



放送型データ受信のためのアクティブデータベース システムの設計と実装

寺田, 努
塚本, 昌彦
西尾, 章治郎

(Citation)

電子情報通信学会論文誌. D-I, 情報・システム, I-情報処理, J83-D-I(12):1272-1283

(Issue Date)

2000-12-25

(Resource Type)

journal article

(Version)

Version of Record

(Rights)

copyright(c)2000 IEICE

(URL)

<https://hdl.handle.net/20.500.14094/90002085>



論文

放送型データ受信のためのアクティブデータベースシステムの設計と実装

寺田 努[†] 塚本 昌彦[†] 西尾章治郎[†]

Design and Implementation of an Active Database System for Receiving Broadcast Data

Tsutomu TERADA[†], Masahiko TSUKAMOTO[†], and Shojiro NISHIO[†]

あらまし 近年、放送衛星や通信衛星の新たな打ち上げや新サービスの開始により、衛星を用いたデータ放送に対する注目が高まっている。このような放送型のシステムが普及することで、放送量、サービスの種類とともに飛躍的に増加する。そのため、一般ユーザが大量に放送されるデータから必要な情報を効率的に抽出し、格納、再利用できるようなシステムが望まれている。このような要求に対し、本論文では、イベント駆動型データベースであるアクティブデータベースを放送型システムに適応させたスーパーアクティブデータベース (SADB: Super Active DataBase system) を提案し、その設計と実装について述べる。アクティブデータベースはシステムの動作が柔軟に記述できるため、様々な種類のデータを受信するのに適している。SADB は、大量に送られてくるデータを高速に処理する必要があるため、受信動作及び受信データ処理をルール処理部から独立に設計した。本論文では実装したシステムの性能を評価して対処可能なデータ量を示す。

キーワード 放送型データ、アクティブデータベース、ECA ルール

1. ま え が き

近年、放送衛星や通信衛星の新たな打ち上げ、デジタル放送の開始などにより、衛星を用いた放送型システムに対する注目が高まっている。今後新たに衛星が打ち上げられることで、放送されるコンテンツの種類、量ともに大幅に増加することが予測できる。また、DirecPC (ヒューズ・ネットワーク・システムズ社) や Sky PerfecPC! (日本デジタル放送サービス) などに見られるように、デジタル放送を利用することで、コンピュータで用いるデータなどのデジタル情報を放送するといった新たなサービスも多数提供されるようになると考えられる。そのため、受信側には様々な種類のデータが大量に放送されることになり、ユーザがこの大量の受信データの中から必要な情報を見つけ出し、効率的に利用することは困難となる。

これまで、放送型システムに関する研究はサーバ側

のデータ放送戦略に関して多く行われている [4], [9]。しかし、大量のデータが放送される環境においては、データ放送戦略とともに、いったんデータを蓄積し、必要なデータを再利用することの重要性が非常に高い。また、データ放送においては、コンピュータの実行プログラムなどのように最初から蓄積を前提としたデータが放送される場合が多い。

このような観点から、筆者らはアクティブデータベースを放送型システムに適応させることで、大量の放送型データを効率良く受信・蓄積し、ユーザが蓄積された情報に手軽にアクセスできるようなシステムを提案する。提案するアクティブデータベースは、簡単な言語を用いて動作を記述することで、多種のデータを受信可能な放送型データ受信基盤として利用することをねらいとする。今回実現したデータベースシステムは、受信データを高速に処理する必要があるため、受信動作記述や受信データ処理部分をルール処理部から独立に設計して高速化を行っている。また、本論文ではいくつかのアプリケーションを構築し、提案するアクティブデータベースを用いることで放送環境における様々なサービスが実現できることを示す。更に、

[†] 大阪大学大学院工学研究科情報システム工学専攻, 吹田市
Department of Information Systems Engineering, Graduate
School of Engineering, Osaka University, Suita-shi, 565-0871
Japan

SADB の性能を実測評価することで、提案方式の有効性を示す。

以下、2. では放送型システムの特徴について述べ、3. ではアクティブデータベースとその放送型データ受信のための拡張について述べる。4. では、システムの設計と実装について述べ、5. では本論文で提案するシステムを用いたアプリケーションについて述べる。6. で評価及び関連研究との比較を行い、最後に7. でまとめと今後の課題について述べる。

2. 放送型システム

衛星を用いた放送型システムは以下の特徴をもつ。

- 同報性

衛星放送は同報通信であるため、品質を落とすことなく多数のユーザに情報を提供可能である。反面、それぞれのユーザに提供できるサービスの内容は画一的になり、ユーザに特化したサービスの提供は困難となる。

- 広帯域性

衛星を用いることにより、大量のデータを送信することができる（一般に数百 kbit/s から数十 Mbit/s）。

- 帯域幅の非平衡

放送の帯域（下り回線）はたいへん広いが、上り回線は存在しないか、存在しても通常は電話回線などの狭い帯域しかもない。

このように、放送型システムでは、受信される可能性のあるデータがすべて送信されることになる。また、広帯域を利用することで放送される情報量が増えるためにデータ量は膨大になり、データの内容もより細分化されることになる。このような状況では、一般のユーザが番組表を片手に膨大な情報を処理することは非常に困難である。

データ放送では、アンケートや申込書のフォームなどのように一度利用すれば蓄積する必要がないものだけでなく、番組表などのように期限付きであるものや、天気予報などのように同じコンテンツであるが内容だけは頻繁に更新されるものなど様々な種類が考えられる。一度利用すれば蓄積する必要のないデータはクライアントのキャッシュから即座に削除されるべきであるし、番組表は参照されても期限までは残しておくべきである。したがって、放送されるデータを効率的に処理でき、更にデータの種別によってそのデータの扱いを変更できる柔軟な放送型データ受信システムに対する要求が高まっている。

3. アクティブデータベースの拡張

本章では、本論文で提案するスーパーアクティブデータベースの概要について述べる。まず、従来のアクティブデータベースについて説明し、次にスーパーアクティブデータベースにおける拡張の部分について述べる。

3.1 アクティブデータベース

アクティブデータベースとは、従来のデータベースと異なり、データベース内部で起こる事象を監視し、あらかじめ定義された条件に適應する事象の発生に反応して自動的に更新等の操作を行うデータベースシステムである [3]。アクティブデータベースはイベント駆動型データベースと呼ばれ、その動作は発生する事象（イベント）、実行させるための条件（コンディション）、イベントによって発火する操作（アクション）の三つの組からなる ECA ルールによって記述される。

