Proceedings of the
IASTED International Conference

Information Systems and Databases

September 25-27, 2002
Tokyo, Japan

A Publication of The International Association of Science and Technology for Development - IASTED

ACTA Press
Anaheim • Calgary • Zurich

ISBN: 0-88986-362-8
ISSN: 1482-793X

Proceedings of the IASTED International Conference on **Information Systems and Databases**, held September 25-27, 2002, Tokyo, Japan.

Sponsors

The International Association of Science and Technology for Development (IASTED)

- Technical Committee on Databases
- Technical Committee on Information Systems

The Information Processing Society of Japan - Special Interest Group on Database Systems
The Institute of Electronics, Information, and Communications Engineers - Special Interest Group on Data Engineering

Conference Chair

Katsumi Tanaka
Kyoto University, Japan

International Program Committee Co-Chairs

Y. Kiyoki
Keio University, Japan
M. Yoshikawa
Nara Institute of Science and Technology, Japan

Local Arrangement Chair

H. Yokota
Tokyo Institute of Technology, Japan

International Program Committee

M. Adiba - Univ. Joseph Fourier, France
M. Arikawa - Univ. of Tokyo, Japan
B. Benatallah - Univ. of New South Wales, Australia
A. Bouguettaya - Virginia Tech., USA
C.-C. Chang - National Chung Cheng Univ., Taiwan
S. Christodoulakis - Technical Univ. of Crete, Greece
W. Chu - Univ. of California – Los Angeles, USA
M.H. Dunham - Southern Methodist Univ., USA
A.W. Fu - Chinese Univ. of Hong Kong, PRC
T. Hara - Osaka Univ., Japan
H. Ishikawa - Tokyo Metropolitan Univ., Japan
Y. Ishikawa - Univ. of Tsukuba, Japan
C.S. Jensen - Aalborg Univ., Denmark
H. Kang - Chung-Ang Univ., Korea
H. Kangassalo - Univ. of Tampere, Finland
N. Katayama - National Institute of Informatics, Japan
W. Klas - Univ. of Ulm, Germany
R.W.H. Lau - City Univ. of Hong Kong, PRC
C. Lee - National Cheng Kung Univ., Taiwan
D. Lee - The Hong Kong Univ. of Science and Technology, PRC
S.-Y. Lee - National Chiao Tung Univ., Taiwan
W.-S. Li - NEC USA, USA
E.-P. Lim - Nanyang Technological Univ., Singapore
H. Lu - Hong Kong Univ. of Science and Technology, PRC
H. Lu - Oklahoma State Univ., USA
B. Moon - Univ. of Arizona, USA
M. Nakano - Univ. of Tokyo, Japan
S. Nishio - Osaka Univ., Japan
V. Oria - New Jersey Institute of Technology, USA
M.E. Orlowska - Univ. of Queensland, Australia
P. Pouyioutas - Intercollege, Greece
A.K. Pujari - Univ. of Hyderabad, India
D. Saccà - Univ. of Calabria, Italy
Y. Shirota - Gakushuin Univ., Japan
I.-Y. Song - Drexel Univ., USA
K. Sumiya - Kyoto Univ., Japan
K.-L. Tan - National Univ. of Singapore, Singapore
G. Wang - Northeastern Univ., PRC
K.-Y. Whang - Korea Advanced Institute of Science and Technology (KAIST), Korea
K.-F. Wong - The Chinese Univ. of Hong Kong, PRC
P.S. Yu - IBM TJ Watson Research Center, USA
J.X. Yu - The Chinese Univ. of Hong Kong, PRC

Additional Reviewers

M.A. Aboelaze - Canada
K.C. Almeroth - USA
S. Byun - Korea
K. Dahbur - USA
A. Gelbukh - Mexico
J. Jordan - Canada
H. Kosch - Austria
B.G. Lees - Australia
M. Li - UK
A. Nelson - USA
C. Nicolescu - The Netherlands
A. Sanchis - Spain
I. Traore - Canada
B. Vélez - Puerto Rico
A. Zhou - PRC

For each IASTED conference, the following review process is used to ensure the highest level of academic content. Each full manuscript submission is peer reviewed by a minimum of two separate reviewers on the International Program Committee / Additional Reviewers list. The review results are then compiled. If there is a conflict in reviews, the paper is sent to a third reviewer.

