

Web エージェント MiSpider に基づく Web ページ協調編集機構の実装

On a Cooperative Web Pages Editing System Based on MiSpider

深萱裕二郎*¹ 大園忠親*¹ 伊藤孝行*¹ 新谷虎松*¹
Yujiro Fukagaya Tadachika Ozono Takayuki Ito Toramatsu Shintani

*¹名古屋工業大学大学院 工学研究科 情報工学専攻

Dept. of Computer Science and Engineering, Graduate School of Engineering, Nagoya Institute of Technology

In this paper, we propose a Web-based agent system called MiSpider, which provide intelligent web services on web browsers. MiSpider Agent can keep its execution state among HTTP sessions, therefore, it enables MiSpider to move another web page and resume its execution continuously. We implement a cooperative web pages editing system based on MiSpider. In order to promote collecting ideas, we realize seam-less writing to Web pages.

1. はじめに

近年, WWW の発展により, クライアントの OS 上で提供されているようなサービスを Web ブラウザを介して, Web 上で提供する要求が高まっている. Web 上でサービスを提供することで, ソフトウェアのインストールを必要とせずあらゆるマシンから同一のサービスを利用できる利点がある. しかし, Web ブラウザを実行環境とすることで, Web ブラウザ自体の制限, 例えば, セキュリティ上の問題を回避するために, 通信などの計算機資源が制限されている. 通信に関する制約は厳しく, 通信先や通信可能なタイミングが限られており, クライアントアプリケーション並のインタラクションを持つ Web アプリケーションの開発には困難を伴う. また, セッション間において Web ブラウザ上で動作するプログラムが初期化され, 継続的なサービス提供ができない. 本研究では, Web ブラウザ上で動作するエージェントシステムとして MiSpider を開発することでこれらの問題にアプローチする.

MiSpider におけるエージェントは, ユーザが Web ブラウザのリロードが発生する処理を行った場合においても, 内部情報を保持し続け, 継続的にサービスを提供可能である. また, エージェント間で通信を行うためのメッセージ通信能力を持つ. このようなエージェントの機能を利用することで効果的に Web アプリケーションの開発を行うことができる.

以降, 2 章では MiSpider の構成及び機能について述べる. 3 章では, MiSpider を利用したシステムとして, Web ページ協調編集機構について述べる. 最後に 4 章で本稿をまとめる.

2. Web エージェント MiSpider

MiSpider は, ページ上で動作するエージェントとサーバ上で動作するエージェントが協調してサービスを提供する. ここでは, 次のような制約がある. 1) ページ上のエージェントとサーバ上のエージェントの通信には制約がある. 2) ページ上のエージェントは計算資源が限られている. 3) サーバ上のエージェントは計算資源はあるがユーザと直接インタラクションできない.

以下でこれらの制約を考慮したシステム構成について述べる.

連絡先: 深萱裕二郎, 名古屋工業大学大学院 情報工学専攻 新谷研究室, 〒466-8555 名古屋市 昭和区 御器所町 名古屋工業大学, TEL:(052)735-7968, FAX:(052)735-5477, fukagaya@ics.nitech.ac.jp

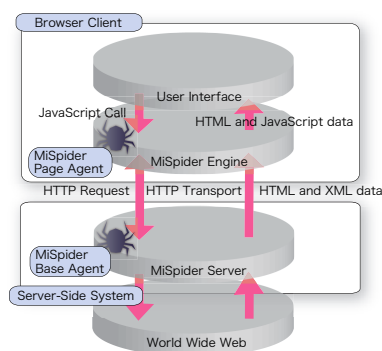


図 1: システム構成図

2.1 システムの構成

図 1 は, MiSpider におけるエージェントの構成を表す. MiSpider におけるエージェントは, ベースエージェントとページエージェントから構成される. ページエージェントとは Web ページ上で動作するエージェントであり, ページ上のデータやユーザに対してサービスを提供する. ベースエージェントとは Web サーバ上で動作するエージェントであり, ページエージェントのタスク実行をサーバ側から支援する. ベースエージェントは, 複数のページエージェントと連携するが, あるページエージェントと連携するベースエージェントは一意に定まる. 図 1 の MiSpider Server はプロキシサーバであり, 取得した Web ページにベースエージェントがページエージェントのプログラムを付加し, Web ブラウザに送信する.

このように, エージェントを 2 つの異なるエージェントから構成するのは, Web ブラウザのさまざまな制限 (通信など) を越えて, より効果的なサービスを提供するためである. 本構成により, エージェント間の通信が可能になり, またページ上のエージェントが異なるセッション間で継続的に動作できる.

