Building the Structure of Specification Documents from Utterances of Requirements Elicitation Meetings

Nobuyuki Miura †*

Haruhiko Kaiya ‡

Motoshi Saeki §

† NTT Software Laboratories
3-9-11, Midori-Cho,
Musashino-Shi, Tokyo 180, Japan
E-mail: miura@slab.ntt.jp

‡ Dept. of Information Science
Japan Advanced Institute of
Science and Technology, Hokuriku
15, Asahi-dai, Tatsunokuchi-Cho,
Noumi-Gun, Ishikawa 923-12, Japan
E-mail: kaiya@jaist.ac.jp

§ Dept. of Computer Science
Tokyo Institute of Technology
2-12-1, Ookayama, Meguro-Ku
Tokyo 152, Japan
E-mail: saeki@cs.titech.ac.jp

Abstract

In the process of requirements elicitation in software development, it is usual that participants with different roles have the series of meetings and requirements analysts compose specification documents between the meetings. There are many studies for supporting these processes, such as cooperative working models in meetings, tools based on those models, and specification & design methods such as Structured Analysis and Object-Oriented Analysis. However there are no studies how to describe specification documents based on contents of meetings. Participants communicate verbally with each other, so we consider that the effective method should be based on verbal histories, i.e. utterances appearing in meetings. We propose a method to write specification documents considering that structures of meetings is reflected into structures of specification documents. Briefly speaking, the assumed basis of our method is that analysts put pairs of subsequently discussed topics (we call them "temporally adjacent topics") into close positions in the tree structures of the specification documents. In this paper, we also assess the feasibility and the effectiveness of the method through several experiments and case studies.

1 Introduction

Software development is cooperative work performed by various kinds of persons, e.g. customers, users, project managers, requirements analysts, designers, programmers and so on. Especially, requirements elicitation phase, which is the earliest phase in the software development process, has two types of human activities ; the activities in meetings which customers, users, analysts (in some cases called, interviewers) and so on may participate in (we call these "*meeting activities*"), and the activities of describing specification documents outside the meetings. These activities are usually repeated alternately and the specification is incrementally constructed. So far, cooperative models and supporting tools based on them for these activities have been studied [1, 2, 3, 4, 5, 6]. Many specification & design method such as Entity Relationship Model, Structured Analysis and Object-Oriented Analysis[7, 8, 9] have been also developed and are putting into practice. However, we don't have methodologies how to describe specification documents based on the records of meeting activities. Almost all meetings for eliciting requirements are *face-to-face* style and the meeting activities are modeled as a sequences of verbal conversation [10]. Therefore we focus on the records of utterances in the meetings since we think that they have useful information to construct requirements.

We think that there are many advantages in describing documents from utterance records. One of them is that we can include all of the discussions in the documents. According to [11], although we take minutes of the meetings, about 30% of the information that were discussed in the meeting is missing in the documents. The second advantage is that we can include very useful information such as decision rationale[12, 13] into specification documents because utterance records contain them. The third one is that we can construct comprehensive requirements specification. In the meetings, the participants that have unclear items about the requirements inquire to explain-

* This research was carried at Tokyo Institute of Technology.

Table 1: Case studies

	Case #1	Case #2	Case #3
System to be developed	Graphic Editor	CASE Tool based on Hyper Cards	Schedule Manager
Number of Workers	6	5	2
Roles of Workers	Customer, Analysts(5)	Customer, Clerk, User, Analyst(2)	Analysts(2)
Number of Meetings	3	4	5
Total Time of Meetings (Hours:Minutes)	4:43	11:36	3:34
Volume of Produced Documents (number of A4-sized sheets)	8	7	15

ers (customers in some cases or analysts in the other cases) and the explainers reply the answers to them so that they can understand their inquiries and make their understanding of the requirements clear. Therefore, if we concentrate on these inquiry-answer cycles, we can make specification documents more comprehensive.

This paper discusses a method to construct specification documents by considering the meeting activities that the participants performed. We assume that the model of meeting activities is a sequence of utterances and the model of specification documents is tree structure of text whose nodes denote chapters or sections. In our method, analysts develop the specification documents reflecting the structure of the meeting activities into structure of documents. As it can be considered that the topics that are subsequently discussed in the meetings (more precisely speaking, the topics that are frequently adjacent on a time axis of a sequence of utterance ; we shorten them to *temporally adjacent topics*) have the semantical relationship, they should be closely positioned in the structure of the document. It enables the person to read an item in the document to refer easily and quickly to the items having semantical relationship. Therefore, our structuring technique is based on the principle that we should make the temporally adjacent topics closer in the structure of documents. The objective of this research is to investigate whether temporally adjacent topics are closely positioned in actual documents and to assess our structuring technique based on the above principle.

