

## 放送型情報配信のためのモデルとシステム

角 谷 和 俊<sup>†</sup> 宮 部 義 幸<sup>††</sup>

本稿では、近年、注目を集めている放送型情報配信技術について述べる。まず、従来提案されている放送型情報配信の方式、システム、およびサービスについて、配信の特性による比較と考察を行う。特に、プッシュ型およびプル型と呼ばれている配信技術を対比し、その方式と実現されているシステム・サービスについて詳しく述べる。次に、放送型情報配信の技術課題を整理し、従来技術の位置づけを明確にする。また、データ管理の側面から解決すべき課題を抽出し、データベース技術の適用についての考察を行う。最後に、今後の放送型情報配信に関する研究の方向性について述べる。

### Models and Systems for Information Broadcast

KAZUTOSHI SUMIYA<sup>†</sup> and YOSHIYUKI MIYABE<sup>††</sup>

In this paper, we survey the recent researches of models and systems for broadcast information delivery mechanisms. We compare the proposed models, systems and services, and discuss the properties of data delivery methods. Especially, we describe the difference between push-based and pull-based data delivery mechanisms. We also discuss the problems of data broadcast and analyze the recent researches based on delivery mechanisms and content management. Finally, we discuss a prospect on the future research on data broadcast.

#### 1. はじめに

近年、インターネットやデジタル放送を用いた情報配信サービスが注目を集めている<sup>1)~3)</sup>。特に、WWW(World Wide Web)による情報提供サービスが急激に増加し、様々な情報を共有することが可能になってきている。しかし、提供される情報が多くなるに従って、欲しい情報を探し出すことが困難になってきている。例えば、WWWではユーザは検索エンジン等を用いて、欲しい情報が含まれるであろうURL(Universal Resource Locator)を探し出し、絞り込みやフィルタリングを行うことにより情報にアクセスしなければならない<sup>4)</sup>。しかしながら、たとえ欲しい情報にアクセスできても、その情報はその後、追加・変更・削除される可能性があり、最新情報を得るためには頻繁に検索を行わなければならない。

このような問題を解決するために、欲しい情報をユーザが毎回探し出すのではなく、情報を自動的に配信する放送型情報配信の枠組みが提案されている<sup>5)</sup>。

すなわち、ユーザが情報を明示的に指定して取り出すのではなく、あらかじめ設定されたプロフィールに従った配信、および情報自身の更新によって引き起こされる配信により、情報が自動的にユーザに配信される技術である。これらの枠組みでは、検索エンジン等を用いた情報の取得に比べるとユーザの負担はかなり減少するが、得られた情報に矛盾が含まれている可能性があること、および配信頻度が多く転送量が増大すること等、いくつかの問題がある。ここで、技術的課題として挙げられるのは、配信された情報の一貫性管理、配信の効率化等である。

本稿で論じる放送型情報配信は、必ずしも通常のテレビジョン放送などで実現可能なもののみではなく、双方向機能や高速ネットワーク通信機能など、現在の放送インフラ以外の部分の発展を前提としている議論も含んでいる。また、インターネットによる放送型情報配信も議論の対象としている。

本稿では、従来提案されている放送型情報配信に関する技術について、配信の特性による比較と考察を行う。また、技術課題を整理し、従来技術の位置付けを明確にする。さらに、データ管理の側面から解決すべき課題を抽出し、データベース技術の適用についての考察を行う。なお、本分野の用語・用法は各従来技術ごとに異なる場合が多いが、出来る限り混乱を避ける

<sup>†</sup> 松下電器産業株式会社 (現在, 神戸大学都市安全研究センター)  
Matsushita Electric Industrial Co., Ltd. (Research  
Center for Urban Safety and Security, Kobe University)

<sup>††</sup> 松下電器産業株式会社 デジタル・ネットワーク戦略推進室  
Matsushita Electric Industrial Co., Ltd.

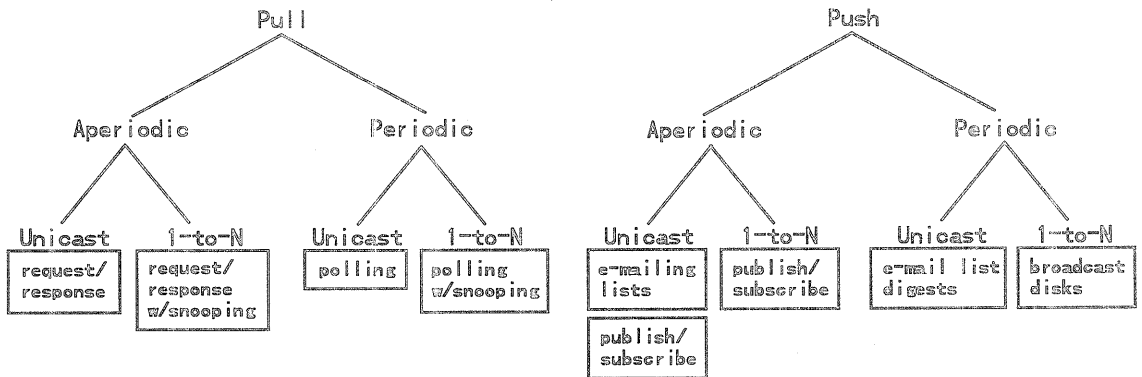


図1 データ配信方式の分類

Fig. 1 Classification of Delivery Mechanisms

ために、本稿で用いる用語・用法は以下の通りとする。  
**方式** ある機能を実現するためのアルゴリズムやメカニズムのこと。

**システム** 方式を実現したソフトウェア（ハードウェアがある場合はそれを含む）のこと。

**サービス** システムによりユーザに提供される（具体化された）機能のこと。

**モデル** いろいろな「方式」を考える上での論理的な枠組みのこと。サーバー・クライアントモデル、検索モデルなどの一般用語として用いる。

すなわち、方式、システム、およびサービスはレイヤーの異なる概念である。

以下、2章では放送型情報配信と呼ばれている技術分野について述べる。特に、プッシュ型、およびプル型と呼ばれている技術について詳しく述べる。3章では、従来、放送型情報配信のために提案されている方式、システム、およびサービスの分類を行なうために、3つの配信特性を基に分類し、特徴と用途について述べる。4章では、提案されている方式、システム、およびサービスについて配信方式と効率の観点から考察を行う。また、5章では、コンテンツ管理の観点から考察を行う。6章では、実用レベルの放送型情報配信の例としてインターネットを用いたシステムとサービスについて紹介する。7章では、現状のインフラにおける技術課題について考察を行なうとともに、筆者らの見解と本分野の今後の方向性についても述べる。

## 2. 放送型情報配信とは

インターネット、デジタル放送、およびモバイルコンピューティングなどのインフラが整備されるに従って、ユーザが取得可能なデータや情報は増加の一途をたどっている。しかし、欲しい情報を取得するためには、検索エンジン等を用いて情報を捜し出さなければ

ならない。検索エンジンを用いた情報取得には以下のような問題点がある。

- どのサーバーにどのような情報があるかをあらかじめ知る必要があり、それぞれ個別に検索を行わなければならない
- 最新情報を得るためには頻繁に検索を行わなければならない

従って、ユーザが常に最新情報を得るためには、サーバーに対して検索要求を常に出し続けなければならない。この問題を解決するために、ユーザがあらかじめ登録した要求（ユーザプロフィールなどに蓄積）、あるいは情報自身の更新により、最新情報を自動的にユーザに配信する技術が提案されている。特に、最新情報が重要な意味を持つアプリケーション、例えば、株価、スポーツの試合結果、交通情報、およびオンライン新聞などでは非常に有用である。最近は、このような情報をリアルタイムに更新し、ユーザに最新情報を提供するサーバーが増えてきている。このような情報配信を**放送型配信**と呼ぶ。この配信の特性として、以下の項目が挙げられる。

- 情報ソースからの下方向転送容量とユーザからの上方向転送容量が非対称（下方向が圧倒的に大きい）である
  - ユーザ数が多いためシステムが大規模である
- 一方、上述のような配信をプッシュ(push-based)型と呼ぶことがある<sup>6),7)</sup>。これに対して、検索エンジン等を用いてユーザ自身が情報を引き出す方法をプル(pull-based)型と呼ぶ\*。

Maryland大学のFranklinとBrown大学のZdonikらは、1998年のACM SIGMOD'98において、プッ

\* 本稿では、「プッシュ型」と「プル型」は、「方式」、「システム」および「サービス」のすべてに対する修飾語として用いる。

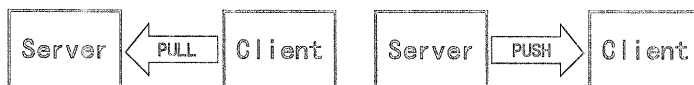


図2 プル型配信とプッシュ型配信  
Fig. 2 Pull-based and Push-based Delivery

シュ型技術 (Push Technology) の展望に関する招待講演<sup>8)</sup>を行なっている\*。ここでは、プッシュ型技術に対する社会的な期待と実際の技術がかなり懸け離れていることを指摘している。すなわち、「プッシュ型サービスと呼ばれているアプリケーションが、実はプル型配信方式」を用いているために混乱を招いていることを指摘している。また、プッシュ型は放送型やマルチキャスト型と同一と考えられているが、実際のインプリメントでは送信先を指定する単一型を用いていること、あるいは、周期的にクライアント側からサーバーにデータ取得要求を出すことで、ユーザには自動的にデータが送信されるように見える方式を用いていることを指摘している。さらに、上記招待講演ではプッシュ型とプル型の分類について論じている。3章では、この論文で分析されている配信の分類について述べるとともに、この他にどのような技術課題があるかについて述べる。

