ICMU • 2004

8-9 January 2004 Yokosuka Japan

Proceedings of

1st International Conference on Mobile Computing and Ubiquitous Networking

January 8-9, 2004

NTT DoCoMo R&D Center, Yokosuka, Japan

Sponsored by

IPSJ SIG Mobile Computing and Ubiquitous Networking

and IPSJ SG Broadcast Communication and Computing

Supported by

IPSJ SIG Intelligent Transport Systems

and IPSJ SIG Ubiquitous Computing System

Executive Committee of International Conference on Mobile Computing and Ubiquitous Networking (ICMU)

Mizuno Laboratory,
Faculty of Information
Shizuoka University
3-5-1 Johoku, Hamamatsu 432-8011, Japan

ISBN 4-902523-00-0
Printed in Japan

Proc. of the 1st International Conference on Mobile Computing and Ubiquitous Networking (ICMU 2004), pp. 166--171 (Jan. 2004)

Composition Order of Filtering Functions for Information Filtering

Rie Sawai, Masahiko Tsukamoto, Tsutomu Terada, and Shojiro Nishio

Graduate School of Information Science and Technology, Osaka University
2-1 Yamadaoka, Suita, Osaka 565-0871, Japan
{rie,tuka,tsutomu,nishio}@ist.osaka-u.ac.jp

ABSTRACT

In our previous works we defined the notion of filtering function that represents filtering as a function, and clarified the properties of filtering functions to establish a mathematical foundation of information filtering. The constructed mathematical foundation makes it possible to qualitatively evaluate various filtering methods, to optimize processing methods in filtering, and to design a declarative language describing filtering policies. In this paper, we investigate the properties in case of changing the composition order of filtering functions to clarify the characteristics of filtering functions that combine two filtering methods. Exploiting the results of this paper, we can qualitatively indicate the effect of the execution order on the filtering results in filtering consisting of some filtering methods.

1 INTRODUCTION

Since mobile computing and ubiquitous networking become widespread in recent years, various types of data can be distributed regardless of time and location. Moreover, the number of broadcast services has increased due to the introduction of new satellite-based services and the digitization of broadcasts[10]. Therefore, broadcast services for mobile computers, such as cellular mobile telephones and PDAs, have also been setting up. In this environment, not only is the amount of data being distributed or broadcast increasing, but so is the variety of data. However, users often only need small amounts of specific data, and it is very difficult to retrieve the information they are interested in from the enormous amounts of data available. As a result, various mechanisms that automatically filter data, and user-request description languages for filtering, have been proposed[1, 2, 3, 7, 9]. These filtering mechanisms filter data by different criteria such as keyword matching or relevance feedback. However, no mathematical foundation for qualitatively representing these filtering processes exists. Thus, it is not possible to qualitatively evaluate various filtering methods, to optimize processing methods in filtering, or to design a declarative language for filtering processes. In [11], we defined a *filtering function* that expresses filtering as a function, and this function made it possible to qualitatively represent several properties of filtering by satisfying relevant constraints.

Filtering methods currently used in practice are generally implemented by combining various other methods whose filtering policies or features are different. In composite filtering, more efficient processes can be achieved by changing the execution order appropriately. For example, when combining a filtering method using simple operations with that using complex operations, the processing cost of the entire filtering can be reduced by firstly executing the filtering method with simple operations, which will reduce the amount of data to be processed by the filtering method with the complex operations. Moreover, by first executing the filtering method that extracts less data and further narrowing down the amount of data at an early stage, the quantity of data that would've been processed by both the first filtering method and its subsequent filtering method can be reduced. Generally, the larger the amount of data to be processed is, the higher the cost of processing them will become. It is therefore effective to change the execution order of composite filtering according to the filtering environment, such as by the contents and structure of the received data.

In some combinations of filtering methods, however, the filtering result may not be consistent if the execution order is changed dynamically according to the environment. For example, consider a user request "I want the data items whose contents are related to economic news, and ranked in the top 10." This request is carried out by a filtering method combining two methods, f and g. One filtering method f extracts the data items whose contents are related to economic news. The other filtering method g arranges the received data in order of importance according to the user's preferences, then extracts the 10 top-ranked data. When the quantity of received data is sufficiently large, if method f is precedently executed, the number of the filtering result set is 10. On the contrary, if method g is precedently executed, the number of the filtering result set is less than 10 because the result of the precedent method g includes data that will not be selected by the subsequent method f. In other words, the filtering results may not be equal if the execution order of the two methods is interchanged. Consequently, in changing the execution order, it is necessary to assure that the filtering results are consistent even if the execution order is changed.