The organization of the paper is as follows. In the next section, we will introduce two kinds of analysis to justify our principle. Concretely speaking, these results involve that temporally adjacent topics in the meetings had often semantical connection to each other and that they were not necessarily closely positioned in the documents. Thus we should make the temporally adjacent topics closer in the documents. Section 3 presents our structuring technique for specification documents by using temporal relationships among utterances occurring in the meetings. And we will apply our technique to the actual meetings and assess the comprehensiveness of the produced documents through the experiments in section 4.

2 Analyzing Actual Meetings and Documents

First of all, we will introduce three examples of the meetings which are used for our analyses. The aim of the analyses is to know the relation of utterances of meetings and the produced documents. In sections 2.2 and 2.3, each analysis procedure and its result will be presented.

2.1 Examples

Table 1 shows three actual meetings which we observed and analyzed. These meetings had common characteristics in the following :

- Meetings was recorded by a video camera.
- There were not so many participants. Thus they always devoted themselves to discuss topics of the meetings without discussing locally.
- The participants sat down around a table in face-to-face style.
- The participants could use a white board and make notes.
- The meetings were not strictly controlled and the participants could talk whenever they intended to do.
- The participants composed specification documents which were written in natural language (Japanese) and contained several drawings.

The participants were university students (including graduated students) of the computer science department. However, they had received an enough education of software engineering and some of them had ever worked for software development in industries.

Thus we consider that these meetings were not so situationally different from the practitioners' meetings.

2.2 Analysis of Temporally Adjacent Relationship among Topics

First of all, we should consider how we model the meetings. We can capture the meetings as a temporal sequence of the discussed topics, roles of the speakers, speech acts of the utterances and so on. In this paper, we abstract a meeting to a sequence of the topics discussed in it because of simplicity. Figure 1 shows an example that a meeting is represented by a sequence of the topics. As shown in the figure, we call a subsequence of the utterances where a specific topic is discussed a *discussion lump*. The time length of a discussion lump is typically from tens seconds to several minutes. We model the meetings as pairs of time adjacent topics with times of adjacency. In this paper, we concentrate on the topics related to the system being developed and omit the utterances not referring to the system, e.g. time schedule, costs, resource (e.g. worker) assignments, and so on.

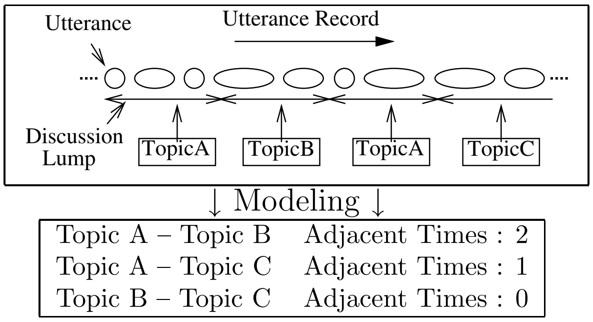


Figure 1: Modeling of structures of meetings

Next, we should make sure that temporal adjacent relationship among the topics can be one of useful information to model the meetings by analyzing the case studies mentioned in section 2.1. We observed the specification documents which were produced as results of the meetings and extracted from the documents pairs of the specification items which had semantical relationship. If we also have to modify the other item to maintain consistency of the document when we modify an item in the document, we can decide that these two items have semantical relationship. That is ; we extract the propagation of modification on the specification items in the document.

The analytic procedure is as follows.

1. We extract the specification items from the document by observing the syntactic structure of the document.
2. By focusing on what specific keywords appear in the specification items and in the utterances of the meetings, we identify discussion lumps and their correspondences to the specification item.

   In the example of the case study #2, we focused on the keywords "data base", "storing cards", and "relationships among cards" in the specification item "Data Base (DB) : Storing cards and the relationships among them", which appeared in the document. Subsequently we grouped the utterances containing the keywords into a discussion lump as follows :

   > "The data base hold the relationship among the cards, doesn't it?"
   >
   > "Umm, it is not good that the DB can identify just the existence of the cards. To hold the relationship is needed and this function should be added."

   We made this discussion lump correspond to the specification item "Data Base(DB) : Storing cards and the relationship among them".
3. We extract pairs of the specification items which have the semantical relationships in the documents by exploring propagation that would occur if items were modified.
4. We calculate how many percentages of the temporally adjacent topics have semantical relationship.

The step 3 was performed by the different persons from the person who performed the steps 1 and 2 to keep fairness of our analysis.