Abstracted by INSPEC

ACTA Press
P.O. Box 5124
Anaheim, CA 92814-5124
USA

ACTA Press
#80, 4500 - 16th Ave. NW
Calgary, AB T3B 0M6
Canada

ACTA Press
P.O. Box 354
CH-8053 Zurich
Switzerland

Publication Code: 367

Proc. of IASTED International Conference on
Information Systems and Databases (ISDB 2002) (Sep. 2002)

An Active Database System for Receiving Broadcast Data

Tsutomu TERADA
Cybermedia Center
Osaka University
5-1 Mihogaoka, Ibaraki, Osaka 567-0047, JAPAN
email: tsutomu@cmc.osaka-u.ac.jp

Masahiko TSUKAMOTO and Shojiro NISHIO
Grad. School of Information Science and Technology
Osaka University
2-1 Yamadaoka, Suita, Osaka 565-0871, JAPAN
email: {tuka,nishio}@ist.osaka-u.ac.jp

Abstract

In recent years, many broadcast and communication satellites have been launched to start various new services, which attracts a great deal of attention on data broadcasting system using satellites. In such systems, the volume and variety of data being broadcast will increase significantly. This results in a growing demand on a system that stores a large volume of data and presents the necessary information for users in an efficient way. Highly motivated by such requirements, we propose the super active database(SADB) system which is a data receiving system based on an active database in this paper. Since the use of active database can increase the flexibility in describing the system behavior, the proposed system is well suited to receive and process many types of data. On SADB, the receiving operation management and the received data processing are separated from the rule processing unit. As a result, the SADB can process a great deal of data faster than the conventional active databases.

Keywords: Active Databases, Data Broadcast, ECA Rules

1. Introduction

In recent years, many broadcast and communication satellites have been launched to start various new services, which attracts a great deal of attention on data broadcasting system using satellites. In such systems, the variety and the volume of data to be broadcast will increase significantly. In this situation, it is very hard for users to extract the valuable data.

Hence, we propose a system for receiving broadcast data based on an active database mechanism. Behaviors of the proposed active database system are described in a simple language called ECA-rules. Therefore, this system can receive and process a variety of broadcasting data by customizing ECA-rules set. Additionally, we propose a rule processing method increasing the processing speed of receiving data for handling a large amount of data.

The remainder of this paper is organized as follows. Section 2 outlines a broadcast data receiving system. Section 3 explains an active database which is the base technology of our system and section 4 illustrates the design and the implementation of our system. Section 5 shows an application using our system. In section 6, we show the evaluation results. Section 7 presents the conclusion and future works.

2. Data Broadcasting System

Data broadcasting using a satellite generally has the following characteristics:

- **Simultaneity**

  A satellite can transmit data with a broadcast channel covering a wide region by which thousands of clients can receive the data simultaneously without degradation of quality. On the other hand, services must be uniform but not personalized to users[7].

- **Broad bandwidth**

  Using a satellite, a sender can broadcast a large volume of data in few minutes. Generally, the transmission speed of satellite broadcasting deverses from few hundred Kbps to tens of Mbps.

- **Unidirection**

  The bandwidth of uplink is unreliable because narrow bandwidth devices such as dial-up access modem are used as an uplink.

With data broadcasting using a satellite, a user receives a great deal of data including a large amount of unnecessary data. In this situation, it is a hard work for user to extract necessary data. At the same time, many data processing methods required such as an expiration date of a TV program guide and questionnaire data which should be stored until user's answer.

Thus, there is a growing demand of a broadcast data receiving system that receives and extracts necessary data automatically and changes the processing method according to the type of received data.

3. Enhancing Active Database

In this section, we explain the super active database (SADB) system. Firstly, we outline the conventional active databases. Then, we explain the enhancement in the SADB.

3.1 Active Database

An active database is a database system which processes prescribed actions in the response to the occurrence of an event generated inside/outside of databases[4][5]. Behaviors of an active databases are described with ECA rules. Each ECA rule consists of three parts: the event, the condition and the action. The event part describes an event that can occur in the system. The condition part describes conditions for executing the following action. The action part describes operations to be carried out when the event occurs and the condition parts are satisfied.