In this paper, we address the effect of exchanging the composition order of filtering functions, which we have defined, on filtering results. Exploiting the results of this paper, we can implement composite filtering considering exchanges of the execution order in proportion to the filtering environment. Furthermore, we can qualitatively evaluate the effect of composition order on the filtering results.

This paper is organized as follows. Section 2 outlines the filtering function we defined in our previous work. Section 3 clarifies the inclusion relation between the filtering results of composite filtering whose composition order is reversed. Section 4 considers the filtering methods currently applied in practice through the results clarified in this paper. Finally, we conclude our paper in Section 5.

2 PRELIMINARIES

In this section, we outline the filtering function described in [11], which is the foundation of this study.

2.1 Categorization of the Filtering Processes

In this paper, we categorize the filtering methods that exist in the real world into several patterns by the number of filtering processes and receivers, as follows:

- Sequential processing

  In a system using sequential processing, newly received data and previously filtered results, which have already been stored, are merged and filtered every time new data is received.

- Batch processing

  In a system using batch processing, a receiver accumulates broadcast data and filters them out in bulk.

- Distributed processing

  In a system using distributed processing, the received data set is divided into multiple arbitrary data subsets, and each subset is filtered separately before the results are merged.

- Parallel processing

  In a system using parallel processing, the merged filtering results of distributed processing are re-filtered.

2.2 Properties of Filtering Function

Let $\mathbf{T}$ be a set of data items. A filtering function is defined as a function f on $2^{\mathbf{T}}$ that satisfies the following two properties for an arbitrary $T \subset \mathbf{T}$ [1]:

D: decreasing $f(T) \subset T$.
ID: idempotent $f(f(T)) = f(T)$.

The following properties of a filtering function are defined:

M: monotone
$if \quad S \subset T \quad then \quad f(S) \subset f(T)$.
DI: distributed increasing
$f(S \cup T) \subset f(S) \cup f(T)$.
DD: distributed decreasing
$f(S \cup T) \supset f(S) \cup f(T)$.
DE: distributed equivalence
$f(S \cup T) = f(S) \cup f(T)$.
PI: parallel increasing
$f(S \cup T) \subset f(f(S) \cup f(T))$.
PD: parallel decreasing
$f(S \cup T) \supset f(f(S) \cup f(T))$.
PE: parallel equivalence
$f(S \cup T) = f(f(S) \cup f(T))$.
SI: sequential increasing
$f(S \cup T) \subset f(S \cup f(T))$.
SD: sequential decreasing
$f(S \cup T) \supset f(S \cup f(T))$.
SE: sequential equivalence
$f(S \cup T) = f(S \cup f(T))$.
C: consistency
$f(S) \supset f(S \cup T) \cap S$.

Here, S and T are arbitrary subsets of $\mathbf{T}$. The sequential equivalence property (SE) signifies that the filtering results of batch processing and sequential processing are equivalent. Similarly, the distributed equivalence property (DE) signifies that the filtering results of batch processing and distributed processing are equivalent, and the parallel equivalence property (PE) signifies that the filtering results of batch processing and parallel processing are equivalent. In [11], we clarified the relationship between the properties: the sequential increasing and distributed increasing, parallel increasing, consistency properties are equivalent; the distributed decreasing and monotone properties are equivalent; the sequential equivalence and parallel equivalence properties are equivalent (SI⇔DI⇔PI⇔C, DD⇔M, SE⇔PE).