Table 2 shows the result of the analysis. In the example of the case study #1, in fact, 70% of the topics which were subsequently discussed in the meetings (i.e. temporally adjacent topic pairs) had the semantical relationship. In addition, observing the other results, we can conclude that the topics which were subsequently discussed in the meetings have the semantical relationship. These results lead to the usefulness of the temporally adjacent relationship among

Table 2: Analytic results of the relations of temporally adjacent topics

Case	#1	#2	#3
Number of Topic	22	22	19
Pairs of Topics	231	231	171
Pairs of Temporally Adjacent Topics (A)	30	87	55
Pairs of Topics Having Semantical Relationship (B)	70	153	71
A∩B (C)	21	78	40
Percentage(%) (C÷A)	70.0	89.7	72.7

topics for modeling the meetings.

2.3 Analysis of the Structure of Specification Documents

In this paper, we consider specification documents that are written in a natural language and that contain several drawings. Generally speaking, they consist of several sections hierarchically and they can be modeled as a tree whose nodes are the chapters, sections, subsections, paragraphs and so on. Figure 2 illustrates the tree structure of a document.

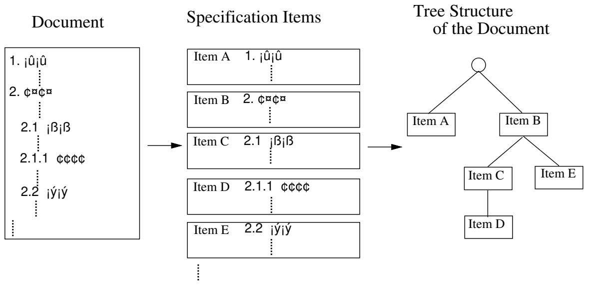


Figure 2: Modeling of structures of specification documents

In this section, we introduce an additional analysis on whether the specification items which had been subsequently discussed in the meetings were closely located in the documents. The point of our method is that the items that were subsequently discussed at many times should be written closely in the structure of the documents. From the previous analysis, we have a finding that the topics that were subsequently discussed in the meetings have semantical relationship. The closer the items having semantical relationship are written in the documents, the more comprehensive the documents are for readers. If the documents would usually be written satisfying the above constraints, we needed nothing and the comprehensive documents had already been obtained. Thus we need check whether specification documents satisfied the constraints or not. That is ; we investigate how close the items that were subsequently discussed in the meetings appeared in the documents. The structural close position on the documents is shown in Figure 3. The pairs of the nodes in the same level or having parent-child relationship were located in close position.

The following is our analytic procedure.

1. As mentioned in the analysis of section 2.2, we identify discussion lumps and extract pairs of the topics that were subsequently discussed in the meetings (i.e. extracting temporally adjacent pairs).
2. We structure the documents in a tree form and extract the specification items closely located in the tree.
3. We calculate the percentage of the items that were subsequently discussed and that were closely located.

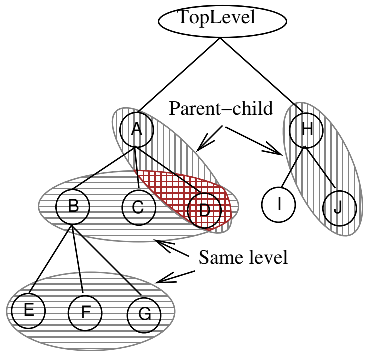


Figure 3: Close positions in tree structures

Table 3 shows the analytic result of our three case studies. In the table, "B∪C" (the addition of the cardinalities of B and C) denotes the number of the items closely located in each specification document. "A∩(B∪C)" means the temporally adjacent topic pairs which are closely located as specification items in the document.

In the case #3, we have extracted 19 topics from its document and we could have 171 ($= {}_{19}C_2$) pairs of the topics in theory. Actually, 55 of these pairs were subsequently discussed in the meetings and just 12 of the 55 pairs were closely located in the document. That is ; 21.8 % of the temporally adjacent topics closely appeared in the structure of the document.

Table 3: Analytic results of positions of temporally adjacent topics in tree structures

Case	#1	#2	#3
Number of Topics	22	22	19
Pairs of Topics	231	231	171
Pairs of Temporally Adjacent Topics(A)	30	87	55
Pairs of Topics Located in the Same Level in a tree (B)	78	76	26
Pairs of Topics Located in the Parent-Child Relationship in a Tree (C)	15	1	13
A∩(B∪C)	16	37	12
Percentage (%) A∩(B∪C) ÷ A	53.3	42.5	21.8