アクティブデータベースを放送型データ受信に利用することで、以下のような利点が生まれる。

- イベント駆動型であるため、データの到着をイベントとして検出する機構を用意しておけば、到着データに対する要求をルールに書いておくだけで、必要な動作が自動的に実行される。システムの機能はこのような要求の集合として表現されることになる。

- アクティブデータベースの機能はすべて ECA ルールの組で表現されるため、ECA ルールを部分的に変更することで機能のカスタマイズが可能である。また、ECA ルールの追加、削除により容易に機能の追加、削除が行える。

- データベース内容の変化を検出することができるため、データの期限切れや時間によるデータの価値の変化などに対する処理が自由に記述できる。また、データの参照をイベントとして検出できるため、例えば「有料データのアクセスによる課金」といった要求が記述できる。

3.2 スーパーアクティブデータベース

本節では、放送型データ受信基盤として利用するために拡張したアクティブデータベースについて述べる。衛星を用いた放送環境では、取り扱うデータの量が増大するため、大量のデータを高速に処理することが目標となる。そこで、提案するシステムをスーパーアクティブデータベースシステム（Super Active Database System : SADB）と名づける。SADB は通常のアクティブデータベースの機能に加え、以下の機

能をもっている。

- 放送データ受信機能

次々と放送されるデータを受信し、データ受信イベントを発生させる機能をもつ。データ受信イベントを処理する ECA ルールのアクションによって到着データを格納でき、その ECA ルールのコンディション記述を用いて、自動的にデータの取捨選択ができる。

- ECA ルール受信機能

送信元が ECA ルールを送信できる。放送元は、放送データとともに、その放送データがある条件を満たしたときに削除するルールや、データが参照されたことを確認し、放送元に伝える ECA ルールを送信できる。この機能により、従来困難であったデータ送信側からのシステム操作が可能になる。

ルール受信判定は受信側の ECA ルールにより行うため、機能グループや送信元によってルールを受信するかどうかを判断することができる。受信判定は受信側ですべて行うため、送信側は受信側にあるルールセットの整合性を保証することはできない。不整合が起こった場合の検出や修正は今後の課題である。

- タイマ処理機能

タイマ処理を可能にすることにより、一定時間ごとに自動的にデータベースを最適化したり、データの新鮮さを表すパラメータの減少処理や、有効期限付きのデータに対する処理を行うことができる。タイマの条件設定には様々な方法を提供しており、柔軟な条件指定が可能である（詳細は付録参照）。

- データパケットの送受信機能

LAN やインターネットを介して接続されているシステム間でデータベース内のデータや ECA ルール等を送受信できる。例えば、あるシステムで最適化されたフィルタリング条件を他のシステムに渡したり、必要とするデータがあったときに、放送されるまで待つのではなく、他のシステムから取ってくる事が可能になる。

そのほかにも、ECA ルールをグループ化して機能ごとにまとめたり、それぞれの実行可能・不可能を切り換える機能により、ユーザごとに実行できるルールを切り換えて複数ユーザの利用に対応するなど、放送データ受信のための様々な機能を提供している。

3.3 SADB の ECA ルール

前節で述べた機能を実現するため、SADB では、従来のアクティブデータベースがサポートする ECA ルール記述に拡張を加えている。SADB における ECA

ルールの記述構文を図 1 に示す。「イベント名」には、ルールが対象とするイベントの名前を記述し、「対象テーブル」には、イベントが対象とするテーブルを指定する。「変数型宣言」では、ECA ルール中で使用するローカル変数を定義する。宣言する変数は、文字列変数、タプル変数またはタプルの組を表す変数のいずれかである。「コンディション」にはルールの発火条件を記述する。記述は、「< 左辺 > < オペレータ > < 右辺 >」の形の羅列で行う。両辺には、データベース属性や変数を指定する。「アクション」にはルールが発火したときに行う動作について記述する。

SADB で利用することができるイベント、アクションを表 1、表 2 に示す。表 2 に示したアクションのうち、SET_TIMER アクションのタイマ条件設定の詳細については、付録に示す。また、NEW データと OLD データをシステム変数として用意する。これは、到着したデータの内容を用いて処理を行うルールでは、イベント対象となったタプル情報などが必要になるためである。イベント発生時にこれらの変数に必要な情報が自動的に格納され、ルール中で自由に使用することができる。各イベントに対する NEW データ、OLD データの内容を表 3 に示す。

3.4 ECA ルール例

SADB の ECA ルール記述例を図 2、図 3 に示す。ルール 1 は、データを受信したときに、その内容が「HotTopics」であればデータの格納を行う、というルールである。このようなルールを複数用意しておくことにより、フィルタリング機能が実現できる。ルール 2 は有料データに対する課金を行うためのルールであり、

```
CREATE RULE ルール名 ON イベント名
[ TO 対象テーブル ]
[ 変数型宣言 ]
[ WHERE コンディション ]
THEN DO アクション
```

図 1 ECA ルールの記述構文
Fig. 1 Syntax of ECA-rule.

表 1 SADB のイベント
Table 1 The events of the SADB.

名称	内容
SELECT	テーブルに対するデータ参照
INSERT	テーブルに対するタプルの挿入
DELETE	テーブルのタプル削除
UPDATE	テーブルのタプル更新
RECEIVE	データパケットの受信
TIMER	設定したタイマの発火

表2 SADBのアクション
Table 2 The actions of the SADB.

名称	内容
QUERY ([クエリ内容])	データベース操作
SEND ([宛先], [送信内容])	データの送信
INSERT_ECA ([ルール内容])	ECA ルール格納
DELETE_ECA ([ルール識別子])	ECA ルール削除
ENABLE_ECA ([ルール抽出条件])	ECA ルール有効化
DISABLE_ECA ([ルール抽出条件])	ECA ルール無効化
GROUPING_ECA ([グループ名],[条件])	ECA ルールグループ化
ADDTO_GROUP ([グループ名],[条件])	グループにルール追加
RELEASE_GROUP ([グループ名])	グループ化解除
SET_TIMER ([タイマ条件])	新たなタイマの設定
KILL_TIMER ([タイマ識別子])	タイマの削除
VBSCRIPT ([スクリプト文字列])	スクリプト実行

表3 NEW データと OLD データの内容
Table 3 Data contained in the NEW-data and OLD-data.