### 3. 配信方式の分類

#### 3.1 配信の特性

Franklin らは、配信の特性を3つの観点で分類している。

- (1) プル型 vs. プッシュ型
- (2) 周期的 (periodic) vs. 非周期的 (aperiodic)
- (3) ユニキャスト型 vs. 1-to-N 型

本節では、それぞれの分類の観点について述べる。これらの特性は直交しているので、組み合わせとしては8種類に分類される。これらの分類を図1に示す。

##### 3.1.1 プル型 vs. プッシュ型

データの配信方式には、プル型とプッシュ型の2種類がある(図2)。プル型とは、サーバーやリポジトリに対してクライアントから検索要求があった場合に、サーバー側がその要求に対して処理を行うという、通常のサーバー/クライアントモデルに基づく方式である。プル型の場合、サーバーからクライアントへのデータ配信は、クライアントの要求のみによっておこる。一方、プッシュ型とは、サーバーがクライアントに

対して、自動的にデータ配信を行う方式のことである。すなわち、クライアント側からの要求によってデータ配信が行われるのではなく、サーバーが能動的にデータ配信を行う。プッシュ型の場合、サーバーがどのような情報を、どのタイミングで、どのクライアントに送付するかについては、あらかじめユーザプロファイルなどをサーバーに登録する必要がある。

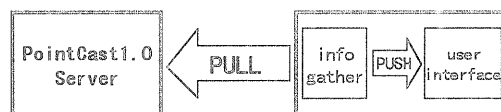


図3 PointCast 1.0 のデータ配信  
Fig. 3 Delivery Mechanism of PointCast 1.0

PointCast 社のニュース配信 PointCast Network<sup>10)11)</sup>, Microsoft 社の Internet Explorer<sup>12)</sup> のチャンネル機能, InterMind 社の InterMind Communicator<sup>13)</sup>, BackWeb Technologies の BackWeb<sup>14)</sup>, および Marimba 社の Castanet<sup>15)</sup> は、一般的にはプッシュ型サービス (システム) と呼ばれている。しかし、データ配信の観点から見れば、プル型に分類されるべきである。これらのサービス (システム) は、ユーザにはサーバーからデータが自動的に配信されてくるように見えるが、それはクライアント内でのアプリケーションとユーザインタフェース間の処理によるものである。すなわち、サーバーとクライアント間のデータ配信はプル型である。図3に、PointCast 1.0でのサーバー/クライアント間のデータ配信の様子を示す。また、その他のサービス (システム) についても同様にプル型である。このように、現在、実現されているサービス (システム) のほとんどは、プル型の配送方式に基づいている。従って、本稿ではこれらのプッシュ型と呼ばれるサービス (システム) はプル型配信方式を用いているとの立場をとる。個々の配信方式については後述する。

##### 3.1.2 周期的 vs. 非周期的

データ配信のタイミングとしては、周期的配信と非周期的配信に分けられる。周期的配信は、あらかじめ決められた時間間隔によりサーバーからクライアントへデータ配信が行われる仕組みである。一方、非周期的配信は、イベント駆動型である。すなわち、非周期

\* 1998年の第1回マルチメディア・コンテンツの高次処理に関する国際会議 (AMCP'98) においても、データ放送と配信 (Data Broadcast and Dissemination) に関する招待講演<sup>9)</sup>を行なっている。

的プル型の場合は、ユーザのデータ要求により配信が行われる。また、非周期的プッシュ型の場合は、データ更新により配信が行われる。

例えば、株価のアプリケーションの例では、一定間隔で株価情報が送られてくるのが周期的、ある銘柄の株価に変動があった場合のみ、その株価情報が送られてくるのが非周期的である。

### 3.1.3 ユニキャスト型 vs. 1-to-N 型

配信先については、ユニキャスト型と 1-to-N 型に分けられる。ユニキャスト型は、サーバーから、特定された一つのクライアントへの配信である。一方、1-to-N 型は複数のクライアントに同一内容を配信する。

1-to-N 型には 2 つの種類がある。マルチキャスト型と放送型である。マルチキャスト型は特定多数のクライアントに同一内容を配信する。マルチキャストをサポートするシステムでは、まず、ルーターにデータ配信を行い、次にそのルーターがあらかじめリストアップしているクライアントに対しデータを配信することで特定多数への送信を実現している。マルチキャストについては、配信されるべきクライアントを特定し、転送を保証するネットワークプロトコルが開発されている。

一方、放送型は受信可能な不特定多数のクライアントに対してデータ配信を行う。例えば、モバイル環境では、サーバーからの電波が届く範囲内のクライアントが、サーバーから発信されたデータを受信する。また、インターネット放送などで動画などを配信する場合も放送型である。

マルチキャスト型と放送型は、課金サービスなどを考える上では区別する必要がある。すなわち、全ての受信者が誰かわかっているマルチキャスト型では、例えば、有料メーリングリストサービスのようなことが可能である。一方、放送型では、CA(Conditional Access) という有料サービスを実現している。これは、スクランブルをかけたデータを不特定多数に放送し、お金を払った人にだけ鍵を渡してスクランブルを解かせる方式である。

## 3.2 データ配信の型

本節では、前節のデータ配信の特性に基づき、その配信方式と周期性の組み合わせによる、4 つの基本的な配信方法について述べる<sup>8)</sup>。

**非周期的プル型** 典型的なサーバー/クライアントモデルで用いられている。

**周期的プル型** リモートセンシングなどの監視用アプリケーション等で用いられている。また、3.1.1 節でも述べたが、WWW やインターネットベースの

プッシュ型システムでは、サーバーとクライアント間のデータ配信はこの型の方式を用いている。ただし、図 3 にも示すように、ユーザからはプッシュ型と見えるので、この型を用いたサービス(システム)は、一般的にはプッシュ型サービス(システム)と呼ばれている。

**非周期的プッシュ型** publish/subscribe プロトコルに基づくアプリケーションは非周期的プッシュ型に基づいている<sup>16)~18)</sup>。publish/subscribe システムでは、ユーザはプロフィールなどのユーザ情報をサーバーに呈示する。このユーザ情報には、そのユーザが受信したい情報に関する記述が含まれている。publish/subscribe プロトコルは、その性質上、本来は 1-to-N 型になるはずである。しかし、実際のシステムでは、現在のインターネットの制約により個々にユニキャストを行うことで実現されている。このようなシステムの例として、電子メールのメーリングリストや IP マルチキャストを用いたシステムが挙げられる。

**周期的プッシュ型** 周期的プッシュ型はさまざまなシステムで用いられている。ユニキャスト型配信の周期的プッシュ型の例では、定期的にダイジェスト版を送付する電子メールのメーリングリストが挙げられる。例えば、ある決まったスケジュール(例えば、一週間)に基づいたダイジェストをおくる場合などである。周期的プッシュ型のメーリングリストを非周期的プッシュ型のメーリングリストと比べると、非周期的プッシュ型では投稿者からのメールが頻繁に送られてくるのに対し、周期的プッシュ型では投稿されたメールが一旦管理者により編集されて送付されるので、投稿者から直接送付されるメールを排除し、効率的に情報を取得することができる。1-to-N 型配信の周期的プッシュ型の例では、TeleText<sup>19),20)</sup>、Data-Cycle<sup>21)</sup>、Broadcast Disks<sup>22)23)</sup>、モバイルデータベース<sup>24)</sup>などが挙げられる。

以上、4 つのデータ配信の型について述べたが、各データ配信方式はネットワークなどの配信インフラストラクチャなどの外部環境や技術レベルの変化により、それぞれ個別の研究の経過をたどってきている。

## 4. 配信方式と効率

### 4.1 配信システムの構成要素

放送型情報配信システムは、サーバー側で、情報ソースを蓄積・管理し、あるタイミングでその情報をクライアント側に配信するものである。以下の 3 つの構成要素に分類される。

- (1) サーバー: 配信される情報ソースの管理
- (2) クライアント: 情報ソースの受信
- (3) ブローカー: サーバーとクライアント間の調整

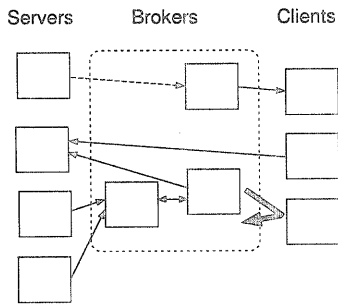


図4 システムの構成要素  
Fig. 4 A System Architecture

図4は一般的な配信システムの構成要素を示す。右側への矢印はデータの配信を、左側への矢印はデータの配信要求を示している。この図では、様々な配信経路と配信要求経路があるが、実際のシステムではこの中の一部の構成要素・経路が用いられている。配信システムは、基本的にはサーバーとクライアントから構成されるが、付加的な構成要素としてブローカー\*が存在するシステムもある。ユーザからは、サーバーからデータが配信されているように見えるが、実際はブローカーがクライアントへ配信を行っている。ブローカーを用いる理由は、配信の効率化、およびIPやHTTPプロトコルの制約などである。

放送型配信システムの課題の一つは、配信サービスを如何に効率的に行なうかである。現在、インターネット、無線LAN、CATV、およびデジタル衛星放送などの進化により、伝送帯域の大幅な拡張が行われている。しかし、ネットワーク資源は有限であるので、如何に少ない伝送帯域で、より多くのデータを配信するかという問題は重要な論点である。そこで、配信するデータの組み合わせ、頻度、および順序を工夫することで、必要なデータを効率的に伝送する方式がいくつか提案されている。また、配信するデータの冗長性を軽減し、効率化する方法も提案されている。

#### 4.2 配信方式とシステム

放送型情報配信のために、いくつかの配信方式とシステムが提案されている。本節では、これまでに提案されている主なものについて紹介する。また、エージェント技術を用いた放送サービスの枠組み、および大規

