

フレームワークをベースにした情報集約・配信システム構築方式

高木浩則⁺, 日浦章英^{*}, 小泉泰則⁺

⁺NTT 東日本 研究開発センタ

^{*}NTT 西日本 研究開発センタ

本稿では、情報集約・配信システムのベースとなるカスタマイズ容易な情報集約・配信フレームワークを提案する。また、このフレームワークを基にした情報集約・配信システム構築方式を提案する。本フレームワークは、情報の集約方法、加工方法、配信方法、並びにそれらを開始するタイミングや条件が容易にカスタマイズ可能であるという特徴を持つ。また、本システム構築方式は、システム構成を動的に構築可能であり、かつシステムの機能を動的に拡張可能であるという特徴を持つ。

本提案手法によって、情報の集約・加工・配信という一連の動作からなる情報集約・配信システムを短期に低コストで構築することができるだけでなく、システムの維持管理も容易に行うことができる。

A Framework based System Construction Method for Information gathering and delivery Systems

Hironori TAKAKI⁺, Akihide Hiura^{*}, Yasunori Koizumi⁺

⁺NTT East Research and Development Center

^{*}NTT West Research and Development Center

We propose a framework and a system construction method for information gathering, processing, and delivery systems. Our framework has following five hotspots that are likely to change at the system construction. These are (1)an information gathering mechanism, (2) information processing mechanism, (3)information delivery mechanism, and (4)the condition or the start timing of information gathering/processing/delivery. A system constructed by our method is easy to modify system configuration and increment functionality of systems dynamically.

Using our method, we reduce the cost and the period of developing information gathering, processing, delivery systems and these constructed systems are easy to maintain.

1. はじめに

現在、個人や企業間の経済活動効率化並びに経費削減を目的とし、情報提供手段や情報取得手段にインターネット技術を利用したサービスやシステムが多数提供されている。例えば、インターネットオークション、インターネット上で投票結果を集約・配信する投票サービス[1]、時間限定型特売商品のようなお買い得商品情報のリアルタイム通知サービス[2]、並びに、為替、株式等の市況状況の通知サービスなどもその一種である。また、Web上のオンライン店舗で顧客から直接受注し、バックエンドの業務システムと連携させる企業も増加している。

これらのサービスを提供するシステムは、人間や

プログラムが介在し、その間で情報が流れつつ、途中で情報が加工されていき、その結果、全体としてのサービスが提供される形の情報流通加工型システムであるといえる。

ワークフロー製品やオークション専用ライブラリなど一部には基盤となるソフトウェアが存在するものの、広範囲の情報流通加工型システムに対して適用可能なソフトウェアは存在しなかった。また従来これらのサービスを提供するためには、ソフトウェア開発に対してかなりの投資が必要であった。今後益々増加すると考えられる多数の中/小規模な企業や個人の要求にも容易にカスタマイズして安価にシステムを構築できる技術が必要である。

本稿では、情報流通加工型システムの効率的構築

を目的とし、広範囲の情報流通加工型システム構築時のベースとして利用可能な情報集約・配信フレームワークを提案する。またそれを基にしたシステム構築方式を提案する。システム構築の際に、本フレームワークを組み込むことにより、情報流通加工型システムを容易に低コストで構築することができる。

以下、2章では検討に当たり考慮した要求条件について述べる。3章ではシステムのアーキテクチャを明らかにする。4章では情報流通加工型システムのベースとなる情報集約・配信フレームワークについて述べる。5章ではシステム構築手順を説明し、請求書発行業務を例にあげる。6章では有効性、適用上の課題、関連技術との比較について述べる。

2. 要求条件

ここでは、本システムで対象とする集配信対象を明確にする。また、方式検討に当たり、考慮した要求条件について述べる。

本システムで扱う集配信対象は、これまでのプッシュ型情報配信技術[3]同様、テキストやHTMLを扱うメッセージ型、並びにファイルを扱うバルク型とし、映像や音声を扱うストリーム型は対象外とする。

