Automatic animation of discussions in USENET

Jun Yabe* Shin Takahashi Estuya Shibayama
Department of Mathematical and Computing Sciences
Tokyo Institute of Technology
2-12-1 Ookayama Meguro-ku Tokyo,152-8552, JAPAN
yabe@sslab.sony.co.jp, shin@is.titech.ac.jp, etsuya@is.titech.ac.jp

ABSTRACT

This paper proposes a technique for generating more comprehensible animations from discussions, which are often hard to follow, in USENET. This technique consists of two steps. In the first step, our prototype system generates a scenario from articles in a news thread using the quote relationship. In the second step, it generates an animation based on the scenario, casting 3D avatars as the authors of the articles. We also implemented a prototype system based on this technique and made several animations from articles posted to USENET.

Keywords

Automatic Animation, USENET, using 3D Avatar

1. INTRODUCTION

Asynchronous discussion systems such as USENET, mailing lists, and electronic meeting rooms on BBS and the WWW have been widely accepted in the Internet community. These asynchronous discussion systems have an advantage of high-quality information over synchronous text communication systems such as chat systems. The reason is that users can take enough time to write high-quality articles. These articles have been archived in various sites, so that we can search and read past articles.

When users read these articles, they usually use news/mail readers such as GNUS or Microsoft Outlook. These traditional news/mail readers usually have two window panes. One pane shows the structure of the whole thread, and it is so called "summary pane". The other pane displays one selected article.

Many researchers have pointed out the problems of this style of reading and have proposed alternative layouts. Most of this past work consisted of improvements to the summary window, and created visualizations of news groups and threads, so that the user could easily find articles of interest among massive numbers of articles of various quality. Galaxy of News[12] showed USENET articles clustered by keywords extracted from the articles and supported users in searching for articles of particular interest. There are also systems that visualize the transition of topics in the discussions. These systems can show subtle differences in topics between the articles in a thread. HISHO[14] shows the transition of topics on a reference tree, using the similarities between articles in a news thread.

However, these systems do not help users to read each individual article. The only way to understand a discussion in USENET is to read each article, but this is difficult. One reason for this is the particular form of netnews articles, which is different from the form of general documents such as a novel or a report. Another reason is that a news thread consists of articles written by multiple authors who are not consistent. Therefore, the following problems still remain.

(1) The authors use quotes in order to clarify which sentence their discussion follows. The "References:" field recorded in the headers of articles is also used to suggest context, although quotes can suggest more concrete relations. However, it is a nuisance to read copies of sentences repeatedly. Moreover, multiple quotes tend to let one article include many different topics, and impose a burden on the readers.

(2) It is difficult to remember the authors and their statements because articles do not have enough characteristic information such as face, voice, or shape.

Figure 1 shows a typical example of the first problem. Readers usually read each article in a thread in a depth-first manner on the reference tree. In this example, the user may read articles A, B, C, D, .., E in order. B-1 may be related strongly to B-2, but D-1 may have a different topic from D-2. If so, the user must read multiple.topics simultaneously.

The second problem is caused by the lack of enough information to distinguish the authors for readers. Identification of authors is necessary to understand their intentions. Articles contain information to identify the author such as the "From:" header in each article and signatures, but they do not have enough characteristics to remember the authors and their statements only by reading a thread. In face-to-face discussion in the real world, we can easily remember each speaker and statement because we can relate his or her face, voice, or shape to his or her statements.

*The author's curret address is SUPC MW dept. Sony Corporation.

AVI 2000, Palermo, Italy.
© 2000 ACM 1-58113-252-2/00/0005..$5.00

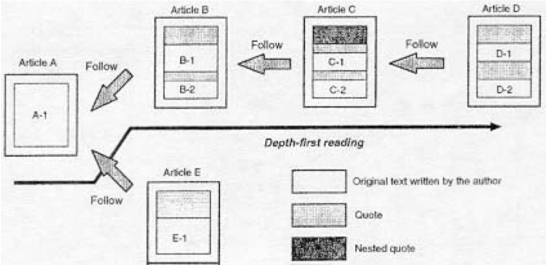


Figure 1: Depth-first reading

To solve these problems, we propose a technique that generates animations of discussions from articles in a news thread in USENET, using 3D avatars allocated for each author. This technique consists of two steps shown in Figure 2. (1) The first step is the generation of the scenarios on which the animation is based. (2) The second step is generation of animation with a 3D avatar allocated to each author.

