

## 情報配信サービスを利用した情報源統合環境の構築

水口 弘紀<sup>†</sup> 北川 博之<sup>††</sup> 石川 佳治<sup>††</sup> 森嶋 厚行<sup>††</sup>

<sup>†</sup> 筑波大学 工学研究科 <sup>††</sup> 筑波大学 電子・情報工学系

e-mail: {hironori, kitagawa, ishikawa, mori}@dbl-lab.is.tsukuba.ac.jp

### 概要

ネットワークの普及に伴い、各種情報源へのアクセスが容易になり、これらの統合利用の要求が増大している。このような要求に対応するため、異種情報源統合システムの研究が各種行なわれている。しかし、情報源の形態は多様化しつつあり、必要な情報を見つけるためにユーザ自身が要求を出すのではなく、ユーザに直接情報を配信する情報配信サービスも数多く見られるようになってきた。したがって、これら情報配信サービスに対応した異種情報源統合システムの構築が今後重要になると考えられる。異種情報源統合システムにおける情報配信サービスの利用形態としては、(1) 統合利用対象の情報源としての情報配信サービスの利用、(2) 統合結果の情報配信手段としての情報配信サービスの利用が考えられる。これらの情報配信サービスの利用はその一方、または両方を独立に選択可能である。しかし、いずれにおいても、記事の到着や時間の経過といったイベントに連動して、情報統合や配信サービスといったアクションをシステムが能動的に実行することが必要である。本論文では、我々の研究グループで研究開発をしてきた異種情報源統合システム *InfoWeaver* の持つ情報統合機能に、ECA ルール処理システムを組み合わせることで、上記二形態の情報配信サービスの利用を実現する環境について述べる。

## Utilization of dissemination-based information service in information integration environments.

Hironori Mizuguchi<sup>†</sup>, Hiroyuki Kitagawa<sup>††</sup>, Yoshiharu Ishikawa<sup>††</sup>  
and Atsuyuki Morishima<sup>††</sup>

<sup>†</sup>Doctoral Degree Program in Eng., Univ. of Tsukuba <sup>††</sup>Institute of Info. Sci. and Elec., Univ. of Tsukuba

### Abstract

Integration of heterogeneous information sources has been one of important research issues in recent advanced application environments. In these days, more and more information sources of various types are available. Dissemination-based services that autonomously deliver information from the server sites to users are among the useful and promising information sources. In this paper we present incorporation of dissemination-based information services into information integration environments. The integration here has two aspects: (1) Users can utilize dissemination-based services as other information sources such as databases and the web. Namely, they can be sources of information integration. (2) Users can obtain integrated information through dissemination-based services. We explain now this requirement can be met by a combination of an information integration engine and an ECA rule processing subsystem. We also explain prototype system development. We have used *InfoWeaver*, which is a prototype information integration system developed in our research group, as core engine.

### 1 はじめに

ネットワークの発達により各情報源へのアクセスが容易になり、これらを統合利用したいという要求が増大している。この要求に対応するため異種情報源統合シス

テムに関する研究が各種行なわれている [1, 6]。一方、ますます情報源は多様化しており、その一つとして、ユーザに直接情報を配信する情報配信サービスが注目されている [2, 3]。情報配信サービスの特徴はサーバクライアントに能動的に情報を配信することである。これに

よって、ユーザは情報がどこにありいつ更新されたかなどを気にせずに、新鮮な情報を受信することができるようになる。ネットワークを介した配信や、データ放送による電波を用いた配信など具体的な配信形態は多様であるが、今後の情報伝達手段として重要な役割を担うと考えられる。したがって、異種情報源統合システムにおいても今後このような情報配信サービスに対応することは重要であると考えられる。

異種情報源統合システムにおける情報配信サービスの利用形態としては、(1) 情報配信サービスを情報源としてその配信情報を統合利用する、(2) 統合システムで作成した統合結果を情報配信システムを用いて配信する、という二形態が考えられる。前者は、従来の情報配信サービスでは実現が困難だった、各ユーザの好みに応じた配信情報の選択や、複数情報配信サービスの統合は勿論のこと、より一般的に他の情報源と情報配信サービスの統合利用を可能にする。また、後者は、ユーザの指定したタイミングに合わせた統合処理結果の配信など、効果的な情報の配信を可能とする。

これらの二形態は、本来それぞれ独立に選択可能であるべきであり、情報配信サービスで到着した情報を問合せで選択したり、既存データベース等のデータを定期的に配信するといった利用が考えられる。しかし、いずれにおいても、このような環境を実現する上では、記事の到着や指定した時間の経過などのイベントに起因して、能動的に統合処理や統合結果の配信を行なう仕組みが必要である。本研究では、メディエータ、ラッパーに基づく異種情報源統合システムにECAルール [4] に基づく処理システムを付加することによってこのような環境を実現する。異種情報源統合システムとしては我々の研究グループで開発している *InfoWeaver* [5, 6] を用いる。より具体的には、情報配信サービスラッパー、ECAルールサブシステム、配信サブシステムを新たに導入し、ECAルールサブシステムと情報配信サービスラッパーで上記(1)の利用形態を、ECAルールサブシステムと配信サブシステムで上記(2)の利用形態を実現する。

