A Groupware for Face-to-face Meetings to Develop Software Specifications

Haruhiko Kaiya Motoshi Saeki
Dept. of Electrical & Electronic Engineering, Tokyo Institute of Technology
Ookayama 2-12-1, Meguro-ku, Tokyo 152, Japan

Phone: +81-3-3726-1111(ext. 2192)
Fax: +81-3-3729-1399
E-mail: `kaiya@cs.titech.ac.jp` `saeki@cs.titech.ac.jp`

In this paper, to support the cooperative works, we introduce a new type of a groupware based on the experimental results. In the meetings to develop software specifications, we have found many items which had been discussed in the meetings, were missing in the specifications, and that the meetings contained conversation to hinder the advance of the meetings, and conclusions without consistency in the meetings. Our groupware will support the users to decrease such faults in the meetings.

1 Introduction

In the requirements analysis stage of software development, works are essentially cooperatively performed by many different roles of workers, and normally they advance their works through their communications such as verbal conversations in meetings. In the meetings we observed, workers had mostly spent their time on verbal communications. In the Gary's paper [1], he reported that "the participants in 8 of these 10 meetings spent on average more than 90% of the time talking to each other".

To support the workers in such meetings, several groupware tools have been developed [2], [3], [4]. However these tools are based on the mechanism that force users to follow some working criteria, hence users cannot work on their own style. Therefore, to provide the comfortably group work environment to human workers, we must capture the feature of the human activities from the analysis of actual meetings in developing software specifications [5].

In this paper, we introduce a new type of a groupware based on the results from the analysis of actual meetings in developing software specifications. In section 2, we present the type of meetings where we intend to support. Section 3 presents the analytic results from the observations of the meetings. In section 4, we introduce the design of our groupware tool to support the workers in the meetings, which functions are obtained from our analytic results. Section 5 presents the user interfaces to use our groupware tool. Finally, we discuss the way to evaluate the applicability of this groupware tool to the actual meetings.

2 Cooperative Processes in Software Development

Our goal is to develop a technique to support the workers in the meetings to develop software specifications. A software specification is the document to specify functions, behaviors and structures of the software system to be developed. We focus on the specifications described in natural languages because specifications have been generally described in natural languages in actual software developments.

2.1 Role

In software development processes, many workers participate in the development of the system. For example, following workers would participate in the meetings for developing requirements specifications. In this paper, we call such workers as *roles.*

Customer: they order the target system. Normally, they limit the resources, such as money, time and so on.

User: they use the completed products of the software development process.

Designer: they actually describe the specification documents.

Implementer: they develop source codes, such as programmers.

Coordinator: they manage the progress of the meetings.

Maintainer: they maintain the products.

2.2 Meetings

In a meeting, workers will develop ideas about the target system during the conversation, and it would be difficult for them to put down its definition as a document completely. So outside the meetings, workers will arrange the ideas and describe a specification document and/or intermediate documents. We consider a software specification process as the sequence of both the meetings and the works outside the meetings as such.

2.3 Steps

Even in one meeting, several types of discussions are held in turn. We call these types of discussions as *steps*. The kinds of steps are as follows:

Order-step: a customer presents his/her requirements to workers. The workers may ask unclear points in his requirements to him.

Planning-step: workers are scheduled to engage in the software processes. Normally, a coordinator manages the workers' interests.

Explanation-step: designers explain the specification document they developed.

Review-step: workers review the specification document and revise it if necessary. For example, workers will trace the behavior of the system and check the functions, i.e. walkthrough.

During the software process, many intermediate products are produced, and the kind of intermediate products depends on the step above.

3 Results of our analysis

We have observed several meetings to develop software specifications. To record them, we used with video camera. Analyzing the data of the records of the meeting activities, we have clarified the points which should be supported. The points are as follows:

1. Missing conclusion which the workers had: We consider the final specification as the collection of the conclusions about the software to develop. And in the meetings to develop the specification, workers have the conclusions through the discussions. According to the final version of the specification, we have found much missing conclusions, which have been discussed but have not been rejected in the meeting.