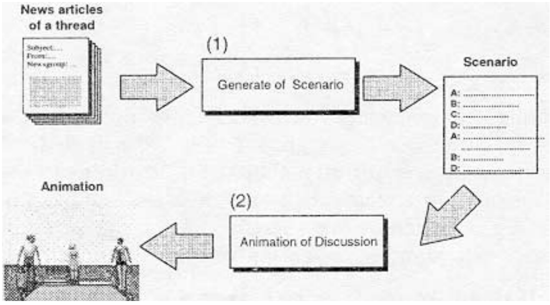


Figure 2: Abstract of the System

The generation of the scenarios solves the problems caused by quotes because the user can understand a discussion without repeated reading of copies or reading different topics simultaneously. Moreover, the animation of discussion solves the problem of identification of authors because 3D avatars have much information to associate authors and their statements such as face, voice, and shape.

We have developed a prototype system based on this technique, that automatically generates a script written in TVML[5; 3]. TVML is a script description language to produce a TV program using 3D avatars and a voice synthesizer, and can generate a TV program from a script.

2. GENERATION OF SCENARIOS

This section describes a technique for generating a scenario based on the relationships between netnews articles.

2.1 Relation of text segments

In a log of chat communication or a scenario of a TV program, each speech is basically related to the speech immediately before it. Even when a speech is related to a speech earlier than the last one, the relation between them is obvious from the contents or the scenario includes some explanation of the relation (Figure 3).

On the other hand, relations between the articles in a news thread are difficult because they are expressed by *quote* and *reference*. A reference shows a relation between articles, and a quote shows relations between parts of articles. For example, in Figure 2.2 article B refers to article A and quotes two segments from article A. Article B consists of the two quoted text segments and two other text segments commenting on the two quoted text segments.

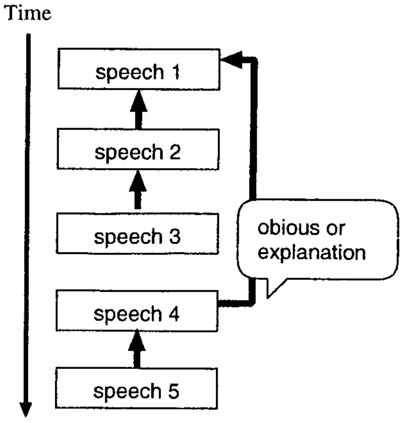
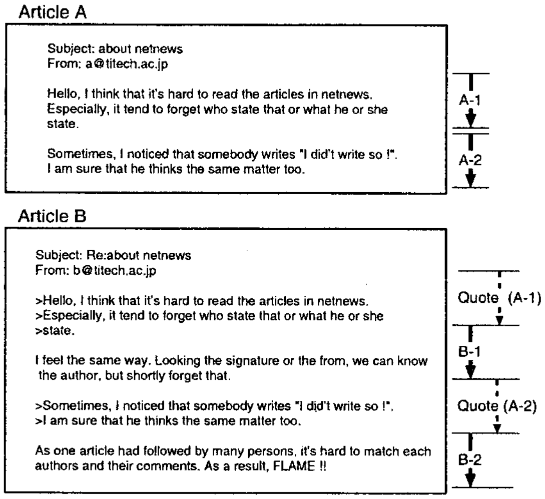


Figure 3: relation of speech



Subject: about netnews
From: a@titech.ac.jp

Hello, I think that it's hard to read the articles in netnews.
Especially, it tend to forget who state that or what he or she
state.

Sometimes, I noticed that somebody writes "I did't write so !".
I am sure that he thinks the same matter too.

Subject: Re:about netnews
From: b@titech.ac.jp

>Hello, I think that it's hard to read the articles in netnews.
>Especially, it tend to forget who state that or what he or she
>state.

I feel the same way. Looking the signature or the from, we can know the author, but shortly forget that.

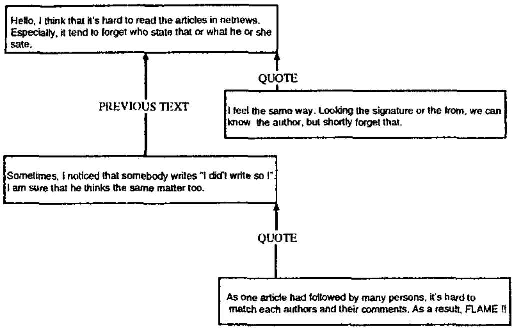
>Sometimes, I noticed that somebody writes "I did't write so !".
>I am sure that he thinks the same matter too.