以下では、2節で情報配信サービスとその情報統合利用について例を示す。3節で本研究におけるアプローチを示した後、4節でルールの役割と必要なイベントを明確にする。5節、6節、7節で、*InfoWeaver* と情報配信サービスラッパー、ECAルールサブシステム、配信サブシステムのそれぞれについて述べる。8節でプロトタイプシステムの実現方法を示した後、システムの利用例、関連研究、まとめの順に述べる。

## 2 情報配信サービスとその情報統合利用例

情報配信サービスは、その配信内容から見ると、Point-Cast のように主に HTML などの文書を配信するものや、Castanet のような文書に加え Java プログラムなどのバイナリも配信できるものがある。配信媒体も、インターネットを利用したメールでの送信、Web サーバを

利用した配信、地上波を利用した bitcast や ADAMS に見られるような情報放送など様々である。これらの情報配信サービスはいずれもサーバがクライアントに能動的に情報を配信するという特徴を持っている。よって、ユーザは何の手間もなく、新鮮な情報を閲覧することができる。以下では、配信される単位を配信単位と呼び、個々のサービスをチャンネルと呼ぶ。

情報配信サービスの異種情報源統合システムにおける利用形態は、先にも述べたように、統合対象の情報源としての利用、統合結果の情報配信システムを利用した配信があり、これらは独立に選択可能である。以下では、この両方の利用形態を用いた情報統合利用例を示す。

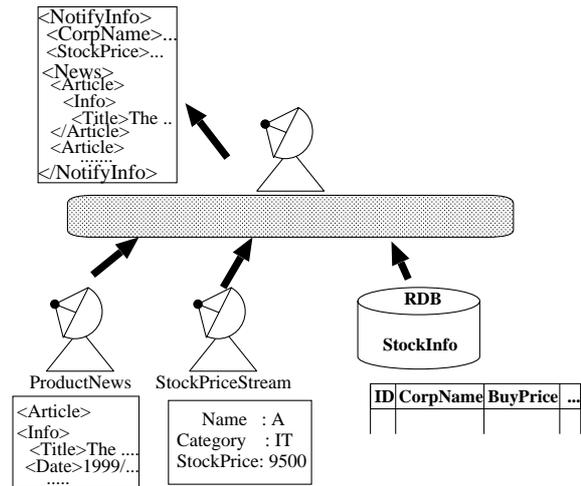


図 1. 情報配信サービスの利用例

メールを用いて製品情報ニュースを配信しているチャンネルと、Web 上で株価情報を配信しているチャンネルがあると仮定する。リレーショナルデータベースには、現在自分が保有している IT 関連の株の情報が格納されている。各社の購入時の株価、株数、閾値として注目している株価が格納されている。製品情報ニュースチャンネルはその配信単位である文書をメールで配信するチャンネルである。株価情報チャンネルは会社名、その業種、株価を配信単位とするチャンネルであり、その配信タイミングは不特定であるとする。これらの情報源の統合利用を考える。ここでは例として、株価情報チャンネルに IT 関連の会社に関する株価の報告があった場合、この会社の株価が自分の注目している閾値よりも高かった時これを報告する。その時、その会社の過去一週間の製品情報と、自分が保有しているその株に関する情報を配信して欲しいという要求を考える (図 1)。このような要求の場合、情報配信サービスを情報源として製品情報ニュースを一週間分保持しておくことが必要である。さらに、リレーショナルデータベースとの統合処理を行ないこの結果を情報配信サービスを使って配信することが必要である。

### 3 アプローチ

本研究では、メディエータ / ラッパーに基づく異種情報源統合システムを想定し、これに情報配信サービスラッパー、ルールサブシステム、配信サブシステムを付加することで、情報源としての情報配信サービスの利用、統合結果の情報配信サービスを利用した配信を実現する(図2)。情報源としての情報配信サービスを扱うためのラッパーは、配信単位をメディエータに提供する。また、配信単位の到着に応じてイベントを発生する。ルールサブシステムは、配信単位の到着や時間経過などのイベントに起因した処理を記述したECAルールを評価する。配信サブシステムは、配信指定された統合結果をクライアントへ配信する。その際、統合結果をその配信方法における適切な形式に変換する。