2.3 Composition of filtering functions

A composite function of filtering functions is not necessarily always a filtering function. In [13], we clarified the conditions needed for a composite filtering function to be a filtering function as follows: For filtering functions f and g, we say "*f is filtering composable with g*" when the composite function $f \circ g$ is a filtering function. When $f : D_1 \rightarrow D_2$, we designate $Im(f) \triangleq \{f(X) | X \in D_1\}$ as the range of f. In addition, we define that "*f is g-invariant*," as $f(X) = g(X)$ is satisfied for all $X \in Im(f \circ g)$. Here, we proved the following theorem:

Theorem 1 *For filtering functions f and g, the fact that f is filtering composable with g is equivalent to that f is g-invariant.* □

2.4 Selection Function and Ranking Function

Regarding general filtering, many filtering methods are based on selection and ranking methods. In [12],

[1] In this paper, $A \subset B$ means that A is a subset of B (including the case where A = B).

we defined *selection function* and *ranking function* as follows:

Filtering by selection is a method that specifies whether each broadcast data item is to be stored. Examples of filtering by selection include keyword matching and filtering by a threshold. Keyword matching carries out the logical operation on keywords included in the data and those representing a user's preference, storing only the data that includes particular keywords. Filtering by a threshold gives an evaluation value to each data according to its content, and stores the data only if its evaluation value is larger (or smaller) than the threshold. Assume that there exists $X \subset \mathbf{T}$. We define a selection function of X, B_X, as $B_X(S) \stackrel{\triangle}{=} S \cap X$ for all $S \subset \mathbf{T}$. We call X the *potential set* of this selection function, meaning the set of data items that satisfy the selection condition. It is clear that B_X is a filtering function. Also note that every selection function satisfies $X = B_X(\mathbf{T})$.

Filtering by ranking is a method that arranges the received data in order of importance according to the user's preferences, and extracts a particular quantity of top-ranked data. Suppose a total order $R = (\mathbf{T}, <)$ is given. A function f is an *n-ranking function* for a total order R if and only if, for all $S \subset \mathbf{T}$, f is represented as $f(S) \stackrel{\triangle}{=} \{x \in \mathbf{T} | x < a\} \cap S$ for a certain $a \in \mathbf{T}$ and the *cardinality* of f is n. We define that the cardinality of a function f on $2^{\mathbf{T}}$ is $n(\in \mathbf{N}$, which is the set of 0 and all positive integers) if and only if

$$\begin{cases} |f(S)| = n & \text{(if } S \text{ is an infinite set,} \\ & \text{or } S \text{ is a finite set and } |S| \geq n) \\ f(S) = S & \text{(if } S \text{ is a finite set and } |S| < n) \end{cases}$$

is satisfied for all $S \subset \mathbf{T}$. We call a filtering function that satisfies this condition a *cardinality function*.

The following theorems on selection function and ranking function were clarified in [12]:

Theorem 2 *A filtering function f is a selection function if and only if f satisfies the distributed equivalence property.* □

Theorem 3 *A filtering function f is an n-ranking function for a total order $(\mathbf{T}, <)$ if and only if f satisfies the sequential equivalence property and the cardinality of f is n.* □

3 INCLUSION RELATION OF COMPOSITE FUNCTIONS

In this section, we clarify the effect of exchanging the composition order of composite filtering functions on filtering results. In Subsection 3.1, we show the effect of that for the composite functions of filtering functions that satisfy the increasing or decreasing properties. In Subsection 3.2, we present the effect of that for the composite functions of filtering functions that satisfy the equivalence properties.

3.1 Filtering Functions that Satisfy the Increasing or Decreasing Properties

For the increasing and decreasing properties denoted in Section 2, the monotone (M), sequential increasing (SI), sequential decreasing (SD), and parallel decreasing (PD) properties are not equivalent to each other. In this subsection, we reveal the inclusion relation between the filtering results of the composite filtering functions whose composition order is reversed for filtering functions that satisfy those four properties, and introduce the following lemmas. We omit the proofs of the lemmas.

Lemma 1 *If filtering functions f and g satisfy the monotone property, and f is filtering composable with g, g is filtering composable with f, then $f(g(S)) = g(f(S))$ is satisfied for all $S \subset \mathbf{T}$.* □

Lemma 2 *For filtering functions f and g, if f satisfies the monotone property, and g satisfies the sequential increasing property, f is filtering composable with g, g is filtering composable with f, then $f(g(S)) \subset g(f(S))$ is satisfied for all $S \subset \mathbf{T}$.* □

Lemma 3 *For filtering functions f and g, if f satisfies the monotone property, and g satisfies the sequential increasing property, f is filtering composable with g, g is filtering composable with f, then $f(g(S)) \supset g(f(S))$ is not necessarily satisfied for all $S \subset \mathbf{T}$.* □