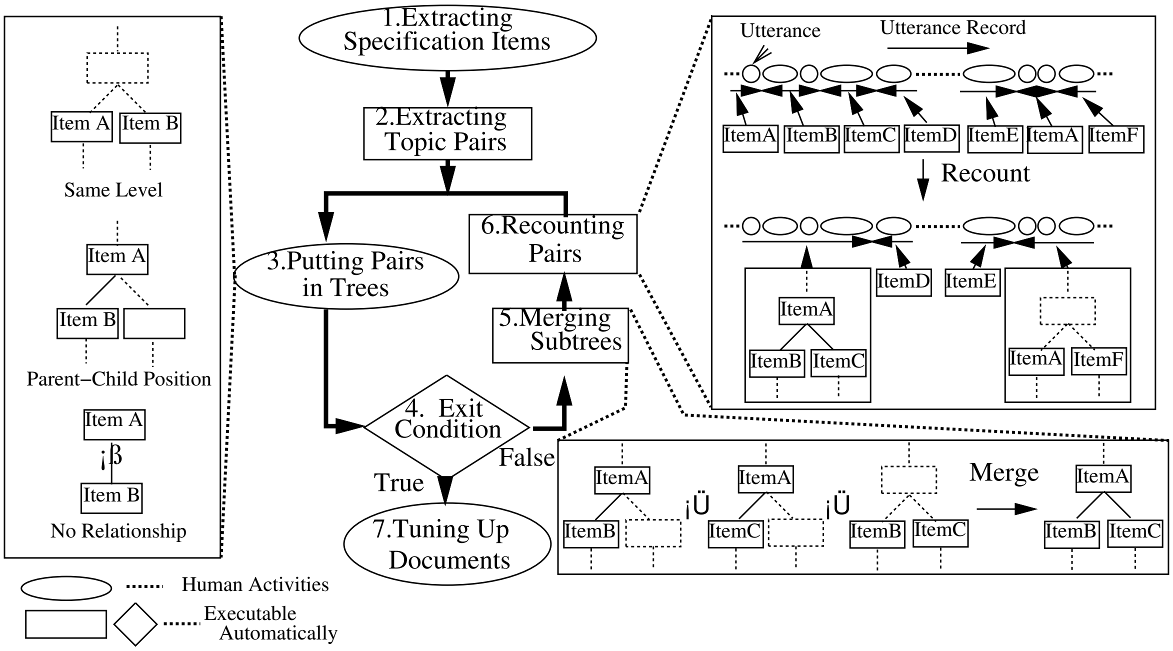


Figure 4: Outline of a method of structuring specification documents by using temporally adjacent topics

By observing each of the case studies, we can conclude that the topics subsequently discussed in the meetings were not necessarily closely located in the specification documents. This conclusion leads to the possibility to get more comprehensive documents by putting the topics subsequently discussed in the meetings to the close position in the documents.

3 Method for Constructing Specification by using Utterance Records

As we pointed out in the section 2.2, the items subsequently discussed in the meetings have semantical relationship. Therefore it seems to be easier to read the specification if a specification writer writes it with these items closely located in it. We propose a method to construct a specification document based on temporally adjacent relationship on utterances of the meetings. An outline of the method is shown in Figure 4.

In the figure, ovals stand for the activities performed by requirements analysts, i.e. human workers, while rectangulars and rhombuses express the activities automatically executed by a supporting tool. We explain the detail of these activities.

1. **Extracting Specification Items**

   We group the utterances, recorded (e.g. by a video camera), into discussion lumps by identifying the topics that were discussed, and extract the specification items which were presented in the grouped utterances.

2. **Extracting Pairs of Temporally Adjacent Topics**

   As shown in Figure 1, we extract the pairs of the topics that were subsequently discussed in the meetings, and count the occurrences of the extracted pairs in the utterance records.

3. **Putting Pairs of Temporally Adjacent Topics in Trees**

   We pick up the pairs of the temporally adjacent topics that occurred most frequently in the utterance records, and build trees whose nodes are the selected pairs. Each pair is located in the same level (brother position) or in parent-child position in a tree. If a pair has no relationship, the pair is not located in the tree. Which alternatives should be taken depends on the workers. The above procedure is repeated in the order of descending the occurrence times of the topic pairs. As a result, we have many small subtrees.

5. **Merging Subtrees**

   The subtrees obtained in the previous step have been built from binary relations of the specification items. When we regards a subtree as a specification item, we can extend temporally adjacent relationship to relationship with more than 3 arities by using transitivity. As shown in Figure 4, we can merge the built subtrees to a larger tree.

6. **Re-count Temporally Adjacent Pairs**

We regard a merged tree as a specification item and re-count the occurrences of the pairs of temporally adjacent "new" items. In the example of Figure 4, we regarded the items A, B and C as a new item, and A and F as an item. It is considered that the "new" item "A+B+C" is adjacent to the item D once, and E is adjacent to the A+F once.

