tagging, arranging and referring functions by the computer system during the meetings. And they may complete the minutes during the meeting if possible. Agenda and minutes are prepared as data in the system.

By observing the behaviors in the experiments and products, we analyze the advantages and disadvantages of the computer system with the following viewpoints.

- From the observation of Experiment#1, we shall identify the problems in a normal meeting, and discuss the possibility for computer support. We assumed the following problems.
  - Repeated discussion: We assume that a conclusion is caused by a discussion. So if a conclusion, which is already existing, is discussed during a discussion, or if some contents, which appeared in the previous discussion, appear again, we regard the spending time of the discussion wastefulness. We call such kind of discussion *repeated discussion*.
  - Discussion without conclusion: In the same way, if no product is produced during a discussion, we may regard it wasteful task. And we assume that the contents of such discussion tend to be repeated in another discussion later. Also if the decision of the conclusion is ambiguous, it causes the inconsistency among the products.
  - Missing the existence or a conclusion of a discussion from the product: If the output of discussion, e.g. its conclusion or its contents with design rationale, is not included in the product, the spending time for discussion seems to be wasteful, and inconsistency or wasteful repetition would be raised.
- We shall identify the system's contributions to describing the minutes or final specification after the meetings. From this analysis, we will confirm the usefulness of the system for constructing products which are completely reflected the contents of discussion in the meetings.
- We shall confirm the influence by the usage of the computer system during the meetings. Concretely, we observe how much did subjects refer or edit the minutes and agenda provided by the system during the meetings. From this analysis, we can identify the trade-off between the traditional way of work and the usage of the computer system.

The data of the computer system and the behavioral protocol of subjects recorded by the video and audio recorder, are used for these analysis.

3 An Overview of The System

In this section, we present a brief overview of a computer system which is applied to our experiments. This system is designed for a small group to develop documents e.g. minutes during face-to-face meetings. For example, this system would be helpful for analysts to elicit requirements for a software system from customers at the upper stream of a software process. This system has the following features;

- It totally supports the sequence of several meetings for develop shared products. It also provides the support to the users before and after a meeting as well as during the meeting.
- It stores utterances of participants and writings on the chalkboard during a meeting as the meeting history. And the meeting histories have several tags for identifying the topic and the break of discussion.

- It also stores products of meetings, e.g. minutes or final documents, and manages the relationships between products and meeting histories.
- For editing products, tagging keys on the meeting histories, linking relationships among them and referring them, it provides graphical user interfaces (GUI).

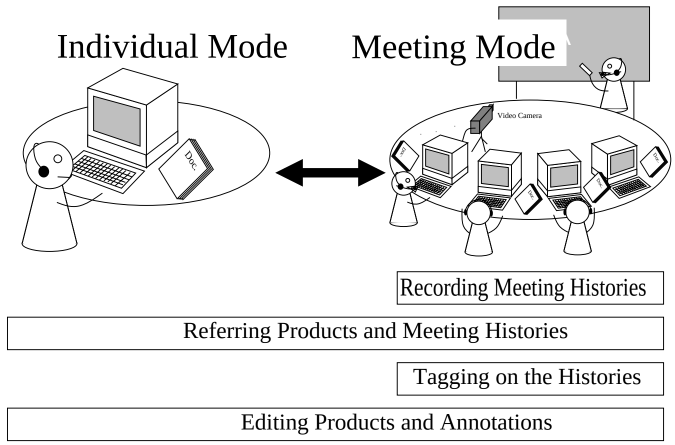

Figure 1: Typical Usages of the System

Figure1 shows the typical usages of the system. The system can be used both before/after meetings and during meetings so as to maintain the seamlessness among successive meetings. In meetings, participants have his own computer terminal and may refer the products and meeting histories. At the same time, the system records the utterances of each participant through the terminal. Users who can use a computer terminal as well as they talk, may edit the products or tag the annotations on the meeting histories, but other users do not need to touch the terminal, namely they may act as same as the meeting without computer system. Normally after a meeting, a scribe should write the minutes for the subsequent tasks. And before a meeting some of participants will write an agenda for the the meeting. The system lets the writers refer the meeting histories related to the current products easily, hence they can write the minutes or agenda completely.