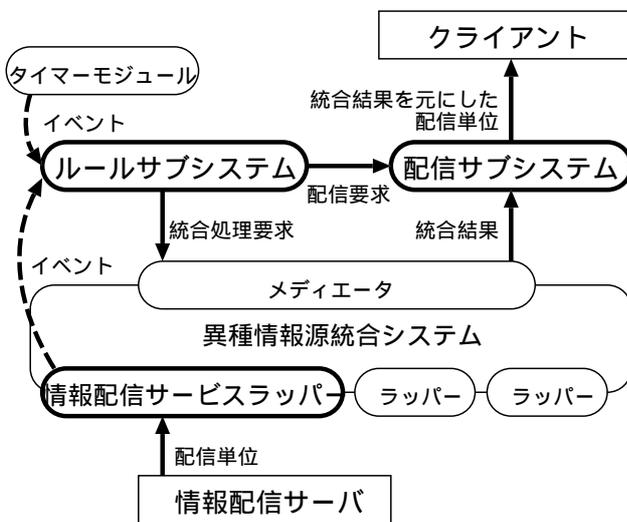


図 2. アーキテクチャ

より具体的には、それぞれのサブシステムは次のように機能する。(1) 情報源として情報配信サービスを用いた統合処理の場面では、異種情報源統合システムとルールサブシステムが用いられる。情報配信サーバから配信単位が送られると、情報配信サービスラッパーはその到着をイベントとしてルールサブシステムへ通知する。ルールサブシステムでは、それらイベントが報告されると、これに対応するルールの評価を開始し、統合システムへ統合処理要求を発行する。(2) 統合処理結果を情報配信サービスを用いてユーザへ配信する場面では、ルールサブシステム、配信サブシステムが用いられる。ルールサブシステムへイベントが報告されると、配信サブシステムへ配信要求が発行される。配信サブシステムでは異種情報源統合システムから統合結果を取得し、この結果を配信要求に沿った形態に変更しこれを配信する。たとえば、メールによる配信要求であれば統合結果を適切な文字列に変換して送信する。

本研究では、異種情報源統合システムとして我々の研究グループで開発している *InfoWeaver* を用いる。以下では、まず、本研究におけるルールの役割と想定してい

るイベントについて述べる。その後、*InfoWeaver* への情報配信サービスの統合と情報配信サービスラッパーの役割について述べ、6節でルール記述とルールサブシステム、7節で配信サブシステムの機能について述べる。

### 4 ルールの役割とイベント

本節では、本研究におけるECAルールの役割を明らかにし、また、想定すべきイベントを明確にする。本研究では情報の統合処理、情報の配信処理という二つの利用形態を考慮し、これらの処理をECAルールを用いて記述することは先に示した。この二つの利用形態を考慮すると、ルールは以下の場面で用いられる。

1. 配信単位の到着時処理を記述するためのルール (Receiving Rule)  
情報配信サービスに配信単位が到着したことに起因し、到着した情報をどのように処理するかを記述するルールである。例えば、到着した配信単位を異種情報源統合システムに保持しておくように指示するなどである。
2. 統合処理を行なうためのルール (Integration Rule)  
情報単位の到着や時間的要因に起因し、Receiving Ruleによって蓄積された配信情報や各種情報源に対する統合処理などを記述するルールである。
3. 配信処理を記述するためのルール (Dissemination Rule)  
時間的要因などに起因し、Integration Ruleで作成された処理結果を配信するルールである。配信タイミングを調節し、スケジュールを決めるルールである。

実際には各ルールは厳密にこれらの一つに分類されるわけではなく、これらを組合せた役割を担うことがある。例えば、あるルールで一週間分の配信単位を蓄積しておき、別なルールで、ある時刻になった時点でこの蓄積情報に対して統合を行ないこの結果を配信するといったことが想定される。しかし、一方で、一つのルールのみを用いて、配信単位が到着した時点ですぐに統合処理を行ないこれを配信するといったことも想定される。本研究では、いずれのルールもその記述方法や処理過程に本質的な違いはなく、全て同一の枠組内で上記役割の選択的な実現を可能にする。

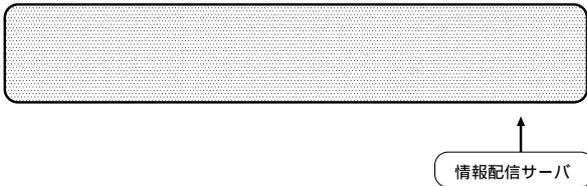
また、ルールサブシステムに通知されるイベントとその発生源には様々なものが考えられるが、本研究では、上記機能を実現するため、イベント発生源として情報配信サービスラッパーとタイマーモジュールを想定する。それぞれ、情報配信サービスからの配信単位の到着、時間的要因に起因するイベントを発生する。

具体的ルール記述方法については6節で述べる。

## 5 情報源としての情報配信サービスの統合

*InfoWeaver* は、情報源として WWW、リレーショナルデータベース、構造化文書を統合対象としていた。本研究では、情報配信サービスを情報源としてさらに統合する(図3)。

*InfoWeaver* では各情報源にラッパーを配置し、これにより各情報源の情報を統合データモデル WebNR/SD に変換する。メディエータはこれに基づいた統合操作の機構を提供する。問合せがメディエータに投入されると、各ラッパーで処理可能な部分問合せに変換される。メディエータは各ラッパーの処理結果から最終結果を作成しユーザーに結果を返す。



