データベースシステム 119−32
（１９９９．７．２２）

Virtual TV Channel: Filtering, Merging and Presenting Internet Broadcasting Channels

MA QIANG ,[†] HIROYUKI KONDO ,[†] KAZUTOSHI SUMIYA [††]
and KATSUMI TANAKA[††]

Broadcast-based information dissemination systems on the Internet are becoming increasingly popular due to advances in the area of web technology and information delivery. In this paper, we propose a concept and a way to construct a *virtual TV channel*, which is a user-defined (virtual) channel from existing push-based Internet channels. Our virtual TV channel is (1) to filter contents of multiple push-based Internet channels, (2) to merge selected articles from different channels, and (3) to present them by a TV-program-like GUI.

1. Introduction

Recent years, broadcast-based (or push-based) information dissemination systems on the Internet are becoming increasingly popular due to advances in web technologies. These systems, such as *Pointcast*[1)2)], *Backweb*[3)], use the push-based technology to disseminate information for users instead of traditional pull browsing paradigm.

In this paper, we propose a personal on-line news broadcasting system *Virtual TV Channel* with new filtering/synthesizing and presentation approaches for information broadcast systems. The virtual TV channel is one broker that searches the information matched with user information needs, and synthesizes them as one virtual channel. The information are also estimated by the similarity with a given user profile, update frequency of a channel, an article's freshness compared with previously selected articles and so on.

The main contributions of this paper can be summarized as follows:

- **Information filtering by user interest, channel update frequency, and information popularity/freshness**
  We propose a new filtering approach that estimate information from both user interests (user profiles) and *information status.* In order to select an information from each channel, we estimate not only the similarity between an article and a user profile, but also how often the article's channel is updated (channel update frequency) , the article's popularity (how much the article is similar to previously selected articles), and the article's freshness (how much the article is different from previously selected articles).
- **Analog-like channel definition to merge different channels**
  The selected articles per each different channel are merged into one virtual channel. In our approaches, it is possible to merge those articles in an *analog manner.* That is, we can specify the merging ration per each channel. For example, we can say *merge contents of channels X, Y, and Z according to the ration 0.2, 0.3, and 0.5.* In the merging phase, each article is also compared with the previously selected articles which has been sent to user. If necessary, the comparison result is fed back to guarantee the given merging ratio.
- **TV program-like GUI for viewing virtual channel content**
  The articles of virtual channel are presented like TV programs. In contrast to typical text-based presentation model, we give a new present model using TVML[5)6)], that is, based on the feature of contents etc., an appropriate TV-program metaphor(news program, drama, entertainment show, and so on.) is automatically selected, and users can enjoy those articles as a TV-program.

The remainder of this paper is organized as follows: In Section 2, we overview the concept of the *virtual TV channel.* The filtering phase is described in Section 3. In Section 4, we present

[†] Department of Computer and System Engineering, Kobe University
[††] Division of Information and Media Sciences, Graduate School of Science and Technology, Kobe University

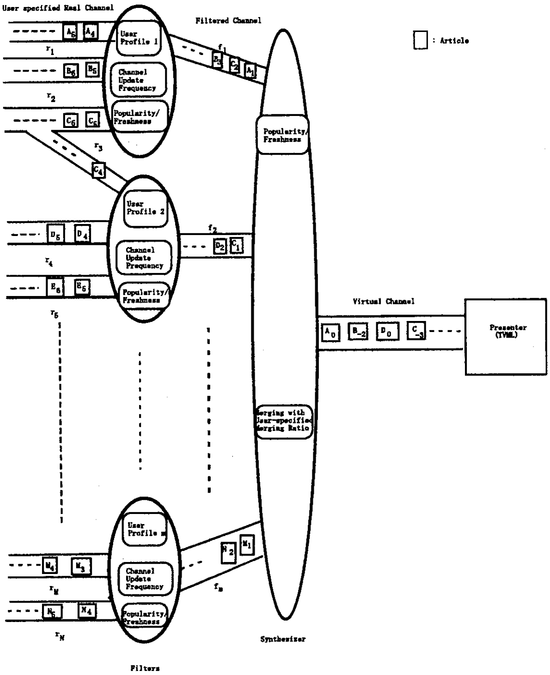


Fig. 1 Concept of Virtual TV Channel

the merging process, which produces a virtual channel. In Section 5, the presentation of a virtual channel using TVML is described. A prototype system is reviewed in Section 6. In Section 7 we discuss previous related works. Section 8 presents our conclusion.