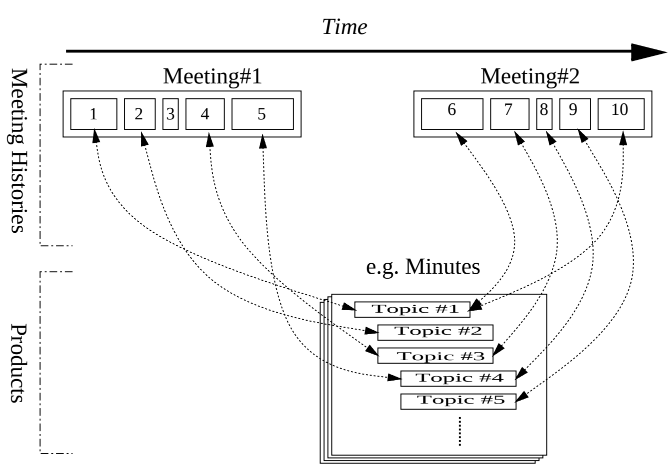

Figure 2: The Relationship between meeting histories and products

Figure2 shows how the system store meeting histories, products and their relationships. A product, e.g. minutes or agenda, is represented by a simple tree structure, each of which leaf has a short sentence. We call the sentence *Topic* in the system.

Each meeting consists of a sequence of *Actions*, which correspond to the utterance, writings on the chalkboard or explicit text annotation during the meetings. Each action at least has the following attributes, starting and ending time-stamps, actors and the raw contents of the action, e.g. computer recorded audio data or image data. Because the action is too small to treat the users, we introduce another large unit for actions, namely *Discussion*. A Discussion denotes an area where one or a few Topics were devotedly discussed. Though each Discussion should decide by the users, annotations tagging during the meetings would be helpful to decide it. In Figure2, Discussion are represented by small numbered rectangles in the Meeting#1 and #2.

The relationships between meeting histories and products are represented by the bi-directed arrows in Figure2. A relationship denotes the facts that the Topic was caused by the contents of the Discussion. For example in Figure2, Topic#1 is a part of the product because it was discussed in Discussion#1, #6 and #10 during the meetings. The relationship has an attribute for representing how the related Topic was discussed. Currently, the attribute value can be taken to whether "proposed", "decided" or "denied". This attribute is helpful for the users to identify the transition of Topic's treatment, e.g. they can easily identify the repeated discussion about some Topic. Also in this system, users can put annotation to the Discussion so as to easily browse the sequence of Discussions. We call such annotation as *Explanation* in this system.

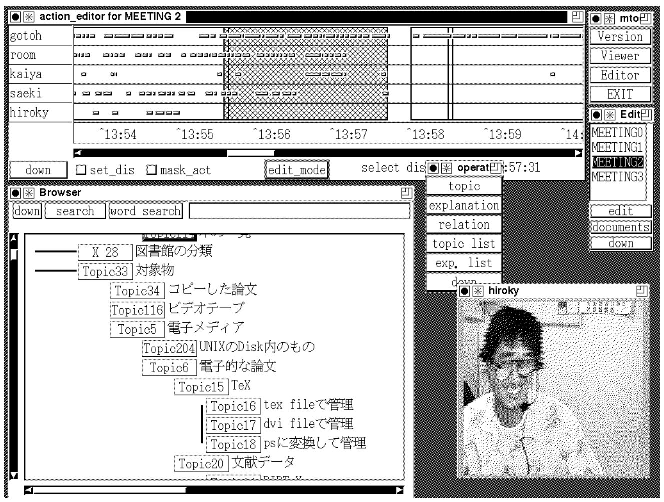

Figure 3: GUIs of the system

The users use the system through the GUIs shown in Figure3. The GUIs consists of several parts and users can select them along their usage, i.e. before a meeting, after a meeting or during a meeting. We will introduce the function of typical part of the GUIs.

Browser (left bottom in Figure3): GUI for representing the tree structure of the product. By clicking buttons labeled by "Topic", one can invoke the list of Discussions related to the Topic. And from the list, one can also invoke the contents of each Discussion which looks like *Action Editor* explaining below, hence one can hear or view the contents of each action again and again. This function helps the user to confirm the design rationale about the topic.

In addition, from the Topic button, one can invoke a list of other Topics, which would be influenced by the Topic. This list is automatically made by the ordering of Discussions in the meetings and the relationships between the Topics and Discussions[13].