Using ECA rules, we can describe system behaviors in the event-driven manner. Further, we can customize the system functions easily by adding ECA rules to the system or by deleting some of them from the system.

3.2 Super Active Database System

To apply the active database mechanism for receiving broadcast data, the following functions should be provided:

- **Receiving broadcast data**

  When the system receives broadcast data, it generates a event called ‘data receiving’. ECA rules that process a data receiving event choose to store or put aside the data.

- **Receiving ECA rules**

  The system can receive ECA rules. A sender can broadcast data with ECA rules such as rules for deleting the data when the data is expired, or rules for reporting user’s data accesses.

- **Timer**

  We can describe ECA rules processing the timer event. Using this function, we can describe various methods such as database optimization for every specific period of time, decreasing freshness of data and deletion of the expired data.

- **Sending/receiving data packets**

  The system can send/receive data packets and ECA rules to another SADB via LAN or Internet.

In addition, several other functions are provided such as grouping ECA rules and enabling/disabling an ECA rule.

3.3 ECA rule

In conventional active databases, only database operations such as SELECT, INSERT, DELETE, and UPDATE are concerned as an event in ECA rules. Further, active databases can carry out actions only concerning database operations. In this study, we enhance the description capability of ECA rules to fulfill various requirements of a broadcast data receiving system. Figure 1 shows the syntax of ECA rules in the SADB.

In the figure, ‘*Event name*’ represents the name of the event which triggers this rule. ‘*Definition of variables*’ defines the local variables used in this rule. ‘*Conditions*’ specifies the condition for executing the following actions. This part is described by the sequence

```
CREATE RULE  Rule name ON  Event name
      [  Definition of variables ]
      [ WHERE  Conditions ]
      THEN DO  Actions
```

Figure 1. Syntax of ECA rules

Table 1. Events provided by SADB

Name	Content
SELECT	Data retrieve
INSERT	Data insert
DELETE	Data delete
UPDATE	Data update
RECEIVE	Receiving a data packet
TIMER	Firing a timer

<*Expression 1*> <*operator*> <*Expression 2*>. Expressions 1, 2 are composed by database attributes or constant values. In ‘*Action*’, the operations to be carried out are described. Table 1, 2 show the events and the actions provided by SADB.

Moreover, SADB provides two system parameters, *NEW data* and *OLD data* for each event. The system sets the information shown in Table 3 to these parameters when the event occurs. These parameters can be used in ECA rules.

Figure 2, 3 show examples of ECA rules. In figure 2, *Rule1* is a rule to store the data whose content-type is ‘HotTopics’. These ECA rules realize filtering functions. *Rule2* is a rule for accounting. This rule writes accounting information when a user expends a pay data. *Rule3* is a rule for adding new functions. This rule is activated on receiving a broadcast ECA rule, then, the system stores the received rule if the sender is ‘Nishio-lab’. *Rule4* is a rule for managing a freshness of data. This rule decreases a freshness parameter of data in a regular period and deletes the data which is too old to be kept in the client’s database. If we use these ECA rules, we can achieve flexible cache management because we can change the handling method according to the contents of data and user’s interest.

On the other hand, it is possible to describe complex behaviors by chaining the execution of ECA rules. The set of rules in Figure 3 realizes the function to request specific information to another host by chaining three rules. *ClientRule1* is activated when a user refers to the data on the client, Then, the system requests the detail of the referred data to the server. *ServerRule1* is activated when receiving a request packet from *ClientRule1*. The rule retrieves the detail of information and sends a reply packet to the client. This packet activates *ClientRule2* and the detail information is displayed on the display of the client.