2. Concept of Virtual TV Channel

As shown in **Fig. 1**, our *virtual TV channel* consists of *filters*, an *synthesizer*, and a *presenter*. Basically, a user of the virtual TV channel gives the following information:

- Real channels of his interest.
  User-selected real channels are organized into several groups according to their genres etc.
- A user profile per each real channel group, and
- A merging ratio for filtered channels

For each real channel group, a filter behaves as a broker which filters information of the channel according to a given user profile. A synthesizer merges those filtered information as a virtual channel by a user-specified merging ratio. Then, the information of the virtual channel are presented in the form like a TV program by the *presenter*.

In our system, we mainly handle news articles that are disseminated by push-based channels. So, hereafter, we say *articles* as an information unit of a channel.

In the phase of filtering articles, the process is done per each real channel group. That is, each article in a real channel is compared with the corresponding user-profile, and the similarity of the article with the user-profile is computed. Basically, an article with higher similarity with the user-profile has a high possibility to be selected. Also, the update frequency of the real channel that the article belongs to is monitored and it is also considered as a criterion to select the article. In most push-based systems, the contents of a real channel are periodically updated. But, update frequency of some content is unpredictably changed because of urgency. An article whose channel has high update frequency has a high possibility to be selected. Furthermore, we compute the similarity/dissimilarity of an article with some previously selected articles are also computed for estimating the *popularity* and the *freshness* of the article. In summary, if the article has high similarity with user profile, high update frequency, and high popularity or freshness, then the article is selected as a candidate content of the virtual channel.

After the above filtering process is over, we have multiple series of candidate articles in filtered channels. The synthesizing process is to select articles from those candidates and merge them as the contents of the virtual channel. In this synthesizing process, the system automatically merges selected articles from each filtered channel by user-specified merging ratio. The selection of the articles to final virtual channel is also based on the similarity/dissimilarity of the article with previously selected articles.

Finally, in the presentation phase, the articles of virtual channel are presented in the form like TV programs. Based on the feature of contents, etc., an appropriate TV-program metaphor(news program, drama, entertainment show, and so on.) is automatically selected, and users can enjoy those articles as a TV-program.

3. Filtering Process

Filtered channels consist of articles selected from each real channel group. The filter for each filtered channel has three functions: (a) user profile matching, (b) channel update frequency monitoring, (c) popularity and freshness calculating. For each filtered channel, the real channel group and the user profile are specified by a user. When the user profile for the filtered channel is q, the score of article a via channel c is calculated by following equation:

$$fc_{score}(a) = \alpha * sim(a,q) + \beta * freq(a) + \gamma * popfresh(a) \quad (1)$$

where sim(a, q) is the similarity between article a and user profile q, freq(a) is the update frequency of channel c which article a belongs to, popfresh(a) is the popularity/freshness of article a against previously selected articles in the filtered channel (α, β, and γ are weight values for each term).

(a) user profile matching

The similarity of article a is calculated by inner product of article's keyword vector k_a and user profile's keyword vector k_q.

$$sim(a,q) = \frac{k_a * k_q}{|k_a||k_q|} \quad (2)$$

(b) channel update frequency monitoring

The update frequency of the channel that the article belongs to is monitored using its update duration. Each real channel generally has its default update duration. However, in some cases, the duration is shorter than the default duration because of urgency. For example, the channel of weather forecast would change their update frequency when typhoon warnings are announced. In this case, the default update duration is six hour and the urgent update duration would be one hour. The frequency of channel c which article a belongs to is evaluated as follows:

$$freq(a) = e^{\lambda_1 \sigma_c} \quad (3)$$

$$\sigma_c = \frac{D_c - d_c}{d_c} \quad (4)$$

where σ_c is the ratio of update frequency of channel c, D_c is the default duration of channel c, d_c is the latest duration of channel c, and λ_1 is a weight value.

(c) popularity and freshness calculating

In order to select *valuable* articles from the real channel groups, the similarity and dissimilarity of the article to previously selected articles should be also evaluated. Articles that are quite similar to almost of the previously selected articles would be also valuable. For example, when an incident happens, the series of the report articles would be sent continuously. That is, the articles are one of the *hottest* information at that time.

On the other hand, articles that are quite different from previously selected articles would be also valuable. In other words, we can say that the articles have their freshness and uniqueness. That is, the articles may be *scoop* news. To calculate popularity and freshness of articles, we use the similarity ratio, which is computed by the number of the similar articles in the previously selected articles. If the number of articles in filtered channel buffer is m, the similarity ratio is basically defined as follows:

$$k = \frac{n}{m} \quad (5)$$

where n is the number of the similar articles in the filtered channel. If k is considerably high, we can say that the popularity of the article should be evaluated. On the contrary, if k is very low, we can say that the freshness of the article should be evaluated. Therefore, popularity/freshness of article a is calculated by following equation:

$$popfresh(a) = \max(e^{\lambda_2 k}, e^{-\lambda_3 k}) \quad (6)$$

where λ_2 and λ_3 are weight values.

