Semantic Issues in Multimedia Systems, IFIP TC-2 □
Working Conference (Sep. 1999)

A User Customized Selection and Categorization for Broadcast Data

Somnuk SANGUANTRAKUL† Tsutomu TERADA† Masahiko TSUKAMOTO† Shojiro NISHIO†
Kouji MIURA‡ Satoshi MATSUURA‡ Takeshi IMANAKA‡

†Department of Information Systems Engineering, Graduate School of Engineering, Osaka University
‡Central Research Laboratories, Corporate Research Division, Matsushita Electric Industrial Co., Ltd.

Abstruct: Recently, many broadcast satellites have been launched to provide data broadcasting services for public users. Although the provided services can cover many kinds of data and a wide range of user interest, it is considered that, in general, a user is interested in only some specific genres of data. Consequently, storing all received data is considered to be inefficient and only wasting a large amount of memory. This motivated us to introduce an information filtering system into a broadcast data receiving system. The use of filtering system increases the efficiency of memory usage and reduces time for searching interesting data. In this paper, we propose a filtering method that uses a tree structure to represent user preferences. The use of this filtering method enable the receiving system to select only data that match user's interest and classify the stored data in the way that suits the user's access pattern. We also describe the design and implementation of our broadcast data receiving system that makes use of the proposed filtering method. Further, we evaluate the performance of our method by showing some simulation results.

1 Motivation

In recent years, several broadcast satellites have been launched. Using these satellites, various types of broadcast services can be provided. The contents of these services vary from conventional stream information such as DIRECTV[10] to digital data. One advantage of broadcast service is that it can simultaneously provide services to a large number of users without any quality degradation. On the other hand, it is difficult to customize the contents of services to match the need of each user. Moreover, using the broad bandwidth of downlink channel, broadcast services can provide a large volume of data which cover an extensive area of interest. To manage and reuse these large volume of data efficiently, it is general for data providers to use some taxonomy trees to classify their broadcast data.

On the other hand, a user of broadcast services is generally interested only in some specific genres of broadcast data. Considering the service charge, the capacity of memory in case of storing broadcast data for reutilization, and the time needed for finding interesting data, it is not efficient to store all of the data that the system receives. Therefore, a broadcast receiving system must have some methods of selecting only the data that its users are interested in.

Several researches concerning methods of selecting data that fit the interest of users have been done in the field of information filtering and various filtering methods have been proposed so far [1, 3, 4, 5, 6, 8, 9]. However, most of them do not take into account the categorization of the received data. In case of broadcast services using satellite, the number of data that match user's interest is still expected to be very large due to the large volume of broadcast data that are provided. Therefore, it is difficult for a user to access the data all at once. It is considered that the users have to selectively access only the important data when they have not much time, and access the other data lately. Categorizing the selected data is necessary to aid the user to select data of his/her interest.

Among the proposed filtering methods, the method proposed by Stevens[9] fills up the gap between the classification of data at the sender side and that at the receiver side, but the users have to define their categorization manually. Therefore, the users have to maintain their ways of categorization over time not only when their interests and access patterns change, but also when the categorization of broadcast data changes.

In the field of categorization, there is also an algorithm of learning to create rules from a training sample and using the created rules to classify incoming electronic mails[2]. However, like Stevens' method[9], the modification and maintenance of the rules is a troublesome task.

In this paper, we propose another method that use the taxonomy trees to filter and categorize the broadcast data at the same time. The remainder of the paper is organized as follows: In the next section, we explain our filtering method in detail. In section 3, the design and implementation of a prototype system are explained. In section 4, we describe the evaluation of our method by showing some simulation results. Finally, we summarize our work and discuss about the future work.

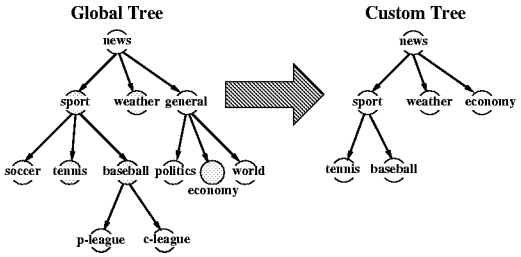

Figure 1: Global tree and custom tree.