As one article had followed by many persons, it's hard to match each authors and their comments. As a result, FLAME !!

Figure 4: Relation by quote

The relations between the text segments in Figure 2.2 are shown in Figure 5. There are two kinds of links that connect two text segments. One is a link by quote that connects the text segment quoted by an article and the text segment just after the quoted text. Another link specifies that the two text segments are originally arranged in sequence in the original article. This link is generated to maintain the relationships between text segments divided by multiple quotes. Compared to relations of speeches in TV programs or chat communication, text segment relations in netnews have the following features.

1. Sentences in a scenario of a TV program and a log of chat are already arranged in sequence. However, in order to serialize them, text segments in netnews articles must be ordered, and the ordering is not obvious.
2. Relations in a scenario of TV program and chat be-



Hello, I think that it's hard to read the articles in netnews. Especially, it tend to forget who state that or what he or she sate.

I feel the same way. Looking the signature or the from, we can know the author, but shortly forget that.

Sometimes, I noticed that somebody writes "I did't write so !". I am sure that he thinks the same matter too.

As one article had followed by many persons, it's hard to match each authors and their comments. As a result, FLAME !!

Figure 5: relation of speech in netnews articles

tween speeches can be understood naturally by reading or listening to them. However, in USENET articles, relations between text segments are expressed by quotes, and it is difficult to understand them.

2.2 Serialization of text segments

To generate a scenario from a thread of news articles, we must consider the two features mentioned in the previous section.

As for the first point, text segments in netnews articles must be ordered while keeping the two kinds of relations extracted from articles in a thread; these are quote relations and the order of text segments in the original article.

R1 The text segments quoted by some articles should be placed before the text segments in the articles that quoted the text segments. In addition, the quoted text segments and the segments in the articles quoting them should be arranged as close together as possible.

R2 The order of text segments in one article should be kept.

For example, in Figure 2.2, the following constraints must be kept.

- $time_{A-1} < time_{B-1}$, $time_{A-2} < time_{B-2}$
- $time_{A-1} < time_{A-2}$, $time_{B-1} < time_{B-2}$

The result of serialization of all text segments in Figure is a text consisting of A-1, B-1, A-2, B-2 in sequence. In these two rules, the quote relation (R1) has precedence over the text segment relation (R2). This is because a relation by a quote is explicit and is hard to understand from other information (e.g., contents).

When several articles quote one text segment, these quote relations form a branching structure. Another serialization technique is necessary for serializing these articles. Consider the serialization of articles in Figure 6. This sample shows that two articles (B and C) quote A-1 from article A. Article D quotes B-1 from article B.

According to the above priority rule, the following constraints should be satisfied.

1. $time_{A-1} < time_{B-1}, time_{C-1}, time_{D-1} < time_{A-2}$
2. $time_{B-1} < time_{D-1}$
3. B-1 and D-1 must be placed as close as possible.

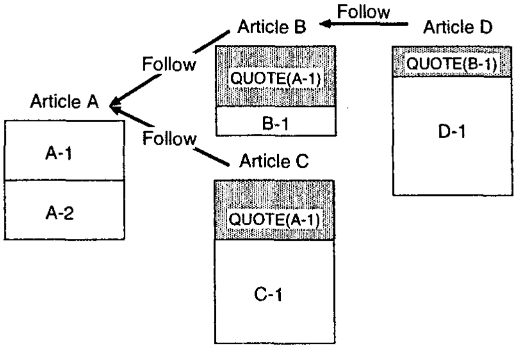


Figure 6: Branching points

4. A-1 and B-1 must be placed as close as possible.
5. A-1 and C-1 must be placed as close as possible.

To satisfy these constraints, the following two choices are possible.

- $A-1 \rightarrow (B-1 \rightarrow D-1) \rightarrow C-1 \rightarrow A-2$
- $A-1 \rightarrow C-1 \rightarrow (B-1 \rightarrow D-1) \rightarrow A-2$

Although these choices fully satisfy constraints 1, 2, and 3, neither of the two constraints 4 and 5 is satisfied. Which choice has higher priority should be determined based on the meaning of articles. However, it is hard to examine the meaning of articles. We therefore introduce an additional priority rule about serialization at branch points.