Table 2. Actions provided by SADB

Name	Content
QUERY([*expression*])	Database operation
SEND([*opponent*], [*expression*])	Send a data packet
INSERT_ECA([*description*])	Store an ECA rule
DELETE_ECA([*identification*])	Delete ECA rules
ENABLE_ECA([*identification*])	Activate ECA rules
DISABLE_ECA([*identification*])	Deactivate ECA rules
GROUPING_ECA([*expression*])	Grouping ECA rules
ADDTO_GROUP([*expression*])	ADD rules to group
RELEASE_GROUP([*expression*])	Remove rules from group
SET_TIMER([*condition*])	Set a new timer
KILL_TIMER([*identification*])	Remove a timer
VBSCRIPT([*expression*])	Execute a script

Table 3. Contents of NEW data and OLD data

Event	NEW	OLD
SELECT	Retrieved data	–
INSERT	Inserted data	–
DELETE	–	Deleted data
UPDATE	Data after update	Data before update
TIMER	Timer identification	–
RECEIVE	Received data	–

4. Design and Implementation

In this section, we explain the design and the implementation of SADB.

4.1 Assumptions on Environment

We assume the following characteristics:

Transmission rate: approximately 1Mbps (transmission rate of CS digital broadcasting in Japan).

Data format: multimedia files(HTML file, MPEG file, executable file and so on).

Data size of an item: dozens of kilo bytes (popular data size on data broadcasting services in Japan[2][6]). However, in this study, we suppose also a few mega bytes such as a large MPEG file.

Frequency of data arrival: approximately seven times per second calculated from **transmission rate** and **data size**.

Store rate: if a system stores data that users need, the store rate is approximately 1/400 calculated from **frequency of data arrival**.

4.2 System structure

Figure 4 illusrated the system structure of the SADB. The *Data Receiving Part* receives the broadcasted data and generates an event of data receiving. The *DB Management Part* accepts the demands of operation in terms of the database and processes the database operation.

```
create rule Rule1 on RECEIVE
 where new.header = 'HotTopics'
 then do
  QUERY( "INSERT INTO DataStore
         VALUES( new.id,new.data )" );
create rule Rule2 on SELECT
 where new.price > 0
 then do
  QUERY( "UPDATE PayInfo SET ACOUNTING
         = ACOUNTING + %s", new.price );
create rule Rule3 on RECEIVE
 where new.header = 'ECA_RULE'
      new.from = 'Nishio-lab'
 then do
  INSERT_ECA( new.eca_difinition );
create rule Rule4 on TIMER
 where new.name = 'UPDATE_TIMER'
 then do
  QUERY( "UPDATE DataStore SET Verdure
          = Verdure - 1" );
  QUERY( "DELETE * FROM DataStore
          WHERE Verdure = 0" );
```

Figure 2. An example of ECA-Rules

```
create rule ClientRule1 on SELECT
  then do
   SEND( 'DataServer', 'RequestDetail',
         'name:', new.name );
create rule ServerRule1 on RECEIVE
 String ResponseData
 where new.header = 'RequestDetail'
 then do
  ResponseData =
    QUERY( "SELECT FROM DetailData
           WHERE Name=%s", new.name );
   SEND( new.from, 'ResponseDetail',
        'data:', ResponseData );
create rule ClientRule2 on RECEIVE
 where new.header = 'ResponseDetail'
 then do
  VBSCRIPT( "MSGBOX \"%s\"", new.data );
```

Figure 3. An example of chain ECA-Rules

Then, it generates a database event. The *Rule Processing Part* detects events generated on the system and processes them. This part retrieves ECA rules in response to an generated event and evaluates the condition of these ECA rules. Then, it processes the action of the ECA rule. The *Application Interface Part* communicates with other SADBs or applications. The *User Interface Part* provides a graphical user interface for user to input ECA rules and operate the database.

4.3 Rule Processing Method

We implemented the *Rule Processing Part* with a conventional technology decreasing the processing cost of ECA rules such as a transformation technique for storing ECA rules into an intermediate format and a clustering technique of the ECA rules.

