

## エージェントに対応したパーソナルWWWサーバ

毛利 隆夫 牛嶋 悟 高田 裕志

(株) 富士通研究所 ネットメディア研究センター  
〒 814 福岡市早良区百道浜 2-2-1

Email: {tmohri, ushijima, yuji}@flab.fujitsu.co.jp

IntelliWeb はエージェントとの連携機能を持った個人専用の WWW サーバである。IntelliWeb は CGI 相当の付加機能を常駐しているエージェントとの通信により実現しており、CGI で問題となる負荷の集中が回避できるほか、複数のエージェントの協調による高度なサービスも提供できる。IntelliWeb はエージェント管理機能を持ち、エージェントの自動起動や自動更新が可能である。IntelliWeb では WWW サーバを個人専用としてブラウザと密接に連携させるため、サーバからブラウザへの通知機能を用いた対話的なシステムが構築できる。

## A Personal WWW Server with Agent Support

Takao Mohri Satoru Ushijima Yuji Takada

Netmedia Laboratory, FUJITSU LABORATORIES LTD.

2-2-1 Momochihama, Sawara-ku, Fukuoka 814, Japan

Email: {tmohri, ushijima, yuji}@flab.fujitsu.co.jp

We have developed an agent oriented WWW server, IntelliWeb. In addition to usual functions of WWW servers, IntelliWeb relays services from agents to WWW browsers. Agents are realized as resident processes, and answer requests from users without invocation cost. They can offer advanced services by communicating to each other. IntelliWeb manages these agents by invoking, stopping, and automatic update. IntelliWeb is used personally to make sure of privacy of personal data, and supports notification from agents to WWW browsers, which enable flexible interaction.

### 1 はじめに

WWW(World Wide Web) が普及するにつれ、一般のアプリケーションも WWW サーバと連携し、GUI として WWW ブラウザを使うことが多くなっている。WWW ブラウザは多くの人が日常的に使っているツールであり、WWW ブラウザをユーザインタフェースとすることによって、WWW ブラウザだけから、つまり、オペレーティングシステムを意識することなく、しかも一般のホームページを操作するのと同じような方法で、アプリケーションを操作することが可能となる。このような WWW ブラウザをユーザインタフェースとして利用するシステムは、基本的には WWW サーバを中心としてサービスを提供しているが、サーバの機能やシステム構成に起因する問題点がいくつか存在する。

本稿では、まず従来の WWW を用いたシステムの問題点を整理し、それらを解決するために我々が開発した WWW サーバ IntelliWeb の特長や機能について述べる。

## 2 従来の WWW を用いたシステムの問題点

従来の WWW サーバを中心としたシステムの問題点としては、以下のような点が挙げられる。

**サーバの機能拡張の難しさ** 従来の WWW サーバの機能を拡張する方法としては、CGI(Common Gateway Interface) プログラムによる方法が広く用いられている。しかし、この方法では CGI プログラムをあらかじめサーバにインストールしておく必要がある。また、ブラウザからのリクエストのたびに、CGI プログラムが、WWW サーバと同じホスト上で起動されるために、サーバの動作しているホストの負荷が大きくなってしまふと同時に、複数のホストへの負荷分散が難しい。

CGI の他には、NSAPI[5] などを使って WWW サーバを機能拡張する方法があるが、これらは拡張機能を提供するライブラリを WWW サーバにリンクする必要がある。また、Jeeves[3] のように WWW サーバを Java で構築し、Java のクラスをダウンロードして機能拡張する方法もある。しかし、基本的に一方のダウンロードをサポートするのみで、逆方向のアップロードを行ったり、通信によって任意のプロセスから機能を提供するといったことは容易ではない。

**データのプライバシー確保の難しさ** 現在、複数のユーザのホームページが、同一の WWW サーバ上で開設されている場合が多い。通常、ユーザのデータにはアクセス制限をかけることができるが、サーバマシンの特権ユーザが悪意を持てば、これらのデータを盗み見ることは可能である。このように、他人と共有して使う WWW サーバでは、ユーザのプライバシーの確保に限界がある。