方式検討に当たり、以下の3つの要件を考慮した。

- (1) 協調動作する複数サーバによるサービス提供が可能
- (2) 同一サーバ上で複数サービスの同時提供が可能
- (3) 構成管理の容易化

上記3つの要件を考慮した理由を示す。

本稿で対象とする情報流通加工型システムは、一つのサーバ上で閉じた形でサービスを提供するものだけではない。例えば分散投票システムなどのように複数サーバが協調動作してサービスを提供するものも対象としたい。また多数の中/小規模の企業や個人の要求に答えるため、一つのサーバ上で多数のサービスを同時に提供できる構成としたい。さらに、一つの構成管理用サーバから各サーバを設定することにより、協調動作するサービス提供ネットワークを容易に構築したい。構築後も、利用者数増減や負荷に応じてシステムのサーバ構成を変えたり、期間限定でのサービスなど新たなサービスを動的に容易に追加することが可能な設計としたい。

3. システムのアーキテクチャ

3.1 システム構成要素

本システムで対象とする情報流通加工型システムは、一つ以上の複数サーバが協調動作してサービス

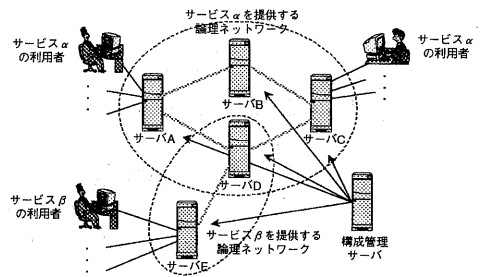


図1 システムのサーバ構成

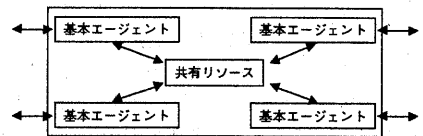


図2 サービスエレメント内の構成

を提供する(図1)。協調動作するプログラムを構成管理サーバから各サーバに送り込むことによって、サービスを提供する論理ネットワークを構築する。

ここでは、サーバ上で一つのサービスを提供する概念的な単位をサービスエレメントと呼ぶ。このサービスエレメントは、構成管理サーバから各サーバに送り込まれる基本単位である。図1の例では、サーバD上には送り込まれたサービスα、βの二つのサービスエレメントが同時に存在している。

サービスエレメント内の構成を図2に示す。サービスエレメントは、複数のオブジェクトからなるソフトウェアモジュールを意味する。図2中の基本エージェントとは、情報の集約、加工、配信という一連の動作を行う基本単位である。共有リソースは基本エージェント間で共有される情報の一時保存機能やタイマ機能を提供する。結果として、一つ以上の複数の基本エージェントと一つの共有リソースで一つのサービスエレメントが構成される。各サーバ上のサービスエレメントが連携して協調動作することにより、一つのサービスが提供される。

3.2 システムアーキテクチャ

一つのサーバ内の、概念的な機能階層図を図3(次頁)に示す。各レイヤの役割を以下に示す。

(1) エージェントレイヤ

情報集約、加工、配信という一連の動作を行うエージェントが存在するレイヤである。

(2) サービスエレメントレイヤ

あるサービスを提供するサービスエレメントが存在するレイヤである。図3に示すように、一つのサービスエレメントは、一つ以上のエージェントを管理する。また図のように、投票サービスのサービスエレメ