Table 1: Rate of discussions with missing conclusions(%)

	total time (hour)	number of meetings	fully	partially
Project1	5	3	36.3	22.4
Project2	12	4	6.7	22.8
Project3	10	9	7.7	31.0

Table 1 shows the rate of discussions with missing conclusions in our observation. We observed three projects (project1, 2 and 3 in the first column of table 1) to develop the software specification. The fourth column of table 1 labeled "fully", shows the rate of discussions, where the conclusion is fully missing in the final specification. The fifth column of table 1 labeled "partially", shows the rate of discussions, where the conclusion is partially missing. For example, in the meetings of project3, 31% of all discussions were partially missing. In the meetings of project1, 36.3% of all discussions were fully missing.

2. Discussion to hinder the advance of the meetings: We consider that the following type of discussion hinder the advance of the meetings.

   (a) Repetition: the same topic is repeatedly discussed, and the conclusions have been always same.

   (b) Upsetting: current conclusion is always replaced by the others.

(c) No conclusion: no conclusion have been made from discussions.

We have found many occurrences of such discussions to hinder the advance of the meetings. Table 2 shows the time rate of such discussions in our observation. For example, in the meetings of project1, workers spent 21.7% of time repeating the same discussions. In contrast with the meetings of project1, workers only spent 1.8% of time on the repetitive discussions in project2. This difference between project1 and 2 would come from the participation of "a clerk" or "a scorekeeper", who had recorded the advance of the meetings.

Table 2: the time rate of discussions to hinder the advance of the meetings(%)

	repetition	upsetting	no conclusion
Project1	21.7	2.5	25.0
Project2	1.8	5.0	2.0

3. Inconsistency coming from conclusion-changes: Even when a conclusion was changed in the meetings, some conclusions, that are semantically related to the changed conclusion, remained without the corresponding changes in the specification. So these remaining conclusions results in inconsistency. We have found such inconsistency in the observation.

Taking account of the results of these analysises, we introduce the design of our groupware tool in the following section.

4 Tool Design

4.1 Overview of the Tool

And from the results of our analysis in section3, workers in the meetings should perform the following tasks:

1. Workers should find out and record both the discussions and their conclusions to find the missing conclusions in discussions.
2. Workers should know whether each item has been already decided or not yet to avoid conversations to hinder the advance of the meetings.
3. Workers should have the information about semantical relationships between a conclusion and the others to check whether a conclusion should be changed or not.

It is impossible for the workers to construct tool data completely along the progress of the meetings. And normally some meetings are held to develop a specification. So in the meetings, workers should operate the groupware tool as far as the operations do not hinder the progress of their discussions. And the rest operations of tool, such as structuring data, can be performed outside the meetings. Our groupware tool supports the workers both in and outside the meetings incrementally as follows:

1. Our tool records almost all records in the meetings, such as utterances and demonstration on the blackboard.
2. Our tool supports workers who construct the structure of the records and the components of a specification, normally outside the meeting time.
3. Our tool supports workers who retrieve the records and components.
4. Our tool generates template of the final specification from the structured information, i.e. shown in figure1.

Specification:
Card based specification describing tool

⋮

item1: Content button:

explanation1: Push here if you wish to write the content of this Card.

item2: Content window:

⋮

Figure 1: Example of the template

To perform these tasks, workers can use graphical user interfaces in section5.

Figure2 shows the overview of tool usage.

4.2 Data Structure

Figure 3 shows the data structure of our tool. The center column in this figure shows the hier-

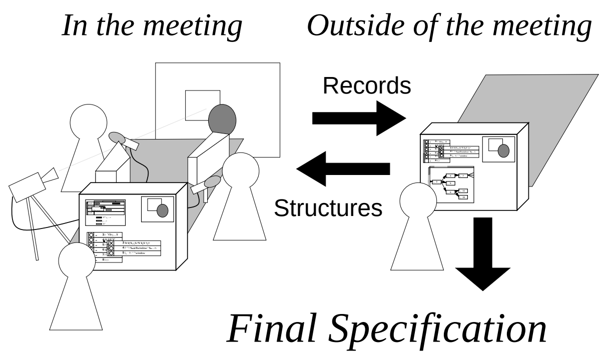


Figure 2: Overview of tool usage

archy of the specification process, the left column shows the hierarchy of workers and the right column shows the hierarchy of the products.