また、電子メールシステムのようなプライベートな情報を提供するシステムを、WWW サーバとの連携で WWW ブラウザからアクセスできるようにした場合、たとえシステム自体はアクセス制限を設けていても、WWW サーバを他人と共有していると、WWW サーバが行うキャッシュや通信経路の割り当てに不安を抱くことになる。

**情報の流れの一方方向性** WWW サーバと WWW ブラウザの間では、通常 HTTP により通信が行われるが、この場合にはブラウザからサーバに要求を発行することはできても、サーバ側からブラウザ側に通知や割り込みやを行うことはできない。このような情報の流れの一方方向性は、対話型のシステムを構築する際に大きな障害となる。

**提供される情報が固定的** 現状では、WWW サーバから提供される情報の更新は、サーバ上の HTML ファイルを書き換えるなどの複雑な操作を、ユーザが直接行う必要がある。しかし、ユーザのスケジュールや、ユーザ自身の現在位置などのように、頻繁に更新され、なおかつ他人に発信したい情報も存在する。このようなデータの更新を受け付ける機能は、現在の WWW サーバには備わっておらず、動的に変化する情報の発信を難しくしている。

## 3 IntelliWeb

我々は上記の問題点を解決するために、WWW サーバ IntelliWeb [6] を開発した。IntelliWeb は HTTP/1.0 [2] という標準的な規格に準拠した WWW サーバであるが、それに加えて「エージェントとの連携」と「パーソナリ化」という二つの工夫を行っている(図1)。

IntelliWeb はプログラミング言語 April [1, 7] で実現されている。April は、移動コード、独自のマルチプロセス、グローバルネットワークに対応したプロセス管理、プロセス間の非同期通信などをサポートするネットワーク指向のプログラミング言語である。IntelliWeb では、April の機能を生かしてエージェントに対応し、なおかつパーソナルな WWW サーバとして、ブラウザと密接に連携した運用を行うことで、上記の問題点を解決している。さらに IntelliWeb は、連携するエージェントの起動、停止、更新といったエージェント管理機能も提供している。この機能によって、IntelliWeb から外部エージェントとの連携を動的に

制御することができる。なお、本稿でいう「エージェント」とは、単独でひとつの独立したサービスを行うシステムであり、かつ他のシステムとの連携が行えるものを指している。

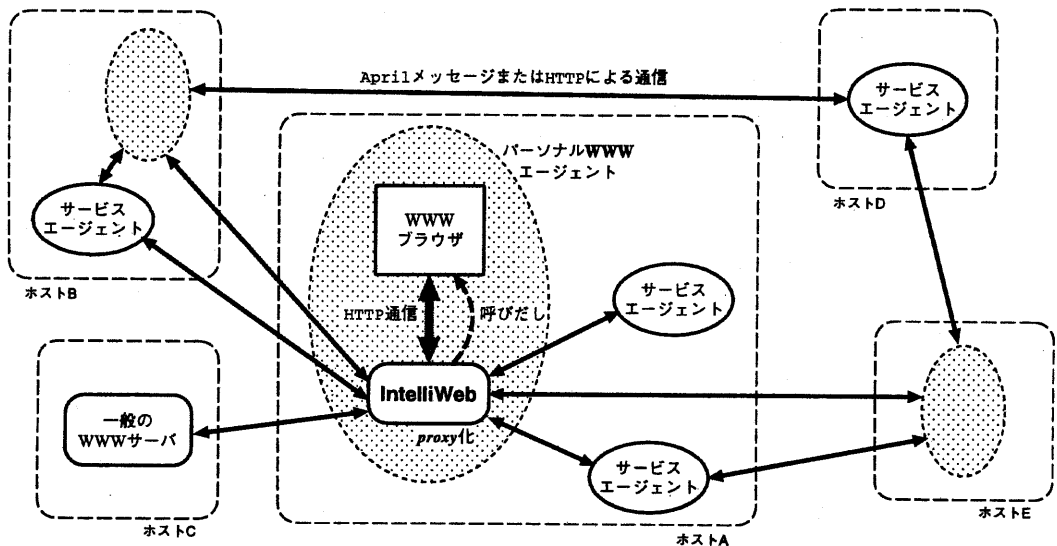


図 1: IntelliWeb でのパーソナリ化とエージェント連携

### 3.1 エージェントとの連携

IntelliWeb は April で構築されたシステムと通信する機能を持っている。この機能によって、April で構築された外部エージェント（ここではサービスエージェントと呼ぶ）とメッセージ送受信によって連携することが可能である（図 1 参照）。