In our approach, the estimation of popularity and freshness are measured with same ratio. This approach is considered as a heuristic way. However, the characteristic of popularity and freshness should be mixed☆.

4. Synthesizing Process

After the filtering processes are over, we have some filtered channels which is a series of candidate articles for virtual TV channel. The *synthesizer* selects articles from those candidates and merge them as the contents of the virtual TV channel.

In short, the synthesizer computes the score of these candidate articles and selects the high score articles to virtual TV channel. The score of candidate article a is calculated by following equation:

$$vc_{score}(a) = p_{fc}(t_i) * PopFresh(a) \quad (7)$$

where p_{fc} is the current time priority of filtered channel which a belongs to, PopFresh(a) is the popularity and freshness of a against previously selected articles in the virtual TV channel.

(a)priority of filtered channel

In order to guarantee the user specified merging ratio, the priority of a filtered chan-

☆ It's possible that we can calculate popularity using the word frequency, freshness using tf/idf.

nel, i.e., the priority of the all articles in the filtered channel, should be adjusted by the its user specified ratio and the actual proportion of the filtered channel in virtual TV channel. Therefore, the current time priority of the filtered channel fc that the article a belongs to is calculated by the following formula:

$$p_{fc}(t_i) = \frac{r_{fc}}{ap_{fc}(t_{i-1})} * p_{fc}(t_{i-1}) \qquad (8)$$

where the r_{fc} is the user specified merging ratio of fc, $ap_{(t_{i-1})}$ is the actual proportion of fc, and the t_{i-1} means the latest iteration.

(b)popularity and freshness

To calculate the popularity and freshness of article a, which is the candidate contents of virtual TV channel, we use the similarity ratio, which is computed by the number of the similar articles in virtual TV channel. The similarity ratio is defined as follows:

$$K = \frac{N}{M} \qquad (9)$$

where the N is the number of similar articles of article a in the virtual TV channel, and the M is the number of articles in the virtual TV channel buffer. In contrast to the similarity ratio of filtering phase, which estimates the m and n per each filtered channel, at the synthesizing phase, the M and N are estimated in the virtual TV channel.

Therefore, popularity/freshness of article a is calculated by the following equation:

$$PopFresh(a) = \max(e^{\lambda_4 k}, e^{-\lambda_5 k}) \qquad (10)$$

where λ_4 and λ_5 are weight values.

5. Contents Presentation By TV-program Metaphor

In this section, we review our TV-program metaphor presentation system for the virtual TV channel.

5.1 TV-program Metaphors

We have defined five TV-program metaphors to present the articles of virtual TV channel, where a TV-program metaphor is a pseudo-TV-program described by TVML[5]. The defined TV-program metaphors are listed in the **Table 1**.

5.2 Selection of a TV-program metaphor

TV-program metaphors can be manually selected by a user according to his interests. For example, the *Headline* metaphor seems to be suitable to summarize a vast of articles. On the other hand, the *News* metaphor seems to be more suitable to present less number of articles with high importance. When the articles have high similarity with each other, the *Debate* metaphor seems to be suitable since the presentation can focus on a specific topic.

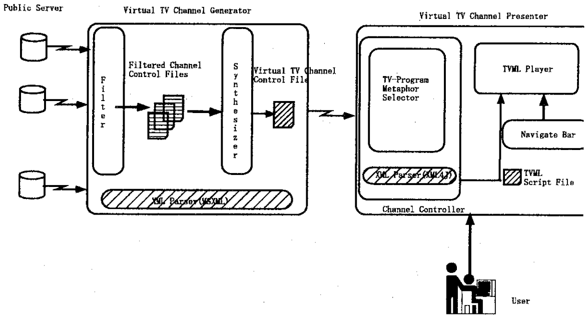

Public Server
Virtual TV Channel Generator
Filter
Filtered Channel Control Files
Synthesizer
Virtual TV Channel Control File
XML Parser(MSXML)
Virtual TV Channel Presenter
TV-Program Metaphor Selector
XML Parser(XML4J)
Channel Controller
TVML Player
Navigate Bar
TVML Script File
User

Fig. 2 System Architecture of Muffin

In this subsection, we describe the ways to automatically select an appropriate TV-program metaphor for the articles of virtual TV channel[7].