2 Filtering Method

2.1 Custom Tree and Global Tree

The filtering method proposed here uses a taxonomy tree to filter and categorize broadcast data. By using tree structure, our method can select and categorize broadcast data at the same time. Here, we propose the use of different tree structure at the sender side and the receiving side for three reasons. First, since broadcast data cover a wide range of interest, the structure of the taxonomy tree at the sender side is considered to be very complicated with a large number of nodes, and therefore it is difficult for a user to find out the interesting data. Secondly, the way that the data provider and the user categorize broadcast data based on different viewpoint and therefore their way of the categorization are often different. Finally, the state of the receiving side and the sender side seperately changes.

Here, we call the taxonomy tree used at the sender side a *global tree*, and the tree used at the receiving side a *custom tree*. Each node at the global tree represents a genre of broadcast data. The data provider can change the structure of a global tree if necessary. The information of the global tree is periodically broadcasts to the users. Each piece of broadcast data is added with the information of its position at the global tree. Figure 1 shows examples of a global tree and a custom tree.

2.2 Construction of a Custom Tree

At the receiving side, a custom tree is constructed using the information of the global tree. Each node at the custom tree represents a category of data that the user is interested in. Note that although the nodes of a custom tree are a subset of the nodes of the global tree, the structure of the custom tree is not restricted by that of the global tree. Its structure is customized to match the access pattern of each user.

2.3 The Use of Custom Tree

Filtering is performed by using the classificational information of broadcast data. In order to perform data filtering in a flexible way, we attach a real number that varies from 0 to 1 to each node of the custom tree to represent the degree of user's interest in that node. We call this number the *profile matching degree*. The receiving system automatically calculates the profile matching degree based on the evaluation of the data that are categorized to that node.

A broadcast data is categorized to a leaf node of the custom tree. Each node in the custom tree has an allotment of the number of data it can store. The allotment depends on the profile matching degree and the access ratio of that node. The allotment is adjusted periodically, say every broadcast cycle, or when a new node is added, or an existing node is deleted. The receiving system stores a broadcast data if there is at least a leaf node to which the data can be categorized and the number of data at the point of time is less than the allotment of that node, i.e., it has a space to store the data.

Moreover, since the structure of the custom tree reflects the access pattern of the user, the receiving system can also use the custom tree as a user interface. In this case, the user can access any data by simply traversing his/her custom tree.

2.4 Reconstruction of custom tree

Generally, the interest of a user, the access pattern, the state of global tree and custom tree continuously change when time passes. For example, a user may be interested in a new category of data or lose his/her interest in some categories, or the degree of interest in a specific category may change. the data provider may add or remove a category, the number of data may changes. the number of data classified in the nodes that have high profile matching degree may become large. On the contrary, the number of data classified in the nodes that have low profile matching degree may become small. The structure of the custom tree may becomes too deep that a user needs more operation to access a data. These changes make the custom tree unsuitable.

Modifying the profile matching degree and the allotment of data can keep up with some of these changes, such as the change of user's interest. However, to cope with the other changes such as the imbalance of the custom tree, the receiving system have to adjust the structure of the custom tree if necessary. The adjustment of the custom tree is done through the following fundamental operations:

- **level up:** move a specific node to the upper level.
- **level down:** move a specific node to the lower level.
- **delete:** delete a specific node.
- **add:** add a new node.

Note that all of the above operations are reversible, i.e., level up is the reverse of level down, and addition is the reverse of deletion. The receiving system combines these fundamental operations to reconstruct the custom tree as follows:

split: When the number of data categorized to a node becomes large, the user may need longer time to decide which data to access. Splitting into detailed categories can decrease the average number of data per category and therefore shortens the selection time. Another case is when a specific node has a low access ratio with a low profile matching degree, which means that the user is actually interested in specific detailed categories of data. Splitting that node into detailed categories can extract the actual interest of the user, i.e., the nodes that the user is not interested is somehow deleted later. An example of the operation is shown in Figure 2.

reduce: When the number of data categorized to each child of a specific node decreased, the detailed categorization of data increases the traversal time while slightly decreases the selection time. Therefore, the operation of summarizing several categories into a more abstract category is needed. An example of the operation is shown in Figure 3.

level up: A node that is frequently accessed is moved to the upper level in order to shorten the the traversal time to that node. An example of the operation is shown in Figure 4.

delete & add: Nodes that have low profile matching degrees are deleted. A "misc" node is added to a node that is frequently accessed to gather other interesting data that may not match the current custom tree.