4. **Exiting a Loop**
   If either of the following conditions are satisfied, we finish the activities.
   (1) We can have no more new pairs of temporally adjacent items.
   (2) All subtrees are merged to a tree.
7. **Tuning Up Documents**
   In our experience, these activities often finish when the condition **(1)** holds, i.e. in many cases we do not have a single tree when we finish. Because we cannot construct a specification document from the only information about temporally adjacent relationship on the items, but other information is needed. In this case, the workers build a tree by connecting and structuring the subtrees.

   In some cases, the constructed tree has some redundant structures, we reduce those structures. The examples are shown in Figure 5.

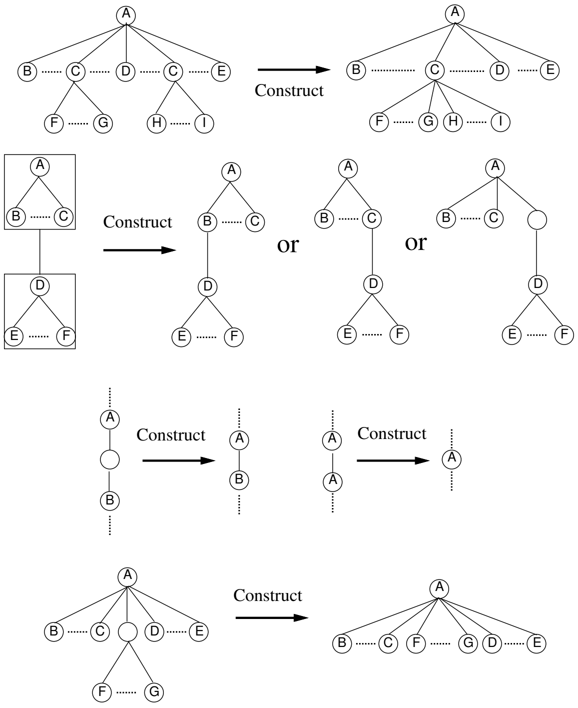


Figure 5: Reduce redundant structures

4 Assessment of Our Method

In this section, we apply our method, mentioned in the previous section, to the case study #2 and assess it. This assessment has been done by experiments on comprehensiveness of the produced specification document by applied our method.

4.1 Applying the Method

4.1.1 Result of Applying the Method

The followings are the steps of applying our method to the case study #2 in Table 1.

- **Extracting Specification Items**
  Originally in our method, we extract specification items from utterance records. But, because documents produced applying method will be used in the assessment which compare the original document and the produced one, we had extracted specification items from the original specification document. The number of the extracted items is 108.
- **Extracting Pairs of Temporally Adjacent Topics**
  We had extracted 341 pairs of the topics that were subsequently discussed.
- **Loops**
  The loop from the step "Putting Pairs of Temporally Adjacent Topics in Trees" to "Recount Temporally Adjacent Pairs" was repeated at three times and we exited from the loop because the condition (1) in "Exiting a Loop" was satisfied at the entrance of the fourth loop.
- **Putting Pairs of Temporally Adjacent Topics in Trees**
  In this step of the first loop, for example, we had the result as shown in Table 6 of Appendix.

  Table 4 shows that extracted pairs decreased as the execution went round the loop.

  For example, before the second loop, we had 278 pairs and put 50 pairs in the same level or in parent-child position in the trees.

Table 4: The results of classification of structures

Loop Count	1	2	3	4
Temporally Adjacent Topic Pairs	341	278	165	136
Pairs used for Building Subtrees	146	50	6	0

- **Merging Subtrees**
  For example, we had merged subtrees in the first loop as shown in Figure 7 in Appendix.

Table 5 shows the result in the step of the merging process.

For example, before the second merge we had already 75 subtrees, and they had been merged to 43 trees. As the execution went round the loop, the trees became larger and deeper.

Table 5: The result of the merging subtrees

Loop Count	1	2	3
Number of Subtrees before Merging	146	75	43
Number of Subtrees after Merging	68	43	39
Average Number of Nodes in a Subtree	2.6	4.9	6.3
Maximum Depth of Subtrees	2	4	7

- **Re-count Temporally Adjacent Pairs**

  Table 7 in Appendix shows the merge of the discussion lumps performed in the step of "Re-counting the temporally adjacent pairs" in the first loop.

  Just after extracting the specification items, we had 387 discussion lumps, and finally obtained 189 lumps through the process of re-counting the temporally adjacent pairs of the items.

  As for the number of the specification items, we had 278 pairs of temporally adjacent items at the first time, could reduce 165 pairs at the second time, and obtained 136 pairs as a final result. (These figures are presented in Table 4.)

- **Tuning Up Documents**

  We obtained 39 subtrees from the information of temporally adjacent relationship and 15 specification items which did not belong to any subtree. We connected them and build into a tree. Finally the 205 specification items were included in the tree. It means that the specification consists of 205 sentences or phrases.