R3 If there are text segments that are quoted by multiple articles, the shorter text segment has priority. If that segment is also quoted, the length of all subsequent text segments must be counted.

This rule can reduce the sum of all gaps between a quoted text segment and the text segments quoting it. In other words, it is a supplementary rule to R1.

In Figure 6, the shorter text segments from $B-1 \rightarrow D-1$ and $C-1$ have priority. Assuming that B-1 has 3 lines, C-1 has 10 lines, and D-1 has 15 lines, the system generates a serialized text with $A-1 \rightarrow C-1 \rightarrow B-1 \rightarrow D-1 \rightarrow A-2$. (Figure 7 top), because C-1 is shorter than $B-1 \rightarrow D-1$.

The serialization technique described above has a problem. The generated script does not have information about several relationships between text segments in their original article. In the above example, text segments B-1 and A-2 are not in relation to the text segment just before them (C-1 and D-1), but both are in relation to the earlier text segment A-1. If the distance between related text segments is not very long, the relation can be understood easily, although as the distance between related text segments increases, this problem becomes worse.

In a TV program scenario or chat log, if the relation is not clear, a chairperson explains it explicitly (e.g., "Let's resume where we left off."). Our system places such explanations into the script in the following style.

- If the length of the gap between two related text segments exceeds a certain limit value, the system adds explanations to the script. The explanations are inserted just after the first text segment and also just before the second text segment.

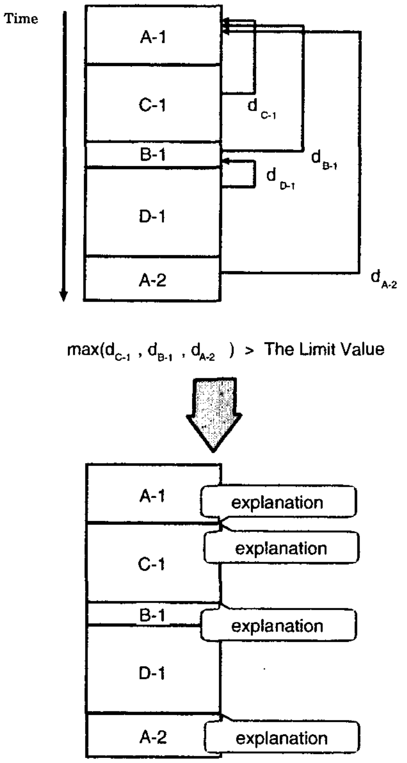


Figure 7: Result of serialization(top), Insertion of explanations (bottom)

For example, when the length of d_{C-1}, d_{B-1}, or d_{A-2} exceeds a certain limit value (Figure 7), the system adds some explanations such as those in Table 1.

Table 1: The points where the chairperson's explanations are inserted.("CP: ..." are the chaiarperson's comments)

Time	Text Segment	Related to
0	A-1	-
CP:"Although (A-1) seems to want to continue ..."		
CP:"Let's talk from (C-1)"		
1	C-1	A-1
CP:"(B-1), Please, talk about (A-1)'s point .."		
2	B-1	A-1
3	D-1	B-1
CP:"(A-2), Please, resume what you were talking about"		
3	A-2	A-1

3. AVATAR'S ACTIONS, STAGE SETTINGS, AND CAMERA CONTROL

In this section, we describe how to represent discussions as animations of 3D avatars. The two important issues are (1) camera control and (2) avatar's actions. We propose two kinds of representation, a basic representation and a representation that shows conversation patterns during discussion. The first one represents each avatar's role such as listener, speaker, and third party by the directions of their faces and also by their actions. The second one aims to explicitly represent typical patterns of conversations. In the following sections, we explain these two representations.

3.1 Basic representation

The purpose of this representation is to help users recognize the role of the participant,(speaker, listener, or third party) by camera control and avatar's actions. This classification of roles in discussion (speaker, listener, third party) was originally used in Inoue's research[6]. Basically, we assign an avatar for each author in a news thread. Avatars repeat their parts in a script in turn. When an avatar is talking, we assign the following roles to the avatars.

Speaker The talking avatar that corresponds to the author of the text segment simply talks.