Since the fundamental operations for tree reconstructions are reversible, all of the operations mentioned above are also reversible too. For example, consider Figure 8, the split operation can be done by adding two new custom nodes, i.e., a "c-league" node and a "p-league" node, to the "baseball" node and then move down all of the data nodes that is categorized to the "baseball" node to either "c-league" node or "p-league" node. On the other hand, the reverse operation of split can be done by leveling up all the data nodes from both "c-league" node and "p-league" node to the "baseball" node and then delete both the empty nodes. However, in the practical system, not all of the pair operation are needed. Proving the sufficiency of the provided operations is one of our future work.

3 Design and Implementation

The system assumes that the information on taxonomy at the global tree is contained in the header of broadcast data. AIS filters the broadcast data based on this taxonomy information. The system is composed of the following 4 components:

User interface:
In AIS, a custom tree is also used as a user interface. The user only traverses through the custom tree to access a data. The information of the access pattern is sent to the custom tree management module and is used in custom tree reconstruction.

Information filtering module:
Broadcast data that match user's interest are selected by using the taxonomy information attached. Selection is made as explained in section 2.

Custom tree management module:
The custom tree is managed by a custom tree management module. The structure of the custom tree is modified to fit user's interest, the access pattern of the user and changes of the global tree.

Active database:
AIS stores the selected data using a super active database(SADB)[7]. SADB is an active database that is extended for broadcast service. It provides the following functions:

- The function of exchanging data or packets among databases.
- The function of sending and receiving ECA rules.
- The function of grouping ECA rules and suspending ECA rules.
- The function of timer execution.

The implemented system uses a server on behalf of the satellite to broadcast data through network. We implemented AIS on a notebook computer with Window95 using the Visual C++ 4.0. The Netscape Communicator 4.03 browser is used as the system interface. Figure 6 shows an example of AIS.

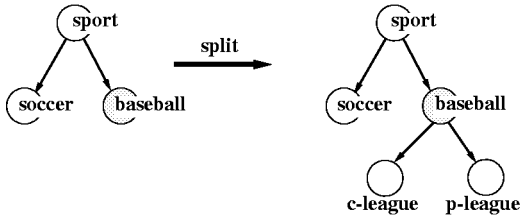

Figure 2: An example of the split operation.

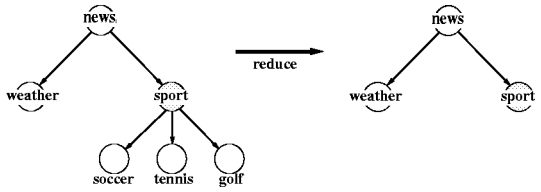

Figure 3: An example of the reduce operation.

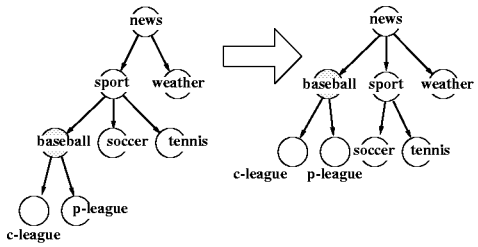

Figure 4: An example of the level up operation.

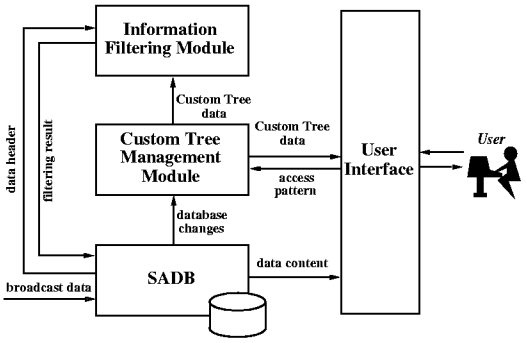

Figure 5: Structure of AIS.

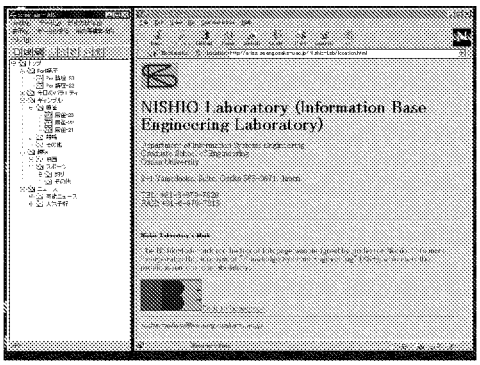

Figure 6: An example of AIS.

4 Evaluation

In this section, we describe the simulation we have done to evaluate the performance of the proposed algorithm. First, we modulate the global tree to simulate a broadcast station. Each node of the global tree periodically creates a number of new data at a specified time intervals. Each data has its lifetime to exist in the broadcast program.