Action Editor (top half in Figure3): GUI for deciding the range of each Discussion and for looking over a stream in a meeting. It is represented as a two-dimensional table, which row denotes the list of participants in the meeting, and which column denotes the passing time from left to right. By clicking small buttons in this GUI, one can hear the utterance of some participant or looking a writing on the chalkboard recorded in the system. The length of each button denotes the time length of utterance if it represents utterance data. One decides the range of a Discussion by hearing and looking the action records and by looking over the patterns of buttons, which reflects the turn-taking of participants.

4 Results and Discussion

In Table1, we present an overview of three experiments. The objective of all experiments was to develop a requirements specification, which was described on papers. The requirements and domain knowledges were

Table 1: Overview of three experiments

Experiment	#1	#2	#3
Target Software	Tool for developing Software Specification by simple hypertext	Simple database system for resources in a laboratory	Supporting tool for program chair of the international Conference
The num. of Pages (A4 paper)	4	12	10
The num. of customer(s)	2	2	1
The num. of designer(s)	2	1	1
The num. of scribe(s)	1	1	1

provided by customer(s), and designers analyzed and arranged them so as to realize the computer system. The scribe recorded meeting histories and minutes, and act as an operator of the computer system if it could be available in the experiment. The subjects in these experiments were graduate school students, undergraduates or staffs at the department of computer science.

In Table2, 3 and 4, we present the results of each experiment, such as spending time of each meeting, the number of Discussions and so on. Because no system was used in experiment#1, the number of Discussions and Topics in Table2, were estimated by observing both subjects' behaviors recorded by a video camera and intermediate and final products. In experiment#2 and #3, the minutes were prepared in the computer system so as to access their design rationale in the meeting histories. Its spending time are listed in a line "Spending time outside the meeting for previous meetings" of Table3 and 4. In experiment#3, the agenda was also prepared in the computer system. Its spending time are listed in a line "Spending time outside the meeting for next meeting" of Table4. As a matter of fact, third meeting of experiment#2 shown in Table3, had about one hour break, so it consisted of two meetings. But we do not distinguish them because no task had been performed during the break.

Table 2: Results of experiment#1

Experiment	#1	#2	#3	#4	Total
Spending time (min.)	165	147	174	210	696
The num. of Discussions	32	67	54	36	189
The num. of Topics	23	38	46	23	130

Table 3: Results of experiment#2

Experiment	#1	#2	#3	Total
Spending Time in meetings (min.)	60	96	125	281
Spending Time outside the meeting for previous meetings (hour)	10	10	7.5	27.5
The num. of Discussions	40	43	68	151
The num. of Topics	127	65	65	257

4.1 Materials of Experiments

The objective of these experiments is to confirm what kind of support by computer systems is helpful. But we can not state the general results from these experiments because a specific task was performed by a specific system in each experiment. So here we discuss the limits and the generalities of our experiments, by compare the settings and others of each experiment.

In our experiments, the tasks for all experiments were specifying requirements of software system, especially small and interactive software system. And the original requirements transmitted from the customer(s) to designers interactively in the meetings. So despite the difference of target software among these experiments, we may compare the three experiments for analysis, but the results are limited to such kind of meetings.

Table 4: Results of experiment#3

Experiment	pre-meeting	#1	#2	#3	Total
Spending time (min.)	-	31	50	30	111
Spending time outside the meeting for previous meetings (hour)	-	2	2	3.5	7.5
Spending time outside the meeting for next meeting (hour)	-	3	1	0	4
The num. of Discussions	-	22	22	24	68
The num. of Topics	58	83	39	7	187

The skill levels of the subjects were not exactly same. And we only had exactly three times of experiments though we had three type of experiments. So our results are limited to the qualitative analysis, especially, for example, we can not conclude the twice effectiveness even if data of an experiment were twice larger than data of another experiment. But they are enough fruitful to discuss the tendencies of human behaviors.

The actual numbers of data in each experiment in Table1, 2, Table3 and 4 are largely different. Especially in experiment#1, spending time in each meeting was long in spite of shorter products than the other experiment shown in Table1. This was why no computer system used in experiment#1 hence the scribe would not have enough data for writing rich documents. In the same reason, the mean time of Discussion in experiment#1 was relatively long.

