

F-008

Web エージェントに基づく Web アプリケーション開発環境の試作 Realizing a Web Application Framework based on a Web Agent

中村正人[†] 大園忠親^{††} 新谷虎松^{††}

Masato Nakamura, Tadachika Ozono, Toramatsu Shintani

1 はじめに

本研究では、迅速な Web アプリケーション開発の為の開発環境の実現を目指す。ここでの迅速な Web アプリケーション開発環境とは、試行錯誤が容易、かつ簡単に習得できる開発環境である。そのような開発環境の実現には、次の 3 点が重要である。1 つ目は、サーバサイドとクライアントサイド、両方のプログラムを単一の言語により記述可能とすることで。2 つ目は、Web サービス間の整合性の確保を容易にすることである。3 つ目は、プログラムのコンパイルやサーバへのアップロードといった手間を省くことである。Web アプリケーション開発は、クライアントサイドの言語とサーバサイドの言語、両サイドの言語を習得する必要があり、教育コストが高い。また、Web アプリケーション同士の連携を図る上で、プログラムを修正しつつ整合性を取るには手間のかかる作業を要する。さらに、Web アプリケーションプログラムが複数のサーバ上に存在する場合、サーバへのアップロードなど余計な手間がかかるため、試行錯誤に時間を要する。これらの問題を解決する手法として、本稿では、Web エージェントに基づく Web アプリケーション開発環境を提案する。

2 Web エージェントのための MiLog の拡張

本論文では、モバイルエージェントフレームワークに基づく Web アプリケーション開発環境を提案する。エージェントとしてプログラムを移動させることで、単一の言語による記述を可能とし、プログラムのアップロードの手間を省く。また、継続性 (Continuation) を実現することで整合性の確保を容易にする。

本研究では、JavaScript で記述したプログラムとデータを一つのモジュールとして Web エージェントと定義する。Web エージェントは、クライアントやサーバを移動し実行することが可能である。サーバ間において Web エージェントの移動を実現するため、本研究室で開発されたモバイルエージェントフレームワークである MiSpider[1] と MiLog[2] を用いる。MiSpider とは、ブラウザ上で動作するモバイルエージェントフレームワークであり、MiLog とは Java で動作するモバイルエージェントフレームワークである。MiLog は、エージェントフレームワークとしての機能のみでなく、HTTP との連携も実現しており、MiLog エージェントを CGI として動作させることも可能である。MiLog エージェントは、MiLog プラットフォーム上のみ移動可能である。MiLog エージェントの動作は論理型言語 Prolog のサブセットで記述し、Java を用いることで述語の機能を拡張することが可能である。

本システムの全体像を図 1 に示す。図 1 において、丸で囲まれた J は、JavaScript により記述された Web エージェントを示す。また、丸で囲まれた M は、MiLog エージェントを示す。クライアントサイドでも動作する Web エージェントは、サーバサイドにおいて MiLog エージェントの一部として移動し、実行される。MiLog エージェントの制御は、Web エージェントが行う。

2.1 JavaScript 実行器

本システムでは、サーバサイドにおける JavaScript 実行フレームワークとして、JavaScript 実行エンジンである Rhino¹ を用いた。Rhino は、JavaScript コード中に記述し

[†] 名古屋工業大学 工学部 情報工学科

^{††} 名古屋工業大学大学院 工学研究科 情報工学専攻

¹ <http://www.mozilla.org/rhino/>

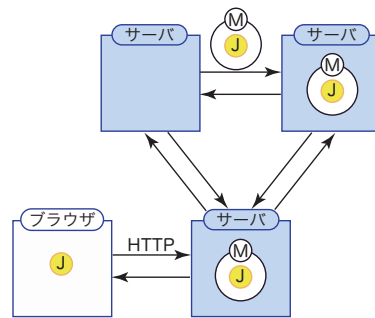


図 1: システムの全体像

た JavaScript プログラムも実行することができる。

本システムにおける JavaScript の実行方法を図 2 に示す。サーバにおいてクライアントから受信した Web エージェントは、MiLog エージェントの述語として保持される。サーバを移動した Web エージェントは、MiLog エージェントの述語から取り出され、Rhino を用いて実行される。

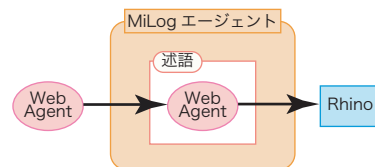


図 2: サーバサイドにおける JavaScript 実行モデル

2.2 変数変換器

本システムでは、Web エージェントのウィークマイグレーションを実現する。そこで、Web エージェントがサーバ間を移動する際、変数情報の保持を行う。

Web エージェント中の変数情報は、MiLog エージェントの述語へ変換して保持する。この変数情報から述語への変数変換は、独自に定義した変数変換器を用いて行う。図 3 に変数変換のモデルを示す。

例えば変数変換器を用いると JavaScript における $a=2$ という変数情報は、MiLog エージェントにおいて以下のような述語で保持される。

```
jsVar(a, 2).
```

Web エージェントから MiLog エージェントへの変数情報の格納は、Web エージェント実行後に行う。また、MiLog エージェントから Web エージェントへの変数情報の読み込みは、Web エージェント実行前に行う。

3 Web エージェントの移動

本システムは、ブラウザにおける Web アプリケーションの開発を可能とする。ブラウザからサーバへ移動した Web エージェントは、MiLog エージェントに梱包された形でサーバ間を移動し、実行される。MiLog エージェントの移動命令は、Web エージェントが行う。