ントである。コンポジットイベントは各ルールのイベント部に記述されたプリミティブイベントの集合であり、プリミティブイベントのいずれかが発生した時、コンポジットイベントが発生する。プリミティブイベントはそれぞれ、(1) arrival(S), (2) interval(int), (3) alarm(T) で記述する。コンポジットイベントはこれらの組み合わせで記述する。

## 6.2 コンディション

コンディションにはアクションを発火させるための条件として、メタデータで評価可能な条件を記述する。

ここでは、リレーションの内容についての条件として、代数演算子の選択演算子におけるパラメタ記法と同じ記法を用いる。例えば、 $R.Title = 'XML'$  は、リレーション R 中の属性 Title が XML のタプルがあれば真となる。また、 $|R| \text{ op } N$  ( op は '=', '<', '>', '≤', '≥' のいずれか ) を用いて、リレーション R のタプル数に関する条件を記述する。ここで、 $|R|$  は R のタプル数を示す。

## 6.3 アクション

アクションとしてメタデータにおける統合処理を記述する。また、統合処理結果の配信指定も記述する。

ここでは、InfoWeaver における統合データモデルである WebNR/SD に基づく代数式を記述する。さらに、図 7 に示される記法を用いることで、リレーションの変更、タプルの追加、削除を記述する。

配信の指定としては Deliver 指定子を用いる。これについては次節で述べる。

|              |                       |
|--------------|-----------------------|
| $R_1 := R_2$ | $R_2$ を $R_1$ とする。    |
| $R_1 += R_2$ | $R_2$ を $R_1$ に追加する。  |
| $R_1 -= R_2$ | $R_2$ を $R_1$ から削除する。 |

図 7. アクションにおけるリレーションに対する記法

## 6.4 ルール記述方法

ECA ルールは以下のように記述する。Rule の後にルール名を指定し、on 節としてイベント部、if 節としてコンディション部、do 節としてアクション部を記述する。do 節には一つまたは複数のアクションを指定することができ、これらは順次実行される。on 節、if 節が省略された場合、イベント部として  $\text{alarm}(t_{inst})(t_{inst}$  はルールの登録された時刻) が、コンディション部には true が指定されたとみなす。

以下の二つのルールを用いてより具体的に説明する。

```
Rule selecting_XML
on: arrival(S)
if: S.Title = 'XML'
do: SelectXML +=  $\sigma_{Title='XML'}(S)$ ;
```

```
Rule clear_data
on: alarm(8 am)
do: SelectXML =  $\phi$ ;
```

ルール selecting\_XML 中 on 節における S は情報配信サービスのチャンネルを表し、if 節、do 節における S は対応する情報配信サービスラッパーによって提供されるストリームリレーションを指す。on 節においてイベントには S に対する 配信単位到着イベント arrival(S) が指定されている。if 節のコンディションにおける  $S.Title = 'XML'$  は、ストリームリレーション S 中のタプルの属性 Title が 'XML' であれば真を返す述語である。ルール selecting\_XML では、S に配信単位が到着し、属性 Title に 'XML' という文字列があれば、アクションを実行する。do 節のアクションでは、タイトルが 'XML' であるタプルを選択しこれをリレーション SelectXML に追加する。

ルール clear\_data では、on 節によって毎朝 8 時になるとルールが適用されることが示されている。if 節が省略されているため、コンディション評価を行わずに do 節におけるアクションを実行する。アクションは 前出のルール selecting\_XML によって蓄積された SelectXML リレーションから全タプルを削除する。

この二つのルールによって、毎朝 8 時までの一日分の A 社の情報のみを SelectXML リレーションに蓄積することを示している。

## 7 配信サブシステム

### 7.1 Deliver 指定子

ECA ルールの do 節には、配信指定子 Deliver を記述することができる。本指定子によって統合結果として何を配信するか、配信方法、配信対象などを指定し、これが配信サブシステムへ渡される。

```
Deliver MAIL('hiro@x.ac.jp','NotifyFromRule',C)[R]
```

本環境では、配信内容としてリレーションを指定する。指定子のパラメタにおいて配信サービスの種類を記述し、その後にそれぞれの配信サービスに必要な情報を記述する。ここでは、メールによる配信を指定し、その宛先とサブジェクトを特定し、配信単位としてリレーション R 中の属性 C の値を配信することを示している。また、この指定子によってリレーションの内容に変化が起こることはない。

### 7.2 データ変換

配信サブシステムでは、Deliver 指定子によって指定された配信単位を、配信方法毎の適切なデータ形式に変換する。例えば、メールであれば属性値を文字列に変換し、これを本文とするメールを配信する。