4.2 Problems in a normal meeting

Table 5: Problems in experiment#1

Experiment	#1	#2	#3	#4	Total
About Discussion					
Missing its conclusion from the Product	34	27	28	25	28
Repeated	28	43	15	36	31
Without conclusion	25	6	7	6	10
About Topic					
Missing itself from the Product	35	39	30	26	33

We identify the problems in normal meetings by observing the experiment#1. And we discuss how these problems can be overcome by applying computer systems. In Table5, we show the kinds and amounts of problems observed from the experiment. First three kinds of problems are about Discussion as mentioned in Section2. For example in Table5, 32 Discussions were detected in the first meeting of experiment#1, and the conclusions of 11 Discussions among them, i.e. 34% of all, were not listed in the product, i.e. the requirements specification. And last one kind of problems is about Topic. Because these kinds of problems are not orthogonal with each other, total number of column direction do not become 100%.

In spite of a scribe was participated in the meetings, both the conclusions and even the Topics itself, were missed from the product, about 30%. It is too large to guarantee the quality of the product because the intention of participants seems not enough to be reflected to the product. This type of problem can be overcome by the records of audio or video recorders, which are already used in a normal meeting today. But the cost does not seem to be enough reasonable to overcome the problem completely. For example, we introduce another observation for protocol analysis in some meetings [13]. In the study, we partially introduced the technique of conversation analysis[14] for precise analysis. In the process of analysis, an experimenter made transcriptions from the video and audio record, decided the discussions from the transcriptions, summarized the contents of each discussion and finally describe a document like minutes. The spending time for the transcriptions, became about 50 times as the meeting length, and for summary and minutes became about 30 times. Of course, it is too expensive to apply this type of protocol analysis for only writing a minutes, hence these magnifications do not directly give evidences of high cost for video and audio records in the meetings. But these data would suggest us the difficulty for the usage for direct meeting records.

From the results in Table5, 31% of all Discussions were repeated discussion. Generally, meetings include redundant discussions, and the redundancy sometimes contributes to the effectiveness of the meeting. For example, by repeating the same kind of discussion during a meeting, one can represent the importance of the discussed Topic, or can relate it to the other Topics around it. But with respect to the efficiency of meetings, reducing the repeated discussions contribute to it, hence such kind of support may be provided. By observing the contents of repeated discussions in this experiment, their redundancy did not contribute to the effectiveness of the meeting. For example, some of repeated discussions were occurred because a conclusion was not provided in the previous discussions. For eliminating this type of redundancy, contents of the previous discussion and its summary, e.g. the reason why was the topic suspended, should be easily referred by all participants during the meetings.

From the discussion above, computer systems should provide the records of meeting for supporting the users, so as to develop the complete products and to make meeting processes efficiently. Especially in the software development, the records of upper stream are said to be useful for the lower stream, e.g. implementing, maintaining and revising the system. Based on the similar concepts, IBIS[5] and QOC[15] are used for recording the *design rationale*. But systems based on such concepts, are said that the structure of records is too complex to refer or browse it. One reason of its complexity is, that conclusion and the others are all located in the same level on the structure, and another reason is that the structure is too isolated from the real world structure, e.g. the event sequences in the meetings and products such as documents, in other words, the structure is too abstract or conceptual to understand. So for eliminating such complexity, the structure for the users should reflect the objects in the real world, e.g. meetings and products.

From the result in Table5, discussions without conclusion were not so many, about 10% of all. In such kind of discussions, the decision was postponed in the discussion, and sometimes this caused the repeated discussion. This type of discussions can not be directly eliminated, but by recording their contents, repeated discussions can be eliminated or decreased. The discussions without conclusion also caused the inconsistencies in the products. For eliminating such inconsistencies, the recorded data should manage the alternatives of a decision. IBIS[5] and QOC[15] has such system in their system, also our system used in experiment#2 and #3 has. In our system, the alternatives are represented by, for example, “Topic16”, “Topic17” and “Topic18” in Figure3.

The occurrences of above problems largely depend on the skill of subjects, and in this experiment, they were not real customers, designers nor a scribe but subjects. But these problems are caused by a essential function of human, i.e. conversation. So in varying degrees, such problems would generally arise in the meetings. Moreover, limiting the meeting for software development, they would arise more easily because of the complexity of products. And reporting in [16], except very big enterprises, the development processes are very ad hoc, hence our experiments were not unrealistic at all.