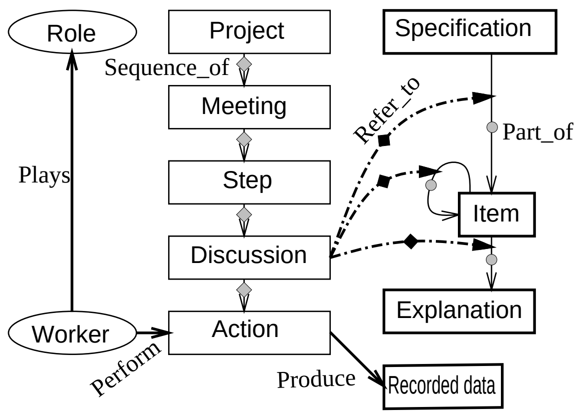


Figure 3: Data structure

Item: Label of specification parts. It might be the "title" of a section, a subsection or a paragraph in the specification document.

Explanation: Sentence and/or figure to explain the items. It corresponds to the "content" of a section, a subsection or a paragraph in the specification document.

Action: Label of workers' utterances and/or demonstration on the blackboard in the meetings. Each action has a worker who acts it.

Discussion: Sequence of actions. It is related to the item and/or the explanation in the specifications.

Step: Steps of meetings defined in section2.3. For example, order-step, planning-step, explanation-step and review-step.

Role: type of workers defined in section2.1.

In figure3, a "worker" performs an "action" which produces "recorded data" such as utterances and demonstrations.

Sequence_of: Related entities are temporally ordered of their occurrence time. For example, a "discussion" is a sequence of "actions".

Refer_to: In a discussion, an "item" becomes a part of a "specification", an "item" becomes a part of another "item" or an "explanation" becomes a part of an "item". We define a relationship between the "discussion" and the "part_of" relationship as "refer_to". "Refer_to" relationship has the several types as follows:

- Create: In a discussion, an "item" is created and it becomes a part of a "specification". In a discussion, an "item" is created and it becomes a part of an "item". In a discussion, an a "explanation" is created and it becomes a part of an "item". The relationship btween them has the "create" type.
- Touch: In a discussion, an "item" is referred and it is a part of a "specification". In a discussion, an "item" is referred and it is a part of an "item". In a discussion, an a "explanation" is referred and is a part of an "item". The relationship btween them has the "touch" type.
- Delete: In a discussion, an "item" is deleted and it is not a part of a "specification". In a discussion, an "item" is deleted and it is not a part of an "item". In a discussion, an a "explanation" is deleted and is not a part of an "item". The relationship btween them has the "delete" type.

Figure4 shows an example of data. In this example, a step of "ordering" the system is the sequence of discussion1, 2, 3, 4 and 5. In the "discussion1", an "item1" is created as a part of specification. "Discussion2" refers the relationship between the "item1" and "explanation1" and so on. The "discussion5" consists of five actions. In this discussion, Three workers participate, and the customer-worker had the major responsibility of conclusion here because he had