## 8 各サブシステムの実現

各サブシステムの実現方法を図9に示す。InfoWeaverにおいては、Web、リレーショナルデータベース、文書リポジトリに対するラッパー、メディアータは既に実現されている。本研究では新たに情報配信サービスラッパーを追加する。ルールサブシステムは、ルールの管理を行なうルールマネージャ、イベントの監視通知を行なうイベントマネージャ、コンディション、アクションの評価を行なう演算処理マネージャからなる。配信サブシステムでは、配信マネージャによって、ルールサブシステムからの配信要求を取得し、メディアータから指定されたリレーションを得てその内容を配信する。さらに、配信マネージャではリレーションを各配信方法毎に対応した形式に変換し、実際に配信を行なう配信エージェントを使い情報を配信する。

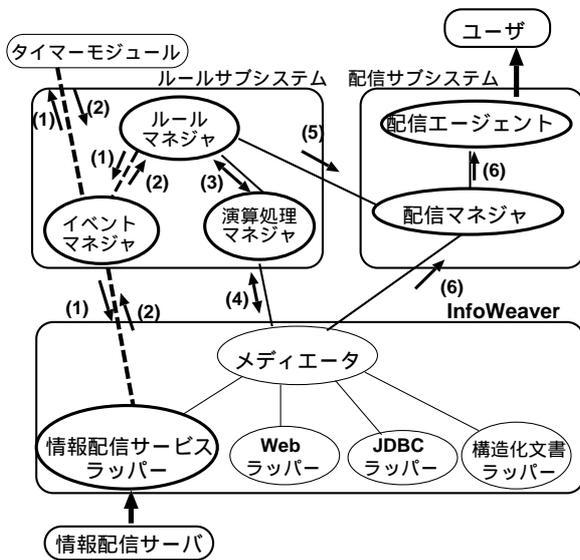


図9. 各サブシステムの実現

ルールの登録から実行までの流れを以下で示す。ルールが登録されると、(1)ルールマネージャはそのイベント部をイベントマネージャに送る。イベントマネージャではイベントを分類し、対応するイベント発生源へイベントの監視を登録する。ここでイベント発生源は情報配信サービスラッパーとタイムラップである。(2)各イベント発生源では登録されたイベントが起こると、イベントマネージャに通知する。イベントマネージャでは対応するルール名を探し、ルールマネージャに通知する。(3)ルールマネージャは演算処理マネージャに、対応するルールのコンディション部、アクション部を渡す。(4)演算処理マネージャでは、メディアータを用いてコンディションが満たされているかを判別し、そうであればアクションを実行する。(5)配信指定がある場合には、配信マネージャへ配信要求を出す。(6)配信マネージャでは、この要求から指定されたリレーションを各配信方法に対応した形式に変換し、配信エージェントを使ってクライアントへ配信する。

## 8.1 情報配信サービスラッパー

情報配信サービスラッパーのアーキテクチャを図10に示す。情報受信部では情報配信サーバから配信単位を受け取る。ストリームリレーション作成部は、受信した配信単位からストリームリレーションを作成する。リレーション管理部は、ストリームリレーションの保持を行なう。メタ情報管理部はストリームリレーションのスキーマや、その数など本ラッパーにおけるメタ情報を保持し、メディアータからのメタ情報取得要求に答える。演算実行部ではラッパー管理部を通してメディアータからの演算要求を受けとり実行する。イベント通知部は、ラッパー管理部を通して、ルールサブシステムへイベントを発生させる。ラッパー管理部は、メディアータとの演算処理に関する通信と、ルールサブシステムとのイベント処理に関する通信を行なう。

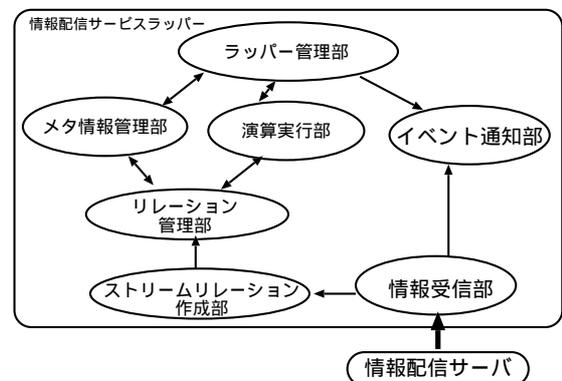


図10. 情報配信サービスラッパーの実現

本ラッパーでは、(1)情報配信サーバからの配信単位の到着、(2)メディアータからの演算要求、(3)ルールサブシステムからのイベント通知要求の三つの場面で処理が行なわれる。

(1)情報配信サーバから配信単位が到着すると、情報受信部では、ストリームリレーションのタブルを作成し、arrival イベントを発生する。まずストリームリレーション作成部に配信単位、到着時刻を渡す。ストリームリレーション作成部では、配信単位と配信時刻をもとにタブルを作成しストリームリレーション中にこれを登録する。次にイベント通知部へ配信単位の到着があったことを知らせる。イベント通知部では通知要求のあったarrival イベントであればこれを発生させる。