Lemma 4 *For filtering functions f and g, if f satisfies the monotone property, and g satisfies the sequential decreasing property, f is filtering composable with g, g is filtering composable with f, then $f(g(S)) \subset g(f(S))$ and $f(g(S)) \supset g(f(S))$ are not necessarily satisfied for all $S \subset \mathbf{T}$.* □

Lemma 5 *For filtering functions f and g, if f satisfies the monotone property, and g satisfies the parallel decreasing property, f is filtering composable with g, g is filtering composable with f, then $f(g(S)) \subset g(f(S))$ and $f(g(S)) \supset g(f(S))$ are not necessarily satisfied for all $S \subset \mathbf{T}$.* □

Lemma 6 *If filtering functions f and g satisfy the sequential increasing property, and f is filtering composable with g, g is filtering composable with f, then $f(g(S)) \subset g(f(S))$ and $f(g(S)) \supset g(f(S))$ are not necessarily satisfied for all $S \subset \mathbf{T}$.* □

Lemma 7 *For filtering functions f and g, if f satisfies the sequential increasing property, and g satisfies the sequential decreasing property, f is filtering composable with g, g is filtering composable with f, then $f(g(S)) \subset g(f(S))$ and $f(g(S)) \supset g(f(S))$ are not necessarily satisfied for all $S \subset \mathbf{T}$.* □

Lemma 8 *For filtering functions f and g, if f satisfies the sequential increasing property, and g satisfies the parallel decreasing property, f is filtering composable with g, g is filtering composable with f, then $f(g(S)) \subset g(f(S))$ and $f(g(S)) \supset g(f(S))$ are not necessarily satisfied for all $S \subset \mathbf{T}$.* □

Table 1: The inclusion relation between $f \circ g$ and $g \circ f$ for f and g that satisfy the decreasing or increasing property

$f \setminus g$	M	SI	SD	PD
M	$=$	$\subset, \neg \supset$	$\neg \subset, \neg \supset$	$\neg \subset, \neg \supset$
SI	$\supset, \neg \subset$	$\neg \subset, \neg \supset$	$\neg \subset, \neg \supset$	$\neg \subset, \neg \supset$
SD	$\neg \subset, \neg \supset$	$\neg \subset, \neg \supset$	$\neg \subset, \neg \supset$	$\neg \subset, \neg \supset$
PD	$\neg \subset, \neg \supset$	$\neg \subset, \neg \supset$	$\neg \subset, \neg \supset$	$\neg \subset, \neg \supset$

Lemma 9 *If filtering functions f and g satisfy the sequential decreasing property, and f is filtering composable with g, g is filtering composable with f, then $f(g(S)) \subset g(f(S))$ and $f(g(S)) \supset g(f(S))$ are not necessarily satisfied for all $S \subset \mathbf{T}$.* □

Lemma 10 *For filtering functions f and g, if f satisfies the sequential decreasing property, and g satisfies the parallel decreasing property, f is filtering composable with g, g is filtering composable with f, then $f(g(S)) \subset g(f(S))$ and $f(g(S)) \supset g(f(S))$ are not necessarily satisfied for all $S \subset \mathbf{T}$.* □

Lemma 11 *If filtering functions f and g satisfy the parallel decreasing property, and f is filtering composable with g, g is filtering composable with f, then $f(g(S)) \subset g(f(S))$ and $f(g(S)) \supset g(f(S))$ are not necessarily satisfied for all $S \subset \mathbf{T}$.* □

Table 1 shows the inclusion relation between the filtering results of composite filtering functions whose composition order is reversed for filtering functions that satisfy the increasing or decreasing properties as proved by the above lemmas. In Table 1, "=" means that $f \circ g(T) = g \circ f(T)$ is satisfied for all $T \subset \mathbf{T}$. Moreover, "$\subset$" means that $f \circ g(T) \subset g \circ f(T)$ for all $T \subset \mathbf{T}$, and "$\neg \subset$" means that $f \circ g(T) \not\subset g \circ f(T)$ for a certain $T \subset \mathbf{T}$.

From Table 1, only the composite function of filtering functions that satisfy the monotone property is necessarily commutative. Additionally, we can see that in filtering that combines the filtering method satisfying the monotone property and that which satisfies the sequential increasing property, the result of filtering precedently using the filtering method which satisfies the monotone property includes that of filtering precedently using the filtering method satisfying the sequential increasing property. However, in the other combinations of filtering methods, there is no inclusion relation between the filtering results whose composition order is reversed. Therefore, in such composite filtering methods, if the execution order is changed in the filtering process, there is no guarantee that the data to be stored before conversion is also continuously stored after conversion.

