6P-4

Web エージェント MiSpider におけるサービスの記述法

深萱裕二郎 † 大囿忠親 † 伊藤孝行 † 新谷虎松 †
† 名古屋工業大学大学院 工学研究科 情報工学専攻
e-mail: {fukagaya, ozono, itota, tora}@ics.nitech.ac.jp

1 はじめに

近年，Web ブラウザをプラットフォームとして扱い，Web 上のプラットフォームで様々なサービスを提供することが注目されている [2]．Web ブラウザをプラットフォームとして扱うためには，継続的なサービスの提供，およびサービスを容易に記述し，実行可能である枠組みが必要である．本研究では，Web ページ上を実行環境とするソフトウェアエージェントとして MiSpider を実装した．MiSpider におけるエージェントは，異なる HTTP セッション間で，内部情報を保持したまま継続的に動作する [1]．本稿では，XML 形式によるサービスの記述法，およびエージェントがサービスを実行する枠組みについて述べる．

第 2 章では，MiSpider の概要と構成について述べる．第 3 章では，MiSpider におけるサービスの記述法と具体例について述べる．最後に第 4 章で本論文をまとめる．

2 MiSpider の概要と構成

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．

これらの制約を考慮し，ページ上のエージェントは，ユーザに対するセンサ及びアクチュエータのような役割を担当する．また，サーバ上のエージェントはページ上のエージェントを遠隔操作する役割を担当する．

2.1 アーキテクチャ

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．

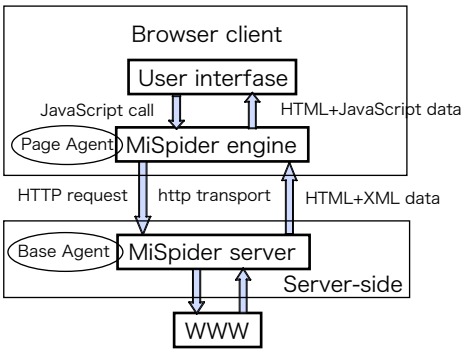

図 1: システムの構成

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 ページエージェント

ページエージェントの定義は，ページエージェント内の 4 つのモジュールを定義することで実現される．すなわち，センサ定義，アクチュエータ定義，イベント定義，そして推論定義から構成される．センサ定義とアクチュエータ定義は，ページエージェントの低レベルの処理を定義する．センサ定義は，ブラウザ上の情報の変化やユーザからのインタラクションをイベントに変換する．アクチュエータ定義は，高レベルの処理を低レベルに変換し，ページエージェントの外界に対して変化を与える手続きを定義する．イベント定義は，マウスクリックのような低レベルのイベントを，ユーザからの情報要求のようなより高レベルなイベントに変換する手続きを定義する．推論定義では，ページエージェントの推論機構を決定するための定義をする．

3 Web エージェント MiSpider におけるサービス提供

ユーザの行動を MiSpider のエージェントが観察し，ユーザの行動に基づいてサービスを提供する．サービス内容とサービス提供のトリガーを XML で定義する．ユーザの行動の観察としては，マウスの移動，クリック，ドラッグ，フォームへの文字入力などを観察し，行動に応じたアクションを起こす．

On Service Description Method of Web Agent System MiSpider
Yujiro FUKAGAYA, Tadachika OZONO, Takayuki ITO, Toramatsu SHINTANI

3.1 具体例：フォームへの入力の監視

サービス提供について具体例を示す．例として以下の HTML ソースで表されるサインアップ用のフォームを監視する．

```
<form>
 アカウント: <input id="account" type="text">
 <span id="account_status"></span><br>
 パスワード: <input id="password" type="password">
 <span id="password_status"></span><br>
 パスワード (確認):<input id="confirm" type="password">
 <span id="confirm_status"></span><br>
 <input type="submit" value="サインアップ">
</form>
```

アカウントへの入力を完了すると (フォーカスが他に移動)，同一アカウントがすでに登録されていないかをサーバ上のベースエージェントに問い合わせる．すでに同一アカウントが登録されていれば警告を表示する．パスワード (確認) に入力した時にパスワードと一致しない場合は警告を表示するようにする．

フォーム監視はページエージェントで行われるため，ページエージェントを定義する．

```
1: <pageagent>
2:  <meta></meta>
3:  <body>
4:    <sensers>
5:      <senser name="form"/>
6:      </sensers>
7:      <actuators>
8:      <actuator name="check"/>
9:    </actuators>
10:  </body>
11: </pageagent>
```

２行目のメタ要素内ではページエージェントの名前などの情報の記述を行うが，ここでは省略する．４－６行目ではページエージェントの持つセンサーを定義している．７－９行目ではページエージェントの持つアクチュエータを定義している．次にセンサーの定義を示す．

```
1: <senser name="form">
2:   <elements>
3:     <element><id>account</id>
4:     <event>onBlur</event></element>
5:     <element><id>confirm</id>
6:     <event>onChange</event></element>
7:   </elements>
8: </senser>
```

３－４行目で id が account である HTML 要素からフォーカスが外れた時に検知する．５－６行目で id が confirm である HTML 要素の値の変化を検知する．次にアクチュエータの定義を示す．

```
1: <actuator name="check">
2:   <elements>
3:     <element>
4:       <id>account</id>
5:       <if_added>
6:        if(!request("account", $account)) {
7: $account_status=$account+"は利用できません";
8:        }
9:        </if_added>
10:    </element>
11:   <element>
12:     <id>confirm</id>
13:     <reqs><req><eq>$password</eq></req></reqs>
14:     <if_added>
15:      if(!req(0)) {
16: $confirm_status = "パスワードが一致しません";
17:      }
18:     </if_added></element></elements></actuator>
```

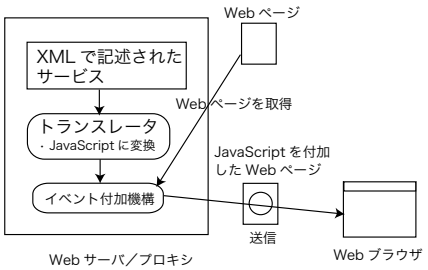

図 2: サービス提供の流れ

フォームの input 要素に対して知識表現としてのフレーム表現が可能である．３行目の element は４行目の id の値と同じ id を持つ HTML の要素を表す．５－９行目及び，１４－１８行目の if_added は input 要素の値が定まった時に起動する付加手続きである．６行目の $account のように，$[id] で表現される文字列によって対応する id を持つ HTML 要素の値を参照できる．６行目の request() はベースエージェントへの問い合わせを示す．返り値は，この例では同一アカウントがすでに登録されていれば true を返す．１３行目の req は入力文字に対する制約を表す．制約は複数登録可能である．１５行目の req(0) は１つ目の制約を満たしていれば true，満たしていなければ false を返す．

サービス提供の流れを図２に示す．XML で記述されたサービスは図２中のトランスレータで JavaScript 言語に変換される．図２中のイベント付加機構で Web ページに JavaScript 言語のイベント，およびイベントに対応して実行されるプログラムが Web ページに付加され，Web ブラウザへ送信される．

4 まとめ

本稿では，Web ページ上を実行環境とするソフトウェアエージェント MiSpider におけるサービスの記述法について述べた．サービスの表現に XML を用いることで知識表現を用いたサービスの記述が可能になる．

参考文献

[1] Y. Fukagaya, T. Ozono, T. Ito and T. Shintani, ”MiSpider: A Continuous Agent on Web Pages,” WWW2005, pp.1008-1009, 2005.

[2] O’Reilly: What is Web 2.0 http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html