模分散データベースを用いたデータ放送についても述べる。

以下、4.2.1節で周期的プッシュの例として Broadcast Disks を、4.2.2節では Broadcast Disk を改良し、データ配信の一部に非周期的プル型を用いる Balancing Push and Pull を紹介する。また、4.2.3節では非周期的プッシュの例として CQ Server を、4.2.4節では非周期的プッシュを実現するためにエージェント技術を用いた AgentCast について紹介する。

##### 4.2.1 Broadcast Disks

Broadcast Disks<sup>22)</sup> は、周期的プッシュ型の一つである。サーバーは連続的かつ定期的に、クライアントにデータ配信を行う。図5に示すように、配信されたデータ(放送チャンネル)は部分的に送信されているが、クライアントからはまとまった”データディスク”と見える。クライアントは、この仮想的なディスクを通してサーバーにアクセスすることになる。データはサイズ、およびスピードが異なるディスクに配置されて、放送チャンネルはこれらのディスクを多重送信する。想定している環境は以下の通りである。

- クライアントの数と、クライアントがどのデータの取得を要求しているかというアクセスパターンは不変である。
- サーバー側のデータは不変である。
- クライアントがデータを取り出すのはオンデマンド(on-demand)方式である。
- クライアントは上方向通信(upstream)を利用しない。

これらの想定環境において、データを効率的に配信するためには、データを送信する手順とデータの組み合わせ、およびキャッシュ戦略などを決定しなければならない。すなわち、決定すべき方針は以下の2点である。

- (1) サーバは、クライアント数とデータへのアクセスパターンを与えられた場合、いかに放送プログラムを組織化するか
- (2) クライアントは、サーバーが放送プログラムを送信した場合、ローカルキャッシュを管理していかにパフォーマンスを向上させるか

まず、一つ目の方針について述べる。サーバーは、クライアントの要求を満たすために放送プログラムの内容、順序などを構成しなければならない。一方、クライアントはデータを取得する場合、まずローカルメモリの検索を行う。取得すべきデータが無い場合は、放送データを監視することによりデータを待つ。サーバーが構成する放送プログラムは、図6に示すように、

\* システムによっては、ファシリテータ、メディアータなどと呼ばれる。

(a)flat 型, (b)skewed 型, および (c)multi-disk 型の 3 種類が考えられる.

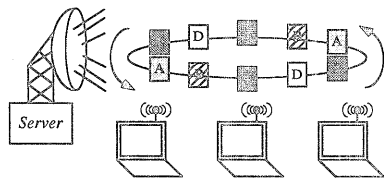


図 5 Broadcast Disks の構成  
Fig. 5 Architecture of Broadcast Disks

flat 型の場合は, flat 型は各データを単純に繰り返して配信するためデータの待ち時間は各データとも同一である. 一方, skewed 型と multi-disk 型では, データを異なる頻度で放送するので, 特定のデータを強調することができる. skewed 型と multi-disk 型の差は, 同じデータ (図 6 では A) をまとめて配置するか, 分散させるかの違いである. 理論的には, このような non-flat 型の放送プログラムの生成は, バンド幅の割当問題として取り扱うことができる. 一般に, アクセスが均一の場合は flat 型の効率が良い. しかし, アクセスの歪みが増加すると共に, non-flat 型の方が効率が良くなる. non-flat 型の中でも, 待ち時間の期待値としては, multi-disk 型はいつも skewed 型より効率が良い.

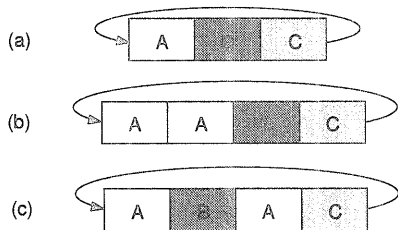


図 6 放送プログラムの例  
Fig. 6 An Example of Broadcast Programs

次に, 二つ目の方針であるキャッシュについて述べる. クライアントがローカルキャッシュに保存しておくデータは, データ取得要求の多いデータ (hottest data) より, そのクライアントにとってアクセスする可能性が高いデータのほうが望ましい. ここでは, PIX(P Inverse X) が提案されている. つまり, キャッシュのアクセス可能性 (P) / データの放送頻度 (X) の小さい値を取るデータを入れ替える方法である. しかし実際は, インプリメントが難しいので, PIX の代替手段として, LRU(Least Recently Used) / データの放送頻度 (X) の小さい値を取るデータを入れ替える方式

LIX(LRU Inverse X) が提案されている. キャッシュの詳しいアルゴリズムについては, 他の文献<sup>25)</sup>に詳しい.

#### 4.2.2 Balancing Push and Pull

前節で述べた Broadcast Disks のように, 定期的に配信する周期的プッシュ型とは対照的に, 従来のクライアント/サーバーデータベースなどでは, プル型の request/response 方式を採用している. プル型はクライアントの多様なニーズに対応するには有利であるが, サーバーの負担, 無線通信や CATV などの back-channel 増設のコストがかかるという問題点がある.

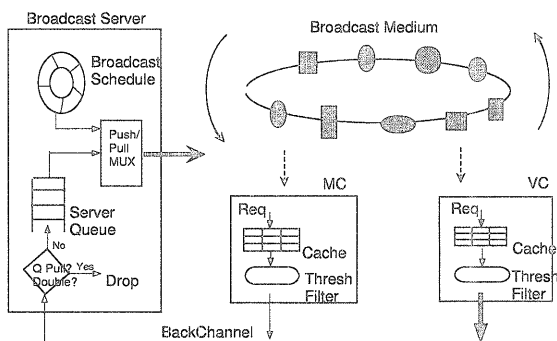


図 7 Balancing Push and Pull の構成  
Fig. 7 Architecture of Balancing Push and Pull

Balancing Push and Pull<sup>23)</sup> は, Broadcast Disk に back-channel を追加し, プッシュ型とプル型の融合を図るものである. back-channel は point-to-point 方式でサーバーと接続されている. クライアントは back-channel を介して, サーバーにリクエストを出することができる\*.

この方法では, プル型バンド幅 (PullBW) と呼ばれるパラメータの設定によって, プル型とプッシュ型のバランスをとっている.

**Pure-Push** すべてのバンド幅をプッシュ型周期配信に使い (PullBW=0%), back-channel を使用しない.

**Pure-Pull** すべてのバンド幅をプル型周期配信に使い (PullBW=100%), プッシュ型周期配信を使用しない.

**Interleaved Push and Pull (IPP)** プッシュ型とプル型をミックスする (0 < PullBW < 1)

IPP の改善として, ある閾値を設定してクライアントの back-channel の利用を制限することも考えられる.

\* すなわち, back-channel はプル型である.

あるページがプッシュ型周期配信に予定される前に、すでにスロットが不足している場合に限り、プル可能であるとする方法である。閾値はサイクル長の百分率(すなわちプッシュ期間)で表される。プッシュ型とプル型の統合については、この他にも様々な方法が提案されている<sup>26)</sup>。

4.2.3 CQ Project

CQ(Continual Queries: 連続的検索)<sup>27),28)</sup>は、サーバーからクライアントに配信されてくるデータを監視(update monitoring)することにより、イベント発生を検知し、あらかじめユーザが設定した処理を行う方式を提案するものである。本方式では、時間経過とともにサーバーで更新されるデータに対する検索として、全てのデータに対して毎回質問を行うのではなく、前回の質問で得られた結果をクライアントで保持し、更新された部分に対してのみ新たに質問を行うことで、効率的にデータを取得することが可能である。

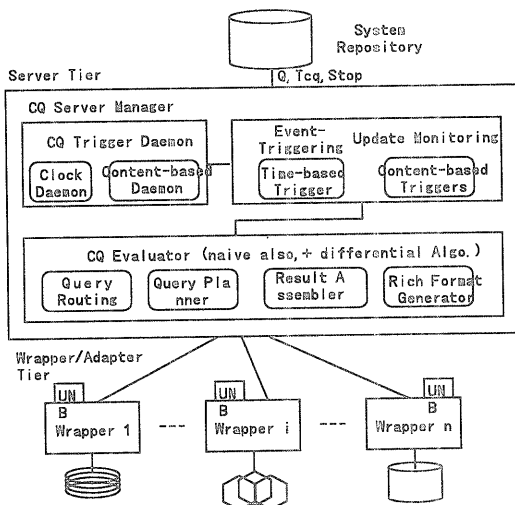


図 8 CQ サーバーの構成  
Fig. 8 The CQ Server

CQ Project の目的は、インターネットやイントラネットのようなオープンかつ動的環境における、イベント駆動型情報配信でのデータ監視方式の提案とツールキット開発である。図 8 に、CQ サーバーの構成を示す。

CQ は、 $(Q_{cq}, Trg_{cq}, Term_{cq})$  のように定義される。ここで、 $Q_{cq}$  は質問 (すなわち、SQL など)、 $Trg_{cq}$  はトリガーを起こす条件、 $Term_{cq}$  は停止条件である。CQ は、 $Trg_{cq}$  が真となる時点から連続的に実行され、 $Term_{cq}$  が定義しているイベントが発生すると終了する。CQ は Time-based Event Triggering と Content-

based Event-Triggering の 2 つのタイプの  $Trg_{cq}$  をサポートしている。

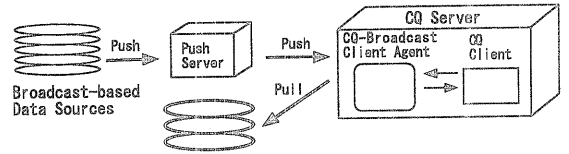


図 9 CQ サーバーを用いたデータ配信  
Fig. 9 The Data Delivery by CQ server

図 9 は CQ サーバーを用いたデータ配信を示している。放送型データソースからプッシュサーバーにデータが配信され、プッシュサーバーが CQ サーバーにデータを配信する。CQ サーバーでは CQ 放送型クライアントエージェントがデータの監視や質問処理を行い、CQ クライアントにデータを配信する。また、CQ サーバーは、必要に応じて他のデータソースに対しデータ要求を行いデータを取得する。