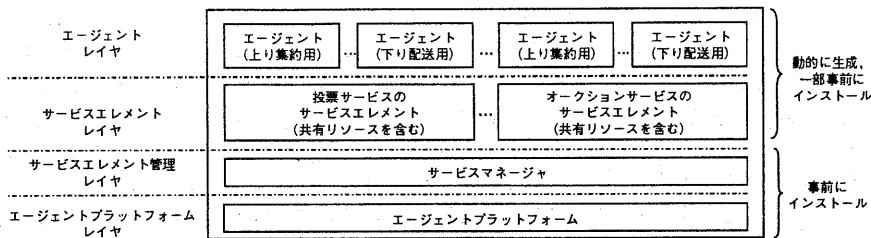


図3 サーバのアーキテクチャ

ントとオークションサービスのサービスエレメントなど、複数のサービスエレメントが同一サーバ上に同時に存在できる。

(3) サービスエレメント管理レイヤ
一つのサーバ上に存在する各サービスエレメントを管理するレイヤである。

(4) エージェントプラットフォームレイヤ
構成管理サーバからのエージェントの移動を実現するためのプラットフォームとなるレイヤである。例えば、Voyager(Objectspace) [4] や Aglets (IBM) [5] などのエージェントフレームワークと呼ばれるものがこれに該当する。

4. 情報集約・配信フレームワーク

ここでは、サービスエレメントの雛型となる情報集約・配信フレームワークの設計目標、構成要素、クラス構成について述べる。

4.1 設計目標

情報流通加工型システムの特徴として以下の2点が挙げられる。

[特徴1] 情報の伝送, 情報の集約・配送, 情報の蓄積や検索, 情報の変換処理が存在する。

[特徴2] 処理形態はサービスにより様々である。

そこでフレームワークの設計に当たり、以下の3つの要件を考慮した。

- (1) 情報集約・配信形態として表1に示すような各種方式に対応する。
- (2) 情報伝送手段として、HTTP、FTP、Mailなどの手段に対応する。
- (3) 情報集約、加工、配信を開始する条件や動作タイミングを容易にカスタマイズ可能とする。

4.2 構成要素

上記の設計目標に加えて、理解容易性も考慮した結果、図4に示すように、大きく分けて基本エージェント部と共有リソース部の二つの部分から情報集約・配信フレームワーク構成した。基本エージェント部は、集約処理部、加工処理部、配信処理部の3つの

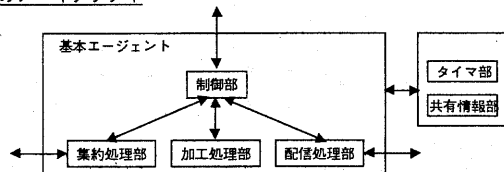


図4 情報集約・配信フレームワークの機能ブロック図

表1 情報集約・配信形態

| 動作主体 | 情報の受入側 | | 情報の送出側 | |
|--------|---------|--|---------------------------|--|
| | 受動的 | 能動的 (半能動的を含む) | 受動的 | 能動的 (半能動的を含む) |
| 動作概要 | 情報の受信待ち | 自発的に送出依頼を送る (内部/外部イベント、 タイムイベントによる情報送 出への送出依頼を含む) | 送出依頼受理 後、依頼元 に情報を送る | 自発的に情報を送る (内部/外部イベント、 タイムイベントによる 情報送出を含む) |
| 動作開始契機 | 情報受信 | 自動 (半自動) | 送出依頼受信 | 自動 (半自動) |
| 通信形態 | 1対1通信 | 1対1通信 N対1通信 | 1対1通信 | 1対1通信 1対N通信 |
| サービス形態 | Wait型 | Periodic/Aperiodic Push型 | Request/ Response型 | Periodic/Aperiodic Push型 |

アクション規定部、並びにそれら間の連携・タイミング制御を司る制御部、の4つの要素から構成される。共有リソース部は、タイマ部と共有情報部の2つの要素から構成される。

各機能ブロックの役割や機能について述べる。

(1) 制御部

制御部は、一つの基本エージェント内の、集約処理部、加工処理部、配信処理部の各アクションを開始するタイミングや条件判断を行い、処理全体の流れや動作を制御する役割を負う。また、他の基本エージェントからのアクション開始イベントを受け取る窓口の役割も果たす。