Listener The avatar that corresponds to the author of the article referred to by the current speaker.

Third Party The other avatars.

The role of each avatar is represented in the following way.

- The camera always aims at the speaker. When the speaker changes, the camera pans to the new speaker.
- If there is a listener, the speaker looks at him or her. If not, the speaker looks to the front.
- All avatars (except the speaker) look at the speaker.

Figure 8 illustrates a scene of a discussion that uses the above representation techniques. Users can recognize each avatar's role by watching this scene. The camera aims at the speaker so that the speaker is always positioned at the center of this scene. In addition, all participants except the speaker look at the speaker. Therefore, the avatar at the middle position is the speaker. The speaker looks at the listener, that is, the avatar at the left position is the listener. The other avatars are third parties.

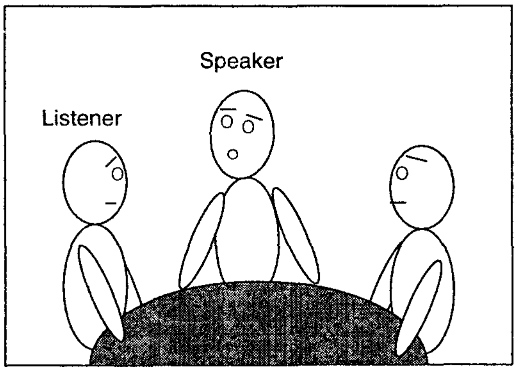


Figure 8: Basic Representation

3.2 Representing conversation patterns

In real discussions, there are various conversation patterns. For example, there is a pattern with two participants talking to each other in turn. Another pattern is for many participants to reply to one participant's speech. Some conversation patterns can be found in the discussions in netnews as well. For example, it is usual that one author's comment is problematic and many authors post objections against the article.

Our system tries to represent these conversation patterns. There are two advantages in representing conversation patterns. Firstly, conversation patterns are helpful for readers to understand the relations between each speech in a discussion. Secondly, it may supplement information lost during the serialization of articles. Although the inserted explanations about relationships between text segments and basic representation method show those relationships, conversation patterns can also help show these patterns.

3.2.1 Extraction of conversation patterns

The extraction of conversation patterns from a generated script is difficult, because it fundamentally requires the system to analyze scripts semantically. Therefore, we simply extract conversation patterns using quote relations and authors.

Our system can extract two conversation patterns, the *Two participants' conversation pattern* and the *Followed by many persons pattern*. In the following, we describe how to extract these patterns from articles in a thread, and how to represent them on the screen.

3.2.1.1 Two Participants' Conversation Pattern.

In real discussions, there are cases where two persons talk to each other. For example, there are conversations that consist of one person's questions and the other person's answers. In another type of conversation, one person speaks an affirmation and the other speaks a negation in turn as in a debate.

This pattern appears in netnews as a sequence of follow-up articles posted in turn by two persons. We analyze this pattern more strictly by handling a text segment as a unit of speech and making a table as shown in Table 2. This table shows each change of speaker and listener in an example script, and also shows the ID of the text segment being spoken by the speaker at the time and the ID of the related text segment.

Table 2: Example: Two participants' conversation by author A and B

time	speaker	listener	segment	relates to
0	A	-	0-1	-
1	B	A	1-1	0-1
2	A	B	2-1	1-1
3	B	A	3-1	2-1
4	A	B	2-2	2-1
⋮	⋮	⋮	⋮	⋮

In this example, authors A and B talk to each other in turn from time 0 to time 4. In addition, they refer to the speech just before their speech (time 1 to time 3). At time 0, author A speaks to all participants, so there is no listener at time 0. The speech of author A at time 4 follows speech 2-1. In this case, only the sequence from time 1 to time 3 can be regarded as a pattern. In general, when conversation consists of two persons' speeches to each other, and each speech follows the text segment just before, we assume that the conversation forms a two participants' conversation pattern.

In order to represent this pattern, the system should show only the speaker and the listener, and third parties should not appear in the representation of this pattern. That is, we represent a conversation of this type as a 'two-shot' of the speaker and the listener, as in Figure9.