2.2 永続性

MiSpider では, セッション間における継続的なサービス提供を実現している. ページエージェントの実装には, Web ブラウザ上でインタラクティブな処理を行うことができる JavaScript 言語を用いる. Web ブラウザ上で動作するプログラムは, ユーザが閲覧ページを移動した際に, 実行情報が初期化される. 閲覧ページを移動した後も継続的にエージェントを行わせるためには, 前のページを閉じる時 (ページを移動することも含む) に, ページエージェントが実行状態を復帰するための手法が必

要である。

MiSpider では、プロキシを通して Web ページにアクセスする。プロキシでは、JavaScript の埋め込みやリンク先の書き換えなどを行う。埋め込まれた JavaScript ファイルには、ページ読み込み時の処理と、閲覧ページ移動時の処理を記述する。そのための手続きは次の通りである。Web ページを読み込む時に、エージェントの内部状態を記述した XML ファイルを同時に読み込む。取得した XML 文書を構文解析して、JavaScript オブジェクト *obj* に変換する。*obj* から得られた実行状態をエージェントの実行状態として再設定した後に、エージェントの処理を再開する。ページを閉じる時には、*obj* を XML 文書に変換し、内部状態を XML ファイルとして保存する。エージェントの内部状態の表現に XML を用いることで、JavaScript のオブジェクトや配列などのデータ構造を表現できる。

3. Web ページ協調編集機構

3章では、MiSpider を利用したシステムとして、Web ページ協調編集システムについて述べる。本システムは関連研究 [2] と同様に Web ブラウザ上から直接に Web ページを作成、編集が可能である。[2] との違いはリアルタイムの協調編集機能を持っている点である。ページエージェントの機能として Web ページ編集機能、ベースエージェントの機能として編集者同士の Web ブラウザの表示を同期させる機能を持たせている。編集を開始するには編集するページを開いて、ブックマークレットと呼ばれる小さな JavaScript プログラムから編集画面に移動する。

3.1 本システムの機能

本システムの具体的な機能として、文章の編集、文字色、文字の背景色、文字フォント、書体 (ボールド、イタリックなど)、文字サイズの変更、リンクの作成、削除線、アンダーバーをつける、レイアウトの変更 (センタリング、右寄せ、左寄せ)、画像の貼付けを図 2 のメニューバーを使って利用できる。

3.2 協調編集

編集者がページ *p* を編集している時に、他のマシンから *p* を複数人で編集することができる。*p* の内容が編集がされると、編集された内容が、ブラウザのリロードを行う必要なく、即座に他の編集者の Web ブラウザの表示にも反映される。

協調編集機構

協調編集を実現するための機構について述べる。ページの編集が開始される時に、編集するページのキャッシュページ *c* を生成する。編集 CGI では、ページ *c* にページを編集するためのスクリプトを内包したページエージェントを付加する。編集者は編集 CGI へ移動するためのリンク (システムのトップページに生成される) からページの編集に参加することができる。編集が行われた時に、編集者全員の Web ブラウザの表示に編集結果を反映するために、サーバ側にあるページ *c* への書き込み、及びページ *c* からデータをロードして Web ブラウザの表示に反映させる処理が必要となる。ページ *c* への書き込みは、編集が行われた時になされる。文字入力や画像などのオブジェクトの移動のような連続した編集では、書き込みのための通信が頻繁に発生するため、入力開始から一定時間後 (3 秒後) に書き込みが行われる。書き込みが行われるまでの間はページ *c* のデータのロードは停止される。ロードは定期的 (10 秒毎) に行われるが、編集者の操作中には、操作の妨げになるのを防ぐためにロードが延期される。ロードが発生してから、次のロードが発生するまでの間に他者による編集が行われてい



図 2: Web ページ編集画面

ない場合 (編集中のデータとページ *c* のデータが等しい場合) には、表示の更新は行われず、キャッシュデータのロード、セーブのタイミングについてはサーバの負荷とインタラクションのトレードオフにより決定している。

また、文字入力や画像などのオブジェクトの移動のような連続性のある編集は一度に複数のユーザが同時に編集しようとする、編集内容が正確に反映されないなどの不都合が発生する恐れがある。それを防ぐために連続性のある編集は同時に複数のユーザができないようにロックする。

3.3 オリジナルのページへの編集内容の反映

メニューバーからセーブボタンをクリックすることでオリジナルのページと編集が行われたキャッシュページとを置き換える。システムと同一のサーバ上のページを編集する場合は、システムの設定にドキュメントルートのパスを設定しておく必要がある。他のサーバ上のページを編集する場合は、サーバ側のプログラムで FTP を用いてページを置き換えるために、さらに、アカウント名とパスワードが必要となる。

4. まとめ

本稿では、Web エージェント MiSpider 及び、MiSpider に基づく Web ページ協調編集機構について述べた。MiSpider が提供する継続的なサービスを実現する永続性や、メッセージ通信機能を利用することで、効果的な Web アプリケーション開発が可能になる。

今後の課題として、サーバへの負荷低減手法の実現する必要がある。不特定多数のエージェントからの頻繁なアクセスによってサーバが過負荷になり、メッセージ通信に支障を及ぼす可能性がある。解決のためにメッセージの送受信タイミングを調整するような仕組みが必要である。Web ページ協調編集機構については、セキュリティについて十分な考慮がなされていないため、現状ではパスワードをかけたディレクトリ上で運用する必要がある。

参考文献

- [1] Y. Fukagaya, T. Ozono, T. Ito and T. Shin-tani, "MiSpider: A Continuous Agent on Web Pages," WWW2005, pp.1008-1009, 2005.
- [2] 西健太郎, 新谷虎松, 松尾徳朗, 田代慎治, 伊藤孝行, "既存 Web ブラウザを利用したオンライン編集可能な Web ページの実現," 電気学会論文誌 (部門誌)C, 電気学会, Apr. 2005.