イベント	NEW	OLD
SELECT	参照タプル	-
INSERT	挿入タプル	-
DELETE	-	削除タプル
UPDATE	対象タプル (更新後)	対象タプル (更新前)
RECEIVE	到着パケット内容	-
TIMER	タイマ識別子	-

データが参照されたときに、そのデータが有料であればユーザに対して課金情報を書き込む。ただし、実際に課金管理を行うには、このようなルール以外にセキュリティ管理など数多くの機能が必要となる。ルール3は機能追加のためのルールであり、ECAルールが放送されてきたときに、その放送元が‘Nishio-lab’であればそのECAルールを信頼してシステムに格納するルールである。ルール4は、データの新鮮さを管理するルールであり、一定時間ごとにデータの新鮮さを減少させ、古くなったデータは自動的に削除する。データのキャッシング戦略において、従来研究では主にデータの内容や関連性を考慮せずに行うものがほとんどであった[6]。文献[10]では、データアイテム間の相関性を考慮するなど、従来研究を拡張したキャッシュ戦略を提案しているが、本論文で述べるように、ECAルールを用いたキャッシュ管理を行うことで、データの内容やユーザの興味を考慮したキャッシュ管理を行うことができる。

ECAルールは連鎖的に実行させることにより、複雑な動作記述可能である。図3に示す三つのルールは、連鎖的に実行させることで、他のサーバへの情報要求機能を実現している。クライアント上でユーザがデータを参照したとき、クライアントルール1が発火

```
create rule ルール1 on RECEIVE
where new.header = 'HotTopics'
then do
  QUERY( "INSERT INTO DataStore
VALUES( new.id,new.data )" );
create rule ルール2 on SELECT
where new.price > 0
then do
  QUERY( "UPDATE PayInfo SET ACCOUNTING
= ACCOUNTING + %s", new.price );
create rule ルール3 on RECEIVE
where new.header = 'ECA_RULE',
new.from = 'Nishio-lab'
then do
  INSERT_ECA( new.eca_definition );
create rule ルール4 on TIMER
where new.name = 'UPDATE_TIMER'
then do
  QUERY( "UPDATE DataStore SET Verdure
= Verdure - 1" );
  QUERY( "DELETE * FROM DataStore
WHERE Verdure = 0" );
```

図2 単体で動作するECAルールの例
Fig. 2 An example of standalone ECA-rules.

```
create rule クライアントルール1 on SELECT
then do
  SEND( 'DataServer', 'RequestDetail',
'name:', new.name );
create rule サーバルール1 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 クライアントルール2 on RECEIVE
where new.header = 'ResponseDetail'
then do
  VBSCRIPT( "MSGBOX %s", new.data );
```

図3 連鎖的に動作するECAルールの例

Fig. 3 An example of chain ECA-rules.

し、ユーザの参照したデータに関する詳細情報をサーバに要求する。要求パケットによりサーバルール1が自動的に発火し、詳細情報を検索して結果パケットを返す。結果パケットにより、クライアントルール2が発火し、受信した詳細情報を表示する。

4. システムの設計と実装

本章では、まずSADBを実現する上で想定する環境について述べ、次にシステムの設計及び実装とルール処理の詳細について述べる。

4.1 想定する負荷とデータ量

本節ではSADBの要求仕様を明確にするため、SADBの想定環境を示す。

● 伝送速度

実環境において、現在CSデジタル放送の伝送速度は1Mbit/s程度であり、本研究で想定している放送の伝送速度も同様に1Mbit/s程度とする。

● データ形式

放送データの形式は、PC用のデータやHTMLデー

タなどであり、一般の番組を放送する場合も、MPEGなどのファイルとして送られるとする。また、それぞれのデータアイテムにはキーワードなどのメタ情報を示すヘッダが付加されているとする。

● データサイズ

現在の地上波データ放送などでは、一般に一つのデータアイテム (HTML データや画像データなど) は、数十 k バイト程度である。本研究においても 1 アイテムのデータサイズは数十 k バイト程度と想定するが、番組を動画ファイルとして送る場合のようにファイルサイズの上限は数百 M バイトまで大きくなる可能性があるとする。

● データ到着頻度

伝送速度を 1 Mbit/s, データサイズを 20 k バイトから 100 M バイトとした場合、データの到着頻度は 1 秒当たり 1/1000 回程度から 7 回程度になる。

● データの格納率

受信したデータの格納する確率はアプリケーションによって様々であるが、大きく二つに分けられる。

ー データの閲覧用アプリケーション

ユーザが放送データを閲覧するために用いるアプリケーションでは、ユーザが見ないデータを蓄積する必要がない。したがって、データの格納率はユーザが閲覧可能なデータ量に依存する。放送データが HTML データであった場合、ユーザが 1 アイテムを閲覧するのに 1 分かかったとすると、格納率は 400 分の 1 程度となる。実際は常にユーザがアプリケーションを利用しつづけるわけではないので、平均すると格納率は更に下がると考えられる。

ー 集計などの機械処理を行うアプリケーション

放送データ受信システムでは、ユーザがデータを閲覧するためだけではなく、集計処理を行ったり、データの一部を利用するといったことも行う。このような場合、格納率はユーザがデータを閲覧する量ではなく、クライアントの空きディスク容量等の要因によって制限される。

ヘッダを利用して統計処理などを行う場合、もしヘッダ自体を格納しなくてよい場合は格納率は 0 となる。一方、ヘッダ情報は常に格納する場合の格納率はほぼ 100% となる。また、到着データの一部を利用し、データの再構成を行ってから格納するようなアプリケーションでは、データを単に閲覧するためのアプリケーションに比べて高確率でデータが蓄積されることになる。

4.2 システムの構成

SADB のシステム構成を図 4 に示す。データ受信部では、放送されるデータを受信し、データ受信イベントを発生させる。DB 管理部では、データベースに対する要求を受け、実際にデータベースの操作を行い、データベース操作イベントを発生させる。ルール処理部では、システム上に起こるイベントを検出し、ECA ルールの検索、コンディションの評価を行い、アクションを実行する。アプリケーションインタフェース部では、オンラインでつながる他の SADB やアプリケーションとの通信を行う。ユーザインタフェース部では、ユーザにグラフィカルなインタフェースを提供し、ECA ルールの入力や、データベース操作などを容易に行える環境を提供する。

4.3 SADB のルール処理

SADB 上で ECA ルールが入力されたり、ECA ルールを受信すると、そのルールは、高速化のため中間形式に変換された後に ECA ルールデータベースに格納される。更に格納 ECA ルールのクラスタリングを行うことで高速な ECA ルール検索を実現する。何らかのイベントが発生した場合、ECA ルールデータベースを検索し、ヒットするルール群を抽出してコンディションの評価、実行を行う。

しかし、このようなルール処理機構では、衛星放送を想定した場合、ルール評価のプロセスがボトルネックとなり、実用に耐え得る処理速度が得られない可能性がある。4.1 で示した想定環境においては 1 秒間に 7 回程度データを受信することになるが、データ受信ごとにイベントが発生するため、それぞれのイベントに対していちいちルールデータベースを検索し、マッチするイベントを探すことはコストの高い処理である。

そこで、SADB に放送に特化した処理モード (高速

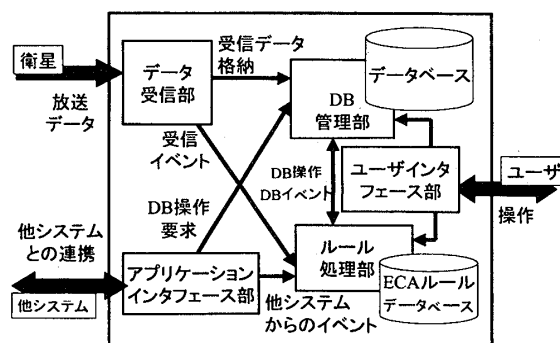


図 4 SADB のシステム構成

Fig. 4 The system structure of the SADB.