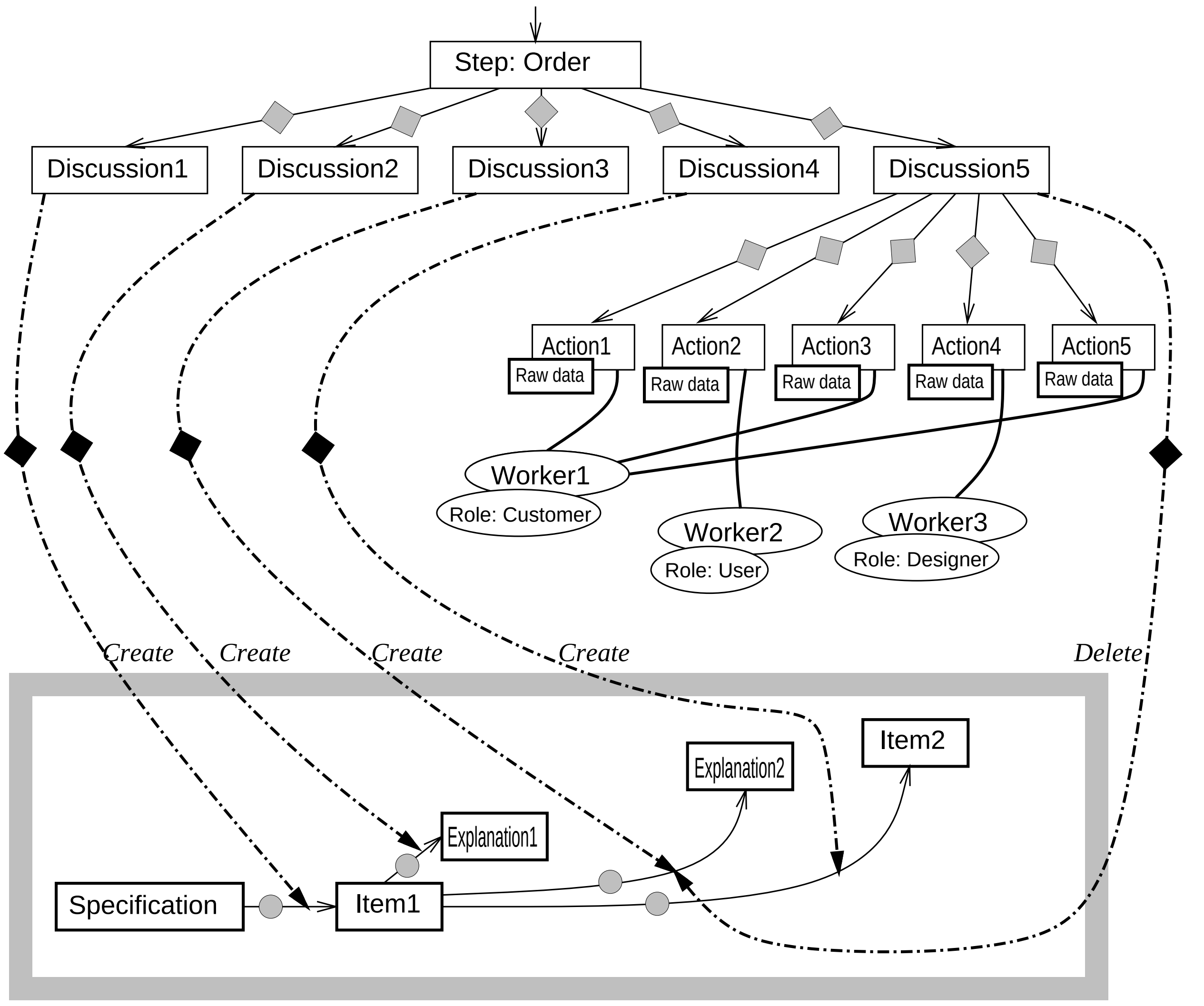


Figure 4: Example of Data

acted many actions and he also acted the last action.

Figure1 shows the example of the product from the step in figure4. In this specification, "explanation2" is not listed because it was cancelled in the the step in figure4. The words in this specification are described by the workers using the "recorded data" in the discussions.

5 User Interface

We introduce several graphical user interfaces for workers to construct the data of our tool.

1. Relate the "workers" with the "roles":

   This relation is defined on the "project" level, so the workers should put their roles on our tool before starting a "project".

2. Identify the "actions" and "recorded data":

   In the meetings, The workers have a face-to-face meetings normally and each worker uses his microphone to have utterances shown in figure5. Each utterance of them is recorded with a microphone which is connected to a computer system. Some of pictures on the blackboard are also recorded with a video camera which is also connected to a computer system. So the workers can identify the "actions" and "recorded data" semi-automatically.