(2) 集約処理部、並びに配信処理部

集約処理部、配信処理部は情報の集約、配信、並びにその通信を司る部分である。それぞれ他の基本エージェントとの情報集約方法、情報配信方法に関して全責任を持つ。集約処理部、配信処理部共に、同一サーバ内並びに他サーバ内の複数エージェントとの通信が可能である。集約処理部は受信した情報の保存や加工前の前処理を行う。前処理を施した情報は加工処理部に渡される。配信処理部は配信待ち情報の保存や配信後の後処理を行う。

(3) 加工処理部

加工処理部は、情報の加工処理、並びに加工した情報の送り先決定に関する全責任を持つ。また、加工処

理部は集約処理部から送られてきた加工待ち情報の保存も行う。情報を加工する際には、単なる情報形式の加工並びに情報の分割・統合だけでなく、条件によって情報の配信先を変えたり、同一情報を複数の基本エージェントに送ったり、同じ情報を異なる形式の情報に加工して別々の基本エージェントに送る等が可能である。配信先を設定した情報は配信処理部に渡される。

(4) タイマ部, 共有情報部

タイマ部は、時間に関する情報、並びにタイマスケジューリングに関する全情報を持ち、基本エージェントからあらかじめ指示されたスケジュールに基づいて、基本エージェントの特定のメソッドを呼び出す役割を持つ。共有情報部は、基本エージェント間で共有される情報の一時保管を行う。

4.3 フレームワークのクラス構成

図5に情報集約・配信フレームワークのクラス構成図を示す。以下の7つのクラスを定義した。

- (a) サービスエレメント部: ServiceElement クラス
- (b) タイマ部: TimerManager クラス
- (c) 共有情報部: DataManager クラス
- (d) 制御部: AgentTemplate クラス
- (e) 集約処理部: ReceiveTemplate クラス
- (f) 加工処理部: ProcessTemplate クラス
- (g) 配信処理部: SendTemplate クラス

4.1節で述べた3つの要件に対して、以下の3つの方針によって対処した。

(1) 表1に示す各種情報集約・配信サービス形態に対応するために、Wait型、Periodic/Aperiodic Pull型、Request/Response型、Periodic/Aperiodic Push型に対応するメソッドとして、receiveByOne

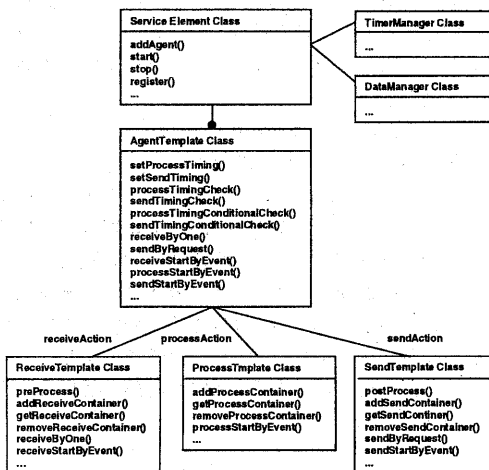


図5 フレームワークのクラス図

(), receiveStartByEvent(), sendByRequest(), sendStartByEvent()を用意した。すなわち、集約処理として、(a)他の基本エージェントから送られてくる情報を一つ一つ受け取るメソッド receiveByOne(), (b)内部的な条件やタイマあるいは他の基本エージェントからの指示をトリガーにして能動的に情報取得依頼を出し情報を受け取るメソッド receiveStartByEvent()の2種類を定義した。配信処理として、(a)他の基本エージェントからの情報配信依頼を受けてそのエージェントに対して情報を配信するメソッド sendByRequest(), (b)内部的な条件やタイマあるいは他の基本エージェントからの指示をトリガーにして情報を配信するメソッド sendStartByEvent()の2種類を定義した。