モード)を実装した。一般に、アクティブデータベースではECA ルールの最適化や実行順序の制御により高速化を行っている[2],[11]。しかし、SADBは放送に特化したアクティブデータベースであるという特徴をもつため、イベントの発生頻度に注目した高速化を行う。衛星放送をターゲットとした本システムでは、受信イベントが頻繁に、また継続的に発生する。想定環境においては1秒間に7回ものイベントが発生するが、データ閲覧アプリケーションではデータ格納イベントはほとんど発生しない。また、検索イベント等のユーザやアプリケーションによって起こるイベントも、データ受信イベントに比べて非常に少ない頻度でしか発生しないことになる。

以上のことから、データ受信イベントの処理時間がシステムのパフォーマンスのボトルネックとなる。そこで、高速モードでは、受信イベントの1次フィルタリングをデータ受信時に行い、ボトルネックの解消を図る。

高速モードのルール処理の流れを図5に示す。データ受信イベントに対するECAルールをシステムに格納するとき、そのルールは、一般のECAルールデータベースには格納せず、中間言語に分解されてデータ受信部がもつルールリストに登録する。ルールリストの格納形式は、ECAルールデータベースに格納する情報からイベント情報を省いたものであり、ルールリストを利用すること自体にはそれほど高速化の期待はできない。そこで、更にそのコンディションが単純なヘッダ比較であった場合、そのルールをヘッダ比較リストに登録する。一般に、データ受信時にはヘッダ比較によってフィルタリングを行うため、このタイプのECAルールがほとんどとなる。あるデータが到着

したとき、データ受信部ではまずヘッダ比較リストを参照して適合するものがないか調べる。その後自分もつルールリストを検索する。データが不要と判断された場合、ルール処理部にイベントを通知せず、次のデータ処理に移る。こうすることで、イベントとしてルール処理部にデータを渡し、ECAルールデータベースを用いて処理を行う手間が省け、効率の良い処理が実現できる。また、1次フィルタリングにマッチせず、ルールリストを用いて処理を行う場合も、1次フィルタリングの結果を用いてコンディション判定を行うため、従来の手法に比べて多少の高速化が期待できる。本方式の実測による評価については6.で述べる。

5. アプリケーション

本章では、SADB及びSADBが提供するルールシステムを用いることで、放送環境における様々なサービスが実現できることを示すため、SADBを用いたいくつかのアプリケーションについて述べる。まず、データ受信側のアプリケーションとして、木構造をもつ情報フィルタリング機構とSADBを統合した放送データ受信、閲覧アプリケーションであるアクティブ情報ストアについて述べる。次に、放送側のアプリケーションとして、ECAルールを用いてユーザに自動的に情報を配信するオーサリングアプリケーションについて述べる。

5.1 アクティブ情報ストア

SADBのルール機構を用いることで、放送されるデータを柔軟に取り扱えるようになった。そこで本研究では、実際にルール機構を使用したアプリケーションの例として、SADBにユーザ適応型のフィルタリング機構を統合し、必要な情報を容易に取得、閲覧できるアクティブ情報ストア(Active Information Store: AIS)を構築し、プロトタイプの実装を行った。

AISは、データ受信基盤にSADB、フィルタリングアルゴリズムに木構造をもつ情報フィルタリング機構を用いることにより、放送型データを受信、格納し、ユーザに対して木構造型のインタフェースを用いて容易なデータアクセスを提供する。

現在開始されている放送型サービスを受信するためには、専用のアプリケーションが必要であり、データを閲覧するためにはそれぞれの閲覧用ソフトを利用する必要があった。そこで、AISでは、様々な情報ソースから放送されるデータをSADBによって統合し、シームレスに扱えるようにした。AISを用いるこ

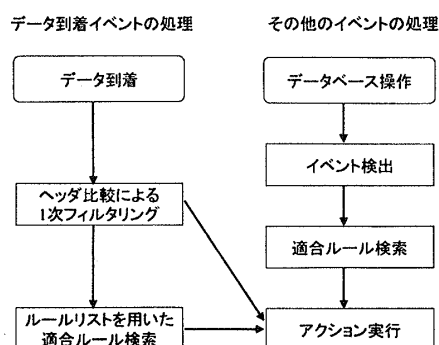


図5 高速モードのルール処理

Fig. 5 Rule processing method in the high-speed mode.

とで、情報源の違いを意識することなくデータにアクセスできる。

5.1.1 情報フィルタリングアルゴリズム

AIS はユーザインタフェースとして木構造をもつビューを用いる。木の各節点はデータ分類の概念を表し、葉に向かうにつれてより詳細な概念になる。例えばある節点‘ニュース’があるとすると、その葉節点として‘政治’や‘スポーツ’が考えられる。

ある放送データを受信したとき、そのデータは一つまたは複数の節点に分類される。そのデータが実際に格納されるかどうかは、その節点の価値や更新頻度、そのジャンルに対するユーザの興味などを総合的に判断して決定される[7]。データの分類、格納に利用される木の構造は、木の偏りやアクセスの変化などの要因に応じて動的に再構築されるためユーザに特化したデータの分類ができ、ユーザは常に木をたどる操作のみで必要な情報にアクセスできる。

5.1.2 システムの構成

AIS のシステム構成を図 6 に示す。まず、放送データを SADB によって受信し、データ内容やサイズ等の情報を情報フィルタリング部に通知する。情報フィルタリング部においては、そのデータの評価値を計算し、情報の必要性を判断する。

ユーザが情報を必要とするときは、ユーザが木構造の節点においてデータを指定すると、フィルタリング部が SADB に情報を要求し、アクティブデータベース部が必要とされた情報を適切な形式で返し、それをユーザに提示する。