3. Relate the "actions" with "workers":

   This relation can be constructed automatically because our tool can recognize the speaker of an utterance by examining which microphone the utterance has been inputted. Pictures drawn on the blackboard

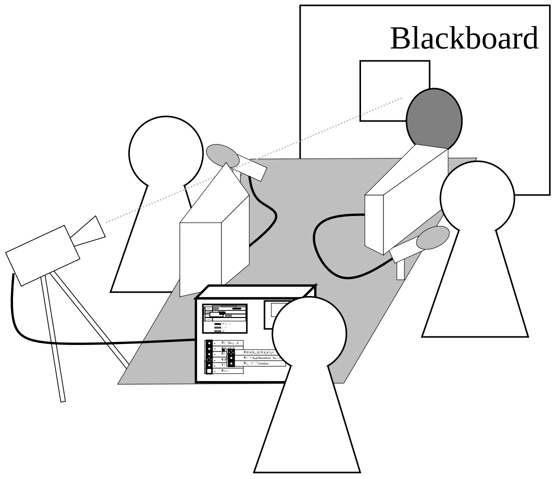


Figure 5: Microphone and Video camera

should be related manually.

4. Identify the "discussions":

   Workers should group several "actions" together into a meaningful "discussion" shown in figure6. By pointing the area of the "PushButton" in this figure, workers can hear or see the corresponding "recorded data". Our tool will suggest the break of the discussions. For example, if two "actions" have performed at intervals of 10 seconds, our tool suggests the break of current "discussion" in figure6.

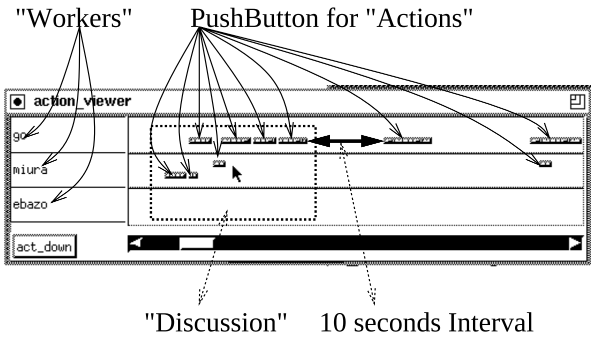


Figure 6: Interface: action viewer

5. Relate a "discussion" with an "item":

   With the interface in figure7, workers can create an "item" from a "discussion". Workers may not write a label of the "item" in the `field_editor` during the meeting but must write it until the next meeting.

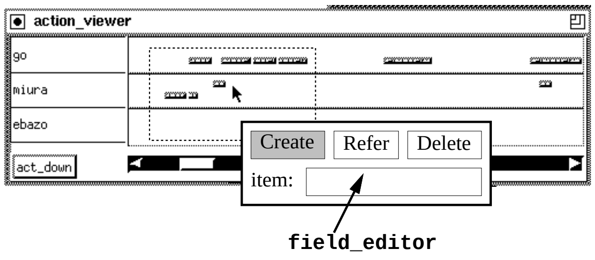


Figure 7: Interface: item/explanation creator

6. Retrieve the "items":

   As shown in in figure8 and figure9, workers can retrieve the existing "items". In figure8, existing "items" and their "explanations" are listed. The symbol "◯" denotes that the corresponding "items" or "explanations" are concluded and the symbol "×" denotes that they were discussed but deleted. For example, "explanation3" in "item5" was previously discussed but has been deleted. In figure9, workers searches for the existing "items" with respect to the structure of the product.

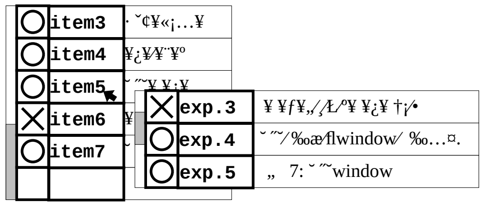


Figure 8: Interface: item menu

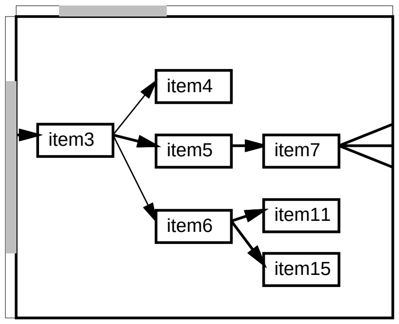
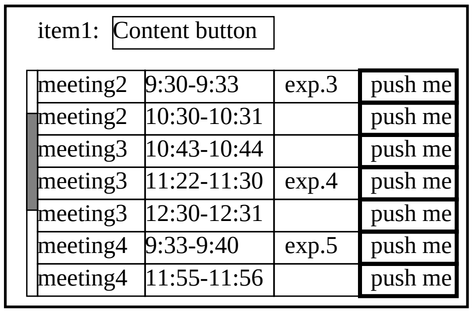