(2) 情報伝達手段としてHTTP, FTP, Mailなどの手段に対応するために、それらをライブラリとして用意した。

(3) 情報集約, 加工, 配信を開始する条件や動作タイミングを容易にカスタマイズ可能とするために、動作タイミング指定として、即実行開始, 条件判断により実行開始, イベント受理により実行開始, の3つを指定できるようにし、条件判断の場合の開始条件判断メソッドをフックメソッド[6]として用意した。メソッド processTimingCheck()は、集約処理終了後に呼び出される処理であり、動作タイミング指定に基づき加工処理を開始するか否かの判断を行う。processTimingConditionalCheck()は条件判断により開始可否を決定する場合の判断メソッドである。メソッド sendTimingCheck()は、加工処理終了後に呼び出される処理であり、動作タイミング指定に基づき配信処理を開始するか否かの判断を行う。sendTimingConditionalCheck()は条件判断により開始可否を決定する場合の判断メソッドである。この動作タイミング指定を変更したり、開始条件判断メソッドを必要に応じてオーバーライドすることによって、情報集約・加工・配信の各処理の流れを制御できるようにした。また、情報を加工するメソッド processStartByEvent()を定義した。この処理の中で、情報形式の加工、情報の分割・統合などの情報の変換処理、並びに加工情報の送り先の決定などを行う。必要であれば、他のデータベースシステムに対する検索などもこのメソッド内で行う。この処理自身をシステムに合わせてオーバーライドすることでカスタマイズを行う。

この他、受け取った情報の前処理を行うメソッド preProcess()や配信後の後処理を行う postProcess()を定義した。これらをオーバーライドすることでカスタマイズを行う。

5. システム構築例

システムの構築手順について述べる。また本フレームワークを基にしたシステム分析例について述べる。

5.1 システム構築手順

システム構築手順について述べる。

(1) 要件分析

業務上の情報の流れに着目し要件を分析する。

(2) 基本エージェントの役割の明確化

情報の集約,加工,配信といった一連の流れに着目し,サービスエレメントを構成する基本エージェントを決定する。また基本エージェント間でやり取りする情報を決定する。一連の流れを考慮する際に,情報集約後,保存する場合や,検索処理後,配信する場合等も一つの基本エージェントとして割り当てる。

(3) カスタマイズ方法の決定

基本エージェントの変可部(ホットスポット)に着目し,どのようにカスタマイズするか決定する。

(4) サーバ構成の決定

利用可能なサーバ数を考慮した上で,サービスエレメントの役割やサーバの業務上の役割を考慮してサーバ構成を決定する。この際,負荷分散などのシステム要件も考慮してサーバ構成を決定する。

(5) サービスエレメントの移動

構成管理サーバ上で,各サーバ向けに,ホットスポットをカスタマイズした基本エージェントを含むサービスエレメントを生成し,目的のサーバに移動させる。移動したサービスエレメントは,目的のサーバ上のネーミングサービスに自分自身を登録する。すべてのサーバ上で登録が終了すると,サーバの論理ネットワークが完成する。

(6) サービスエレメントの起動

各サーバに対してサービスエレメントの実行を開始する指示を送る。各サーバ上のサービスマネージャが指示を受け取ると,ネームサービスからサービスエレメントをロックアップして各エージェントを起動し必要な処理を実行する。

5.2 請求書発行業務の分析例

利用者がコンテンツを利用することに課金され,一ヵ月ごとに利用者に対してまとめて請求書を発行するという単純な請求書発行業務を考える。システムはコンテンツを管理し利用状況を報告するコンテンツ管理サブシステムと,請求書発行業務を行うサブシステムからなる。ここでは請求書発行サブシステムを構築対象とする。このサブシステムでは,顧客は顧客識別子によってそれぞれ請求書発行のタイミ