4.3 Suitable Supports outside the meetings

Though annotating and editing functions were available for all meeting participants in experiment#3, only annotating function was used for a scribe to ease the labor after the meeting. We give the following explanation of this result. Mostly in current electronic meeting systems [1], [2], [3], [5], [8], [18], the support for verbal conversation is not systematically integrated with *the computerized support*, e.g. editing files, voting functions and so on. For example, the functions for editing texts, sharing chalkboard or spreadsheets, are used apart from the support for communication channels, e.g. verbal and textual conversation or viewing the others face expressions. In other words, these communication channels are used as auxiliary function for their cooperation and moreover their outputs are volatile, hence logical and persistent products are not produced by such functions. Though the logical or linguistic aspects of conversation are used for the support in several tools [10], [20], suitable supports could not be accomplished only with such aspects. From the results of our observations or a report[21], the role of conversation is qualitatively and quantitatively large in the meetings. So, if both verbal conversation and computerized support are available for participants to achieve logical and persistent products, they would preferably use the former support.

From these discussion, the technologies for recording and referring the histories of conversation, seem to be important for the meeting support. This technologies result in the method for decreasing the amount of the labor for recording and referring the histories. The difficulties for these technologies are as follows;

- It is difficult to relate the textual annotations to the raw meeting histories, e.g. verbal histories.
- It is difficult to browse the histories alone at a glance.
- It is difficult to construct a structure on the meeting histories so as to easily develop products, e.g. minutes or agenda.
- It is tiresome to hear or refer a part of the histories repeatedly for developing products.

Some of current technologies can overcome some of them. For example, it is easy to record and to store the verbal and graphical data in a computer system for sequential and random accessing. And computer systems can recognize or pick up a word from continuous speech semi-automatically within some limited words[17]. So computer supports can be available for recording and referring the verbal histories in the meetings.

These computer supports for recording histories are divided into two types; (semi-)automatic support and augmenting support. In automatic support, the system used in our experiments provides a function to split utterances of each participant into a small segment with starting and ending time and the name of the speaker, by measuring their amplitudes. And in augmenting support, the system records textual annotations of each participant with time stamp and the name of the writer. So in meetings of experiment#2, automatic supports were mainly available, and in meetings of experiment#3, augmenting supports were also available in addition. This difference seems to affect the efficiency of the tasks after the meeting. In experiment#2, it took for a scribe to refer the contents of previous meetings as six times long as meeting length itself, and in experiment#3, four times long. Because augmenting support was not used by all participants in experiment#3 as mentioned above, the difference would become small.

By having interviews from subjects in experiment#3, we have the following five sub-tasks after the meetings, which would be improved by the system functions in the meetings.

1. Identifying the Discussions from the meeting histories: For summarizing the contents of meetings without missing, and easily browsing them, it is useful to segment the meeting histories into the Discussions, a semantic chunks like conversation. But from the experiences of protocol analysis, is is not easy to segment them without any support.

2. Putting an Explanation on each Discussion: An annotation for a Discussion would be helpful for browsing the stream of meetings at a glance. So we want to make annotations as easily as possible.

3. Making relationships between a Discussion and a Topic: We believe that the products of meetings should be caused by the Discussions in the meetings, and in our system, the fundamental parts of the products are Topics. So one will make mappings from the Discussions to a Topic in the first step for developing the products, hence the efficiency will give influences on all steps for products.

4. Making relationships among Topics: This is the second step after the mapping above.

5. Summarizing a Topic as a sentence: Because each Topic has its reasons or design rationale on some Discussions, one should carefully refer the contents of the Discussions while the contents of the Topic are represented as a text. This task is not difficult but tedious.

Table 6: System's Contributions to these tasks in experiment#3(%)

	Experiments			
Tasks	#1	#2	#3	Total
Identifying Discussions	22	50	38	40
Putting Explanation	-	-	71	71
Rel. between a Topic and a Discussion	36	27	63	43
Rel. among Topics	51	24	39	36
Summarizing a Topic	7	8	14	8

In Table6, we show how did our system in the meetings contribute to these tasks after the meeting. After each meeting, a scribe achieve these tasks for next meetings. And some of these tasks would be accomplished with the help of contributions by the system's functions in the meetings, but other would be accomplished only by the scribe's efforts. The data in Table6 tell the rates of contributed tasks. We decide whether a task was contributed or not, by the interview for the subject. From the interview, annotations with time stamp, both textual and verbal, were mostly helpful for these tasks. Here we discuss each task respectively;

1. Identifying the Discussions from the meeting histories: Data in the first line of Table6 tell us about 40% of all tasks for identifying Discussions were contributed by the system's functions. In our system, during the meeting, one can mark the range of a Discussion by put annotations on the action editor in Figure3. And action editor also provides the view of actions' distribution over the meeting, hence one can easily identify the turn taking in the participants' conversation. From the interview of the subject, we find that these functions mainly contributed to this task. As mentioned in the studies of conversation histories [14], [22], the distribution of actions over the time tells us the semantic breaks in the meetings. For example, During a customer's continuous talking, if another, e.g. a designer, interrupts the customer, new or another topic may start from the interruption. Action editor visually provides this kind of information.