IntelliWeb はブラウザからのリクエストである URL と、それに対するサービスを提供するサービスエージェントとの関連付けを行う。次のように URL のファイル名の部分が /april で始まっている場合には、その次の agent で示される名前のエージェントがサービスを行う。

```
http://host/april/agent? key1=value1&...
```

各サービスエージェントは、「ローカルプロセス名@ホスト名」という形式の名前を持っている。「ローカルプロセス名」は、各ホスト内で登録する名前であり、ホスト名はそのエージェントの動作するホスト名である。IntelliWeb とエージェントが同一ホストで動作している場合には「@」以下は省略できる。エージェント名の後ろには、エージェントに与えるパラメタのキーと値のセットを記述する。例えば次の URL は、ホスト localhost の IntelliWeb を経由して、remotehost 上の calendar という名前のエージェントに、パラメタ month を 12 としてアクセスするという意味である。

```
http://localhost/april/calendar@remotehost?month=12
```

IntelliWeb はこの連携機能を使い、常駐型の April プロセスとの通信によってサービスを提供する。これにより、リクエストのたびにプロセスが起動されるという CGI プログラムの問題が解消される<sup>1</sup>。また、April でのプロセス間通信は、通信相手が同一ホスト上であるか、異なるホストに存在するかをほとんど意

<sup>1</sup> エージェント内では、連続して要求を受け取るために、要求毎に内部的なプロセスを生成している。しかし、内部プロセスの生成は通常のプロセス生成に比べて遥かにコストが小さい。

識する必要がないため、IntelliWeb はローカルなエージェントだけでなく、リモートなエージェントとも透過的に連携することができる。このため、IntelliWeb では、従来の CGI では WWW サーバの動作するホストに集中していた負荷を、複数のホスト間に分散できる。別の見方をすれば、IntelliWeb を用いることにより、ネットワーク上に分散した様々なエージェントによるサービスを利用できるようになる。

サービスエージェントが常駐型のプロセスである利点は他にもある。たとえば、サービスエージェントは IntelliWeb を通して、または独自に、外部からの要求を受け付けられる。そのため、動的に変化する情報の更新を受け付けて、提供するサービスに反映させることができる。また、サービスエージェント間で情報の交換が可能になる。我々はこの特長を利用して、グループスケジュールの調整を実現している [9]。例えばあるユーザが自分のスケジュールを管理するサービスエージェントに対して、他のユーザとの打ち合せの設定を要請した場合、そのユーザのサービスエージェントは、他のユーザのサービスエージェントと対話して、空き時間を見つけて打ち合せを設定できる。サービスエージェント間の通信には、HTTP に加えて、April メッセージも用いられる。

### 3.2 パーソナル化

WWW サーバをパーソナル化して運用する点も、IntelliWeb の大きな特長のひとつである。パーソナル化とは、具体的には次の 3 つの点からなる。

- 個人専用の WWW サーバとして運用する。

IntelliWeb は多人数でも共用できる機能を持つが、あえてユーザー一人にサーバをひとつ立ち上げる。IntelliWeb に蓄えられる HTML ファイルなどの情報は、そのユーザのものだけになり、サーバの管理者と情報提供者が同じであるため、管理者が情報を盗み見る問題は起こり得ない。ユーザは、自分で IntelliWeb の設定を管理することによって、ユーザ自身の責任でプライバシーを守ることができる。

- IntelliWeb を proxy サーバとして動作させる。

WWW ブラウザと密接に連携を行わせるため、IntelliWeb を proxy サーバとし、ブラウザから外部へのすべてのアクセスを IntelliWeb 経由にする。この構成では、ユーザが頻繁に参照しているページの記録などが簡単に取れるため、ユーザの興味のある分野を分析したり、定期的にアクセスするページをあらかじめ取得しておくなどの付加機能が実現できる。