Web エージェントの移動は、独自に定義した関数 (moveFromClientToBaseServer(), moveTo(), moveToClient()) を用いて行う。

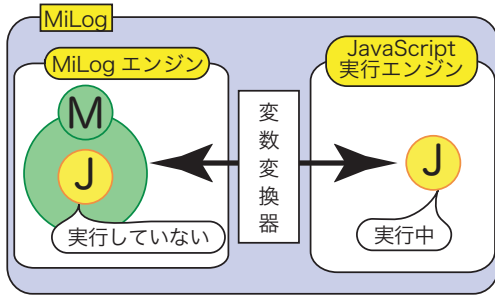


図 3: 変数変換器を用いた変数情報の保持

3.1 継続 (Continuation)

プログラムを継続する際、スタック情報を取得することが望ましい。しかし JavaScript では、詳細なスタック情報を取得することは不可能である。そこで本研究では、JavaScript コードの継続点となる MiLog エージェントの移動命令を関数外でのみ実行可能とする。実行可能な範囲を関数外に限定することで、スタック情報の取得が不要となる。

継続する際にコードの継続点を識別するため、コードを実行する前にクライアントにおいてコードの変更を行う。つまり、JavaScript の移動関数に対し引数としてそれぞれ個別の ID を付加する。

Web エージェントの移動処理の流れを以下に示す。

3.2 クライアントからの受信

サーバにおいてクライアントから Web エージェントを受信した後の処理の流れを以下に示す。

- a-1 受信した Web エージェントを MiLog エージェントの述語 (code/1) として保持する。また、Web エージェント中の実行済み移動命令 (moveFromClientToBaseServer()) の ID を MiLog エージェントの述語 (moveID/1) として、Web エージェントの変数情報を述語 (jsVar/2) として保持する。
- a-2 MiLog エージェントのコードを保持する述語 (code/1) からコードを読み出し、これをコード A とする。
- a-3 MiLog エージェントの述語 (moveID/1) として保持した、直前に実行した移動命令の ID と一致する ID を持つ移動命令をコード A から検索し、その移動命令以前のコードを削除する。
- a-4 変数情報を保持する MiLog エージェントの述語 (jsVar/2) から変数情報を取得し、コード A の先頭に付加する。
- a-5 Rhino を用いて Web エージェントの実行を行う。Web エージェントにおいて、移動命令 (moveTo()) を呼び出して別のサーバに移るならば、以下の「3.3 サーバ間の移動」を実行する。また、移動命令 (moveToClient()) を呼び出してクライアントに移るならば、以下の「3.4 クライアントへの移動」を実行する。

3.3 サーバ間の移動

Web エージェントがサーバ間を移動する流れを以下に示す。

- b-1 Web エージェント中で実行した移動命令 (moveTo()) の ID を MiLog エージェントの述語 (moveID/1) へ格納する。
- b-2 変数情報を MiLog エージェントの述語 (jsVar/2) へ格納する。
- b-3 MiLog エージェントを移動させる。
- b-4 a-2 以降を実行する。

3.4 クライアントへの移動

Web エージェントがサーバからクライアントへ移動する流れを以下に示す。

- c-1 MiLog エージェントの述語 (code/1) から Web エージェントを取得する。これをコード B とする。
- c-2 直前に実行した移動命令の ID を保持する MiLog エージェントの述語 (moveID/1) から ID を、変数情報を格納する MiLog エージェントの述語 (jsVar/2) からそれぞれ取得する。それらを JSON 形式としてコード B に付加し、クライアントへと送信する。
- c-3 クライアントへ移動した web エージェントは、直前に実行した移動命令 (moveToClient()) 以前のコードが削除される。また、代わりに変数情報の更新作業が付加され、処理が再開される。

4 Web アプリケーション開発環境

4.1 システムの機能

本システムは、ブラウザにおける Web アプリケーションの開発を可能とする。本システムを用いると、CGI のようなサーバ待機型アプリケーションや、サーバで処理を実行した後にクライアントへと戻ってくる帰還型アプリケーションなど、様々な形態の Web アプリケーションを JavaScript のみで作成することが可能である。

4.2 簡易チャットシステムへの応用

本システムを用いて、簡易チャットシステムを作成した。プログラムのソースを以下に示す。

```

01: var msg=""; //メッセージを保持する
02: while(true){
03:     //クライアント
04:     msg = prompt("メッセージを入力してください");
05:     moveToServer(); //サーバへ移動
06:     //サーバ
07:     saveMsg(); //メッセージをサーバ上に保存
08:     //別クライアントへ移動
09:     moveToClient(getAnotherClientAddress());
10:     //相手のクライアント
11:     displayMsg(); //メッセージ表示
12: }
```

5 おわりに

文献 [3] では、エージェントの協調動作に基づく Web アプリケーション開発が提案されている。しかしこのモデルはサーバサイドにおけるエージェントシステムであり、クライアントサイドへと移動することは不可能である。本研究では、JavaScript のみで開発可能な Web アプリケーション開発環境を試作した。クライアントとサーバを移動可能な Web アプリケーションを開発できることで、Web アプリケーションの開発者の負担を軽減することが可能と考えられる。

参考文献

- [1] Y Fukagaya, T Ozono, T Ito, T Shintani: "MiSpider: a continuous agent on web pages", International World Wide Web Conference, 2005
- [2] N Fukuta, T Ito, T Shintani: "A Logic-based Framework for Mobile Intelligent Information Agents", WWW10,2001
- [3] Y Qi, D Kung, E Wong: "An Agent-Based Testing Approach for Web Applications", Computer Software and Applications Conference 2005