The one is to select a TV-program metaphor from the distribute status of articles. The basic idea is as follows:

- Cluster the articles according to their features, and then
- Analyze the clusters (the number of clusters, the size of each cluster etc.) , and
- Select an appropriate TV-program metaphor for the obtained clusters of articles.

The another one is to select the appropriate TV-program metaphor for virtual TV channel articles based on the feature of their contents. At first, we extract the keywords of a TV-program metaphor from the summaries of sample TV-program, and map each article to a TV-program metaphor with the keywords of the article and metaphor.

In addition, the attributes of an article(title, characters and so on) are also considered when selecting a TV-program metaphor in our approaches. For example, when a user just wants to know the title of articles, the *Headline* metaphor will most possibly be selected. On the other hand, when a user is interested in an actor(actress), the *Drama* metaphor is more reasonable.

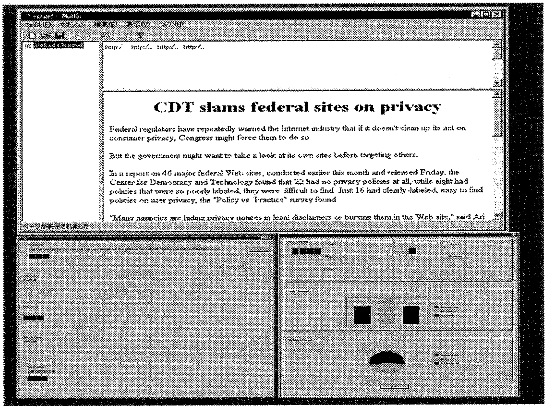
6. Prototype System

We are developing a prototype system for the virtual TV channel, called *Muffin*. As illustrated at **Fig. 2**, *Muffin* has a 3 tiers structure,

(1) Public Server
(2) Virtual TV Channel Generator

Table 1 TV-program metaphors

TV-program metaphor	Actions
News	A newscaster character and a sub-newscaster character argue some article
Headline	Headline news program style. In this metaphor, a newscaster just summarize the news without giving any comment
Debate	One theme is discussed by some characters
Drama	A drama style program that represents a series article
Entertainment show	The articles are reviewed as an entertainment, comic etc.



http://.. http://.. http://.. http://..

CDT slams federal sites on privacy

Federal regulators have repeatedly warned the Internet industry that if it doesn't clean up its act on consumer privacy, Congress might force them to do so

But the government might want to take a look at its own sites before targeting others.

In a report on 46 major federal Web sites, conducted earlier this month and released Friday, the Center for Democracy and Technology found that 22 had no privacy policies at all, while eight had policies that were so poorly labeled, they were difficult to find. Just 16 had clearly-labeled, easy to find policies on user privacy, the "Policy vs. Practice" survey found

"Many agencies are hiding privacy notices in legal disclaimers or burying them in the Web site," said Ari

Fig. 3 An Execute Example of Virtual Channel Generator

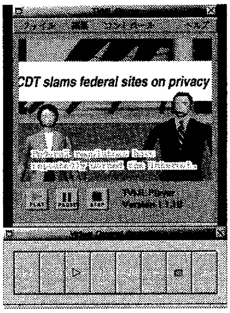


Fig. 4 An Execute Example Of Virtual Presenter

(3) Virtual TV Channel Presenter

The *public server* means the data sources on the Internet, and corresponds for a real channel. In our implementation, we assume that the broadcast articles are formatted with XML(eXtensible Markup Language)-the universal format for data on the Web.

The *Virtual TV Channel Generator* is the core of our prototype system. At first, the *filter* ranks the broadcast information of *public server*, and constructed the higher rank articles to filtered channels, and use the XML formatted *filtered channel control files* to describe the filtered channels. Secondly, the *synthesizer* ranks articles of filtered channel, and add the higher rank articles to the *virtual TV channel control file*. The *MSXML*[8] is used to parser the source articles, the *filtered channel control files* and the *virtual TV channel control file*.

The *synthesizer* analyzes the *virtual TV channel control file* as a feedback to itself in order to adjust the priorities of filtered channels, which is used to guarantee user-specified merging ratio.

We are developing the *Virtual TV Channel Generator* on a Windows NT Workstation machine, the Microsoft Visual Studio Enterprise Edition Version 6.0 and MSXML version 2.0 have been used. **Fig. 3** is an executed example of the current version *virtual TV channel generator*. In Fig. 3, the top screen is the image of *muffin*'s browser, the bottom left screen is the image which shows the behavior of *virtual TV channel generator* as an animation, and the bottom right screen is the image of the proportion monitor. The left window of browser is the channel control, the right top window is the link viewer which list the link information of the viewing article, and the right bottom window is the article viewer.