142 of 205 links between nodes in the tree have been structurally connected by means of the only information on temporally adjacent topics. It means that our method could build about 70% of the structure of the specification document.

4.1.2 Discussion

The fact that we could structure the 70% of the specification by our method shows its effectiveness and applicability.

The specification document (344 lines) that has been newly produced by applying our method included more sentences than the old document (188 lines) that was produced before. The reason is that the specification items were duplicatedly described in multiple positions of the document. However we guess that the new document may be more comprehensive because the other descriptions that are needed to understand an item are positioned near it. Furthermore 45% of the structure of the new document is equal to the old one from the viewpoint of tree structure. It means that the new document is not quite different from the document that we ordinary produce. Especially the leaves in the new document are almost equal to the old one. Our method helps us to build comprehensive structure of specification rather than to compose comprehensive sentences.

Investigating the new document in details, it has the new sections or subsections as follows :

- the section where backgrounds of the system to be developed are briefly described.
- the section where the important and essential parts of the system are outlined.
- the sections where the functions of the system are grouped based on similarity. In these section, the other similar functions are duplicatedly and briefly described.

  (Appendix shows the example of the function " Buttons for adding and changing links between cards".)
- the section where the behavioral scenario of the system is described.

All of these sections or subsections came from structuring the document based on the temporally adjacent topics and it is considered that they contribute to increasing comprehensiveness of the document. This basis will be discussed in the next subsection.

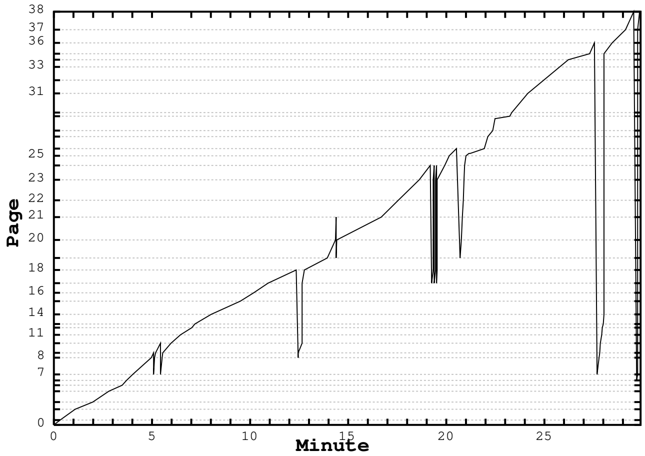
4.2 Assessment of the Method by Experiments

4.2.1 Experiments

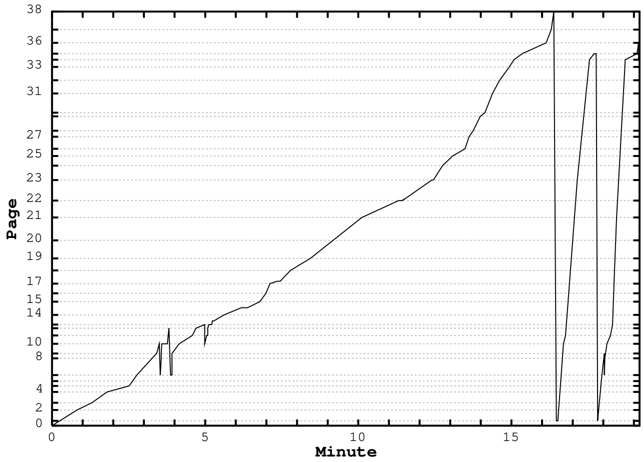
The problem is the criterion of the comprehensiveness of specification. That is ; what documents are easy to read and to understand. In this paper, we adopt the following criterion of comprehensiveness.

> When readers read a document, the less they should read it back again (backtrack) or look it ahead to understand, the more comprehensive it is.

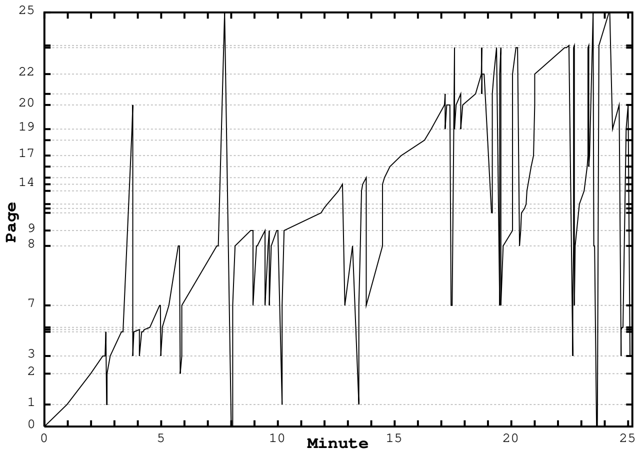
Based on this criterion, we made a comparison of the document that was newly produced in 4.1.1 with the old document produced in the case study #2. This comparison was done by the experiments on the behavior of reading them. We observed the readers' be-