また、必要か不要かが確実にわかっているデータに関するフィルタリング情報は ECA ルールとして記述され、SADB に格納される。

時間の流れにより情報の価値が変化するデータに関しては自動的に評価の値を更新し、期限付きデータの

期限が切れた場合にはデータを削除する。

送信元から ECA ルールが送られてきた場合、受信判定を行い、適合するルールであった場合はシステムに格納する。

5.1.3 アクティブ情報ストアの実装

アクティブ情報ストアのプロトタイプとして、実際に衛星を利用した放送型データを受信するのではなく、(1) 地上波データ放送サービスである ADAMS (朝日放送) の放送データ、(2) インターネットを利用したデータ放送サービスである PointCast の放送データ、(3) 独自に開発したロボットを用いて収集したインターネット上のホームページデータの 3 種のデータを放送するサーバプログラムを開発し、放送されるデータを受信しユーザに提供するシステムを実装した。

システムの実行例を図 7 に示す。放送されるデータは、データタイプなどファイル情報を表すテキストや、HTML 文書、画像ファイル、音声ファイルなど多岐にわたる。実際に放送サーバが放送するデータ量は、ADAMS, PointCast, ロボットによる収集データを合わせて 1 日当り 1000 アイテム、8M バイト程度であった。これらのデータを受信し、情報フィルタリング部においてジャンルごとに分類し、ユーザに提示する。ユーザインタフェース上では、どの情報源からのデータも同様に扱われており、ユーザは放送元の違いを意識することはなく情報を利用できる。ユーザがあるコンテンツを選択したときは、選択したコンテンツをブラウザ上に表示する。ブラウザとして Netscape 社の Communicator4.03 を用いた。

システムの機能は、データ受信処理、ルール受信処

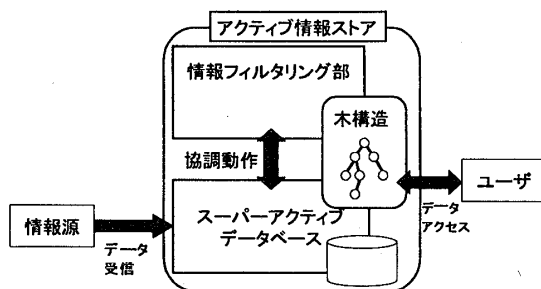


図 6 アクティブ情報ストア (AIS) の構成
Fig. 6 The structure of the Active Information Store.

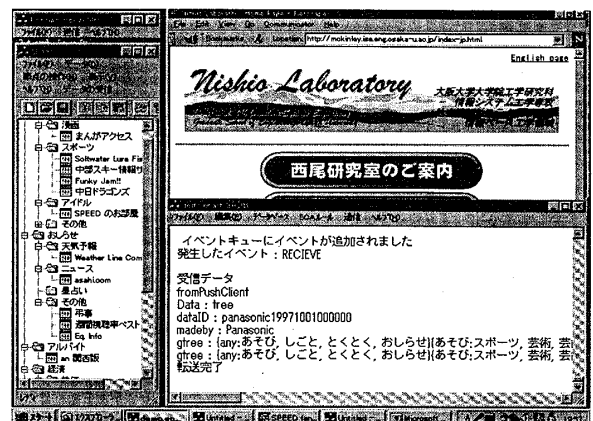


図 7 システム稼働図
Fig. 7 A snapshot of the prototype system.

理, データ表示処理などを行う 10 個の ECA ルールにより実現されている. 1 週間程度利用した後, システム内の ECA ルール数は 18 個に増加していた. これは, フィルタリングのためのルールや, 送信側がデータの自動削除を行うために送信してきたルールがシステムに格納されたためである. 放送サーバでは, 天気予報など有効期限をもつデータを送信する際にはそのデータを自動的に削除するためのルールも併せて放送している. これらのルールは 1 日に数個が放送されているが, ルール自体の有効期限がくれば削除されるため, システム内のルール数をそれほど増加させることはない. したがって, フィルタリング条件の追加を行うことでルール数は増加するが, 一般に SADB を用いてアプリケーションを構築するために必要なルール数は数十個以内に収まると考えられる.

5.2 ECA ルールオーサリングシステム

次に放送するコンテンツを作成する立場から考える応用システムである ECA ルールオーサリングシステムについて説明する.

SADB の大きな特徴として, 放送される ECA ルールを受信できることが挙げられる. この機能を利用することで, ユーザにさまざまなサービスを提供できる. 例えば, アクティブ情報ストアのトリー上に, ‘アウトドア’に関する節点があり, 頻繁に参照されているが, ‘4WD 車’に関する節点がないユーザがいたとする. そのとき, ‘4WD 車’に関するデータはトリーの節点にないため受信されないが, 両者は関連性が高いので, 受信さえされれば見てもらえる可能性が高い. そこで, 「もしユーザのトリーの節点に ‘アウトドア’ があれば, ‘4WD 車’のデータも受信する」というルールを配信しておけば, この問題は解決される. しかし, コンテンツ作成側で放送側の要求に合わせて ECA ルールを作成することは面倒である. したがって, ユーザからの情報や, コンテンツ制作側のターゲット要求, グローバルトリーの情報などをもとに, 配信用の ECA ルールを自動作成する ‘ECA ルールオーサリングシステム’を提供することが有効であるものと考えられる.

オーサリングツールのイメージ図を図 8 に示す. オーサリングツールは, クライアントの SADB に置いたアンケート回収 ECA ルールや, ユーザのリクエスト, カスタムトリー情報などから得るユーザ情報と, コンテンツ制作側のターゲット要求, 更にグローバルトリーなどから情報の関連性を見つけ, その関連性に基づいて ECA ルールを出力する.

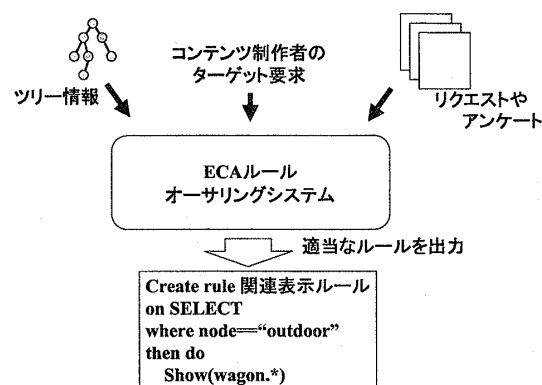


図 8 オーサリングシステムの概要

Fig. 8 The conceptual design of the authoring system.

6. 考 察

本章では SADB におけるルール処理の評価及び他のシステムとの比較を行うことで, 実装したシステムについて考察する.