CQ は、 $Trg_{cq}$  が真になる度に、 $Q_{cq}$  をデータベース全体に対して再評価を行なう。しかし、前回変化したデータに対してだけに処理を行なうことで、効率化を図っている (Differential Evaluation Method<sup>29)</sup>)。

4.2.4 AgentCast

AgentCast<sup>30)</sup> は、モバイルエージェントによるデータ配信を実現するシステムである。エージェントは、必要に応じて自動的に移動し、協調することで多様な番組を提供することが可能となる。AgentCast はこれまでに述べた方式とはエージェントモデルを用いている点で異なる。今後、ネットワーク型のマルチエージェントモデルは、放送型情報配信の分野への適用が注目されている。

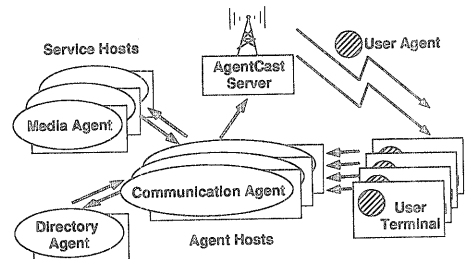


図 10 AgentCast の構成  
Fig. 10 Architecture of AgentCast

エージェントには 3 種類ある。User Agent はサービス提供者によって作られる。Media Agent は放送局にある様々な資源を制御する。Communication Agent は User Agent と Media Agent の間に立ち、主とし

てユーザの応答の集約・集計を行なう。処理の流れは以下の通りである(図10)。

- (1) サービス提供者が AgentCast Server から User Terminal に向けてエージェントを配信する
- (2) User Terminal に到着したエージェントがサービスを開始する
- (3) サービスを終えたエージェントは、User Terminal で消滅、Agent Host に移動してユーザの指定したサービスを実行、あるいは AgentCast Server に戻って再び User Terminal に配信される

4.3 配信の効率化

効率化については、データの配信順序をデータ間の相関関係を用いることで決定する方式、ハイパーテキストを配信する時に複数のハイパーテキストから参照されているデータを共有データとして配信順序を決定する方式、およびハイパーメディアを送信する際に関連する部分ごとに分割し配信する方式などが提案されている。本節では、それぞれの方式の代表的な方式とシステムについて紹介する。

4.3.1 データ間の相関性を考慮した方式

CBS 法 (Correlation-based Scheduling)<sup>31)</sup> は、データ間の相関関係を考慮したスケジューリング方式である。本方式では、データの配信順序の決定にアイテム間の相関関係を用いる\*。相関関係とは、例えば、サーバーが様々なサイトのホームページの HTML ファイルを配信している場合には各ホームページから異なるページへのリンクが張っており、リンクしている複数のページに対し、まとめてアクセス要求を出す確率が高いものと考えられる。この確率は、相関関係を表し、各ページ間で相関性の強さは異なる。

クライアントがまとめてアクセス要求を出す確率が高いアイテムを、放送プログラム内に隣接して配置することで、アクセスの平均応答時間の短縮を図るヒューリスティックなアプローチである。図11はCBS法を適用した様子を表している。a から f のアイテムを左図のように完全グラフで表し、グラフ内のハミルトン閉路のうちで、閉路上の重み  $W_{ij} = 1 - P_{ij}$  の総和が最小となる閉路を選択する。相関性が強いアイテム間は太線で示している。

アルゴリズムは以下の通りである。

- (1) 各データアイテムを節点、節  $i, j$  間の重みを  $W_{ij} = 1 - P_{ij}$  とした完全グラフを用意する。ただし、 $P_{ij}$  はクライアントがアイテム  $i$  と  $j$

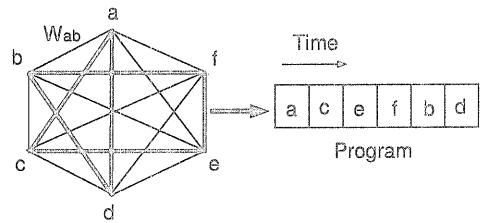


図11 CBS 法  
Fig. 11 Correlation-based Scheduling Method

- に一括して同時にアクセス要求を起こす確率とする ( $\sum_{j=i+1}^M \sum_{i=1}^M P_{ij} = 1$  かつ  $P_{ij} = P_{ji}$ ).
- (2) すべての節点を一度ずつ通り、かつ、その道に含まれる辺の重みの総和が最小となる閉路を見つける。
  - (3) 閉路上のアイテムを順に配置する。

また、放送頻度を考慮しCBS法を拡張したFCBS法 (Frequency-and Correlation Based Scheduling)<sup>31)</sup> も提案されている。

4.3.2 共有を考慮したハイパーテキスト伝送方式

配信データがハイパーテキストの場合、データが共有されていることが多いことに着目し、共有データを1つにまとめて伝送する順序配意決定方式<sup>32)~34)</sup>が提案されている。

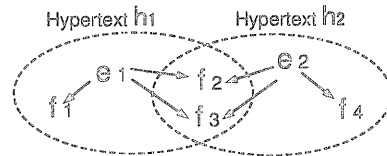


図12 共有データを考慮したハイパーテキスト伝送方式  
Fig. 12 Method for Hypertexts with Shared Files

図12のように、2つのハイパーテキスト  $h_1$  と  $h_2$  があり、それぞれの構成要素は  $\{e_1, f_1, f_2, f_3\}$ ,  $\{e_2, f_2, f_3, f_4\}$  とする。ファイル  $f_2$  と  $f_3$  はいずれも2つのハイパーテキストで共有されているものとする。 $h_1$  と  $h_2$  を別々にパッケージングして伝送すると、 $f_2, f_3$  を2回送ってしまうことになり効率が悪い。そこで、 $h_1$  を配信する場合、非共有部分の  $e_1, f_1$  と共有部分の  $f_2, f_3$  に分けて配信する。同様に、 $h_2$  を配信する場合、非共有部分の  $e_2, f_4$  と共有部分の  $f_2, f_3$  に分けて配信する。すなわち、配信パッケージは  $e_1, f_1, e_2, f_4$ 、および  $f_2, f_3$  の3つのパッケージ  $H_1, H_2, F$  として配信する。この時、(1) 非共有パッケージ  $H_i$  を共有パッケージ  $F$  にできるだけ接近させる、(2) 各非共有パッケージ  $H_i$  のアクセス確率により共有パッケージ  $F$  からどれくらい離れた配信順番にするかを

\* 本方式では配信されるデータの単位をアイテムと呼んでいる。



決定する, という方針により配置を決定する.

#### 4.3.3 時系列ハイパーメディア

DVX\*は, デジタル放送において映像と同期したデータを送出するための伝送フォーマットである<sup>36),37)</sup>. この伝送フォーマットでは, 映像と同期した付加情報として場面对応したハイパーメディアを, そのシーンごとに時分割して送ることが可能である. 例えば, 図13の上部に示すようなハイパーメディア  $H$  を映像の付加情報として配信する場合は, シーンに応じてセグメント  $S_1, S_2, S_3$  に分割し, 分割したセグメントをシーンに同期するように配信する. 図13の下部は, 分割されたセグメントが時間経過とともに順次配信されている様子を示している. ここでは,  $S_1$  と  $S_2$  の2つのセグメントが配信されているが,  $S_1$  については, そのバージョン  $S_1$  (バージョン1) と  $S_1$  (バージョン2) が切り替わって配信されている. バージョンは, あるノードだけが変更されているがセグメントの構造が同じ場合に用いられる. 例えば, ノードに野球放送におけるストライクカウントなどの数値データが含まれている場合は, この数値だけを変更し, 新しいバージョンとして配信する. このように, DVX では, ハイパーメディアをセグメントとバージョンの2つの概念で分割し時系列配信を行うことが可能である.

DVX で配信される各セグメントは, NI(Navigation Information) と呼ばれ, ユーザのインタラクションのためのアイコン, ボタン, スクリプト, およびデータが含まれる. NI の伝送フォーマットは DSM-CC データカールセル<sup>38)</sup> と互換性がある.

DVX の利点は, 映像と同期した付加情報にリアルタイムにアクセスすることが出来る点, および, ハイパーメディアを時間軸分割することで帯域およびクライアントの受信メモリを低減できる点である.

### 5. コンテンツ管理

4章では, 配信方式と効率について述べた. 情報配信システムのもう一つの大きなトピックは, 配信されるコンテンツの管理である. コンテンツの管理は, サーバー側とクライアント側の2ヶ所での処理が考えられる. サーバー側では, コンテンツは常に更新されている場合が多いためコンテンツ自身の管理や, 新たにどのコンテンツを配信すべきかという差分配信の問題がある. 一方, クライアント側では, 受信した大量のコンテンツの中からユーザの所望しているコンテンツを

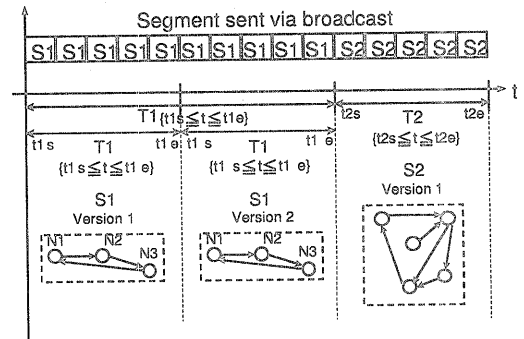
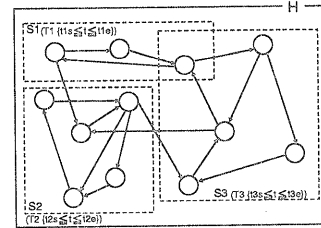


図13 時系列ハイパーメディア伝送フォーマット DVX  
Fig. 13 DVX: An Delivery Format for Time-lined Hypermedia

選択する必要がある. この選択要求に対応するために, コンテンツを組織化あるいは構造化しユーザの選択処理を支援する方式が提案されている.