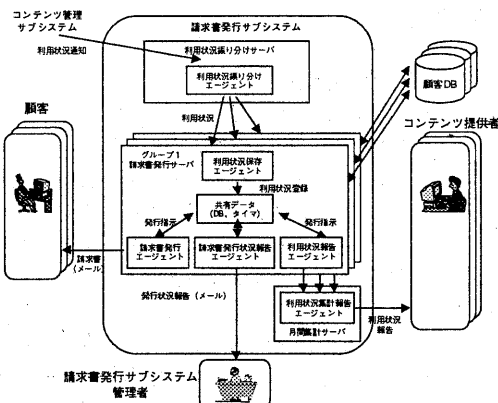


図5 請求書発行サブシステム構成、並びにサービスエレメント内の構成

表2 各エージェントの処理内容

| エージェント名 | 典型的処理 | 加工処理開始条件 | 加工処理 | 配信処理開始条件 | 配信処理 |
|-----------------|-------|---------------|----------------|----------|--------------|
| 利用状況振り分けエージェント | Wait型 | 即実行開始 | 振り分け処理 | 即実行開始 | Push型(HT TP) |
| 利用状況保存エージェント | Wait型 | 即実行開始 | 保存 | なし | なし |
| 請求書発行エージェント | なし | Event受渡後即実行開始 | 検索実行,請求書作成 | 即実行開始 | Push型(Mail) |
| 請求書発行状況報告エージェント | なし | Event受渡後即実行開始 | 検索実行,請求書発行状況作成 | 即実行開始 | Push型(Mail) |
| 利用状況報告エージェント | なし | Event受渡後即実行開始 | 検索実行,利用状況報告作成 | 即実行開始 | Push型(HT TP) |
| 利用状況集計報告エージェント | Wait型 | 実行開始後(3件待ち受け) | 利用状況集計報告作成 | 即実行開始 | Push型(Mail) |

ングが異なる3つのグループに分けられる。顧客に対して請求書はメールで発行し,請求書発行終了後,発行状況を請求書発行サブシステムの管理者へメールで通知する。またコンテンツ提供者には,3グループひとまとめにした一ヵ月ごとのコンテンツの利用状況をメールで通知する。

検討する請求書発行業務においては,簡略化のため以下の仮定をおく。

(1)コンテンツの利用状況は,コンテンツ管理サブシステムから請求書発行サブシステムへ自動的に通知される。

(2)顧客情報を管理するDBサーバは独立して存在し,請求書発行サブシステムからアクセス可能である。

(3)十分な数のサーバが利用できる。

請求書発行サブシステムでは,負荷分散を考慮し,利用状況振り分けサーバ,3つのグループごとの請求書発行サーバ,3つの請求書発行サーバから一ヵ月間のコンテンツの利用状況を集めてまとめる月間集計サーバから構成することとした。請求書発行サブシステムのシステム構成,並びに各サーバの基本エージェントの構成を図5に,各エージェントの処理内容を表2に示す。各エージェントのどちらが主となって情報の授受を行うかは容易に変更できる。

6. 考察

6.1 有効性の評価

本提案方式の利点について述べる。

(1) 生産性の向上, 設計品質の向上

本手法では, 5. 2節の例のように情報集約・配信フレームワークをカスタマイズして組み合わせることにより, 目的とする情報流通加工型システムが容易に構築できる。これにより生産性向上が期待できる。また構築したシステムは設計を再利用することになり, 設計品質の向上も期待できる。

(2) システムの分析・設計作業の効率化

情報集約・配信フレームワーク自体を単純な機能要素として捉えることができる。情報流通加工型システムを分析・設計する際に, この機能要素に着目することにより, システムの分析・設計作業を効率的に進めることができる。

6.2 適用上の課題

実際にこの本方式を適用する際には, セキュリティの機能, 並びにシステム全体の状態を保存する機能が必要である。現在, 本システムではこれらを実装していない。サーバ間のアクセスコントロールや移動するサービスエレメントの認証(シングル・ホップの認証)を実装する必要がある。また, システム全体を停止・再起動する際に, システムの状態を保存・再構築する仕組みを実装する必要がある。