(2)メディアータからの演算要求があると、ラッパー管理部は演算実行部にこの演算要求を渡す。演算実行部では演算要求にしたがって演算を実行し結果を返す。実際にはメディアータから投入される演算はリレーションを取得する演算のみであるので、リレーション管理部から要求のあったリレーションを返す。

(3)ルールサブシステムからイベント通知要求があると、ラッパー管理部ではイベント通知部にこれを登録する。イベント通知部は通知要求のあったイベントが起これば、これを発生しルールサブシステムへ知らせる。

## 8.2 ルールサブシステム

ルールサブシステムは大まかに、(1) イベントの監視、検知を行なうモジュール、(2) コンデション、アクションの評価を行なうモジュールに分割して考えることができる。イベントマネージャが前者であり、演算処理マネージャが後者である。ルールマネージャは、この二つの管理とルールの管理を行なう。

### 8.2.1 イベントマネージャ

ルールマネージャはルールが登録されるとルール名とイベント部をイベントマネージャに送る。イベントマネージャではイベントリスナーを生成しイベントを監視する。イベントリスナーにはコンポジットイベントリスナー (以下 CEL) とプリミティブイベントリスナー (以下 PEL) があり、それぞれ、コンポジットイベント、プリミティブイベントを監視する。

次のようにイベントリスナーによってイベントを監視通知する。(1) イベントマネージャはそのイベント部とルール名をもとに CEL を作成する。(2) CEL はイベント部から監視するプリミティブイベントを特定、これを監視する PEL を作成し、イベント発生源にこれを登録する。(3) イベントが発生すると PEL は CEL にこれを報告し、CEL はルール名と発生イベントをイベントマネージャに報告する。(4) イベントマネージャではルールマネージャにルール名と発生イベントを報告する。イベントの発生にともなって (3)(4) を繰り返す。

### 8.2.2 演算処理マネージャ

イベントが発生すると、ルールマネージャは対応するルールのコンディション部、アクション部を演算処理マネージャに送る。演算処理マネージャではコンディション部、アクション部の評価を行なう。配信指定以外はメディアータを呼び出すことによって評価する。

コンディション評価は、コンディション部を *Info Weaver* における選択演算に変換し、これをメディアータに投入する。例えば、先の `selecting_XML` ルールの `if` 節  $S.Title = 'XML'$  であれば、 $\sigma_{Title='XML'}(S)$  に変換する。

アクション評価は、`do` 節にある *Info Weaver* における代数式を順次メディアータに投入することで実行する。ただし、配信指定がある場合、ルールマネージャを通じて配信要求を配信サブシステムに渡す。

## 8.3 配信サブシステム

配信マネージャでは、配信要求をルールマネージャから受けとり、(1) 配信指定されたりレレーションをメディアータより取得、(2) これを各配信方法での適切な形式への変換、(3) 配信方法に対応した配信エージェントへの変換結果及び配信要求の通知を行なう。

配信エージェントでは、それぞれの配信方法に応じた配信を行ないクライアントに情報を届ける。

本環境では、メディアータによって統合結果としてレレーションが生成され、これを配信マネージャが取得する。実際には、配信方法毎に配信対象となるレレーションにはスキーマに関する制限がついており、メディアータによってこれに沿ったレレーションを作成する。例えば、配信方法がメールであれば、レレーションはコンテンツとして指定された属性のみの単項レレーションであり、その属性の型は SD 型または文字列型でなければならない。

## 9 情報統合利用例の実現

2 節における統合利用例の実現方法を示す。

株式情報データベースは図 11 のスキーマを持ち、会社名、購入時の株価、株数、注目している株価の閾値が格納されている。製品情報ニュースチャンネルは図 12 の DTD を持ち、このような SD 値をコンテンツとするストリームレレーションにラッパーによって変換されるものとする。株価情報チャンネルは、会社名、業種、株価の各項目を配信しており、ラッパーによって図 13 のようなスキーマをもったストリームレレーションとなるものとする。

| ID | CorpName | BuyPrice | Amount | Threshold |
|----|----------|----------|--------|-----------|
|----|----------|----------|--------|-----------|

図 11. リレーション StockInfo

|         |   |                        |
|---------|---|------------------------|
| article | = | seq(info,body)         |
| info    | = | seq(title,date,c-info) |
| c-info  | = | rep(c-name)            |
| body    | = | rep(p)                 |

図 12. ProductNews DTD

| Name | Category | StockPrice | TS |
|------|----------|------------|----|
|------|----------|------------|----|