コンテンツ管理は, 組織化/構造化と一貫性管理の2つに大別することができる. 5.1 ではコンテンツの組織化/構造化, 5.2 では一貫性管理についてそれぞれ述べる. 紹介するシステムの中で, クライアント側のみで処理を行うのは, 5.1.1 節のアクティブ情報ストア, 5.1.2 節の Virtual Channel, 5.2.1 節の Time-Varying Query Expressions 5.2.2 節の Tapestry であり, サーバー側とクライアント側の両方で処理を行うのは, 5.1.3 節の時間依存リンク, 5.2.3 節の配信コンテンツのバージョン管理である.

#### 5.1 組織化/構造化

放送型情報配信システムでは, 大量のデータが常時配信されてくるため, ユーザが自分で整理するには手間がかかりすぎる. 従って, 必要なデータだけを自動的に組織化・構造化する仕組みが必要である. ただし, ユーザが必要なコンテンツを如何に取り出すか, あるいは取り出したコンテンツを如何に構造化するかがポイントとなる.

##### 5.1.1 アクティブ情報ストア

アクティブ情報ストア<sup>39)</sup>は, Super Active Database system (SADB) を用いて, 大量に配信される放

\* 現在は, IplusTV<sup>35)</sup> と改称して, 衛星デジタル放送 DIRECTV Japan にて標準仕様として採用されている.

送型データを効率的に抽出し、格納、再利用できるように設計されたシステムである。SADBはECA(Event-Condition-Action)ルールを用いて記述されるアクティブデータベースを放送型に拡張したものである。

アクティブ情報ストアは、ユーザインタフェースとして木構造を持つビューを用いる。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。受信されたデータは1つまたは複数の節点に分類される。この時、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成されるため、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 Virtual Channel

デジタル放送では、通常の番組放送以外の情報を伝送するデータ放送も可能であり、番組のための付加的なデータを伝送する電子番組ガイド(EPG: Electronic Program Guide)が配信されている<sup>40)</sup>。

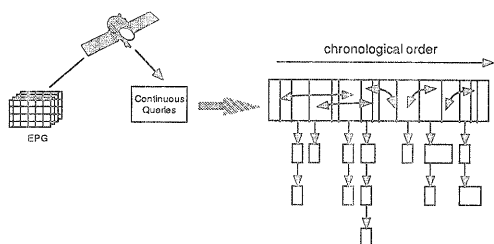


図 14 Virtual Channel の構成  
Fig. 14 Architecture of Virtual Channel

Virtual Channel<sup>41)</sup>は、サーバーから送られてくる EPG データを再構築する枠組である。すなわち、EPG データ上でハイパーリンクを動的に生成することで、各ユーザごとのビューを構築するものである。Virtual Channel は以下の手順にしたがって処理が行なわれる。

図 14 に Virtual Channel の構成を示す。EPG データは放送局のサーバーで管理され、一定間隔で更新が行われる。また、衛星放送などの経路により、クライアントに対して常時配信されている。クライアント側では、追加・変更・削除されたデータに対して連続的の検索を行う。検索された結果は番組の放送時間によって時系列的に配置される。また、それぞれの番組情報に関連している他の番組へのリンクが自動的に生成される。図 14 の右側の部分では横軸は時間軸を、縦方向の矢印は関連する番組情報へのリンクを表す。

### 5.1.3 時間依存リンク

時間の経過に伴って変更される情報、時系列データ、

およびリアルタイム性を持つ情報をハイパーメディアとして配信する場合、情報が頻繁に更新されるために、その情報間の参照情報であるリンクが正しいアンカーを結合しているかどうかを検証しなければならない。

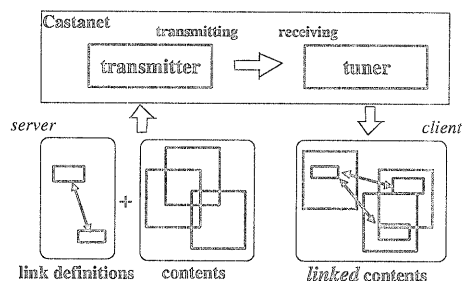


図 15 放送型ハイパーメディア Mille-feuille  
Fig. 15 Mille-feuille: A Broadcast Hypermedia

時間依存リンク<sup>42),43)</sup>は、アンカーや文書に有効時間を付与し、リンクの時間的有効性を管理することが可能である。すなわち、単純に動的リンクを生成するだけでなく、リンク元とリンク先のアンカー、およびリンク元とリンク先のコンテンツの時間的有効性を検査し、時間的矛盾の無いリンクを生成することが可能である。例えば、ある期間だけ存在するリンク(長野オリンピック期間中のみ有効な、文字列「オリンピック」から文字列「長野」へのリンクなど)を動的に生成することが可能である。

ハイパーメディアにおけるリンク構造の時間的 management を行うモデルとして HyTime<sup>44)</sup> や SMIL<sup>45)</sup> のリンク構造が提案されている。しかし、これらで提案されている時間管理はコンテンツの再生時間の管理のみであり、時間依存リンクで扱っているコンテンツ自身の有効時間という概念は含まれていない。

図 15 に時間依存リンクのプロトタイプシステム Mille-feuille を示す。Mille-feuille では、サーバーにおけるリンク定義はコンテンツとは別に管理される。すなわち、従来のハイパーメディアのリンクのようにコンテンツの中にリンクを埋め込むのではなく、サーバー側ではリンクの定義だけを行う。Mille-feuille では、サーバーは Castanet のトランスミッターを用いてリンクとコンテンツの配信を行い、クライアントはチューナにより受信する。クライアント側では、コンテンツとリンクに定義されている時間的有効性を検証し、有効であるリンクのみを生成する。

### 5.2 一貫性管理

コンテンツの一貫性管理では、時系列コンテンツに対する処理が必要になる。配信されるコンテンツには

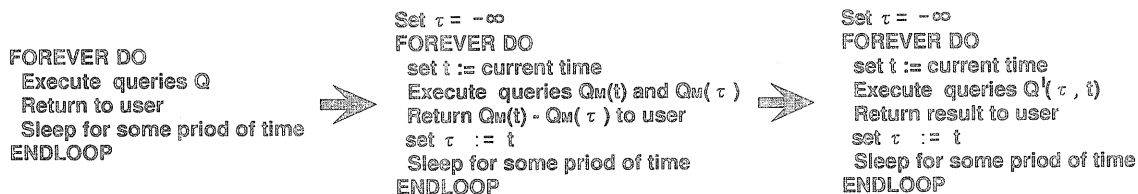


図 16 連続質問の変換

Fig. 16 Convert of Continuous Queries

リアルタイムな情報が含まれており、ユーザのアクセス時間によってコンテンツの有効性が異なる場合が考えられる。従って、時系列コンテンツの時間的一貫性管理機構やバージョン管理が必要となる。また、連続的な処理を逐次的な質問に変換し差分結果を結合することで効率的に行う方法もいくつか提案されている。コンテンツの一貫性管理では、コンテンツに対する処理を如何に減らすことができるかがポイントとなる。

### 5.2.1 Time-Varying Query Expressions

Time-Varying Query Expressions<sup>46)</sup>は、時間的に変動があるデータに対する関係データベースの質問処理のモデルである。時間的に変動のあるデータに対しては、変動分に対して差分のみを計算することは有効である。ただし、質問処理の結果が時間の経過とともに変化する場合には適用できない。この問題を解決するために、スーパービューという概念を提案している。スーパービューは、時間に依存する選択述語を満たす可能性のある全てのタプルを含むビューである。

質問  $Q$  の時刻  $t$  における選択を  $\sigma_{[P]}$  とする。  $P$  は条件である。時間的に変動が無い場合は、  $t_1 \leq t_2$  で以下の式が成り立つ。

$$\sigma_{[P]}(Q \text{ at } t_2) \equiv ((\sigma_{[P]}(Q \text{ at } t_1)) - Q^D) \cup \sigma_{[P]}(Q^I)$$

ここで、  $Q^I$ 、  $Q^D$  は  $t_1 \leq t_2$  での挿入と削除である。次に、時間変動の質問のために、スーパービュー  $(\sigma_{[P]}(Q))_s$  を定義する。

$$t_1 \leq t_2 \Rightarrow (\sigma_{[P]}(Q))_s \text{ at } t_1 \supseteq \sigma_{[P \text{ at } t_2]}(Q \text{ at } t_1)$$

この時、以下の式が成り立つ。

$$(\sigma_{[P]}(Q)) \text{ at } t_2 \equiv ((\sigma_{[P \text{ at } t_2]}((\sigma_{[P]}(Q))_s \text{ at } t_1)) - (\sigma_{[P]}(Q))^D) \cup (\sigma_{[P]}(Q))^I$$

すなわち、  $(\sigma_{[P]}(Q)) \text{ at } t_2$  は、スーパービューと中間の削除の差および中間の挿入の和である。このように、Time-Varying Query Expressions はスーパービューの演算を用いて差分結果を得る方式である。Time-Varying Query Expressions では、ビュー管理や影響伝播についても検討されている。

### 5.2.2 Tapestry

Tapestry<sup>47)</sup>は、データが連続的に追加されるのみ (append-only) のデータベースに対して、連続質問処

理を行なうシステムである。格納されるデータが単純な単項のデータであれば、以前に検索した結果と、その時点から増加したデータに対する検索結果を合わせれば正しい結果が得られる。しかし、その質問に結合演算や時間ファクターを含んでいる場合は、単純な差分処理では正しい結果が得られない。このシステムの本来的対象は、電子ドキュメント、電子メール、電子ニュースの記事のフィルタリングであるが、この方法でいわゆる放送型の配信データ、すなわちサーバーからクライアントに配信されるデータを処理することが可能であると考えられる。