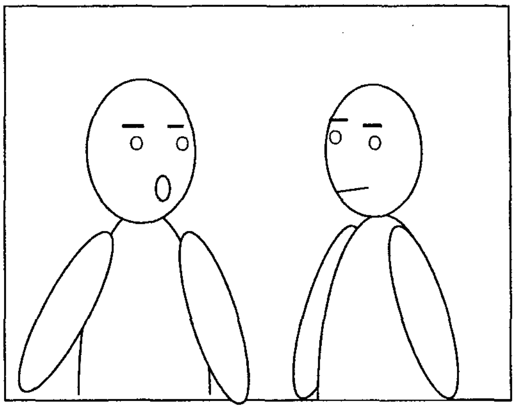


Figure 9: Conversation with two participants

In Figure 9, two participants are positioned at the left and right end of a screen. The speaker and the listener can be distinguished by watching avatars' motions such as nods. The system zooms the camera so that only two avatars appear on the screen. This helps users recognize that the current conversation forms the two participants' conversation pattern.

3.2.1.2 Followed by many participants.

When an article attracts the interest of a number of readers, a number of articles are posted following the original article. However, they do not necessarily refer to the same text segment in the article, but may follow different parts of the article. Our system can automatically judge that the following articles refer to the original article or not by analyzing relationships between text segments, and the system can represent that a speech is being followed by many individuals.

Furthermore, the representation of this pattern has a role in presenting branching information lost during serialization. Although a chairperson can explain branching information, our system currently uses the chairperson only when the gap between related text segments exceeds the limit. We propose a method for presenting branching information in another way.

The same table described in the previous section is also used to extract this pattern from a script. Table 3 suggests the situation where the authors B, C, and D speak to author A's speech 0-1. Whether or not the chairperson's explanations are inserted by the system depends on the length of each text segment 1-1, 3-1, 4-1 written by B, C, and D, but the system always represents conversations of this type in the following way. Firstly, the system changes the camera position so that all avatars in the scene appear in the screen(Figure 10). Secondly, the other avatars act as if they wish to reply to the first speaker. To put it concretely, they wave or raise their hands. By this representation, the system can show the following information.

- Which text segment is followed by many participants.
- Who is following the text segment.

Table 3: Example: Author A's 0-1 is followed by B, C and D

time	speaker	listener	segment	relate to
0	A	-	0-1	-
1	B	A	1-1	0-1
⋮	⋮	⋮	⋮	⋮
5	C	A	3-1	0-1
6	D	A	4-1	0-1
7	A	-	0-2	-
⋮	⋮	⋮	⋮	⋮

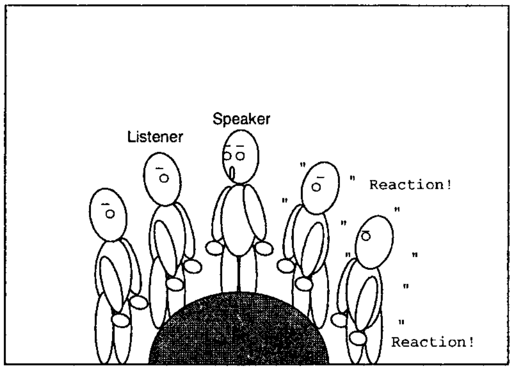


Figure 10: Followed by many participants

4. EXAMPLE

Figure 12 is the example animation generated from a thread[1] that has six articles and four participants. Each article's relationship by references are shown in Figure 11.

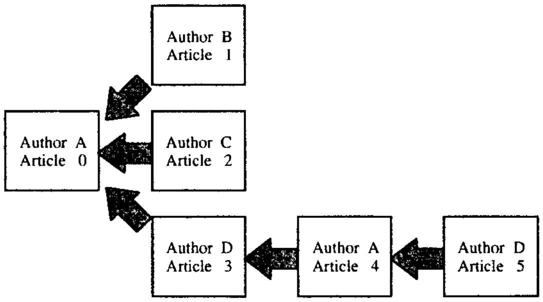


Figure 11: Example: the relationship of the articles

5. DISCUSSION

We have applied the prototype system to some news threads samples, which were extracted from USENET archives. These samples were taken from news groups for discussion such as fj.living and fj.soc.*. There are 10 to 15 articles and five to eight authors in each thread. Most of the scenarios generated by the prototype system are easier to read than the original articles. This is because we can read articles with less 'backtracking' of topics, and there are no quotes in the scripts. The most typical examples that cause 'backtracking' of topics are where the original article has two topics. One is the main topic, and the other is a 'PS', which often is written at the end of the article. It is originally an additional topic, but often becomes so interesting that it is followed by other authors. In such cases, the scenarios generated by our serialization technique were easier to read because the user can read each topic in turn. Moreover, the scenario without quotes also lightens the burden imposed on the reader. We found that the sizes of scenarios were from 40 to 60 percent of the total size of all articles in the thread on average (these did not include the sizes of signatures).