図 13. ストリームレレーション StockPriceStream

まず、製品情報ニュースチャンネルの配信単位の一週間分を保持するためのルールを作成する。次に、株価情報チャンネルに業種が IT である配信単位が到着した時、目的とする統合処理を行なうためのルールを記述する。今回の要求では、株価に変動があるとすぐに配信することが要求されているので、このルールを用いて情報配信も行なう。上記二つのルールを記述することで配信単位受信時の処理、統合処理、その結果の配信を実現可能である。それぞれのルールを図 14 に示す。ルール `OneWeekProductNews` は前者、ルール `NotifyStock` は後者のルールに相当する。ルール `OneWeekProductNews` では `if` 節が省略されており、イベントの発生があるとすぐにアクションが実行される。

この例においては、ルール `OneWeekProductNews` は製品情報ニュースチャンネルの情報を保持する `Receiving Rule` である。ルール `NotifyStock` は株価情報チャンネルの情報の到着をイベントとし、その中で情報

## 10 関連研究

(1) 情報配信サービスと他の情報源の統合利用, (2) 情報配信サービスを用いた統合結果の配信という両者の情報配信サービス利用形態を併せ持った異種情報源統合環境は, 本研究の特徴であり, 同様の機能を持つ情報源統合環境の報告は筆者らの知る限り見当たらない.

情報源の状況を監視し, 一定の条件が成立した時にあらかじめ登録した問合せに基づく結果を返す, イベント駆動型の異種分散情報統合システムとしては, OpenCQ [8] がある. 条件成立時に実行される問合せを Continual Query と名付けている. これは, トリガ条件, 問合せ, 停止条件から成り, ユーザに結果として前回問合せ時との差分を返す. ある条件の成立に起因して問合せが行なわれる点, 異種情報源を対象としている点では本研究と共通点をもつ. しかし, OpenCQ で考慮しているのは上記 (2) の特別な場合のみである. また, メディエータが保持するリレーションのような概念が OpenCQ にはなく, アクションとして記述できるのは問合せのみであるため, 統合操作を記述する上で本アプローチより制限が生じる.

情報配信サービスのクライアント側での配信情報の蓄積, 格納を目的としたアクティブデータベースシステムに SADB [9] がある. SADB は, 配信情報の蓄積管理に ECA ルールを用いている点では, 本研究と関連が深い. しかし, SADB ではある一つの情報配信サービスに対応したクライアントを主な対象としているのに対し, 本研究では複数の情報配信サービスを対象とした情報の蓄積を行なうことも可能である. さらに, 本研究では異種情報源統合システムを利用した他の情報源との統合処理も可能であり, また, 統合結果などの配信を情報配信サービスを用いて行なうこともできる.

既存の情報配信サービスとして 2 節で述べた Point-Cast などがある. これらは, ユーザに情報を能動的に配信することはできるが, 複数の異なる情報配信サービスからの情報選択を行なうことはできない. 本研究では, 各情報配信サービスにラッパーを作成し異種情報源統合システムに追加することでこれを実現可能である. さらに, その配信情報は異種情報源統合システムにおける問合せを用いて, 複数の情報源の情報を統合したものとすることも可能である.

複数の情報配信サービスからの情報選択を目的とした研究としては DBIS [10, 11] や Muffin [12, 13] がある. DBIS では, サーバとクライアントの間にインフォメーションブローカと呼ばれるソフトウェアモジュールをはさむことで, サーバの違いを吸収しユーザプロファイルに基づく情報のフィルタリングを行なっている. Muffin では複数の配信サービスからユーザの欲しい配信情報を探し, これを含む新たな仮想的なチャンネルを構成することができる. 情報を選択する際, ユーザの与えるキーワードと配信情報の内容との類似度のみでなく, 配信頻度, 情報の内容を考慮した新鮮度や人気

```

Rule OneWeekProductNews
on: arrival(ProductNews)
do: Last1WeekNews += ProductNews;
Last1WeekNews :=  $\sigma_{TS > now - 1week}$ (Last1WeekNews);

```

```

Rule NotifyStock
on: arrival(StockPriceStream)
if: StockPriceStream.Category = 'IT'
do: # 一週間分の製品情報から必要な要素構造を抽出し会社毎にまとめる
CorpNews :=
Pns(News) → CorpNews( $\nu_{ns} = (News)$ )( $\pi_{C-name, News}$ (
P(Title, Date, Body) → News( $\mu_{cs}$ (
UC-info → cs(O, C-name)(
UInfo → (Title, Date, C-info)(
UArticle → (Info, Body)(Last1WeekNews )))))));
# 株価が閾値を越えた会社を選択する
NotableStockInfo :=
 $\pi_{CorpName, StockPrice, BuyPrice}$ (StockInfo
 $\bowtie_{Threshold > StockPrice \wedge CorpName = Name}$ (
 $\sigma_{Category = 'IT'}$ (StockPriceStream));
# 閾値を越えた会社の株価情報と製品情報をまとめる
NewsAndStockInfo :=  $\pi_{NotifyInfo}$ (
P(CorpName, StockPrice, BuyPrice, CorpNews)
→ NotifyInfo(
NotableStockInfo  $\bowtie_{CorpName = C-name}$  CorpANews);
# 作成した統合結果をメールで配信する
DeliverMail('hiro@x.ac.jp', 'StockPriceHasChanged',
NotifyInfo)[NewsAndStockInfo];

```

図 14. 利用例における演算式<sup>1</sup>

の再構築を行ない, その結果を配信しており, 4 節で述べた三つの全ての役割を担っているルールである.

