

## 移動エージェントに対応したパーソナル WWW サーバ IntelliWeb \*

牛嶋 悟 市来 宏基 塩内 正利 高田 裕志 毛利 隆夫 和田 裕二

3 T-6

(株) 富士通研究所 ネットメディア研究センター

E-mail: {ushijima, iciki, shiouchi, yuji, tmohri, wada}@flab.fujitsu.co.jp

## 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の GUI として WWW ブラウザが使われる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 ブラウザを利用することによって親しみやすいユーザインタフェースを提供することができる。WWW ブラウザを GUI として使う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する、WWW サーバ依存のシステムである。

このような WWW サーバを中心としたシステムにおいて問題となる点がいくつかある。まず、一つ目の問題点として、従来の WWW サーバでは機能の拡張が容易でないということがあげられる。サーバの機能を拡張する方法として、CGI のプログラムを書く方法がある。しかし、これは CGI スクリプトをあらかじめサーバにインストールしておく必要があり、また、ブラウザからのリクエストのたびにプロセスが起動するため、サーバの負荷が大きくなってしまふ。NSAPI[1] などを使って WWW サーバに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リンクする必要がある。また、Jeeves[2] 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、基本的に一方方向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、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。

二つ目の問題点は、他人と共有して使う WWW サーバの限界で、ユーザのプライバシーが確保できないことである。アプリケーションの中には、アクセス制限を設けてユーザのデータを保護するものもある。しかし、そのアプリケーションへのアクセスのために、他人と共有する従来の WWW サーバを使ったのでは、ユーザのプライバシーを守ることができない。

我々はこれらの問題点を解決するために、WWW サーバ IntelliWeb を開発した。IntelliWeb はプログラミング言語 April で実現された WWW サーバである。April はマルチプロセスや、グローバルネットワークに対応したプロセス管理をサポートする、ネットワーク指向のプロ

グラミング言語である。本稿では、IntelliWeb における上記の問題点の解決手法について述べる。

## 2 WWW サーバ IntelliWeb

前節で述べた従来の WWW サーバの問題を解決するために、IntelliWeb では以下の二つの工夫を行なった。まず一つ目は、外部エージェントとの連携機能と、移動エージェントを受け付ける仕組みを設けたことである。エージェントとは April で書かれた、一つの独立したサービスを行なうプロセスである。WWW サーバに追加するサービスを移動エージェントとして用意し、必要に応じて WWW サーバとのリンク付けを行なう。これによって WWW サーバの機能の追加、削除、更新が簡単に行なえるようになる。もう一つは WWW サーバのパーソナル化である。リモートホスト上で動作する WWW サーバを複数のユーザが共有するのではなく、自分専用の WWW サーバを WWW ブラウザと同じホストで動かす。これでアプリケーションへのアクセス制御をユーザ自身が管理できるため、ユーザのプライバシーを守ることができる。

以下では IntelliWeb が提供する、エージェントによるサービスの仕組みを説明する。

## 2.1 外部エージェントとの連携機能

IntelliWeb は WWW ブラウザとの間は HTTP で通信を行なうが、April で書かれた他のアプリケーションとの間は April のメッセージ形式で通信を行なう。

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外部エージェント名 (April プロセス名) との関連付けを行なう。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 *process-name* で示される名前のプロセスがサービスを行なうというルールを設定している。

```
http://host/april/process-name?key1=value1&...
```

プロセス名の後ろにはプロセスに与えるパラメータのキーと値のセットが記述されている。例えば次の URL は、ホスト localhost の IntelliWeb を経由して、localhost 上の calendar という名前のプロセスに、パラメータ month を 12 としてアクセスするという意味である。

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

\*IntelliWeb: A Personal WWW Server with Mobile Agent; Satoru Ushijima, Hiroki Ichiki, Masatoshi Shiouchi, Yuji Takada, Takao Mohri and Yuji Wada; Netmedia Lab., FUJITSU LABORATORIES LTD. 2-2-1 Momochihama, Sawara-ku, Fukuoka 814, Japan