However, supposing a satellite broadcasting environment, this mechanism may not be suitable for keeping the processing speed, on account of the rule evaluation process, enough to process a large volume of received data. A data receiving event is generated every time when a broadcasting data arrives. Note that we supposed 7 times per second and it is very highly costed for the system to retrieve rules corresponding to every event

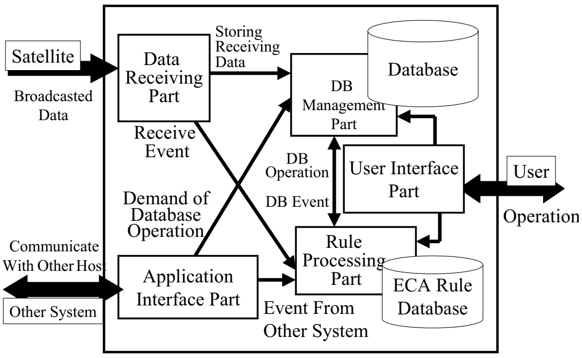

Figure 4. The system structure of SADB

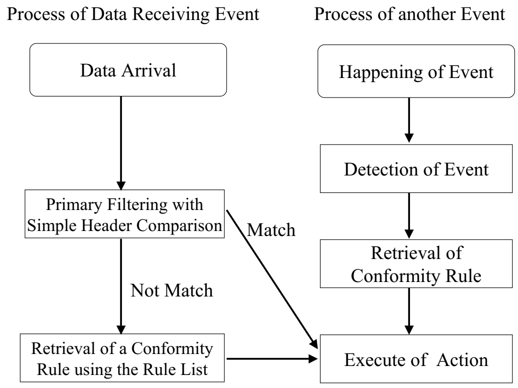

Figure 5. The rule processing method in the high-speed mode

from the *ECA Rule Database.*

Therefore, we implemented the processing mode specialized for a broadcasting environment(high-speed mode). Generally, an active database system can increase its processing speed by an optimization of ECA rules and the control of an execution sequence of ECA rules[1]. However, since the SADB is an active database system specialized for a broadcasting environment, we perform improvement in the speed which directed our attention to the frequency of the event occurence. In this system, we assume that a data receiving event occurs frequently and continuously. Even if events occur no less than 7 times in 1 second in a practical environment, a data storing event hardly occurs in a data receiving system. Moreover, the event invoked by the user's operation or application is generated less frequently compared with data receiving event.

From the above observation, the processing time of a data receiving event often becomes the bottleneck of the system performance. Then, we aim at eliminating this bottleneck in the high-speed mode to perform a primary filtering of a data receiving event.

The flow of rule processing in the high-speed mode is shown in Figure 5. When the system registers a rule to process a data receiving event, the rule is not stored in the *ECA Rule Database* but is translated into the intermediate format, and stored into the *Rule List* on the *Data Receiving Part.* The storing format of the *Rule List* excludes the event information from generally stored in *ECA Rule Database.* Nevertheless, we cannot expect so much improvement in the speed using the *Rule List.*

Then, the system registers that rule into a *Header Comparison List* if the condition part of this ECA rules is a simple header comparison. Generally, the simple header comparison fills the demand of the most of filtering. Therefore, most ECA rules are adapted to this type. When a certain data arrives, the *Data Receiving Part* checks whether there is a rule which suits the received data by referring to the *Header Comparison List.* After that, if the received data does not match the *Header Comparison List*, the *Data Receiving Part* searches the *Rule List.* If the *Data Receiving Part* decides that the received data is unnecessary, it does not notify the *Rule Processing Part*, but the next data is processed.

In this way, it saves the processing time to carry the information of received data to the *Rule Processing Part* and to retrieve rules from the *ECA Rule Database.* Moreover, when received data does not match the primary filtering but the data is filtered using the *Rule List*, we can expect the improvement of the speed compared with the conventional method because the system must evaluate the condition part of each rule using the result of primary filtering.

We describe the evaluation of this method by actual measurement in section 6.

5. Application

In this section, we describe the broadcast data receiving application using the SADB for showing that various services in broadcasting environments can be realized with the SADB.

By using the rule mechanism of SADB, we can deal with broadcasting data flexibly. As an application which actually use the rule mechanism, we propose the Active Information Store(AIS) which stores and accesses required information by integrating the SADB and a user oriented information filtering mechanism with a tree structure. Because the AIS provides a tree structure as a user interface too, users can access information only by tracing the tree structure.