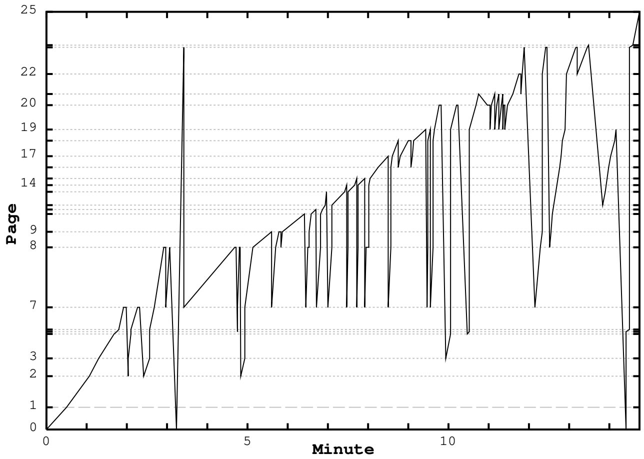
Reader #1 (New Document)



Reader #2(New Document)



Reader #3 (Old Document)



Reader #4 (Old Document)

Figure 6: Results of assessment experiments

havior of reading these two documents and explored how often the readers read them back again and looked them ahead. The experimental procedure is as follows.

- Two documents, new one and old one, were re-described so that a sheet contains just a section or a subsection. We focus on which sheet the readers look and the re-described document enables us to detect the behavior of reading back and looking ahead.
- We had four readers, i.e. subjects — two for reading the new documents that were produced by our method and the other two for the old documents.
- The readers read the re-described document and their behavior was recorded by a video camera. It recorded which sheet they read and their voices.
- After the experiments, we let them see both of the document and had an interview about comprehensiveness.

4.2.2 Experimental Result and Discussion

Figure 6 shows the readers' behavior.

In each graph, the horizontal axis stands for time expiration and the vertical one graduated in sheet numbers. The intervals of the sheet numbers depend on the description volumes of the sheets. For example, if you see the graph of the reader #3, you can find that the eighth page has the volume twice as large as the second page in the old document.

In the case of the reader #3 of Figure 6 he referred to the second page of the old document from one minute to two minutes after starting the experiment. After three minutes passed, he looked ahead to the 20th page while reading the 4th page. Furthermore he read back again to the second page while reading the eighth page after 6 minutes.

The readers #1 and #2, who read the new document, backtracked and looked ahead greatly less than the readers #3 and #4 did. In the sense of our criterion mentioned in 4.2.1, we can conclude that the new document was comprehensive.

Through the interviews, our subjects answered the positive opinions and negative ones to the new document. As for the positive ones, they told that the new document became more comprehensive because the sections and the subsections listed up in the section 4.1.2 were added to the new document.

The negative opinion is as follows. "I had an impression that the new document is comprehensive for new comers to read it first. However once they understand the outline of the system to be developed or when they refer to it during the phases in the lower stream of the development such as testing phase, it might not be so suitable." It includes the several redundancies (i.e. duplicatedlly positioned items) and these redundancies result in the difficulties for them to retrieve the information they want from it. This opinion includes the significant suggestion that the suitable document structure depend on the role of the readers and on the development phases where the document is referred to. We should build need multiple structure of specification documents according to the situation, e.g. by using hyper-text systems.

5 Conclusion

This paper discussed the method to reflect the utterance structure of the meetings to the specification structure. Our basis is that the specification items that were subsequently discussed in the meetings should be closely written in the specification document. To clarify that our method is useful, we had two preliminary analyses of the actual meetings and the produced documents. These analyses suggested that the actual documents did not hold our basis and were not so comprehensive. Thus there were rooms to apply our method effectively to develop more comprehensive documents by reflecting the utterance structure to them. The experiments to apply our method supported its effectiveness.

Our method is the first step towards deriving requirements specification from meeting records in a systematic way. Our method is also effective to reduce missing information from completed specification because the meeting records hold all the information in the form of voice and images. We consider that all the information consists of specification of the system to be developed and call this specification *multi-media specification*. Thus we need to investigate how to store this kind of information in a structural way and to provide the functions to retrieve easily the information that the workers want. To structure the meeting records more efficiently, the emphasis will be on domain-specific keywords and speech acts appearing in the utterances in the future research.

References

[1] C.A. Ellis, S.J. Gibbs, and G.L. Rein. Groupware : Some Issues and Experiences. *Commun. ACM*, 34(1):38–58, 1991.