3.2 Filtering Functions that Satisfy the Equivalence Properties

In this subsection, we clarify the inclusion relation between the filtering results of the composite filtering functions whose composition order is reversed for filtering functions that satisfy the equivalence properties. First of all, we introduce the following lemmas for only filtering functions that satisfy the equivalence properties.

Table 2: The inclusion relation between $f \circ g$ and $g \circ f$ for f and g that satisfy the equivalence property

f \ g	DE (Selection)	SE, PE	Ranking
DE(Selection)	$=$	$\subset, \neg \supset$	$\subset, \neg \supset$
SE, PE	$\supset, \neg \subset$	$\neg \subset, \neg \supset$	$\neg \subset, \neg \supset$
Ranking	$\supset, \neg \subset$	$\neg \subset, \neg \supset$	$\neg \subset, \neg \supset$

Lemma 12 *If filtering functions f and g satisfy the distributed equivalence property, and f is filtering composable with g, g is filtering composable with f, then $f(g(S)) = g(f(S))$ is satisfied for all $S \subset \mathbf{T}$.* □

Lemma 13 *For filtering functions f and g, if f satisfies the distributed equivalence property, and g satisfies the sequential equivalence property, f is filtering composable with g, g is filtering composable with f, then $f(g(S)) \subset g(f(S))$ is satisfied for all $S \subset \mathbf{T}$.* □

Lemma 14 *For filtering functions f and g, if f satisfies the distributed equivalence property, and g satisfies the sequential equivalence property, f is filtering composable with g, g is filtering composable with f, then $f(g(S)) \supset g(f(S))$ is not necessarily satisfied for all $S \subset \mathbf{T}$.* □

Lemma 15 *If filtering functions f and g satisfy the sequential equivalence property, and f is filtering composable with g, g is filtering composable with f, then $f(g(S)) \subset g(f(S))$ and $f(g(S)) \supset g(f(S))$ are not necessarily satisfied for all $S \subset \mathbf{T}$.* □

Next, for selection function, ranking function, and filtering functions that satisfy the equivalence properties, we introduce the following lemmas to the inclusion relation between the filtering results of the composite filtering functions whose composition order is reversed.

Lemma 16 *For filtering functions f and g, if f satisfies the sequential equivalence property, and g is a ranking function, then $f(g(S)) \subset g(f(S))$ and $f(g(S)) \supset g(f(S))$ are not necessarily satisfied for all $S \subset \mathbf{T}$.* □

Lemma 17 *If filtering functions f and g are ranking functions, then $f(g(S)) \subset g(f(S))$ and $f(g(S)) \supset g(f(S))$ are not necessarily satisfied for all $S \subset \mathbf{T}$.* □

Lemma 18 *For filtering functions f and g, if f is a selection function, and g is a ranking function, then $f(g(T)) \subset g(f(T))$ is satisfied for all $S \subset \mathbf{T}$.* □

Lemma 19 *For filtering functions f and g, if f is a selection function, and g is a ranking function, then $f(g(S)) \supset g(f(S))$ is not necessarily satisfied for all $S \subset \mathbf{T}$.* □

From Theorem 2, since selection function and filtering function that satisfies the distributed equivalence property are equivalent, we omit the following lemmas: the lemma on the composite function of selection function and filtering function that satisfies the sequential equivalence property; the lemma on the composite function of ranking function and filtering function that satisfies the distributed equivalence property; the lemma on the composite function of selection functions.

Table 2 shows the inclusion relation between the filtering results of the composite filtering functions whose composition order is reversed for filtering functions that satisfy the equivalence properties as proved by the above lemmas. From Theorem 2, since selection function and filtering function that satisfies the distributed equivalence property are equivalent, we describe them in the same column and row. Table 2 indicates that only the composite function of selection functions is necessarily commutative. Moreover, the inclusion relation is identical between the case where f and g satisfy the sequential equivalence or parallel equivalence property, and the case where f and g are ranking functions. It is therefore clarified that the inclusion relation between filtering results of composite filtering functions whose composition order is reversed does not depend on the cardinality.