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。また、Aprilのプロセス間通信は異なるホスト間でも可能であるため、サービスを提供するプロセスはIntelliWebとは別のホストで動作していても構わないという利点がある(図1)。

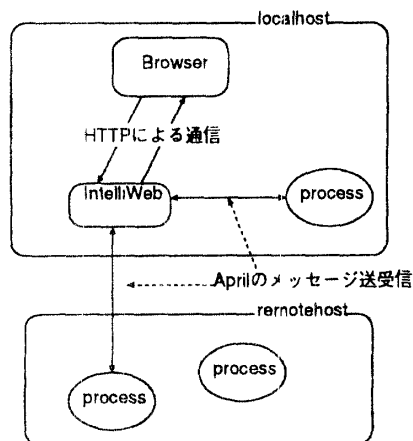


図1: 外部エージェントとの連携

## 2.2 移動エージェントへの対応

IntelliWebは、外部エージェントとの連携機能の他にAprilのプロセスを起動し、管理する能力を持っており、プロセス名と、プロセスの実行に必要なリソースが置かれている場所を管理し、必要に応じてプロセスの起動、停止を行なう。先の例で使ったURL

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

がリクエストとして送られてきた時に、localhostにプロセスcalendarが起動していなかった場合には、IntelliWebは次のような動作をする(図2)。

1. calendarを実行するために必要なコードを取得し、起動に必要なリソースを確保
2. calendarという名前でプロセス起動
3. パラメータをcalendarプロセスに送信
4. calendarプロセスからの返答をブラウザに返信

以後calendarプロセスは常駐し、もう一度calendarへのリクエストがあった場合は、IntelliWebはすぐにこのプロセスと通信を行なって結果をブラウザに返す。そして、長期間calendarに対するリクエストがなかった場

合は、IntelliWebは自動的にcalendarプロセスを停止させる。

Aprilを利用した移動エージェントに対応することにより、IntelliWebが提供する機能の追加、削除、更新を簡単に行なうことができる。また、IntelliWebは個人での利用を前提としているため、移動エージェントをより安全に扱うことができる。そして、HTTPに依存しないため、Aprilで書かれた一般のプロセスからでもIntelliWebへの機能提供を行なうことができる。

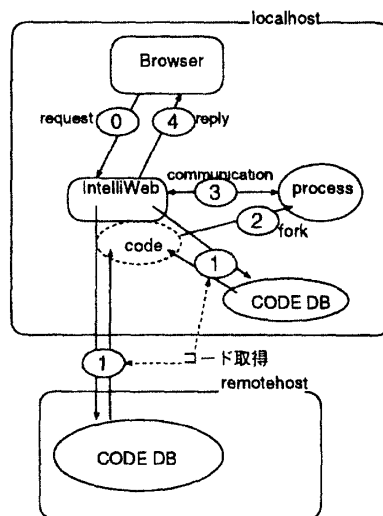


図2: 移動エージェントの仕組み

IntelliWebは現在、マルチエージェント指向グループウェアIntelliTeam[3]のツール群を連携するWWWサーバとして運用されている。

## 3 まとめ

移動エージェント対応のパーソナルWWWサーバIntelliWebをネットワーク指向プログラミング言語Aprilで実現した。WWWサーバへの機能拡張が柔軟に行なえるようになった他、個人ベースの利用を前提としているため、個人データのプライバシーが確保でき、エージェント連携をより安全に扱うことができるようになった。

## 参考文献

- [1] Netscape API Functions, <http://home.netscape.com/comprod/server/central/config/nsapi.html>
- [2] Jeeves Home Page, <http://www.javasoft.com/products/jeeves/>
- [3] 塩内, 市来, 牛嶋, 高田, 毛利, 和田, マルチエージェントモデルに基づくグループウェアIntelliTeam. In 情報処理学会第54回全国大会, 3V-05, Mar, 1997.