[2] Mark Stefik, Gregg Foster, Daiel G. Bobraw, Kenneth kahn, Stan Lanning, and Lucy Suchman. Beyond the Chalkboard : Computer Support for Collaboration and Problem Solving in Meetings. *Communications of the ACM*, 30(1):32–47, Jan. 1987.

[3] J. Conklin and M. Begeman. gIBIS: A Hypertext Tool for Exploratory Policy Discussion. *ACM Trans. on Office Information Systems*, 6(4):303–331, 1988.

[4] Conklin,E.J. and Yakemovic,KC.B. A Process-Oriented Approach to Design Rationale. *HUMAN-COMPUTER INTERACTION*, 6(3 & 4):357–393, 1991.

[5] T. Winograd. Where the action is. *BYTE*, 13(13):256–260, 1988.

[6] Colin Potts, Kenji Takahashi, and Annie I. Anton. Inquiry-Based Requirements Analysis. *IEEE Software*, 11(2):21–32, Mar. 1994.

[7] Peter Chen. The entity-relationship model – toward a unified view of data. *ACM TODS*, 1(1), 1976.

[8] DeMacro T. *Structured Analysis and System Specification*. Yourdon Press, New York, 1978.

[9] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lonrensen. *Object-Oriented Modeling and Design*. Prentice-Hall, 1991.

[10] Gary M. Olson, Judith S. Olson, Mark R. Carter, and Marianne Storrosten. Small Group Design Meetings: An Analysis of Collaboration. *Human-Computer Interaction*, 7(3):347–374, 1992.

[11] H. Kaiya, M. Saeki, and K. Ochimizu. Design of a hyper media tool to support requirements elicitation meetings. In *Proc. of IEEE Seventh International Workshop on Computer-Aided Software Engineering*, 1995.

[12] C. Potts and G. Bruns. Recording the Reasons for Design Decisions. In *Proc. of 10th ICSE*, pages 418–427, 1988.

[13] J. Lee. Extending the Potts and Bruns Model for Recording Design Rationale. In *Proc. of 13th ICSE*, pages 114–125, 1991.

Appendix

We show a part of the case study which we applied our method to (in 4.2.1). After the step "Putting Pairs of Temporally Adjacent Topics in Trees" in the first loop, as shown in Table 4, we performed the steps "Merging Subtrees" and obtained Figure 7. And then the step "Re-count Temporally Adjacent Pairs" was performed and we had Table 7 as a result. This example is a part of the section " Buttons for adding and changing links between cards", where the functions related to link manipulations are listed up.

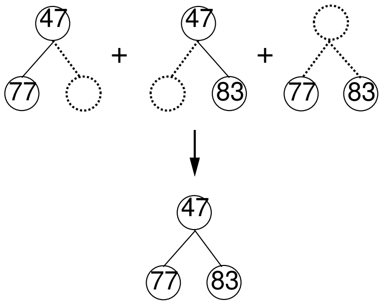


Figure 7: An example of merging subtrees

Table 6: An Example of Putting Topic Pairs in Trees

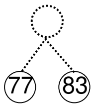
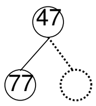
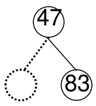
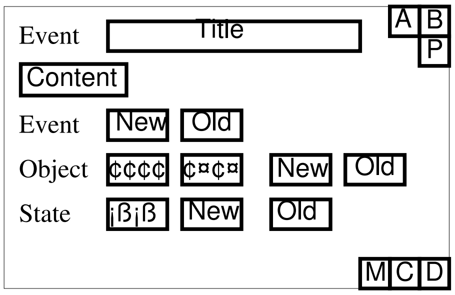
Pair of Specification Items	Items #77 & #83	Items #47 & #77	Items #47 & #83
Number of Occurrences	12	5	5
Building A Subtree			
Item #47	The Layout of a card window is shown below 		
Item #77	A "Cut" button is used for removing the links between the selected card and this card		
Item #83	A "Move" button is used for modification of links		

Table 7: An example of re-counting temporally adjacent pairs

1 hour and 20 minutes has passed in the second meeting

Before Processing Items 47	77	83	After Processing	Utterances
*			*	Well, we will decide the layout of the buttons displayed in a card window.
*			*	Okay.
*		*	*	If we consider link manipulations, Move button ...
*		*	*	Move button? Is it for cutting the links to this card
*		*	*	and then creating new links?
*		*	*	What? I feel that it is not intuitively correct.
*		*	*	Well, I think that it is alright if users became experienced.
*	*		*	What is the rest? Button for removing links?
*	*		*	Cut Button?
*	*		*	Yes. It is similar to Move button. Its position is the right of Move,isn't it?
*	*		*	Well. It is in the right side of Move.

* stands for the range of a discussion lump.