図 16 は、質問  $Q$  を連続的に行うために、最小単調質問  $Q_M$  と逐次的質問  $Q^I$  に変換する方法を示している。左側は、質問  $Q$  を単に周期的に実行する手順である。すなわち、質問  $Q$  を実行し、その結果をユーザに返し、一定時間が経過した後、質問を繰り返すものである。真中は、最小単調質問  $Q_M$  を用いて連続質問を行う手順である。前回の質問実行時刻  $\tau$  における結果と、現時刻  $t$  における結果の差をユーザに返すことを繰り返すものである。右側は、  $Q_M(t) - Q_M(\tau) \subseteq Q^I(\tau, t)$  を満たす逐次的質問  $Q^I$  を用いて、同様に質問結果を得る。Tapestry では、逐次的質問  $Q^I$  の性質と効率についての議論が行われている。詳細については述べないが、Tapestry の有用性は、任意時刻の質問実行の場合にも正しい結果が得られることである。

### 5.2.3 配信コンテンツのバージョン管理

放送型情報配信システムで配信されるコンテンツは、サーバー側で時々刻々と変更される可能性がある。サーバー側ではコンテンツが更新される毎に、新しいバージョンとして配信する。一方、クライアントでは新しいバージョンを受け取り、以前のバージョンと置き換える。しかし、すべてのクライアントが新しいバージョンを受信するとは限らない。すなわち、いくつかのクライアントでは何らかの理由により受信できないことがある。例えば、その時点で受信状態にない場合\*などである。クライアントが受信状態でない状

\* 電源が入っていない、ソフトウェアが起動していない、あるいは

態があった場合には、コンテンツに矛盾が生じないようにバージョン管理を行わなければならない。

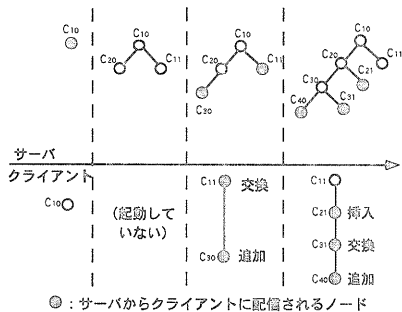


図 17 放送コンテンツのバージョン管理

Fig. 17 Version Control Mechanism for Broadcast Contents

放送型情報配信システムの構成管理を扱うためのバージョン管理方式がいくつか提案されている。ここでは、コンテンツの有効時間に着目した方式<sup>(48), (49)</sup>を紹介する。この方式では、サーバー側でのコンテンツの更新を2つの種類に分けて管理を行なう。一つは、内容の更新、もう一つは有効時間の更新である。コンテンツ  $C_s$  は以下のように定義される。

$$C_s = (id, version, Tr, [V_1, V_2])$$

ここで、 $id$  は識別子、 $version$  はバージョン番号、 $Tr$ 、 $[V_1, V_2]$  はそれぞれトランザクション時間、有効時間である。バージョン番号は、メジャー番号とマイナー番号の2つの組で表される。メジャー番号が内容の更新、マイナー番号が有効時間の更新を表す。メジャー番号が更新された場合は左に、マイナー番号が更新された場合は右に成長するバージョン木が得られる。一方、クライアント側では、バージョンリストとして管理される。

図 17 に配信の例を示す。上部はサーバーでのバージョン、下部はクライアントでのバージョンを示す。左から右に時間的経過を表している。ここでは、4つの時刻におけるサーバー、およびクライアントのバージョンの更新を示している。まず、左から1番目の時点では、サーバーにコンテンツがバージョン  $C_{10}$  として登録され、クライアントに配信されている。2番目の時点では、コンテンツの内容が変更されバージョン  $C_{20}$  となり、かつ有効時間が変更されたバージョンは  $C_{11}$  となる。通常では、 $C_{20}$  と  $C_{11}$  が配信されるはずであるが、クライアントが起動していないために配信されていない。3番目の時点では、更に内容が更新さ

れバージョン  $C_{30}$  となり、かつ有効時間が変更されたバージョンは  $C_{11}$  となる。この時点では、クライアントは起動しているので、 $C_{30}$  と  $C_{11}$  が配信される。4番目の時点では、同様に内容と有効時間が更新されたバージョンが作成され、クライアントに配信されている。

## 6. 放送型情報配信システムの実例

本節では、放送型情報配信サービスの中でインターネットを用いたシステムについて簡単に紹介する。

### 6.1 PointCast

PointCast<sup>(10), (11)</sup> は、チャンネルメタファーを用いたニュース配信システムである。チャンネルは、ニュース、天気予報、スポーツなどのジャンルに分かれている。また、それぞれのチャンネルにはサブチャンネルが用意されている。

ユーザは、どのチャンネルを購読するかをあらかじめ設定しておく。また、サーバーに対し、どのような頻度でデータを受信するかも、あらかじめ設定しておく。既に述べたが、PointCast は実際はプル型のサービスであるので、ユーザが設定するデータ受信の頻度がプルの間隔となる。

用意されているインタフェースは3種類ある。チャンネルビューアー、ティッカー、およびスマートスクリーンである。チャンネルビューアーは階層型の通常のブラウザである。スマートスクリーンは、スクリーンセーバーの代わりにニュースの見出しを表示する。また、ティッカーは、テレビ放送のように横に細いテロップに自動的に見出しが表示される。

PointCast はニュース配信システムとして完成度の高いサービスであるが、配信データは、テキストと静止画に限られている。

### 6.2 Castanet

Castanet<sup>(50), (51)</sup> は、Transmitter(トランスミッタ)と呼ばれるサーバーアプリケーションと、Tuner(チューナ)と呼ばれるクライアントアプリケーションから構成されている。Castanet は、ソフトウェアの配信、自動インストールなどが可能である。配信するコンテンツは、HTML 文書、Java アプレット、Java アプリケーションなどである。

Castanet では、Transmitter と Tuner の間で効率的な配信を行なうために、差分配信のしくみを用いている<sup>(52)</sup>。概要は以下の通りである(図 18)。

- (1) 全体のチェックサムを Transmitter に渡す
- (2) チェックサムを比較
- (3) チェックサムツリー全体を転送

はモバイル環境では受信地域に入っていないなどである。

- (4) ログとプロフィールを取得
- (5) Tuner がログを更新
- (6) チェックサムツリーを比較して、異なるファイルを探し、ファイルを一致させるためのコマンドを作成して配信
- (7) Tuner がコマンドを実行

Castanet は、差分情報を用いて効率的な配信を実現しているが、コンテンツの内容、特に有効時間等を考慮した配信の枠組みは実現されていない。

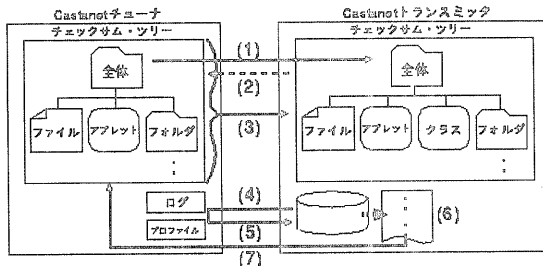


図 18 Castanet の差分配信のしくみ

Fig. 18 Castanet Differential Delivery Mechanism

### 6.3 BackWeb

BackWeb<sup>14)</sup> は、情報配信やソフトウェアの自動インストールを行なうシステムである。特徴としては、配信方式として、ブロードキャスト、ナローキャスト、およびパーソナルキャストの 3 つを選択できることである\*。ブロードキャストは、そのチャンネルの購読者全員に同じコンテンツを配信する方式である。ナローキャストは、ユーザがチャンネルの中で選択したコンテンツのみを配信する方式である。パーソナルキャストでは、hook と呼ばれるプラグインを用いて受信するコンテンツの条件を記述することが可能である。

コンテンツは infoPak 単位で配信される。この infoPak に対して、有効期限を設定することが可能なため、コンテンツの有効性のチェックに用いることが可能である。

BackWeb は複数の配信方式を選択することが可能であり、配信形態を細かく設定することが可能である。しかし、コンテンツ蓄積や構造化などのコンテンツ管理の面では十分とは言えない。

## 7. 今後の課題と展望

前章までの議論では、従来提案されている放送型情報配信に関する技術の分類について述べた。本章では、

技術課題についての考察をおこなうとともに筆者らの見解と本分野の今後の方向性について述べる。

### 7.1 配信方式と効率

配信アーキテクチャについては、伝送経路の性能によるところが大きいが、ここでは現在のインフラにおける課題について考察する。

(1) **完全なプッシュ型配信**: 現在のプッシュ型サービスは、配信において一部分はプル型の配信形態を採用している。これは、ネットワーク帯域の制限による要因が大きいと考えられる。ただし、ネットワーク資源は有限であるので、パフォーマンスとスケーラビリティの問題は残る<sup>8)</sup>。

(2) **プロフィール**: ユーザの興味をモデル化して、必要な情報だけを配信することが求められている<sup>53)~56)</sup>。ユーザプロフィールの定義については、情報検索の分野で、ベクトル空間モデルやソーシャルフィルタリングなどの手法が提案されており、これらの技術を如何に放送型情報配信の枠組みに入れるかが課題となっている<sup>9)</sup>。

(3) **分散データベース**: 従来の分散データベース技術と比べると、前提とする環境やインフラが大きく違っている。特に、ネットワーク環境では、インターネットの進化によりこれまでの分散データベースの様なクローズな環境だけでなく、オープンな環境にも対応する必要がある<sup>57)</sup>。すなわち、従来は異種の分散したデータベースを結合することが目的であった。しかし、現在の環境では、結合対象はデータベースのみならず、多種のコンテンツに対応しなければならない。また、扱うコンテンツもリアルタイム性のあるものが多い。