本環境を Java を用いて実装した (図 15), 情報源として RDBMS Oracle, メールによる情報配信サービス, 今回の実験のために新たに作成したメッセージ配信サービスを対象とし, それぞれにラッパーを配置している. 配信サブシステムにおける配信方法としては, メールによる配信を実装している.

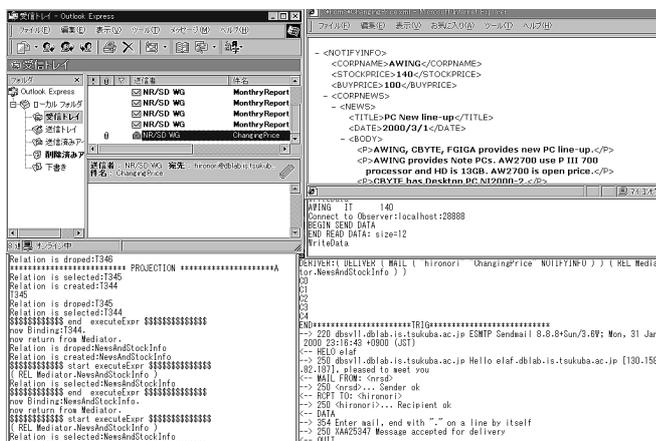


図 15. スクリーンショット

<sup>1</sup>問合せ式中 now は現在の時刻を表す. do 節において行頭が”#”の行はコメントである.

度を指標としていることが特徴の一つである。両研究とも複数の情報配信サービスを含む情報配信サービスの高度利用を目的としているものであるが、本研究におけるような、より一般的な異種情報源の統合利用や、上記情報配信サービスの利用形態 (1),(2) の選択的利用を考慮していない。

定期的な問合せを発行することで情報源を監視するシステムとして Tapestry [14] がある。Tapestry は電子メールやネットニュースを格納する追加のみを考慮したデータベースを想定し、問合せを定期的に行きユーザーに結果の差分を返すものである。配信情報サービスは次々と情報が配信されてくるため、これを追加のみのデータベースと見ることができる。しかし、この研究でのデータ処理は限定的なものであり、イベント駆動型のアクションを記述できない。また、他の情報源との統合も考慮されていない。

## 11 まとめと今後の課題

本研究では、情報配信サービスの異種情報源統合システムにおける利用を目的として、(1) 情報配信サービスと他の情報源との統合利用、(2) 情報配信サービスを用いた統合結果の配信という両者の利用形態を併せ持った異種情報源統合環境を構築した。配信単位の到着や時間的要因をイベントとし、能動的な統合処理や統合結果の配信を実行するルールを用いることで、配信単位の到着に伴う持続的な統合処理や統合結果の効果的な配信を行なうことが可能である。また、本研究では、異種情報源統合システムに、情報配信サービスラッパー、ルールサブシステム、配信サブシステムを付加することで、上記 (1)(2) の利用形態を選択的に利用可能な異種情報源統合利用環境の構築が可能であることを示した。また、*InfoWeaver* を想定した本環境の具体的な実現について述べた。

今後の課題として以下があげられる。分散した異種情報源統合環境では、イベントの通知に遅延が生じることが予想されるため、この点を考慮したイベントモデルを検討する必要がある。また、にルールの効率的な処理方式とそれに関係した問合せ最適化について考えていきたい。

## 謝辞

本環境の構築に関して熱心にご討議いただいた、筑波大学電子・情報工学系データベース研究室の諸氏に感謝する。また、本研究の一部は、文部省科学技術研究費補助金、ならびに次世代情報放送システム研究所の助成による。

## 参考文献

[1] R. Domingue and K. R. Dittrich, "An Overview and Classification of Mediated Query Systems", *ACM SIGMOD Record*, Vol.28, No.3, pp.63-72, 1999.

- [2] M. Franklin and S. Zdonik, "Data in your face : push technology in perspective", *Proc. SIGMOD '98*, pp.516-519, 1998.
- [3] 角谷和俊, 宮部義幸, "放送型情報配信のためのモデルとシステム", *情報処理学会論文誌データベース*, Vol.40, No.SIG8-(TOD4), pp.141-157, 1999.
- [4] N. W. Paton and Oscar Diaz, "Active Database Systems", *ACM CS*,1999.
- [5] A. Morishima and H. Kitagawa, "InfoWeaver: Dynamic and Tailor-Made Integration of Structured Documents, Web, and Databases", *Proc. ACM Digital Libraries '99*, pp.235-236, 1999.
- [6] H. Kitagaw