6.1 評 価

4.2 で述べた高速ルール処理機構を実装し, 性能を評価した. 評価は, 5.1 で述べたアクティブ情報ストアを Windows95 搭載の携帯端末 (CPU: Mobile PentiumII 266 Mhz, Memory: 128 M バイト) 上で実際に動作させて行い, 5.1.3 で述べた擬似的な放送サーバから HTML データ及びイメージデータを繰り返し放送し, 通常のルール処理時間と高速モードのルール処理時間を測定した. 結果を図 9~図 11 に示す. 放送されるデータアイテムはそれぞれ一つ以上の HTML データ及び画像データ (各データのサイズは数 kb から数十 kb) を含む. また, 本測定はデータ閲覧アプリケーションを対象としているため, 4.1 の想定に合わせてデータの格納率は 1/400 程度になるように放送データを調整し, 放送サーバのデータ放送速度は 1 Mbit/s 程度になるようにした. 更に平均 1 秒に 1 回程度, データベース参照などを強制的に行うことで, データ受信以外のイベントを発生させた. 図 9, 図 11 の評価に使用した ECA ルールは, システムの動作に必要なルールを除き, ランダムに生成したものを用いるが, 5.1.3 で実際に AIS を運用した際の比率を参考に, データ受信を処理するルールとそれ以外のイベントを処理するルールを 7:3 の割合で生成するようにした. また, 図 9, 図 10 における受信ルールに関しては, 単純なヘッダ比較を用いたルールと, そうでないルールの発生比率は, 平均 1:1 になるようにした.

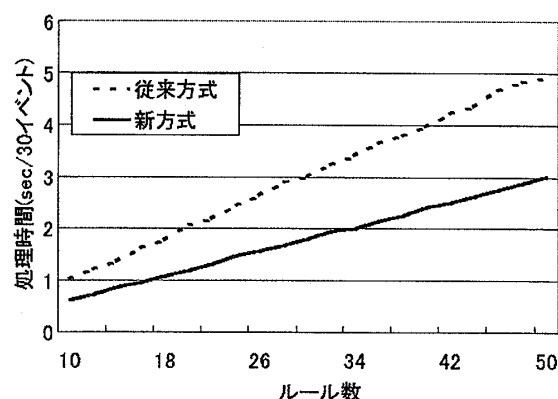


図9 ECA ルール数に対するシステムの性能の変化

Fig. 9 System performance in response to the number of ECA-rules.

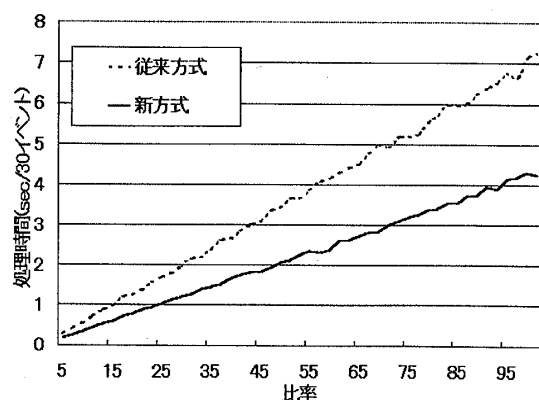


図10 受信ルールの比率に対するシステム性能の変化

Fig. 10 System performance in response to the ratio of receiving rules.

「そうでないルール」とは、単純なヘッダ比較にシステム変数の参照による条件などを加えたもので、計算量はそれほど増えないが、ルールリストを用いた処理を行う必要があるルールである。すべてのグラフにおいて、30 イベント当りの処理時間を縦軸に示す。

図9は、システム内のECAルール数を10から50まで変化させたときの処理時間の変化を示したものである。ECAルールを用いたアプリケーションでは、通常10から20程度のルール数でシステムが動作しているが、フィルタリングの複雑さに応じてフィルタリングのためのECAルール数も増加する。グラフより、高速モードではルール数の増加に対して処理時間の増加が抑えられていることがわかる。

図10は、ルール数を50に固定し、受信イベントを扱うルールとそれ以外のデータを扱うルールとの比率を変化させたときの処理時間を変化させたものである。発

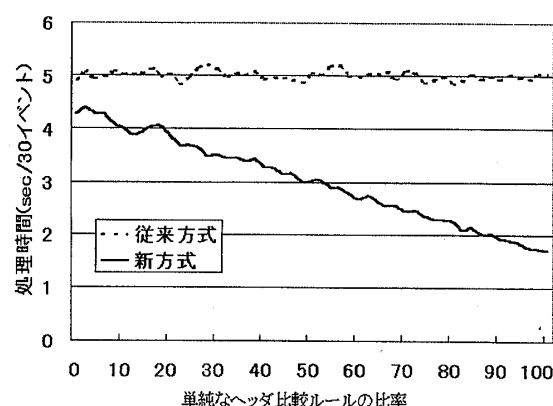


図11 ECA ルールの複雑さに対するシステム性能の変化

Fig. 11 System performance in response to the complexity of ECA-rules.

生するイベントのほとんどがデータ受信ルールであるため、両手法ともデータ受信ルールの割合が増加すると、発火するルールが増えて処理時間が増加している。受信ルールがないときは両方式とも同じ処理を行うため、ほとんど差が現れない。しかし、受信ルールの比率が多くなるにつれて、本手法の効率の違いが見られるようになっていく。

一般に、放送環境ではルールが多くなると受信ルール比率が多くなる。これは、フィルタリング条件を記述するECAルールが多くなるためである。以上のことから、本手法は放送環境で有効であると考えられる。

図11は、ルールをすべて受信に関するルールに限定した場合において、単純なヘッダ比較ルールと、そうでないルールとの比率を変化させたときの処理時間を示したものである。4.3で述べたように、ヘッダ比較ルールはヘッダ比較リストに登録して処理するため高速化が可能である。特に、受信ルールがすべてヘッダ比較であった場合は、従来手法に比べて大幅に効率が高くなる。

想定する環境では、システムに最低限必要なECAルールの総数は少なく、後からデータのフィルタリング用にルールを追加していくといった使い方が一般的である。また、フィルタリング用のルールの多くは、「スポーツのデータが欲しい」「競馬のデータはいらない」といったように、単純な比較ですむものが多い。したがって、提案手法は評価したあらゆる面において、想定する環境にふさわしい手法であるといえる。更に、すべての場合において、最悪の場合でも提案手法が従来手法よりも高速であることがわかる。