6.3 関連技術との比較

類似技術として下記の3つの技術と比較を行う。

(1) PointCast Network (米Point Cast社)やCastanet (米marimba社)に代表されるプッシュ型技術

(2) アクティブネットワーク[7][8]

(3) OpenWebServer[9]

プッシュ型技術は, エンドユーザが特別な操作をしなくても自動的に情報を届けることができる。我々の方式は, プッシュ型, プル型を統合しており, また情報を届けるだけでなく情報が流れて行く途中で加工できるという点で, 単なるプッシュ型技術とは異なるサービスが実現できる。

情報が流れていく途中で加工できるという点では, アクティブネットワークと類似している。アクティブネットワークは, 中間のルータがアプリケーション層までの処理を行うネットワークであり, ユーザやネットワークの運用者がその処理のプログラムをルータに動的に投入することで, ネットワークをプログラムすることができる[7]。しかし, エン

ドユーザが扱う情報を考慮した場合, データベース検索などと組み合わせられる場合も多い。また, エンドユーザの情報がどこのルータを通過するかわからない。そのため, アプリケーションレベルの情報は, アクティブネットワークのようにルータで情報加工するよりも, より上位のサーバネットワークレベルで情報加工を施した方が適していると考えられる。

OpenWebServerは, Webサーバの機能を動的に変更・拡張できるフレームワークである。本提案方式もWebサーバの機能を動的に変更・拡張できるという点では類似している。OpenWebServerは, アプリケーションやドメインに特化しないシステムレベルフレームワークであるのに対して, 本提案方式は, 情報の集配信並びに情報の加工に特化したアプリケーションレベルのフレームワークである点が異なる。

7. おわりに

本稿では, 情報流通加工型システムの基本構成単位となるカスタマイズ容易な情報集約・配信フレームワークを提案した。また, このフレームワークを基にしたシステム構築方式を提案した。本フレームワークでは, 各種情報集約・配信形態を事前に規定し, かつ加工方法自体や集約・加工・配信処理の間の連携タイミングをフックメソッドとして規定している。本システム構築方式では, このフレームワークを基本単位とし, 構成管理サーバから各サーバに動的に移動させることによって論理的なサーバのネットワークを構成する。そのためシステムの構成が動的に構築可能であり, かつシステムの機能が動的に拡張可能であるという特徴を持つ。

本提案手法によって, 情報の集約・加工・配信という一連の動作からなるシステムを短期, 高品質, 低コストで構築することができるだけでなく, システムの維持管理も容易に行うことができる。

【参考文献】

- [1] 小泉, 吉田, 長岡, インターネットにおけるフィードバック型情報集約方式の構想, 情報処理学会研究報告, マルチメディア通信と分散処理 (97-DPS-84) 84-6, pp. 31-36, Vol. 97, No. 87, 1997.
- [2] <http://www.jnews.com/business/digest/199903.html#19990315>
- [3] 高橋, プッシュ型情報配信サービス技術の展望, 情報処理学会学会誌, Vol. 39, No. 11, pp. 1124-1131, Nov., 1998.
- [4] <http://www.objectspace.com/products/vgrOverview.htm>
- [5] <http://www.trl.ibm.com.jp/aglets/>
- [6] デザインパターンプログラミング, W. プリー著, 佐藤・金澤訳, 1996, トッパン.
- [7] 小花, 杉山, アクティブネットワーク: ネットワークをオーダメードに使う, 情報処理学会学会誌, Vol. 40, No. 6, pp. 590-595 June, 1999.
- [8] Konstantinos Psounis, Active Networks: Applications, Security, Safety, and Architectures, IEEE Communications Surveys, <http://www.comsoc.org/pubs/surveys,FirstQuarter,1999>.
- [9] Junichi Suzuki, Yoshikazu Yamamoto, OpenWebServer: An Adaptive Web Server Using Software Patterns, IEEE Communications Magazine, Apr., 1999.