4 OBSERVATIONS

In this section, we address some of the filtering methods currently applied in practice and discuss the processing methods each filtering can apply, based on the properties shown in the previous section.

4.1 Composition of Selection Methods

XFilter[1] filters XML documents, while NiagaraCQ[4] categorizes the queries. They filter by keyword matching, which is a selection method. WebMate[5] and SIFT[14] extract the data if the vector product of the vectors representing the data and the user's preference exceeds a particular threshold. Tapestry[6] filters by appointing the name of other users whose preferences resemble his/hers, which is known as the collaborative filtering method. These methods are also forms of filtering by selection. Table 2 shows that in filtering combining these methods it is assured that the filtering result is consistent even if the execution order is exchanged. The following optimization can be executed by exploiting this result. When the most-received XML data items are in the same structure, XFilter cannot narrow down the data to be stored due to the XML's data structure. In general, the larger the number of data to be processed in filtering is, the higher the processing cost of the data becomes; therefore, if SIFT has previously narrowed down the data by some threshold and decreases the number of the data to be processed by XFilter, the filtering cost of XFilter and the entire composite filtering can be reduced. On the other hand, when there are many keywords in the received data, the processing cost of SIFT becomes high because the vectors representing the data and user's profile become large, and calculating their product becomes complex. Thus, if XFilter has previously narrowed down the data by using the XML data structure, the vectors calculated by SIFT can be made small, and it is possible to reduce the processing cost of the entire filtering processes.

4.2 Composition of Ranking Methods

LIBRA[8] is an example of filtering by ranking. From Table 2, there is not necessarily an inclusion relation between the filtering results of composite filtering whose execution order is inverted. In other words, there is no guarantee that the filtering result is consistent when changing the execution order in the filtering process. Therefore, a more efficient execution order should be decided upon before the filtering process begins according to which filtering method can narrow down the received data in the early stage, and which filtering method can reduce the processing cost in the first process, where there is a large amount of data to be processed.

4.3 Composition of Selection and Ranking Methods

From Table 2, in the filtering combining selection and ranking methods, if we interchange the execution order while the filtering is in process, constant filtering results cannot be obtained. However, it has been proved that the filtering result of the method that precedently filters by selection (functionally $r \circ s$) includes that of the method that precedently filters by ranking ($s \circ r$). Consequently, even if we interchange the method $s \circ r$ with the method $r \circ s$, the data that should be stored by the former method $s \circ r$ is also stored by the latter method $r \circ s$. As such, even if the execution order is changed while the other processes are using the data from the filtering result, it is certain that the same data is continuously available.

On the other hand, if we interchange the method $r \circ s$ with the method $s \circ r$, then the data that should be stored by the former method $r \circ s$, and ranked lower, may not be stored by the latter method $s \circ r$. However, this interchange is efficient in the case where only the data ranked higher are needed from the data that should be stored before interchanging.

Let's consider the case of combining Tapestry and LIBRA. When there are few other users whose preferences resemble those of a user, or when the other users do not provide enough evaluation values for the data to the filtering system (these values are used to decide which other users have preferences that most resemble those of the current user), the method that precedently filters by Tapestry cannot narrow down the data items in the early stage. This occurs because Tapestry uses a collaborative filtering method that has not learned the user's preference adequately nor has obtained the other users whose preferences resemble those of the current user. Thus, by instead using a method that precedently filters with LIBRA's ranking method, it is possible to narrow down the received data to a particular number of data items in advance, and to curb the calculation costs of the entire filtering process. However, if the receivers lack sufficient calculation ability and memory capacity, it is difficult for LIBRA to use all the data items' evaluation values for the current user to calculate those values and store them into local memory. In such a case, if Tapestry precedently narrows down the data items when the system obtains enough other users whose preferences resemble those of the current user, the number of keywords needed in LIBRA can be reduced. In this conversion, it is assured that the data used before conversion can be continuously available after conversion.

4.4 Composition of the Other Methods

If filtering considers the correlation between the contents of the data and upgrades the evaluation value of the data when they are together, then the filtering satisfies the monotone, sequential decreasing, and parallel decreasing properties. On the contrary, if filtering degrades the evaluation value of the data when they are together, then the filtering satisfies the consistency, sequential decreasing, and parallel decreasing properties[11]. Therefore, from Table 1 and Table 2, filtering combining the former methods, and that combining the former method and selection method, are commutative. A consistent filtering result can thus be obtained even if the execution order is exchanged. However, in filtering combining the latter methods and that combining the latter method and ranking method, there is not necessarily an inclusion relation between the filtering results whose execution order is inversed.