2. Putting an Explanation on each Discussion: Data in the second line of Table6 tell us about 71% of all tasks for putting Explanations were contributed by the system's functions. Because no Explanation were put during the first and second meeting, these corresponding cells has no value, labeled '-'. So, Improvement of this task does not affect the whole tasks. Discussions have links to Topics and Topics have their contents as a text, so Topics can play the Explanations of Discussions, hence the pure Explanations of Discussion were not so many. This task was contributed by the annotations during the meeting, especially, textual annotations could directly use the part of an Explanation.

3. Making relationships between a Discussion and a Topic: Data in the third line of Table6 tell us about 40% of all tasks for making relationships between a Discussion and a Topic, were contributed by the system's functions. As mentioned above, we assume that Topics are caused by Discussions. Therefore, a Topic is created by referring the contents of Discussions for the first time. In this case, annotations among the Discussions or an Explanation of a Discussion could be directly used as a part of Topic sentence.

   But as spending many meetings or discussions, Topics which were already discussed, are taken up again. In this case, one should select the corresponding Topic from the current products. Browser of our system shown in Figure3 contributed to such task, because Browser lets the users easily *browse* the products at a glance, and the tree structure provided by Browser lets them find the corresponding Topic even if the appearance of written characters are different.

4. Making relationships among Topics: Data in the fourth line of Table6 tell us about 36% of all tasks for making relationships among the Topics, were contributed by the system's functions. A subject utilized the co-occurrences of words in annotations or co-occurrences and adjacency of annotations for the relations, because related Topics would be discussed adjacently with each other.

5. Summarizing a Topic as a sentence: Only 8% of all tasks were contributed in this case. In other words, a subject should write the contents of Topics as sentences with few system's supports during the meetings. we infer the reasons as follows. First, most Topics were prepared as an agenda in the system, so functions during the meeting did not contribute to this task. Second, participants during the meetings tended to make *empty* annotations so as to mark the range of Discussions. So a scribe should write the contents of Topics or Explanations only by his effort, e.g. hearing and viewing the contents of each actions. This facts reflects that we Japanese are not familiar with typing and that we should have intermediate steps for getting our characters, i.e. Kanji and Kana characters.

4.4 Suitable Supports during the meetings

Table 7: System's contributions during the meeting#3

Experiment	#1	#2	#3
Spending time in a meeting (min.)	31	50	30
Rate of contributed time (%)	71	94	80
Num. of Topics (total until each meeting)	58	141	180
Rate of referred Topics (%)	100	57	17
Num. of new Topics	58	83	39
Rate of referred new Topics (%)	100	96	80

Here we observe what and how were the systems' functions used during the meetings in experiment#3 for discussing the suitable system's support during the meetings. Especially, we discuss the following three tasks during the meetings.

1. Discussion with referring an agenda: In experiment#3, agenda was completely prepared as a form of Topic structure, i.e. Browser in Figure3, in our system. So the agenda were referred in the meetings about 80 % of all spending time as shown in Table7. And at the fourth line in Table7, we show the rates for referred Topics in each meeting, and we can find the rate are decreased along the progress of the experiment, 100, 57 and 17. But we calculate the rate again only focus on the new Topics in each meeting as shown at the sixth line in the Table, the rates are almost same, more than 80%, and high. And also from the rates at fourth and sixth lines in Table7, we may conclude that the repeated discussions about a Topic rarely occur over the meetings. So computer mediated agenda for participants would be useful to progress the meetings. Before normal meetings, some participants also prepare the agenda, normally as

a paper documents, in the same way of our experiments. In contrast with the tasks during the meetings, describing the agenda do not need the real-time feature, hence it is easy to introduce such functions into current meetings.

2. Reconfirmation by hearing the previous actions in Discussions: Our system provides the function to reconfirm the contents of previous actions by hearing or viewing them directly. But the rate of reconfirmed Topics was only about 1% of all and also the rate of spending time for reconfirmation was also about 2% of all. So this function did not contribute to the meetings well. In a few example of this function, action editor shown in Figure3 played an important role, because the subjects appointed the actions for reconfirming not directly but ambiguously like “first half of the previous meeting” or “middle of the first meeting”.