We represent user's interest by using a tree, where the structure is the same as that of global tree and each node has a real number representing the degree of user's interest. Note that user's interest is independent of tree structure. The number of nodes that match user's interest is about 20% of the number of nodes in the global tree.

Moreover, we classify the access pattern into the following 4 types.

- select the node in order of the degree of interest and access all the unaccessed data classified to that node.
- select the node in order of the degree of interest and access a number of data in proportion to the degree of interest.
- select the node sequentially from top to bottom and access all the unaccessed data classified to that node.
- select the node from top to bottom and access a number of data in proportion to the degree of interest.

We also take into account the changes of user's interest. In our simulation, the degree of interest of about 20% of the number of nodes that users are interested

in changes at specific cycles. The change of interest is limited to the node that is semantically near to nodes that users are interested in.

To compare with our method, we also perform a simulation using two naive methods. The first method simply uses the global tree to categorize the incoming data without any filtering performed. The other method is more robust in the point that it removes nodes of which the access ratio is 0, i.e., users do not access any data categorized to that node.

We evaluate the performance of the algorithm by simulating 60 broadcast cycles using 5 randomly created patterns of user's interest, each performs the above 4 access patterns. We use the following factors to evaluate the performance of the algorithm:

- Precision. Here, we define a precision as a percentage of nodes in the custom tree that match user's interest.
- Recall. Here, we define a recall as a percentage of nodes that are included in the custom tree of all nodes that the each user is interested in.
- The number of data accessed compared with the number of data stored.
- The number of operations done to access data compared with the number of data accessed.

The simulation results are shown in Figures 7 to 10. Each result shows the average value taken from the simulations. In every graph, "ctree" is the result of our method using custom tree, "gtree" is the result of the method simply using global tree, and "ztree" is the result of the method that removes nodes of which its data are not accessed. The suffix "-c" means that user's interest is constant, "-t" means that user's interest changes.

As shown in Figure 7 and 8, both cases of our method results in a well-balanced precision and recall. That is, our method is able to select almost all of the data that the user is interested in while select only few data that is out of interest. On the other hand, the naive method using global tree results in obviously high recall but very low precision. That is, even it can select all the data that the user is interested in, it also select a considerable number of data that is out of interest. However, the performance of naive method which removes the non-access nodes is comparable to our method.

The precision of our method is not effected so much when the user's interst changes. As for the recall, it is obviously effected by the change. However, tree reconstruction recovers the recall in some degree.

Figure 9 shows that our method successfully selects only data that match user's interest. On the other