しかし、このようにルール処理を二つのモジュール

に分けて行うことで、ルールの実行順序が保証できなくなる。このことが問題となるようなアプリケーションでは、従来手法を用いる必要がある。

実際のイベント処理能力について、高速モードの処理能力は評価結果から 10 から 50 イベント/s 程度であることがわかる。4.1 で想定するイベント発生頻度が 7 イベント/s 程度であるため、システム中に ECA ルールが 50 程度存在したとしても本方式は十分要求仕様を満たしていることがわかった。しかし今後、想定した帯域幅である 1 Mbit/s より多くの帯域がデータ放送に割り当てられた場合、今以上の処理速度が要求されることになり、更にシステムの高速化を行う必要がある。

次に、提案手法だけを用い、データの格納率を変化させて処理時間を測定する実験を行った。結果を図 12 に示す。4.1 で述べたように、集計処理等を行うアプリケーションではデータの格納率が高くなる可能性がある。格納率の上昇は ECA ルールが発火する割合の増加を意味するため、処理時間もほぼ単調増加している。格納率が 100% 近くになった場合、1 秒当り 5 イベント程度しか処理できていないため、想定環境における要求仕様を満たしていない。3 割程度の格納率のときにちょうど現在の要求仕様の処理速度を示しているため、本システムは格納率が 3 割以下になるようなアプリケーションにおいて要求仕様を満たしているといえる。それを超えるようなアプリケーションを利用する場合には更に高速な処理が要求される。

6.2. 他システムとの比較

本節では、他のアクティブデータベースや放送型システムと、SADB を用いたシステムとを比較する。

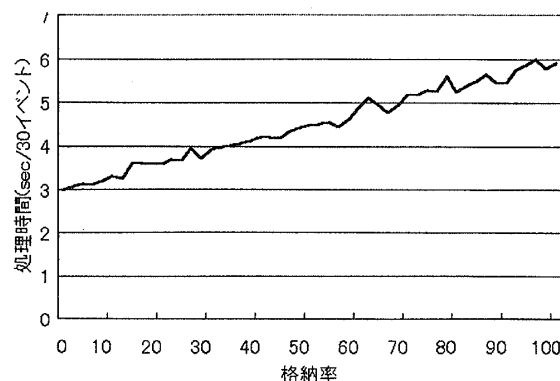


図 12 データ格納率に対するシステムの性能の変化
Fig. 12 System performance in response to the ratio of storing data.

SADB と同様にアクティブデータベースを拡張したシステムとしては、アクティブモバイルデータベース (Active Mobile Database System : AMDS) が提案されている [8]。AMDS は移動体計算環境の基盤システムとして提案されているアクティブデータベースであり、移動体の接続・切断の検出のほかに、パケット到着の検出や、ECA ルールの送受信など SADB と同様の機能をもつ。AMDS が移動体の動作を把握し、移動体からのデータ収集を行うことを主な目的としているのに対し、SADB は、放送型データの受信基盤として、データを受信、格納することを目的としている点と異なっている。

実際の放送型サービスとしては、インターネット上でデータ (文字情報) を配信するシステムである、PointCast (ポイントキャスト社) [5] が挙げられる。インターネットを用いてデータを配信している点で、本論文で述べたアクティブ情報ストアと類似している。PointCast は、コンテンツ作成者が作成したニュースなどの文字ベース情報をインターネットを用いて配信し、専用のブラウザで閲覧するシステムである。ユーザは好きなジャンルを選択することで、その情報を受信することができる。このブラウザの機能は簡潔で操作も軽快であるが、データの分類が最大 2 段階しか存在せず、コンテンツが増加したときに、ジャンル選択の数が非常に多くなってしまふ。アクティブ情報ストアではトリー構造を採用しているため、ジャンル数が多くなった場合に有効である。また、PointCast では、各データはどれも同様に扱われ、一定期間後削除される。したがって、興味のあるデータをまだ見ていなかったとしても削除されてしまう可能性がある。アクティブ情報ストアでは、データやジャンルにそれぞれ特性をもたせることができるため、データによって削除期間や重要度を設けることができる。また、大変興味があると判断されたデータはそれだけ保存期間も長くなるため、ユーザが見落とす可能性が低くなる。

次に地上波テレビ放送を多重化することで HTML データ配信を行うサービスである ADAMS (テレビ朝日) [1] との関連性について述べる。ADAMS は、コンピュータ上に受信ボードを搭載することで、HTML データが受信できるサービスである。このサービスは、文字放送と同様に、地上波放送の帯域を利用したもので、ニュースや天気などの情報をユーザに提供する。配信データは HTML 形式であり、リンクをたどるときはインターネットを利用するという点で、アク

ティブ情報ストアと共通の部分が見られる。ただし、ADAMS は放送されるデータの取捨選択を行わず、すべてのコンテンツをハードディスク上に保持するため、放送されるコンテンツが制限される点が問題である。これに対しアクティブ情報ストアでは、フィルタリングを行うため、大量のコンテンツにも対応できる。

PointCast, ADAMS は、ともに、放送されたどのデータにアクセスが多かったかなどの情報を得る手段に乏しい。アクティブ情報ストアでは、SEND アクションを利用してデータを送信する ECA ルールを受信側に格納しておくことで、放送元からのデータ収集が行える。この機能を利用することで、5.2 で述べたアプリケーション例のように、ユーザの好みを利用したコンテンツ作成を行える。また、ユーザにアンケートに答えさせたり、あるいはリアルタイムで投票を行う、といった機能も実現できる。

アクティブ情報ストアでは、ADAMS 及び PointCast のデータも受信して利用できるので、上に述べたような ADAMS, PointCast の欠点は、アクティブ情報ストアを通してデータにアクセスすることで解消できる。

7. む す び

本論文では、放送型データ受信基盤として拡張したアクティブデータベースの利用について述べた。アクティブデータベースを用いることにより、対応する ECA ルールを記述しておくだけで様々な種類のデータを処理し、ユーザはデータの種別を意識することなく情報を利用できるようになった。また、本システムを利用したアプリケーションについて述べ、それらによって今まで提供が困難だった様々なサービスが容易に提供できることを示した。更に、構築したアプリケーションを用いて SADB の性能評価を行い、提案方式で対処できるデータ量を定量的に示した。

今後は、ECA ルールの処理機構を更に最適化し、トラフィックの増加に対応できるようにシステムを改良する予定である。また、無限ループの検出やアクセスレベルの管理など、SADB の安全性について考察していく予定である。更に、各種のフィルタリングアルゴリズムを ECA ルールに変換するコンパイラなど、様々なアプリケーション及びサービスを構築し、放送環境における本システムの有効性を検証する予定である。