However, there are also some undesirable cases. Our serialization policy gives higher priority to relationships by quotes than to relationships by sequence in the same article. This policy sometimes causes a problem, because even if the relation between two text segments in one article is most significant, they are split and placed at quite separated positions. A typical example of this problem is where multiple quotes from items and lists are used. In this case, the items are placed at quite separated positions, although they are often in strong relation to each other. To solve this problem, semantic analysis is required.

Our approach is to generate animations with 3D avatars in order to easily remember the authors of articles. The users can use information about the authorship of articles to infer the context of the articles. However, in our current implementation, we can only use seven characters because of the limitation of the TVML player. Therefore, the system may allocate one character to multiple authors, which confuses the viewers.

This problem can be easily solved by increasing the number of characters usable in TVML. However, since it is difficult to provide enough avatars to allocate a unique avatar to all authors of articles in all USENET newsgroups, a way to reuse already allocated characters may be necessary.

Our representation method is effective in informing viewers about the current state of a discussion, such as the current role of each avatar and relationships between speeches. Moreover, the representations of conversation patterns are effective in suggesting the relationships between text segments, and can make up for the branching information lost by serialization.

By using other types of representations, more information can be provided to users. For example, it may be possible to represent an author's emotion by changing avatars' faces and actions. Face marks and fixed forms of sentences in articles may be useful for extracting emotions. It is also possible to represent other types of conversation patterns such as many vs. many or affirmation vs. negation.

6. RELATED WORK

In 'visual chat' (Comic Chat[8], Palace[2], etc.) or 'virtual society' (Community Place[9; 10], FreeWalk[11], etc.), remembering each author and statement is easy because the systems use characteristic avatars or snapshots that are allocated to each author. Using avatars is a natural and effective method because the participants or observers remember the statement and the speaker by relating the speaker's figure, voice and features in face-to-face discussions. However, those systems are synchronous discussion systems, and there are no asynchronous systems because it is difficult to

[1] We applied the prototype system to several real news threads, but cannot include them in this paper because of their copyrights. We therefore wrote samples of articles and applied the prototype system to them.

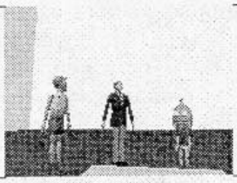
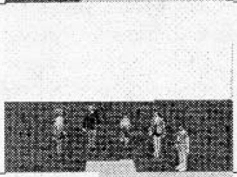
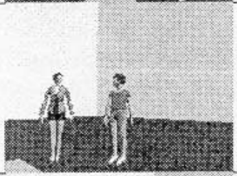
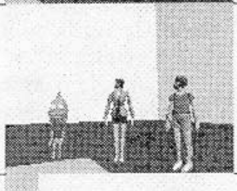
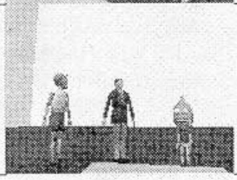
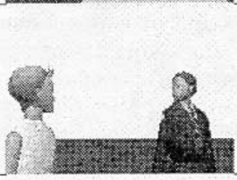
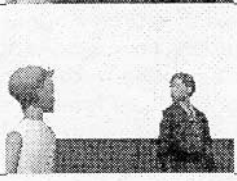
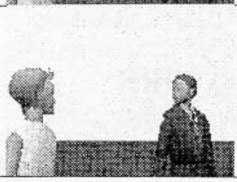
Time	Screen	Speaker: speech / (Explanation)
0		A: *"Hello, I think that it's hard to read netnews articles. There are many useful and useless articles."* (The center avatar is the author A and is speaking.)
0		A: *"So it's hard to find the useful information."* (In the middle of A's speech, two participants (right two avatars)start waving. This action suggests that these authors wish to reply to A's speech.)
1		B: *"I agree. We need a good filtering tool."* (Firstly, B follows A's speech.)
2		C: *"I think all articles in netnews are garbage. Of course, this is my article too. :-p"* (Secondly, C also follows A's speech .)
3		A: *"And unlike face-to-face communication, it's hard to remember who said what. Even though the author's name is described in the header or signature, I quickly forget."* (A starts to speak the rest of the previous speech.)
4		D: *"I think that this is one reason for flames."* (D speaks to A and the camera catches A and D in a two-shot because A and D repeatedly speak to each other from this time.)
5		A: *"I think so. I often notice authors who write 'I never wrote that!'."*
6		D: *"They must have been misunderstood."*