A tree structure expresses a user's interest and the system dynamically change this structure according to the access pattern of the user. The concrete algorithm is described in [8]. Generally, in order to receive the current broadcast service, a dedicated application is needed and a user has to use a different application for data accessing for each service. Then, the AIS unifies the data which are broadcast from various information sources with the SADB and it enables the system to handle broadcast data seamlessly. Therefore, a user can access data without

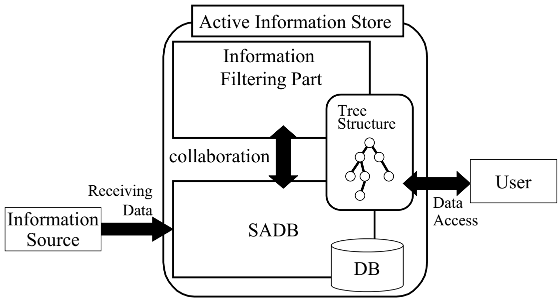

Figure 6. The structure of the Active Information Store

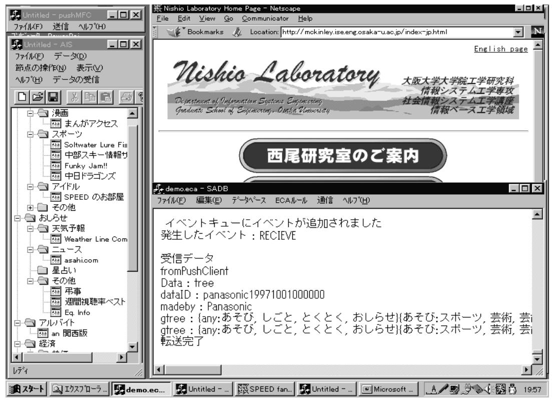

Figure 7. A snapshot of the prototype system

being aware of the difference by the information sources.

The structure of the AIS is illustrated in Figure 6. In the figure, the *Information Filtering Part* generates ECA rules according to the user's interest expressed by the tree structure and sends them to the SADB. Moreover, this part detects the change of the user's interest by access patterns of the user. When this part detects the change, new ECA rules are generated and the system changes the tree structure according to the user's new interest. In this way, the user can access data via the new tree structure. In addition, using ECA rules, the AIS provides various other functions including the function which changes the evaluation value of data in response to progress of time, detects the expiration of time-limited data and deletes it, and exchanges a filter information by exchanging ECA rules with other SADBs.

A snapshot of the prototype system is shown in Figure 7. We should mention that we do not use broadcasting services via satellites in the prototype. In this research, we implemented the data broadcasting server which broadcasts data i) broadcasted with a data-broadcasting service on the internet, ii) broadcasted with ground wave data-broadcasting service, iii) which is the homepage data on the internet collected using the robot program which is developed in this study. Then, the prototype system of the AIS receives and shows data broadcasted from this server.

The broadcasting data covers a lot of types such as an *HTML document*, *plane text*, *graphic file*, and *sound file*. The amount of data which the broadcasting server actually broadcasts was approximately 1000 items / 8M bytes per a day. The AIS receives, filters, and classifies them. The *User Interface Part* handles the received data from different information sources in a uniform manner. Therefore, a user can use data without being conscious of a difference in information sources. When a user chooses an item on the user interface, the content of the selected item are displayed on a WWW browser. As the WWW browser, we use the *Netscape Communicator 5*.

In our experimental use of this system, the number of ECA rule was approximately 20 at the start time. After being used for 1 week, the number became 26. Although the number of rules increases by the change of the user's interest, unnecessary rules are deleted and the number of rules in the whole system becomes stabilized in a long run. In this way, we examined dozens of rules is enough to operate the system.

6. Evaluation and Related Work

In this section, we discuss the proposed system by showing the performance evaluation of the rule processing capability of the SADB.

6.1 Evaluation

We implemented and evaluated the high-speed rule processing mechanism described in section 4. We evaluated the system using the AIS actually running on personal computer(Windows98, Mobile PentiumII 266MHz, 128Mbyte Memory). We measured the rule processing time of a conventional method and the high-speed mode when the implemented broadcast server broadcasts HTML and image data repeatedly.