従来の分散データベース技術は、データの基本的操作に対しては有効であると考えられるが、アプリケーションやシステムにおける特有の処理については研究の余地があると考えられる<sup>58)~60)</sup>。

(4) **放送型ハイパーメディア**: 従来の配信モデルでは、対象とするデータはシンプルなデータストリーム型であった。しかし、今後は HTML などのハイパーリンクを持ったデータや静止画、動画などを扱うことが可能な配信モデルが求められている。また、XML によるチャンネル記述のフォーマット提案なども行われており、動向が注目される<sup>61)62)</sup>。

(5) **テレビジョン放送による情報配信**: 日本におけるテレビジョン放送を用いたデータ放送で既に実用化されているものは、地上アナログ放送を用いたものと、CS(衛星)デジタル放送を用いたものの 2 種類に分けられる<sup>63)64)</sup>。地上波データ放送は、垂直帰線消

\* ここでの用語は本論文での用法とは違っているが、紹介するシステムでの用語であるので、このまま使用する。

去期間 (VBI: Vertical Blanking Interval) を利用してデータ配信を行なう方法が主流である。例えば, IT ビジョン, ADAMS, データパレードなどである。一方, CS デジタルでは, SkyPerfecPC やインタラク TV などのサービスが実用化されている。

テレビジョン放送を用いたデータ放送は今後の重要な技術分野であり, 放送サービスと連携した技術開発が行われている\*。

本稿では詳しく紹介しなかったが, 情報配信の伝送経路として, 今後もっとも有望な配信形態である。理由としては, 数メガから数十メガの広い伝送帯域幅をもっていること, 視聴者数に関わらずインフラの整備費用が一定であることなどが挙げられる。ただし, 対称的な双方向通信ではなく, 実質的に単方向通信となるため, インタラクティブ性を実現する場合などには工夫が必要である<sup>36),37)</sup>。

(6) 課金モデル: 放送型情報配信によって大量の情報がユーザに届けられるが, その情報の価値を正確に表現できる課金モデルが必要になる。

## 7.2 コンテンツ管理

(1) クライアントでのコンテンツ蓄積: これまでの放送型情報配信では, クライアントが受け取ったデータはキャッシュ等に蓄積されるだけで, 永続的に蓄積されることは考慮されていない。これは, もともとモバイル環境での使用など, 2 次記憶が少ない環境を想定していたためである。ところが, 今後は大容量の記憶媒体をクライアントで用いることが考えられる。

(2) 組織化/構造化: 今後, 配信されるデータが大容量になるに伴い, 配信されたデータを構造化してユーザに分かりやすく提示する技術が必要になる<sup>65),66)</sup>。例えば, 全体の傾向を把握できるようなマップやユーザインタフェースのメタファである。特に, 時系列データが多く含まれるため, 時間軸の表現がキーとなる。また, 従来の科学データの視覚化の手法だけでなく, テキスト情報を音声や CG 合成により提示する研究も進められている。例えば, データベースや Web 文書を, TVML(TV Program Making Language)<sup>67)~69)</sup> を用いてテレビ番組風のコンテンツに変換し提示する方式<sup>70),71)</sup> も提案されている。

(3) 一貫性管理: 連続的検索に関しては, 5.2 で紹介した従来の研究で, 検索質問の性質や能力が詳しく検討されている。しかし, コンテンツそのものの有効時間など, コンテンツの内容に関する一貫性管理は十分とは言えない。特に, バージョン管理については,

従来のバージョンモデルの能力では記述できない場合がある。特に, サーバーでのバージョンと, クライアントでのバージョンが同一であると保証できない場合は, かなり煩雑なモデルになると考えられる<sup>48)</sup>。

## 8. おわりに

放送型情報配信技術は, 現在注目を集めている技術の一つである。その一因として, ネットワークや家庭内の情報機器の性能向上が挙げられる。これにより, 一般ユーザがこれまでと比べ物にならないほどの大量のデータに簡単にアクセスすることが可能になっている。

放送型情報配信を実現するための要素技術としては, これまで述べた通り, 従来のデータベース技術が適用できる場合も少なくはない。しかし, トランザクション, 検索モデル, およびコンテンツ管理についてはまだまだ研究の余地があると考えられる。

謝辞 本論文をまとめるにあたり御指導いただいた神戸大学工学部情報知能工学科 (大学院自然科学研究科情報メディア科学専攻) 田中克己教授, 都市安全研究センター都市情報システム研究分野 上原邦昭教授, サーベイに御協力いただいた田中研究室放送型 DB グループの馬強君, 近藤宏行君, 野田玲子君, 服部多栄子君, および赤迫貴行君 (現在, NTT コムウェア) に感謝致します。また, 有益なコメントを頂いた査読者の方々に感謝いたします。

## 参考文献

- 1) 石田晴久: インターネットはここまできた, 情報処理, Vol. 39, No. 5, pp. 393-399 (1998).
- 2) 藤原洋: デジタル放送技術, 情報処理, Vol. 39, No. 9, pp. 936-937 (1998).
- 3) European Broadcasting Union: Digital Video Broadcasting (DVB) Systems, [http://www.dvb.org/dvb\\_index.html](http://www.dvb.org/dvb_index.html).
- 4) 林良彦, 小林喜嗣: WWW 上の検索サービスの技術動向, 情報処理, Vol. 39, No. 9, pp. 861-865 (1998).
- 5) 北村泰彦: インターネット上での知的情報統合, アドバンスト・データベース・シンポジウム'98(ADBS'98), 情報処理学会, pp. 167-174 (1998).
- 6) Gerwig, K.: The Push Technology Rage... So What's Next?, *ACM netWorker: The Craft of Network Computing*, Vol. 1, No. 2, pp. 13-17 (1997).
- 7) 谷萩裕之: プッシュ型インターネット技術入門 -Web が変わる-, ソフト・リサーチ・センター (1998).

\* 詳細については他の文献<sup>63)</sup>を参照されたい。

- 8) Franklin, M. and Zdonik, S.: "Data In Your Face": Push Technology in Perspective, *Proc. of ACM SIGMOD '98*, pp. 516-519 (1998).
- 9) Aksoy, D., Altinel, M., Bose, R., Cetintemel, U., Franklin, M. and Zdonik, S.: Research in Data Broadcast and Dissemination, *Proc. of 1st International Conference on Advanced Multimedia Content Processing (AMCP'98)*, pp. 196-210 (1998).
- 10) Ramakrishnan, S. and Dayal, V.: The Point-Cast Network, *Proc. of ACM SIGMOD '98*, p. 520 (1998).
- 11) PointCast: PointCast Network, <http://www.pointcast.com>.
- 12) Microsoft: Internet Explorer, <http://www.microsoft.com/windows/ie>.
- 13) InterMind: InterMind Communicator, <http://www.intermind.com>.
- 14) BackWeb Technologies: BackWeb, <http://www.backweb.com>.
- 15) Marimba: Castanet, <http://www.marimba.com>.
- 16) Oki, B., Pfluegl, M., Siegel, A. and Skeen, D.: The Information Bus - An Architecture for Extensible Distributed Systems, *Proc. of 14th SOSP* (1993).
- 17) Yan, T. and Garcia-Molina, H.: SIFT - A Tool for Wide-area Information Dissemination, *Proc. of USENIX Technical Conference*, pp. 177-186 (1995).
- 18) Gance, D.: Multicast Support for Data Dissemination in OrbixTalk, *IEEE Data Engineering Bulletin*, Vol. 19, No. 3 (1996).
- 19) Ammar, M. and Wong, J.: The Design of Text Broadcast Cycles, *Perf. Evaluation*, No. 5 (1985).
- 20) Wong, J.: Broadcast Delivery, *Proc. of the IEEE*, Vol. 76, No. 12 (1988).
- 21) Herman, G., Gopal, G. and Lee, K.: The Data Cycle Architecture for Very High Throughput Database Systems, *Proc. of ACM SIGMOD '87*, pp. 97-103 (1987).
- 22) Acharya, S., Alonso, R., Franklin, M. and Zdonik, S.: Broadcast Disks: Data Management for Asymmetric Communication Environment, *Proc. of ACM SIGMOD '95*, pp. 199-210 (1995).
- 23) Acharya, S., Franklin, M. and Zdonik, S.: Balancing Push and Pull for Data Broadcast, *Proc. of ACM SIGMOD '97*, pp. 183-194 (1997).
- 24) Imielinski, T., Viswanathan, S. and Badrinath, B. R.: Energy Efficient Indexing on Air, *Proc. of ACM SIGMOD '94*, pp. 25-36 (1994).
- 25) Franklin, M.: *Client Data Caching: A Function for High Performance Object Database Systems*, Kluwer Academic Publishers (1996).
- 26) Franklin, M. and Zdonik, S.: A Framework for Scalable Dissemination-based Systems, *Proc. of OOPSLA '97*, pp. 94-105 (1997).
- 27) Pu, C. and Liu, L.: Update Monitoring: The CQ project, *Proc. of 2nd Worldwide Computing and Its Applications (WWCA '98)*, LNCS, Vol. 1368, pp. 396-411 (1998).
- 28) Liu, L., Pu, C., Tang, W., Buttler, D., Biggs, J., Zhou, T., Benninghoff, P., Han, W. and Yu, F.: CQ: A Personalized Update Monitoring Toolkit, *Proc. of ACM SIGMOD '98*, pp. 547-549 (1998).
- 29) Liu, L., Pu, C., Barga, R. and Zhou, T.: Differential Evaluation of Continual Queries, *Proc. of 16th IEEE Distributed Computing Systems*, pp. 458-465 (1995).
- 30) 白井剛, 田島孝一, 下條真司, 宮原秀夫: Agent-Cast: 次世代放送サービスを実現するフレームワークの提案, 情報処理学会研究会報告 97-DPS-85, pp. 267-272 (1997).
- 31) 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送受信方法について, 情報処理学会研究会報告 99-DBS-116, pp. 95-102 (1998).
- 32) 石川裕治, 田辺雅則, 箱守聡, 井上潮: ハイパーテキスト間のデータ共有を考慮した放送型情報提供方式, 信学技報 (97-160), pp. 121-126 (1997).
- 33) 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, 情報処理学会研究会報告 (34-8), 情報処理学会, pp. 55-60 (1997).
- 34) 井上潮, 石川裕治, 田辺雅則, 箱守聡: 放送型情報提供システムにおけるキャッシュ管理方式, 信学技報 (97-160), pp. 127-132 (1997).
- 35) 岡村和男, 角谷和俊, 宮部善幸: IplusTV: 時系列ハイパーメディアを用いたデータ放送方式とその実装, 情報処理学会論文誌データベース, Vol. 40, No. SIG5(TOD2), pp. 91-102 (1999).
- 36) 角谷和俊, 楠見雄規, 岡村和男: デジタル放送インタラクティブ・データ配信のためのカラーセル型送出方式 DVX とその応用, アドバンス・データベース・シンポジウム'97(ADBS'97), pp. 23-30 (1997).
- 37) Nabeshima, S., Okamura, K., Kakiuchi, T., Sumiya, K., Takao, N. and Miyabe, Y.: Extended Digital Video Broadcasting with Timed Hypermedia, *Proc. of 1st Int'l Conference on Advanced Multimedia Content Processing* (1998).
- 38) ISO/IEC 13818-6: Generic Coding of Moving Pictures and Associated Audio Information - Part6: Extension for Digital Storage media Command and Control (DSM-CC) (1997).