- IntelliWeb とブラウザとを一対一対応させる。

WWW サーバと WWW ブラウザは、特定の条件を満たしていれば、サーバ側からブラウザ側に、つまり通常とは逆方向に、メッセージを送ることができる。例えば、OS が MS-Windows 95 である場合に、IntelliWeb と WWW ブラウザが同一マシン上で動作していれば、IntelliWeb からブラウザにプロセス間通信を行い、特定の URL を表示させるなどの命令を行うことができる。また、UNIX (X windows) 上でも同様な機能が実現されている。これらの機能はブラウザ依存であるが、主要な WWW ブラウザではサポートされており (Netscape Navigator については [4] を参照)、IntelliWeb ではこの機能を用いてサーバからブラウザへの通知が実現されている。

このように WWW サーバをパーソナル化し、ブラウザと密接に連携させれば、プライバシーを確保しつつ、ユーザとシステムとの間で、双方向の柔軟な対話が可能である。例えば、メールの到着をブラウザに通知することや、時間の掛かる検索などを依頼しておき、処理が終了時点でブラウザに結果を表示することなどができる。また、個人での利用を前提としたことにより、エージェントのセキュリティやリソース管理の問題を、より簡単に扱えるようになる。

### 3.3 例：メールエージェントとの連携

IntelliWeb のパーソナル化とエージェント連携の効果を、April で構築された MH 用メールエージェント apmh (図 2) との連携を例として説明する。apmh は、電子メールハンドラである MH のコマンドを実行す

るエージェントで、apmh を IntelliWeb と連携させることで、任意の WWW ブラウザから電子メールを読み書きすることが可能になる。その際、メール操作はすべて apmh が起動されているホスト上で行われるため、メールがさまざまなホストに分散してしまうことはない。

apmh を利用するには、通常メールを操作するマシン（ここでは mailhost とする）で apmh を常駐させておく。IntelliWeb との連携で apmh を利用するには、URL

```
http://intelliweb/april/mh@mailhost
```

によってアクセスする。

apmh は、ユーザと IntelliWeb を指定することによって、アクセスを制御している。したがって、パーソナル化された IntelliWeb からのアクセスしか受け付けないように設定することによって、IntelliWeb のアクセス制御と合わせた、より強いプライバシーの保護を行うことができる。また、IntelliWeb に proxy を制御させることによって、キャッシュや通信経路によってプライバシーが暴かれることを防ぐことができる。また apmh は、新着メールを検知した場合には、IntelliWeb を通してブラウザ上にその旨を表示させることができる。

## 4 エージェント管理機能

IntelliWeb は、外部のサービスエージェントとの連携機能の他に、サービスエージェントを検知、起動、停止、更新する機能を持っており、エージェント名とエージェントの実行に必要なリソースが置かれている場所も管理している。これらの機能を組み合わせにより、エージェントの自動更新が実現されている。先の例で使った URL

```
http://localhost/april/calendar@remotehost?month=12
```

がリクエストとして送られてきた時に、IntelliWeb は次のような動作をする (図 3)。

- (1) calendar のコードの配布元の場所を管理情報から取得し、最新のコードのバージョンを得る。
- (2) calendar が動作していない場合には、最新のコードを取得し、ローカルでエージェントを起動する。
- (2') calendar が動作している場合には、コードのバージョンを比較し、古いコードで動作している場合にはエージェントを一旦終了させ、最新のコードでエージェントを再起動する。
- (3) ユーザからの要求を calendar エージェントに送信する。
- (4) calendar エージェントからの返答をブラウザに返信する。

なお、エージェントの終了時には、必要に応じてユーザに確認を求めることができる。また終了させられるエージェントは、次の起動に必要な自分の内部状態を外部に出力できる必要がある。