3. Looking up the influences of modifications: It is difficult to find the influence of modifications among the products. But leaving them alone without any care, inconsistent products would be produced. From the observation of normal meetings [13], we found that Topics discussed in adjacent area, tend to have mutual influence of modification. Based on this feature, our system can suggest the influences of modifications to the users, hence users can avoid the inconsistency. In this experiment, only one conclusion were modified during the meeting, and our system suggested four other Topics to the subjects, and they can decide to modify all four Topics in the same way. So we can partially confirm the effectiveness of the function.

5 Conclusion

In this paper, we discuss what and how should computer system support the workers participated in the face-to-face meetings by observing three types of experiments with or without the computer system. From the results of our experiments and analysis, we have the following implications.

- Because histories of normal meetings do not tend to be reflected by the products of the meeting, supports for reflecting the contents to the products are needed. By the computer systems, it is available for such support to record the actions in the meetings directly as meetings histories, and to facilitate browsing or referring the histories.

- It is not useful only with the direct-recorded histories of meetings. Therefore, the computer system lets the users annotate any tags with time stamp with the records so as to easily refer them. The annotation function would be mostly used by the scribe in the meetings, but it is hard to the other participants to annotate such tags because they should discuss the contents.

- Suitable representation for the meeting histories are needed. For example, it should let users easily identify the turn taking in the conversation at a glance.

- Computer mediated agenda is reasonable because normally users only refer it during the meeting without getting behind the current in the meetings. In contrast, one can prepare it without the strict synchronization for the other participants, like during the meeting.

In this paper, we limit the range of meeting in the face-to-face form. But if we expand the range into the geographically distributed meetings, we should take account to the support for the awareness like gestures and eye gazing. These issues can be partly overcome by using the real-time tele-conferencing systems [2], [3] with our system. Also we should support the domain specific issues for providing suitable resources.

Acknowledgments

The authors would like to thank the members of *User Oriented Software Process Group*, International Institute for Advanced Study of Social Information Science, FUJITSU LABORATORIES LTD, for discussions and insightful comments which led to some of the issues raised in this paper. And they would like to thanks Prof. K. Ochimizu, *Japan Advanced Institute of Science and Technology, Hokuriku*, for discussions and insightful comments about the drafts of this paper.

References

[1] J.F. Nunamaker, A.R. Dennis, J.S. Balacich, D.R. Vogel, and J.F. George, Electronic meeting systems to support group work. *Commun. ACM*, Vol. 34, No. 7, pp. 40–61, Jul. 1991.

[2] Hiroshi Ishii. TeamWorkstation: Towards a Seamless Shared Workspace. In *CSCW90*, pp. 13–26, Oct. 1990.

[3] Kazuo Watanabe, Shiro Sakata Kazutoshi Maeno, Hideyuki Fukuoka, and Toyoko Ohmori. Distributed Multiparty Desktop Conferencing System: MERMAID. In *CSCW90*, pp. 27–38, Oct. 1990.

[4] Mark Stefik, Gregg Foster, Daiel G. Bobraw, Kenneth kahn, Stan Lanning, and Lucy Suchman. BEYOND THE CHALKBOARD: computer supported for collaboration and problem solving in meetings. *Commun. ACM*, Vol. 30, No. 1, pp. 32–47, Jan. 1987.

[5] Jeff Conklin and Michael L. Begeman. gIBIS: A hypertext tool for exploratory policy discussion. In *CSCW86*, Dec. 1986.

[6] G. L. Rein and C. A. Ellis. rIBIS: a real-time group hypertext system. *Int. J. Man-Machine Studies*, Vol. 34, No. 3, pp. 349–368, Feb. 1991.

[7] J. Kawakita. Hassou-Hou (in Japanese). Chuko Shinsho, 1967.

[8] Kazuhisa Kawai, Akichika Shiomi, Naohiko Takeda and Hajime Ohiwa. Preliminary Experiments with a Distributed and Networking Card-handling Tool Named KJ-Editor (in Japanese). Journal of Japanese Society for Artificial Intelligence, Vol. 8, No. 5, pp. 47–56, Sep. 1993.

[9] Colin Potts and Kenji Takahashi. An active hypertext model for system requirements. In *7th International Workshop on Software Specification and Design*, pp. 62–68, Redondo Beach,CA, Dec. 1993.