Figure 9: Interface: item tree

By pointing the area, e.g. $\boxed{\texttt{item3}}$, $\boxed{\texttt{exp.2}}$, in figure 8 and 9, workers can display the list of related discussions shown in figure10. Each line in figure10 denotes a discussion in the meetings. In the third column of figure10, "exp.3" denotes the discussion about the explanation of "item1" and blank boxes denote the discusion about "item1" itself. By pointing the area $\boxed{\text{push me}}$ in figure10, workers can display the "action viewer" such as figure6, and they can retrieve the "recorded data" from the "action viewer".



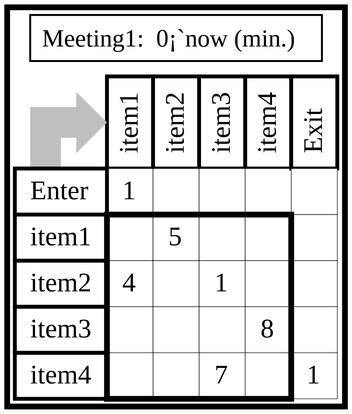
item1: Content button

meeting2	9:30-9:33	exp.3	push me
meeting2	10:30-10:31		push me
meeting3	10:43-10:44		push me
meeting3	11:22-11:30	exp.4	push me
meeting3	12:30-12:31		push me
meeting4	9:33-9:40	exp.5	push me
meeting4	11:55-11:56		push me

Figure 10: Interface: discussion list

7. In the meetings, many topics are discussed in turn. Our tool can display the patterns of topic changes shown in figure 11. In this figure, discussed topics are considered in the same as the "items" in the specification because discussed topics are related to the "items" in our tool. The label of "Meeting1: 0∼now(min.)" shows that workers are now in the meeting1 and that they display the whole time in the meeting. Four items have been created in the meeting, "item1" was discussed first, and "item4" is discussed now. "Item2" has been discussed at five times after the discussions of "item1", "item4" have been discussed at eight times after the discussions of "item3" and so on.

   If a pattern of topic changes occurs frequently, the topics would have a semantical relation strongly. For example in figure 11, "item1" and "item2" would be closely and semantically related to each other. That is to say, if we modify "item3", the modification have much influence on "item4", and vice versa. The pair of "item3" and "item4" is a similar example too. Our tool can display such tables for workers



Meeting1: 0¡`now (min.)

	item1	item2	item3	item4	Exit
Enter	1				
item1		5			
item2	4		1		
item3				8	
item4			7		1

Figure 11: Interface: change viewer

   - to create the relationship among the specification and
   - to detect the influences of modifying the part of specification.

   In contrast with the former, workers may check the pair of items which are not related at all.