これらのエージェント管理機能は、ユーザだけではなくサービスエージェントにも公開されており、エージェントを起動・更新するような操作を自動化することができる。例えば、定期的にエージェントの配布元のコードのバージョンをチェックして、自動的に最新のバージョンに更新しておくことができる。逆にエージェントの配布元からのバージョンアップを通知を受け取ると、コードの更新を行うような設定も可能である。また、ローカルで動作しているエージェントと同じ種類のエージェントをリモートの IntelliWeb 上で起動し、古いエージェントから新しいエージェントにデータをコピーし、その後で古いエージェントを停止することで、エージェントの移動を実現可能である。



図 2: メールエージェント

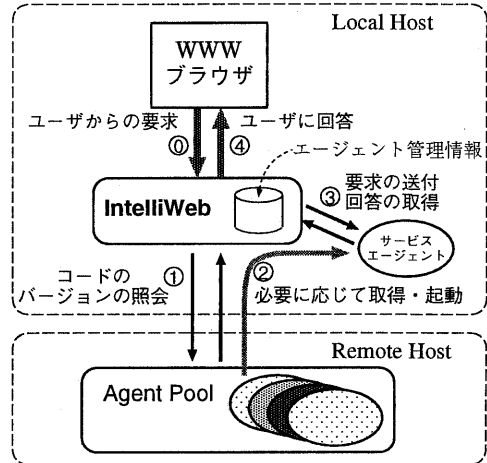


図 3: 移動エージェントの仕組み

## 5 まとめ

今回我々は、エージェント対応のパーソナル WWW サーバ IntelliWeb を開発した。エージェントを用いた WWW サーバの機能拡張は、負荷が小さく、また負荷のホスト間の分散が可能である。エージェントは常駐型のプロセスであるため、情報の更新を常時受け付け可能であり、エージェント間の連携による高度なサービスを提供できる。IntelliWeb は個人専用の WWW サーバであり、個人データのプライバシーが確保できると同時に、サーバとブラウザと一対一対応させることにより、サーバからブラウザへの通知が可能になり、対話的なシステムが構築しやすくなった。

IntelliWeb と WWW ブラウザとの組みを一つの構成単位と考えると、この構成単位は、個人の情報の発信側にも受け取り手にもなり得るような、WWW エージェントとでも言うべき存在である。この WWW エージェントが多数存在するようなマルチエージェント構成では、必要に応じて任意の WWW エージェントが動的に連携を結び、必要が無くなれば連携を解消することができるため、より柔軟で、しかもスケラビリティの高いシステムを構成することができる。

IntelliWeb は現在、マルチエージェント指向グループウェア IntelliTeam[8] のツール群を連携する WWW サーバとして運用されている。

## 参考文献

- [1] April Home Page. <http://www.fujitsu.co.jp/hypertext/Products/Software/April/index.html>.
- [2] Hypertext Transfer Protocol - HTTP/1.0. <ftp://ftp.internic.net/rfc/rfc1945.txt>.
- [3] Jeeves Home Page. <http://www.javasoft.com/products/jeeves/>.
- [4] Netscape Navigator Remote Control. <http://home.netscape.com/newsref/std/>.
- [5] Netscape API Functions. <http://home.netscape.com/comprod/server.central/config/nsapi.html>.
- [6] 牛嶋悟, 市來宏基, 塩内正利, 高田裕志, 毛利隆夫, 和田裕二. 移動エージェントに対応したパーソナル WWW サーバ IntelliWeb. 第 54 回情報処理学会全国大会講演論文集, 第 3 巻, pp. 551-552, 1997.
- [7] 牛嶋悟, 高田裕志. April プログラミングガイド. チュートリアル ネットワーク指向プログラミング言語, pp. 117-156. 日本ソフトウェア科学会, 1996.
- [8] 塩内正利, 市來宏基, 牛嶋悟, 高田裕志, 毛利隆夫, 和田裕二. マルチエージェントモデルに基づくグループウェア IntelliTeam. 第 54 回情報処理学会全国大会講演論文集, 第 4 巻, pp. 231-232, 1997.
- [9] 和田裕二, 市來宏基, 塩内正利, 高田裕志, 毛利隆夫. マルチエージェント技術を用いたスケジュール管理システムの構築. 第 53 回全国大会講演論文集, 第 3 巻, pp. 369-370. 情報処理学会, 1996.