謝辞 末筆ながら、本研究を行うにあたって貴重な御助言を頂いた Somnuk Sanguantrakul 氏、松下電

器産業の今中武氏、三浦康史氏、松浦聰氏に感謝する。また、有益な助言や協力を頂いた西尾研究室の諸氏に感謝の意を示す。なお、本研究は、日本学術振興会未来開拓学術研究推進事業における研究プロジェクト「マルチメディア・コンテンツの高次処理の研究 (Project No. JSPS-RFTF97aP00501) の研究助成によるものである。ここに記して深謝の意を表す。

文 献

- [1] ADAMS Homepage, <http://www.tv-asahidata.com/>
- [2] F. Llirbat, F. Fabret, and E. Simon, "Eliminating costly redundant computations from SQL trigger executions," Proc. the ACM SIGMOD International Conference on Management of Data, pp.428-439, 1997.
- [3] G. Lohman, L. Bruce, P. Hamin, and K. Bernhard, "Extentions to starburst: Object, types, functions, and rules," Commun. ACM, vol.34, no.10, pp.94-109, 1991.
- [4] 箱守 聡, 田辺雅則, 石川裕治, 井上 潮, "放送型通信/オンデマンド型通信を統合した情報提供システム," 情処研報, vol.34, no.8, pp.55-60, 1997.
- [5] PointCast Japan ホームページ, <http://www.pointcast.ne.jp/>
- [6] S. Acharya, M. Franklin, and S. Zdonik, "Prefetching from a broadcast disk," 12th International Conference on Data Engineering, pp.276-285, 1996.
- [7] S. Sanguantrakul, T. Terada, M. Tsukamoto, S. Nishio, K. Miura, S. Matsuura, and T. Imanaka, "A user customized selection and categorization for broadcast data," International Workshop on Multimedia Network Systems, pp.596-601, 1999.
- [8] T. Murase, M. Tsukamoto, and S. Nishio, "A system platform for mobile computing base on active database," Proc. International Symposium on Cooperative Database Systems for Advanced Applications, vol.2, pp.424-427, 1996.
- [9] 矢島悦子, 原 隆浩, 塚本昌彦, 西尾章治郎, "相関性をもつデータ間の放送時間間隔について," 情処学論, vol.40, no.1, pp.188-196, 1999.
- [10] 矢島悦子, 原 隆浩, 塚本昌彦, 西尾章治郎, "データ間の相関性を考慮した放送データのスケジューリング法およびキャッシング法," 情処学論, vol.40, no.9, pp.3577-3585, 1999.
- [11] Y.E. Ioannidis and T.K. Sellis, "Conflict resolution of rules assigning values to virtual attributes," Proc. the ACM SIGMOD International Conference on Management of Data, pp.205-214, 1989.

付 録

SET_TIMER アクションの条件設定方法 (3.3)

SET_TIMER アクションは、タイマを定義するアク

ションである。放送されるデータには有効期限や有効期間があるものが多いため、タイマの設定は大変柔軟に行えるように設計している。「タイマ設定条件」において記述する内容は、有効区間の設定及び有効区間切れの際のタイマ削除タイミング、有効区間内のタイマ発火回数、タイマ発火タイミング指定からなる。以下に構文を示す。

<タイマ識別子>、

[VALID_TIME (<開始タイミング>, <終了タイミング>),]

[TRIGGER_PATTERN (<削除タイミング>, <発火回数>),]

EXECUTE_ON (<時間設定>, <タイミング指定>)

「開始タイミング」「終了タイミング」は「yyyymmddhhmmss」の形で表現され、省略した部分はワイルドカードとみなされる。終了タイミングのみ「+yyyymmddhhmmss」の形で、相対時間を表現できる。「削除タイミング」と「発火回数」は、「ONCE」または「REPEAT」のどちらかで指定し、その意味は表 A・1 に示すとおりである。「時間設定」「タイミング指定」では、タイマが発火するタイミングを指定する。タイミング指定は「ON_TIME」, 「ON_TIME_RELATIVE」, 「ON_DAY」, 「INTERVAL_TIME」のいずれかで指定し、それぞれの時間設定方法は表 A・2 に示すとおりである。

表 A・1 削除タイミング、発火回数の指定

Table A・1 Specification of the timing of deleting and the number of trigger actions.

属性名	ONCE にした場合	REPEAT にした場合
削除タイミング	有効区間が終了した時点でタイマは削除	有効区間が終了しても、次の有効区間のためにタイマは保存
発火回数指定	有効区間内でタイマが一度発火すればタイマは削除	有効区間内であれば何度もタイマが発火可能

表 A・2 タイマ実行タイミングの指定

Table A・2 Specification of the timing of executing the timer event.

タイミング指定	意味	時間設定の内容
ON_TIME	指定時間	直接指定 (yyyymmddhhmmss)
ON_TIME_RELATIVE	相対時間	直接指定 (yyyymmddhhmmss)
ON_DAY	指定曜日	曜日 (MON,TUE,WED...)
INTERVAL_TIME	一定時間毎	時間間隔 (yyyymmddhhmmss)

これらの指定を組み合わせることで、例えば「毎日 8時から9時の間だけ 10分ごとに発火するタイマ」のような、複雑な記述も、

SET_TIMER (VALID_TIME (080000,090000),

TRIGGER_PATTERN (REPEAT, REPEAT),

EXECUTE_ON (1000, INTERVAL_TIME))

と指定することで行える。

(平成 12 年 5 月 9 日受付)

寺田 努 (学生員)



平 9 阪大・工・情報システム卒。現在同大学院情報システム工学専攻博士後期課程在学中。アクティブデータベース、データ放送の研究に興味をもつ。tsutomu@ise.eng.osaka-u.ac.jp

塚本 昌彦 (正員)



昭 62 京大・工・数理卒。平 1 同大学院工学研究科修士課程了。同年シャープ(株)に入社、同社研究員。平 7 大阪大学工学部情報システム工学教室講師。平 8 より同教室助教授、現在に至る。工博。モバイルコンピューティング、分散知識ベースシステムの研究開発に従事。ACM, IEEE 等 7 学会各会員。tuka@ise.eng.osaka-u.ac.jp

西尾章治郎 (正員)



昭 50 京大・工・数理卒。昭 55 同大学院工学研究科博士課程了。工博。京都大学工学部助手、大阪大学基礎工学部及び情報処理教育センター助教授を経て、平 4 より大阪大学大学院工学研究科情報システム工学専攻教授となり、現在に至る。この間、カナダ・ウォータールー大学、ビクトリア大学客員。データベース、知識ベース、分散システムの研究に従事。現在、Data & Knowledge Engineering, Data Mining and Knowledge Discovery, VLDB Journal などの論文誌編集委員。ACM, IEEE など 8 学会各会員。nishio@ise.eng.osaka-u.ac.jp