8. Create the "specification":

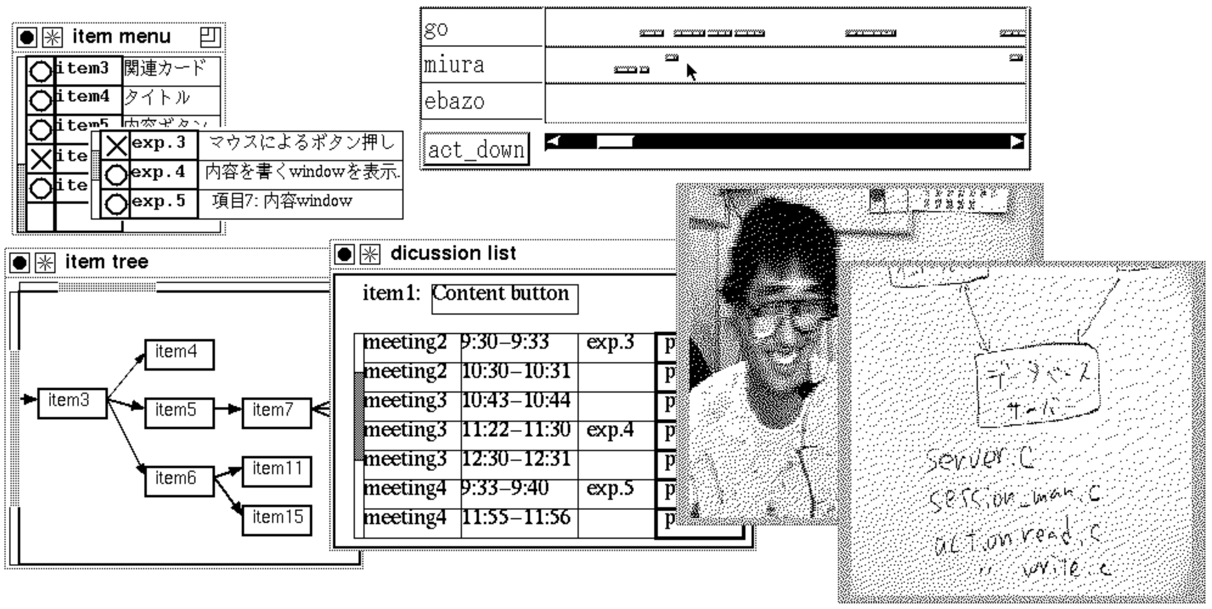
   From the "items" and the "explanations", which do not have deleted through the meetings, workers write the specification document. Our tool creates a template of the documents as shown in figure1.

6 Discussion

We are now developing a prototype of this groupware tool on the networked UNIX-based platform shown in figure12. In figure12, several interfaces and two "recorded data" of video camera are mapped. In the future, we will assess its performance and applicability. The standards of assessment are

- the amount of missing conclusions in the specification,
- the amount of the discussions which hindered the progress of meetings and
- the existence of inconsistency in the specification.

They are the same as the standards of our observation of the actual meetings.



○	item3	関連カード
○	item4	タイトル
○	item5	内容ボタン
×	ite	
○	ite	

×	exp.3	マウスによるボタン押し
○	exp.4	内容を書くwindowを表示.
○	exp.5	項目7: 内容window

item1: Content button

meeting2	9:30–9:33	exp.3	p
meeting2	10:30–10:31		p
meeting3	10:43–10:44		p
meeting3	11:22–11:30	exp.4	p
meeting3	12:30–12:31		p
meeting4	9:33–9:40	exp.5	p
meeting4	11:55–11:56		p

Figure 12: Snapshot of the prototype

It is still not clear what kind of information is effective to both the progress of meetings and the specifications. So we should clarify such features from the use of our tool. For example, we can collect the *excellent* patterns of actions for the each steps like Winograd's conversation diagram[6].

Our tool only support the meetings in the software development process. But the data which our tool record would be useful for the other parts in the process. For example, "recorded data" of our tool easily provide the *design rationale*[7] for the implementers, maintain staffs and/or the workers who should hand over the works.

Acknowledgements

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.

References

[1] Gary M. Olson, Judith S. Olson, Mark R. Carter, and Marianne Storrosten. Small group design meetings: An analysis of collaboration. *Human-Compter Interaction*, 7(3):pp.347–374, 1992.

[2] Jeff Conklin and Michael L. Begeman. gIBIS: A hypertext tool for exploratory policy discussion. In *CSCW'86 Proceedings*, Dec. 1986.

[3] C. Potts and G. Bruns. Recording the Reasons for Design Decisions. In *10th International Conference on Software Engineering*, pages 418–427, 1988.

[4] J. Lee. Extending the Potts and Bruns Model for Recording Design Rationale. In *13th International Conference on Software Engineering*, pages 114–125, 1991.

[5] Haruhiko Kaiya and Motoshi Saeki. An Experimental Results on Meetings to develop Software Specifications. In *Poster Sessions: Abridged Proceedings, 5th International Conference on Human-Computer Interaction*, page 176, Aug. 1993.

[6] Terry Winograd. Where the action is. *BYTE*, 13(13), Dec. 1988.

[7] John M. Carroll and Thomas P. Moran. Introduction to this special issue on design rationale. *Human-Computer Interaction*, 6(3&4), 1991.