The results are presented in Figure 8, 9, and 10. The broadcast item consists of two or more HTML files and image files. The size of each data is approximately dozens of Kbytes. Moreover, we adjusted broadcast data so that the rate of storing data becomes about 1/400 according to the environmental assumption described in section 4.1. Furthermore, the transmission rate of the broadcast server was set to 1Mbps. Events without a data retrieving event were made to occur at random(1 time per second in the average). In all graphs, the rule processing time for 30 events is shown in the vertical axis.

Figure 8 shows the change of the processing time in the case of changing the number of ECA rules in the system from 10 to 50. In general applications of the SADB, a system usually handles approximately 10 to 20 of ECA rules. However, ECA rules for filtering increases according to the complexity of filtering. This graph shows that the increase in processing time in the response to

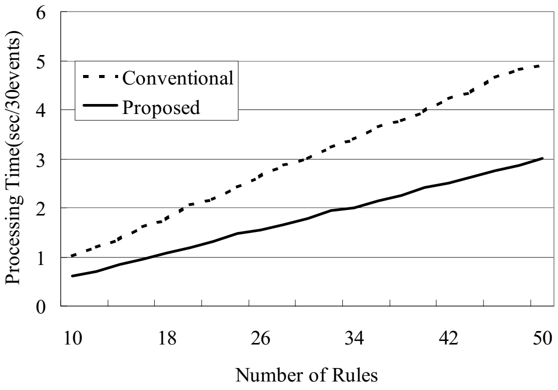

Figure 8. The system performance in the response to the number of ECA-Rules

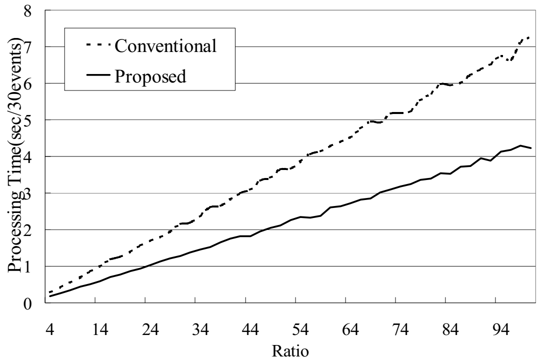

Figure 9. The system performance in the response to the receiving-rule ratio

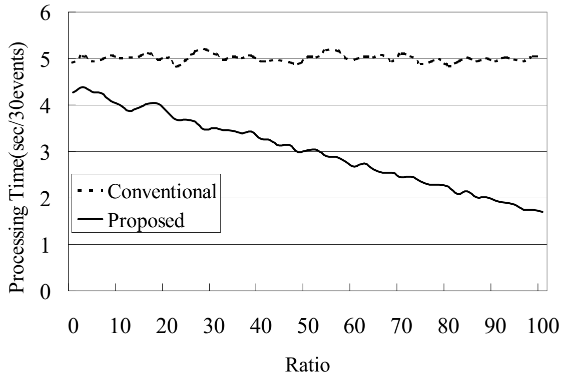

Figure 10. The system performance in the response to the complexity of ECA-Rules

the increase of the number of ECA rules is suppressed in the high-speed mode compared with the conventional method.

Figure 9 shows the change of the processing time in the case that the number of rules is fixed to 50. Here, we changed the ratio of the number of the rule for handling the data receiving event. Since most events are data receiving events, if the rate of the number of a data receiving rule increases, the processing time increases in both methods. If there are few rules for receiving data, the difference of both methods hardly appears because they perform the same processing. However, the merit of this method comes out as the ratio of the number of rules processing a data receiving event becomes large.

In general, if the number of rules increases, the ratio of the number of rules processing a data receiving event becomes high. This is because many additional rules are required for filtering. From the above, the proposed method is effective in the broadcasting environment.

Figure 10 shows the change of processing time in the case that all rules process the data receiving event and we change the ratio of the number of a rule using a simple header comparison regarding to a rule using the *Rule List*. As described in section 4.3, the use of a simple header comparison rule contributes to the improvement of the processing speed. Especially, in the case that all rules are simple header comparison, it is accelerated sharply compared with the conventional method.

In the assumed environment, the minimum number of ECA rules required is small and additional rules are those for filtering received data. Moreover, many filtering rules perform a filtering by simple comparisons such as “sports data is needed” and “data related to animation is unnecessary” in the practical sense. Therefore, the proposed method is suitable to the assuming environment.