[10] Colin Potts, Kenji Takahashi, and Annie I. Anton. Inquiry-based requirements analysis. *IEEE Software*, Vol. 11, No. 2, pp. 21–32, Mar. 1994.

[11] Kouichi Doi, Akira Ohmori, Hisayuki Horai, Isamu Watanabe and Yoshinori Katayama. A paradigm for requirements capture based on the speech act theory (in Japanese). Proc. of 47th conference of IPSJ, Japan. p.125. Oct. 1993.

[12] Haruhiko Kaiya and Motoshi Saeki. A Groupware for Face-to-face Meetings to Develop Software Specifications. In *InfoScience '93, International Conference organized by Korea Information Science Society*, pp. 691–698. Korea Information Science Society, Oct. 1993.

[13] Nobuyuki Miura, Haruhiko Kaiya and Motoshi Saeki. An Experimental Analysis of Verbal Histories in Meetings and Structures of Specifications. IPSJ sig. notes. 94-HI-53, Vol. 94, No. 23, pp. 71–78, Mar. 1994.

[14] Stephen C. Levinson. *Pragmatics*, chapter 6, pp. 284–370. Cambridge Univ. Press, 1983. about Conversation Analysis.

[15] A. MacLean, R. M. Young, V. M. E. Bellotti, and T. P. Moran Questions,options,and criteria: Elements of design space analysis. *Human-Computer Interaction*, Vol. 6, No. 3 & 4, pp. 201–250, 1991.

[16] Watts S. Humphrey and David H. Kitson. A comparison of U.S. and Japanese software process maturity. In *13th international conference on software engineering*, pp. 38–49, May 1991.

[17] Katunobu Itou, Satoru Hayamizu and Hozumi Tanaka. Continuous Speech Recognition by Context-Dependent Phonetic HMM and an Efficient Algorithm for Finding N-Best Sentence Hypotheses (in Japanese). The Transaction of the institute of Electronics, Information and Communication Engineers, Vol. J75-D-II, No. 6, pp. 1023–1030, Jun. 1992.

[18] Joseph S. Valacich, Alan R. Dennis, and J. F. Nunamaker. Jr. Electronic meeting support: the GroupSystems concept. *Int. J. Man-Machine Studies*, Vol. 34, No. 2, pp. 261–282, Feb. 1991.

[19] J.R. Searle. *Speech Acts*. Cambridge University Press, 1969.

[20] Terry Winograd. Where the action is. *BYTE*, Vol. 13, No. 13, Dec. 1988.

[21] Gary M. Olson, Judith S. Olson, Mark R. Carter, and Marianne Storrosten. Small group design meetings: An analysis of collaboration. *Human-Computer Interaction*, Vol. 7, No. 3, pp. pp.347–374, 1992.

[22] Masahiro Nishi, Haruhiko Kaiya and Motoshi Saeki. Interaction Analysis in the meetings among customers & software developers (in Japanese). IPSJ sig. notes. 92-HI-41, Vol. 92, No. 15, pp. 17–24, Mar. 1992.

Biographies

Haruhiko Kaiya is a research associate at Japan Advanced Institute of Science and Technology.

Kaiya received PhD in electrical and electronic engineering from Tokyo Institute of Technology. He is a member of the IEEE Computer Society and ACM. His URL is `http://www.jaist.ac.jp/~kaiya/`, and e-mail address is `kaiya@jaist.ac.jp`.

Nobuyuki Miura is a researcher at Global Computing Laboratory, Software Laboratories Nippon Telegraph and Telephone Corporation.

Miura received M.E.in electrical and electronic engineering from Tokyo Institute of Technology. His e-mail address is `miura@slab.ntt.jp`.

Go Anai received B.E.in electrical and electronic engineering from Tokyo Institute of Technology.

Tsuyoshi Ebata received B.E.in electrical and electronic engineering from Tokyo Institute of Technology.

Hiroki Nagata received B.E.in electrical and electronic engineering from Tokyo Institute of Technology.

Motoshi Saeki is an associate professor at Tokyo Institute of Technology.

Saeki received PhD in computer science from Tokyo Institute of Technology. He is a member of the IEEE Computer Society and ACM. His URL is `http://saeki-www.cs.titech.ac.jp/saeki-home-e.html`, and e-mail address is `saeki@cs.titech.ac.jp`.