Figure 12: An Example of a generated animation

generate an animation of discussion from an asynchronous discussion with branches made from references and quotes. There are also systems that use an avatar as an expressive reporter. PPP Persona[13; 1] is a system that presents multimedia material to the user, using a lifelike character. Peddy[7] is an agent-based interface, and the agent has a rich repertoire of animated bird-like and human-like behaviors. The system helps implementation of applications with high powers of expression by using an avatar. These techniques make use of the avatar's expressive power. Our research also makes use of it, not only with a representation of a single avatar, but also with multiple avatars. The employment of multiple avatars can show more complicated information (e.g., The conflict of multiple opinions), and can apply to user interface agents.

There are also some systems based on film grammars or camera controls on TV programs to visualize avatars' conversation effectively (Virtual Cinematographer [15] and Harada's research [4]). These systems automatically change the viewpoint to the appropriate points according to such events as change of speaker, times of speech and character actions.

7. CONCLUSION

We have proposed a technique for generating more comprehensible animations from discussions in USENET, which are often hard to follow. We have also implemented a prototype system to prove the effectiveness of our technique and applied it to some threads extracted from USENET archives. There are several topics for future work, such as the generation of more natural chairperson's comments such as explanations of context, the generation of more attractive animations, and quantitative evaluations.

8. REFERENCES

[1] E. Andre, T. Rist, and J. Muller. Guiding The User Through Dynamically Generated Hypermedia Presentations with a Life-Like Character. In *IUI'98*, pages 21–28, 1998.

[2] Communities.com. *The Palace Viewer User Guide verion 1.0*, 1999.

[3] M. Douke, M. Hayashi, and E. Makino. A Study of Automatic Program Producing Using TVML. In *Eurographics'99*, pages 42–45, 1999.

[4] Y. Harada and M. Nakamura. Multimedia System with Film Grammar. In *WISS'95*, pages 1–8, 1995.

[5] M. Hayashi, H. Ueda, and T. Kurihara. TVML(TV program Making Language) - Automatic TV Program Generation from Text-based Script -. In *Imagina'99*, 1999.

[6] T. Inoue, K. ichi Okada, and Y. Matsushita. Learning from TV programs : Application of TV presentation to a videoconferencing system. In *UIST'95*, pages 147–154, 1995.

[7] D. Kurlander and D. T. Ling. Planning-Based Control of Interface Anmation. In *CHI'95*, pages 472–479, 1995.

[8] D. Kurlander, T. Skelly, and D. Salesin. Comic chat. In *SIGGRAPH'96*, pages 225–236, 1996.

[9] R. Lea, Y. Honda, K. Matsuda, and S. Matsuda. Community Place: Architecture and Performance. In *VRML97*, pages 41–49, 1997.

[10] K. Matsuda, Y. Honda, and R. Lea. Virtual Society: Multi-user Interactive Shared Space on WWW. In *ICAT 96*, pages 83–95, 1996.

[11] H. Nakanishi, C. Yoshida, T. Nishimura, and T. Ishida. FreeWalk: Supporting Casual Meetings in a Network. In *CSCW'96*, pages 308–314, 1996.

[12] E. Renninson. Galaxy of news - an approach to visualizing and understanding expansive news landscapes. In *UIST'94*, pages 3–12, 1994.

[13] T. Rist, E. Andre, and J. Muller. Adding Animated Presentation Agents to the Interface. In *IUI'97*, pages 79–86, 1997.

[14] K. Uchimoto, H. Isahara, and H. Ozaku. Method for identifying topic-changing articles in discussion-type newsgroups within the intelligent network news reader hisho. In *NLPRS'97*, pages 375–380, 1997.

[15] L. wei He, M. F. Cohen, and D. Salesin. The virtual cinematographer : A paradigm for automatic real-time camera control and directing. In *SIGGRAPH'96*, pages 217–224, 1996.