However, using proposed method, the system cannot guarantee the execution order of ECA rules because the rule processing is done by two modules in the method. Therefore, the system has to use the conventional method if the execution order should be guaranteed.

As for the event-processing capability, the evaluation result shows that the processing capability in the high-speed mode is approximately 10 to 50 events per a second. It turns out that this system can be useful in such cases even if there are approximately 50 ECA rules in the system. However, if the bandwidth increases in the future, more efficient methods are needed.

6.2 Related Work

As related work of the system, there is an extended system of an active database like SADB, Active Mobile Database System(AMDS)[9]. The AMDS is an active database system proposed as a base system of a mobile computing environment. The AMDS has similar functions as the SADB, such as the detection of receiving packet, and the transmission/reception of ECA rules. The aim of the AMDS is the management of mobile hosts and collecting data from mobile host in a mobile computing environment. On the other hand, that of the SADB is a base system in the broadcast environment to receive and filter data. Therefore, the AMDS has the functions of manegement of mobile hosts such as detecting connection to/disconnection from the AMDS's cell. However both of the systems have similar functions, the SADB has to process the great deal of events because the data receiving event of the SADB occurs much more than the connecting/disconnecting event of the AMDS. Consequently, the SADB has the function that process the generated event in the high speed mode.

7. Conclusion and Future Work

In this paper, we designed and implemented the super active database system, a system for efficiently receiving broadcast data. Our system can deal with various kinds of data without being conscious of the difference of data types. Moreover, we evaluated the performance of the SADB and quantitatively shown the amount of data which can be dealt by the proposed system.

In the future, we will improve the processing mechanism of ECA rules corresponding to the increase of traffic. What is more, we will consider the safety of a system such as the detection of an infinite loop of ECA rules, the management of access levels[3][10].

Acknowledgement

This research was supported in part by the Special Coordination Funds for Promoting Science and Technology from the Ministry of Education, Culture, Sports, Science and Technology, Japan and Grant-in-Aids for Scientific Research numbered 13780331 from Japan Society for the Promotion of Science.

References

[1] A. Aiken, J. Widom, J. M. Hellerstein, "Behavior of database production rules: termination, confluence, and observable determinism," *in Proc. ACM SIGMOD International Conference on Management of Data*, pp. 59–68, 1992.

[2] ADAMS Homepage, http://www.tv-asahidata.com/.

[3] F. Llirbat, F. Fabret and E. Simon, "Eliminating Costly Redundant Computations from SQL Trigger Executions," *in Proc. the ACM SIGMOD International Conference on Management of Data*, pp. 428–439, 1997.

[4] G. Lohman, et al. "Extentions to Starburst: Object, Types, Functions, and Rules," *Communications of the ACM*, Vol. 34, No. 10, pp. 94–109, 1991.

[5] H. Ishikawa, et al.: "An active object-oriented database: a multi-paradigm approach to constraint management," *in Proc. the 19th VLDB Conference*, pp. 467–478, 1993.

[6] PointCast Japan Homepage, http://www.pointcast.ne.jp/.

[7] S. Acharya, M. Franklin, and S. Zdonik, "Prefetching from a Broadcast Disk," *in Proc. 12th International Conference on Data Engineering*, pp. 276–285, 1996.

[8] S. Sanguantrakul, T. Terada, M. Tsukamoto, S. Nishio, K. Miura, S. Matsuura, T. Imanaka, "A User Customized Selection and Categorization for Broadcast Data," *in Proc. International Workshop on Multimedia Network Systems*, pp. 596–601, 1999.

[9] T. Murase, M. Tsukamoto, and S. Nishio, "A System Platform for Mobile Computing base on Active Database," *in Proc. International Symposium on Cooperative Database Systems for Advanced Applications*, vol. 2, pp. 424–427, 1996.

[10] Y. E. Ioannidis and T. K. Sellis, "Conflict Resolution of Rules Assigning Values to Virtual Attributes," *in Proc. the ACM SIGMOD International Conference on Management of Data*, pp. 205–214, 1989.