- 39) 寺田努, ソムヌックサグアントラクーン, 塚本昌彦, 西尾章治郎, 三浦康史, 松浦聡, 今中武: 放送型データ受信のためのアクティブデータベースについて, 情報処理学会研究会報告 99-DBS-116(1), 情報処理学会, pp. 119-126 (1998).
- 40) Rosengren, J.: Electronic Programme Guides and Service Information, *Philips Journal of Research*, Vol. 50, No. 1/2, pp. 253-265 (1996).
- 41) Sumiya, K. and Tanaka, K.: Virtual Channel: Dynamic Structuring and Continuous Queries for Data on the Air, *Proc. of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM'97)*, pp. 715-720 (1997).
- 42) Sumiya, K., Noda, R. and Tanaka, K.: Hypermedia Broadcasting with Temporal Links, *Proc. of 9th International Conference on Database and Expert Systems Applications (DEXA'98)*, pp. 176-185 (1998).
- 43) 角谷和俊, 野田玲子, 田中克己: 放送型ハイパーメディアのための時間依存リンク機構, 電子情報通信学会論文誌 D-I, Vol. J82-D-I, No. 1, pp. 291-302 (1999).
- 44) Hoschka, P.: An Introduction to the Synchronized Multimedia Integration Language, *IEEE Multimedia*, Vol. 5, No. 4, pp. 84-88 (1998).
- 45) The XML W3C Recommendation: Synchronized Multimedia Integration Language (SMIL) 1.0 Specification, <http://www.w3.org/TR/REC-smil/>.
- 46) Bækgaard, L. and Mark, L.: Incremental Computation of Time-Varying Query Expressions, *IEEE Trans. Knowledge and Data Engineering*, Vol. 7, No. 4, pp. 583-590 (1995).
- 47) Terry, D., Goldberg, D., Nichols, D. and Oki, B.: Continuous Queries over Append-Only Databases, *Proc. of ACM SIGMOD '92*, pp. 321-330 (1992).
- 48) 野田玲子, 馬強, 角谷和俊, 田中克己: 放送型情報提供システム Mille-feuille における時間依存リンク情報の配信, 情報処理学会研究会報告 99-DBS-116 (1998).
- 49) 野田玲子, 角谷和俊, 田中克己: Eclair: 同期化コンテンツの放送型配信とその版管理方式の提案, 情報処理学会研究会報告 99-DBS-119, pp. 189-194 (1999).
- 50) Lemay, L.: *Official Marimba Guide to Castanet*, Sams.net(松田晃一ほか訳: Marimba オフィシャルガイド Castanet, トッパン, 1997) (1997).
- 51) Goodman, D.: *Official Marimba Guide to Bongo*, Sams.net(石川和也訳: Marimba オフィシャルガイド Bongo, トッパン, 1997) (1997).
- 52) 手島透: Marimba Castanet1.1, 日経インターネットテクノロジー, No. 2, pp. 74-81 (1997).
- 53) 橋本隆子, 白田由香利, 飯沢篤志, 矢野隆志: コンテンツの選択的視聴を可能とする再構成検索方式の検討, 情報処理学会研究会報告 98-DBS-116(2), pp. 1-8 (1998).
- 54) 白田由香利, 飯沢篤志, 橋本隆子: 人気度に依存した放送配信スケジューリング方式のシミュレーション, 情報処理学会研究会報告 99-DBS-119, pp. 7-12 (1999).
- 55) 川口知昭, 土居明弘, 角谷和俊, 田中克己: ScoopCast: ライブ映像ストリームによる動的番組編集システム, アドバンスト・データベース・シンポジウム'98(ADBS'98), 情報処理学会, pp. 167-174 (1998).
- 56) Sumiya, K., Kawaguchi, T., Doi, A., Tanaka, K. and Uehara, K.: ScoopCast: Dynamic Video Production and Delivery from Indexed Live Video Stream, *Proc. of Int'l Computer Science Conference (ICSC'99), IEEE Hong Kong Section*, (to appear) (1999).
- 57) 飯沢篤志, 浅田一茂, 白田由香利: 情報放送のための超大規模分散データベース, 情報処理学会研究会報告 97-DBS-113 (1997).
- 58) 白田由香利, 飯沢篤志, 矢野隆志: 放送型配信機構上での並行制御制御, アドバンスト・データベース・シンポジウム'97(ADBS'97), pp. 15-22 (1997).
- 59) 白田由香利, 飯沢篤志: 情報放送システム上での並行制御方式の考察, 情報処理学会研究会報告 97-DBS-113 (1997).
- 60) Shirota, Y., Iizawa, A., Mano, H. and Yano, T.: The ECHO Method: Concurrency Control Method for a Large-Scale Distributed Database, *Proc. of the 15th International Conference on Data Engineering (ICDE)*, pp. 174-183 (1999).
- 61) Ellerman, C.: CDF: Channel Definition Format, <http://www.w3.org/TR/NOTE-CDF-submit.html>.
- 62) The XML W3C Recommendation: RDF (Resource Description Framework), <http://www.w3.org/Press/1999/RDF-REC>.
- 63) 宮部義幸: 次世代デジタルテレビの全貌 ~放送・通信・コンピュータ業界の挑戦~, 日経ニューメディア別冊, pp. 22-32 (1997).
- 64) 野口英男: 放送のデジタル化とデータベース, 電子情報通信学会夏のデータベースワークショップ'98, <http://nijo.aist-nara.ac.jp/ishikawa/CDROM/contents/talk/talk.html> (1998).
- 65) Ma, Q., Kondo, H., Sumiya, K. and Tanaka, K.: Virtual TV Channel: Filtering, Merging and Presenting Internet Broadcasting Channels, *Proc. of ACM Digital Library Workshop on Organizing Web Space (WOWS)*, (1999).
- 66) 西山揚子, 大和田俊和, 浅田一繁, 飯沢篤志: 距



離の公理に基づいた情報放送フィルタリング, 情報処理学会研究会報告 98-DBS-119, pp. 13-18 (1999).

- 67) TVML: TV Program Making Language, <http://http://www.strl.nhk.or.jp/TVML/>.
- 68) 林正樹: 番組記述言語 TVML を使ったインタラクティブアプリケーション, 第 57 回情報処理学会全国大会論文集 (3), p. 641 (1998).
- 69) Hayashi, M., Ueda, H. and Kurihara, T.: TVML (TV program Making Language) -Automatic TV Program Generation from Text-based Script-, *Proc. of Imagina'99* (<http://www.strl.nhk.or.jp/TVML/Japanese/J09.html>) (1998).
- 70) 近藤宏行, 角谷和俊, 田中克己: 番組メタファーを用いた情報検索結果の提示方式, 情報処理学会研究会報告 99-DBS-119, pp. 419-424 (1999).
- 71) 服部多栄子, 角谷和俊, 灘本明代, 草原真知子, 田中克己: 番組メタファーを用いた情報検索結果の提示方式, 情報処理学会研究会報告 99-DBS-119, pp. 413-418 (1999).

(平成 11 年 5 月 1 日受付)

(平成 11 年 9 月 27 日採録)

(担当編集委員 村田 真)



角谷 和俊 (正会員)

1988 年 神戸大学大学院工学研究科修士課程修了。同年松下電器産業株式会社入社。ソフトウェア開発環境, マルチメディアデータベース, データ放送の研究開発に従事。1998 年 神戸大学大学院自然科学研究科博士後期課程 (情報メディア科学専攻) 修了。1999 年 8 月 神戸大学都市安全研究センター都市情報システム研究分野 (工学部情報知能工学科) 講師。博士 (工学)。ACM, IEEE Computer Society, システム制御情報学会各会員。



宮部 義幸

1983 年大阪大学大学院工学研究科修士課程修了。同年松下電器産業株式会社入社。フォント, 文書処理, ハイパーメディアシステム, データ放送の研究開発に従事。共著「ハイパーメディアとオブジェクトベース」(共立出版), 「次世代デジタルテレビの全貌」(日経 BP)。IEEE, 電子情報通信学会各会員。