5 CONCLUSION

In this paper, we clarified the inclusion relation between the filtering results of composite filtering functions whose composition order is reversed for filtering functions that satisfy various properties. Exploiting the results in this paper, we can implement composite filtering, considering exchanges of the execution order in proportion to the filtering environment. Moreover, we indicated that we can achieve more efficient filtering processes according to the environment by applying the mathematical foundation established in this paper to filtering using multiple methods.

Our future work includes the following:

- Constraints on the composite function

  In the composite filtering functions given in this paper, there is not necessarily an inclusion relation between the filtering results whose execution order is inverted. However, by adding specific constraints to each filtering function in composition, the inclusion relations dealt with in this paper may be satisfied after composition. We will define such constraints.

- Deciding the optimum execution order

  When applying the results in this paper, it is necessary to examine the effect of various factors, such as the broadcast environment and the characteristics of filtering methods, on the processing cost. Therefore, we must establish a mechanism to evaluate the efficiency of each execution order, considering the environmental factors, and to automatically decide the most efficient one.

ACKNOWLEDGEMENTS

This research was supported in part by "The 21st Century Center of Excellence Program," Special Coordination Funds for promoting Science and Technology, and Grant-in-Aid for Scientific Research (B) numbered 15300033 of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

[1] M. Altinel and M. J. Franklin: "Efficient filtering of XML documents for selective dissemination of information," in *Proc. 26th International Conference on Very Large Data Bases* (*VLDB2000*), pp. 53–64 (2000).

[2] N. J. Belkin and W. B. Croft: "Information filtering and information retrieval: two sides of the same coin?," *Communications of the ACM*, vol. 35, no. 12, pp. 29–38 (1992).

[3] T. A. H. Bell and A. Moffat: "The design of a high performance information filtering system," in *Proc. SIGIR '96*, pp. 12–20 (1996).

[4] J. Chen, D. J. DeWitt, F. Tian, and Y. Wang: "NiagaraCQ: a scalable continuous query system for internet databases," in *Proc. ACM SIGMOD2000*, pp. 379–390 (2000).

[5] L. Chen and K. Sycara: "WebMate: a personal agent for browsing and searching," in *Proc. Second International Conference on Autonomous Agents*, pp. 132–139 (1998).

[6] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry: "Using collaborative filtering to weave an information TAPESTRY," *Communications of the ACM*, vol. 35, no. 12, pp. 61–70 (1992).

[7] S. Loeb and D. Terry: "Information filtering," *Communications of the ACM*, vol. 35, no. 12, pp. 26–28 (1992).

[8] R. J. Mooney and L. Roy: "Content-based book recommending using learning for text categorization," in *Proc. 5th ACM Conference on Digital Libraries*, pp. 195–204 (2000).

[9] J. Mostafa, S. Mukhopadhyay, W. Lam, and M. Palakal: "A multilevel approach to intelligent information filtering: model, system, and evaluation," *ACM Transactions on Information Systems*, vol. 15, no. 4, pp. 368–399 (1997).

[10] Satellite Magazine: http://www.satemaga.co.jp/.

[11] R. Sawai, M. Tsukamoto, Y. H. Loh, T. Terada, and S. Nishio: "Functional properties of information filtering," in *Proc. 27th International Conference on Very Large Data Bases* (*VLDB2001*), pp. 511–520 (2001).

[12] R. Sawai, M. Tsukamoto, T. Terada, and S. Nishio: "On Selection and Ranking in Filtering Function," IPSJ Transactions on Databases, vol. 43, no. SIG12(TOD16), pp. 80–91, in Japanese (2002).

[13] R. Sawai, M. Tsukamoto, T. Terada, and S. Nishio: "Composition of filtering functions," In *Proc. 8th International Conference on Database Systems for Advanced Applications* (*DASFAA2003*), pp. 293–300 (2003).

[14] T. W. Yan and H. Garcia-Molina: "The SIFT information dissemination system," *ACM Transactions on Database Systems*, vol. 24, no. 4, pp. 529–565 (1999).