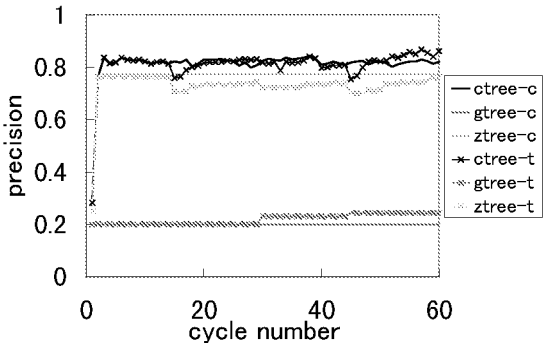

Figure 7: Precision

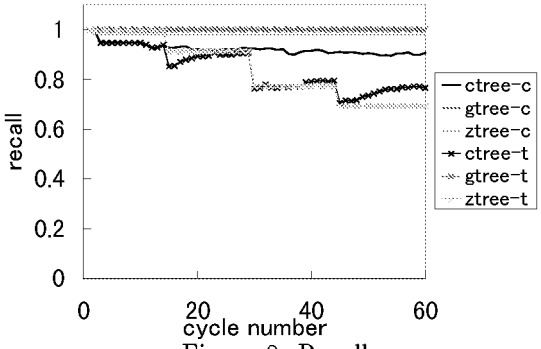

Figure 8: Recall

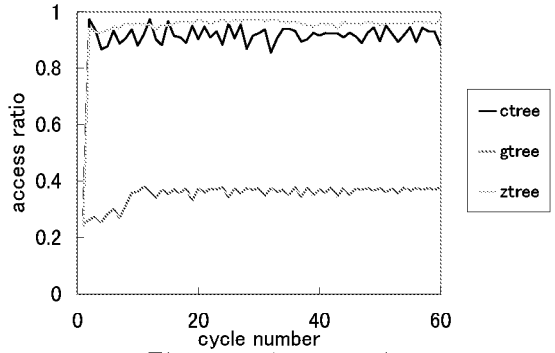

Figure 9: Access ratio

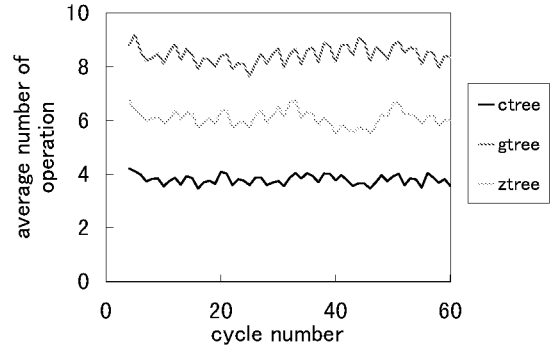

Figure 10: Average number of operation

hand, the naive method using global tree has a very low access ratio while the method which removes non-access nodes performs a little bit better than our method.

Figure 10 shows the average number of operations that is needed to access a broadcast data. The result is smoothed by averaging over every three broadcast cycles. As shown in the figure, our method is obviously better than the other two naive methods. This means that the reconstruction of custom tree is effective to modify the categorization to fit to the access pattern of user.

Although our method achieves better performance than other naive methods in the aspect of average number of operations, four operations per data is considered to be quite large. This is because the hierarchy structure is not always suitable for user interface. That is, when the number of data is small, the operations need for traversal the custom tree account for greater part of all operations, thus the average number of operations that is needed to access a data becomes large. In the case that the number of selected data is small, it is more appropriate to simply list all data to users.

5 Conclusion

In this paper, we have introduced a method of selecting and categorizing broadcast data for the receiving system of broadcast service. Our method constructs a user-oriented category tree named a custom tree based on the taxonomy tree named a global tree that is used to manage broadcast data at the server side. Broadcast data that match user's interest are selected using the custom tree. The structure of the custom tree is constantly observed and modified as necessary to fit to user's interest and access pattern. We have also mentioned the design and implementation of a prototype application system using SADB to store broadcast data called active information store(AIS). Finally, we have described the simulation we have done to evaluate the performance of our method.

Although the evaluation done here is a bit lacking of substantiation, it is enough to discover some weak points of our method. For example, the average number of operation is still high. The performance of our method still drops when the user's interest changes. Further improvement will be made.

Acknowledgement

This research was supported in part by Research for the Future Program of Japan Society for the Promotion of Science under the Project "Researches on Advanced Multimedia Content Processing (JSPS-RFTF97P00501)" and Grant-in-Aid for Scientific Research on Priority Areas from Ministry of Education, Science, Sports, and Culture, Japan under grant number 08244103.

References

[1] Baclace, P.E.: "Competitive Agents for Information Filtering," *Comm. ACM*, vol. 35, no. 12, p.50 (Dec. 1992).

[2] Cohen, W.W.: "Learning Rules that Classify E-Mail," *Proceeding in AAAI Spring Symposium on Machine Learning in Information Access* (Mar. 1996).

[3] Loeb, S.: "Architecting Personalized Delivery of Multimedia Information," *Comm. ACM*, vol. 35, no. 12, pp.39–48 (Dec. 1992).

[4] Maes, P.: "Agents that Reduce Work and Information Overload," *Comm. ACM*, vol. 37, no. 7, pp.31–40 (Jul. 1994).

[5] Mock, K.J.: "Hybrid Hill-Climbing and Knowledge-Based Methods for Intelligent News Filtering," *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI'96)*, Vol. 1, pp.48–53 (Aug. 1996).

[6] Mostafa, J., Mukhopadhyay, S., Lam, W., Palakal, M.: "A Multilevel Approach to Intelligent Information Filtering: Model, System, and Evaluation," *ACM Transactions on Information Systems*, vol. 15, no. 4, pp.368–399 (Oct. 1997).

[7] Terada, T., Sanguantrakul, S., Tsukamoto, M., Nishio, S., Miura, K., Matsuura, S., Imanaka, T.: "A Broadcast Data Storing Method Using Active Database," *IPSJ SIG Notes 97-DPS-85*, pp.243–248(Nov. 1997), in Japanese.

[8] Sheth, B.: "NEWT: A Learning Approach to Personalized Information Filtering," ftp://ftp.media.mit.edu/pub/agents/interface-agents/news-filter.ps.

[9] Stevens, C.: "Automating the Creation of Information Filters," *Comm. ACM*, vol. 35, no. 12, p.48(Dec. 1992).

[10] DIRECTV Homepage: http://www.directv.com/.