The *Virtual TV Channel Presenter* is composed of the *Channel Controller*, *Navigate Bar*, and the *TVML player*. At first, The *channel controller* receives the *virtual TV channel control file* from *virtual TV channel generator*. Then the *TV-program metaphor selector* parses the *virtual TV channel control file*, and creates a scenario: *TVML Script File*. Finally, a TV program, which represents the articles of virtual TV channel, is presented with the *TVML Player*. The *Navigate Bar* can be used to control the TVML Player just like the remote-control of TV.

The *Presenter* has been developed on a SGI O2 machine with JAVA2, XML4J and the TVML Player. **Fig. 4** is an executed example of the current version *presenter*.

7. Related Work

Pointcast[1)2)] is a dissemination service that has attracted a large population of users. It obtains profiles from users in which they subscribe their interest channels (sub-channels), and then uses these profiles to assemble and update customized "newspaper" from a database of current stories. As the customizing of a user is limited to add(remove) the server defined channels(sub-channels), the user interests would not be well represented. In other word, the Pointcast is a more popular information service than personal.

On the other hand, SIFT[4], a tool developed for wide-area information dissemination at Stanford University, combines data management ideas from information retrieval with a public/subscribe model for dissemination. The approach taken by SIFT, which uses the publish/subscribe model for wide-area information dissemination, requires users to explicitly submit their profiles and update those profiles using relevance feedback. SIFT can answer for the complex interests of a user well, but the omission of information status, update frequency, dissimilarity etc., may raise a problem that miss some user needs(both interesting and importance) information.

Shapiro has developed the techniques that can create "channels" by dynamically filtering the large-scale web contents[9]. Its goal is to change the way information is presented to users from the traditional Web pull-based browsing model to a push-based channel paradigm. But, it also ignores the information status, and creates channels directly form retrieval results without restructuring.

The ANATAGONOMI[10] is a novel push-type news delivery system that use the user behavior as a feedback to make an automatic presentation of user interesting information. The key idea of them is that the user behaviors can be accumulated as user profile and present the relevant information for user automatically. Another work, which has been done at Incubation Center of NEC corporation, developed a Theater-style Web Browsing System *SiteCruise Theater*[11]. In SiteCrusit, Web information are classified by category or theme and provided just like a movie or TV program automatically.

8. Conclusions

In this paper, we have proposed a personal on-line news broadcast system virtual TV channel with new filtering, restructuring and presentation approaches. The scheduling problem of virtual channel is untouched in this paper. We'll do some work on it as our future work. Also, the verification of virtual TV channel generation, and TV-program metaphor selection is also planned to do.

Acknowledgments

This research is partly supported by the Japanese Ministry of Education Grant-in-Aid for Scientific Research on Priority Area (A):"Advanced databases," area no. 275 (08244103). This research is also supported in part by "Research for the Future" Program of Japan Society for the Promotion of Science under the Project "Advanced Multimedia Contents Processing" (Project No. JSPS-RFTF97P00501).

References

1) Pointcast Network: http://www.pointcast.co.jp (1999).
2) Ramakrishnan, S. and Dayal, V.: The PointCast Network, *SIGMOD Record (ACM Special Interest Group on Management of Data)*, Vol. 27, No. 2, p. 520 (1998).
3) BackWeb:http://www.backweb.com (1999):
4) Yan, T. W. and Garcia-Molina, H.: SIFT — a Tool for Wide-Area Information Dissemination, *Proceedings of the 1995 USENIX Technical Conference: January 16–20, 1995, New Orleans, Louisiana, USA* (USENIX Association(ed.)),pp. 177–186 (1995).
5) TVML: http://www.strl.nhk.or.jp/TVML/ index.html (1999).
6) Hayashi, M., Ueda, H. and Kurihara, T.: TVML (TV program Making Language) - Automatic TV Program Generation from Text-based Script-, *Proc. of Imagina'99* (1999).
7) Kondo, H., Sumiya, K. and Tanaka, K.: A TV-program Metaphor Based Presentation System for Retrieval Results (in Japanese), Technical report, IPSJ SIGDBS Technical Report, Okinawa (1999).
8) Microsoft: http://www.microsoft.com/XML (1999).
9) Shapiro, D. M.: Push-Based Web Filtering Using PICS Profiles, http://www.w3.org/TandS/ Public/Theses/DavidShapiro/ thesis-dshapiro.html (1999).
10) Kamba, T., H.Sakagami and Y.Koseki: Automatic personalization on push news service, *W3C Push Workshop*, http://www.w3.org/ architecture/9709_Workshop/paper02/paper02.html (1997).
11) SiteCrusie Theater: http://www.sitecruise.com (1999).