

# MiSpider: A Continuous Agent on Web pages

Yujiro Fukagaya      Tadachika Ozono  
fukagaya@ics.nitech.ac.jp      ozono@ics.nitech.ac.jp

Takayuki Ito      Toramatsu Shintani  
itota@ics.nitech.ac.jp      tora@ics.nitech.ac.jp

Dept. of Computer Science and Engineering  
Nagoya Institute of Technology  
Gokiso-cho, Showa-ku, Nagoya, Aichi, 4668555, JAPAN

## ABSTRACT

In this paper, we propose a Web based agent system called MiSpider, which provides intelligent web services on web browsers. MiSpider enables users to use agents on existing browsers. Users can use MiSpider all over the world only to access the Internet. MiSpider Agent has persistency, and agents condition doesn't change if users change a browsing page. Moreover, agents have a message passing skill to communicate among the agents.

## Categories and Subject Descriptors

H.4.3 [Communication Applications]: Information browsers;  
H.5.3 [Group and Organization Interfaces]: Web-based interaction; I.2.11 [Distributed Artificial Intelligence]: Intelligent Agent

## General Terms

Design

## Keywords

Browsing Support, Multiagent System, and Information System

## 1. INTRODUCTION

Our aim is developing MiSpider, which is an agent system on web pages. MiSpider Agent provides intelligent web services on web browsers. The following problems occur in achieving intelligent web services on web browsers. First, it is difficult to provide continuous services. Second, it is difficult to provide real-time services for user interaction. Third, the resources on web browsers are limited, and web browsers need to cooperate with a web server, however, it is difficult to communicate with a web server freely. MiSpider solve these problems.

Using MiSpider, users can use agents on existing browsers. So, users can use MiSpider all over the world only to access the Internet. Developers can easily create agents for some web sites. MiSpider enables to obtain the input values of the form and link access, and developers can develop agents to support browsing interactively to user's input.

Copyright is held by the author/owner.  
WWW 2005, May 10-14, 2005, Chiba, Japan.  
ACM 1-59593-051-5/05/0005.

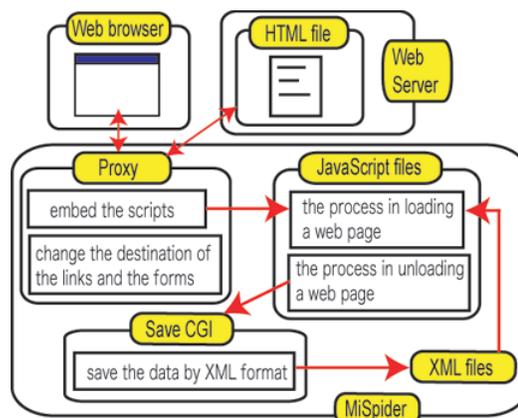


Figure 1: System Architecture

We propose a innovative idea to support users continuously on web browsers. MiSpider Agent has persistency, and agents condition doesn't change if users change a browsing page. Not being affected by loading pages, MiSpider Agent supports user's browsing continuously. Moreover, agents have a message passing skill to communicate between agents.

MiSpider enables to support user's browsing effectively by using these functions.

## 2. AN AGENT SYSTEM ON WEB PAGES

### 2.1 The process of MiSpider

In the figure 1 shows system architecture. MiSpider consists of JavaScript and perl CGI script. In proxy, the system embeds the JavaScript code to cached original HTML source, and rewrite link destination via proxy. Web browser loads the modified HTML source and show this page. The system scribes the process in loading a web page and the process in unloading a web page. In the process in loading a web page, the system embeds an agent in the web page by using the dynamic positioning technology of DHTML, and obtain information from the XML file. The agent runs by using the obtained information. In the process in unloading a web page, the system sends agent's information to Save CGI. Save CGI rewrite the XML file.

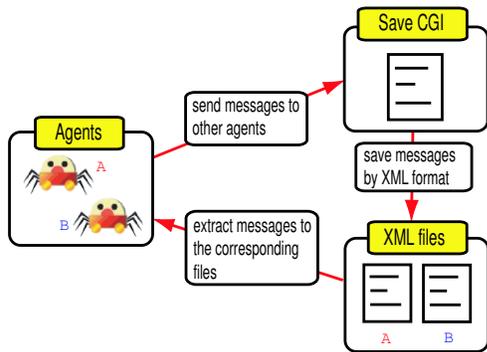


Figure 2: Message Passing

```

[JavaScript]
kbase.hoge_site.items[0].title = "item0";
kbase.hoge_site.items[0].url = "http://www.hoge.co.jp/item0.html";
kbase.hoge_site.items[1].title = "item1";
kbase.hoge_site.items[1].url = "http://www.hoge.co.jp/item1.html";

[XML]
<kbase>
<knowledge>
<name>hoge_site</name>
<items>
<item>
<title>item0</title>
<url>http://www.hoge.co.jp/item0.html</url>
</item>
<item>
<title>item1</title>
<url>http://www.hoge.co.jp/item1.html</url>
</item>
</items>
</knowledge>
</kbase>
  
```

Figure 3: the JavaScript object to the XML source transformation

## 2.2 Message Passing

Agents can send the messages to the arbitrary agents. In figure 2 shows the mechanism of message passing. The *agentA* sends a message to the *agentB*. Agent data are saved on *serverS*. *agentA* submits the name of *agentB* and a message to the CGI on *serverS*. The CGI on *serverS* save the message contents at the (*agentB* name).xml file by XML format. Agents regularly access the (each agent name).xml file on *serverS* and agents receive the messages. The problem that agents are affected by reloading a page occurs when agents send or receive the messages. The communication method e.g. `location.href(cgi name)`; the browsing page is reloaded when the contents in the page are updated. Therefore, we use the XMLHttpRequest[4] object from the JavaScript code when agents send or receive the messages. Using the XMLHttpRequest object, web clients can retrieve and submit the XML data directly without reloading the page.

## 2.3 Persistency

In moving a user's browsing page, agent information is initialized. Agents run continuously after moving a user's browsing page, the method that agents have to send the condition in unloading a old web page and receive the condition in loading a new web page is needed.

The system accesses the XML file and obtain the XML source in loading a web page. The system transforms the XML source to JavaScript object. Agents start running after obtaining the condition.

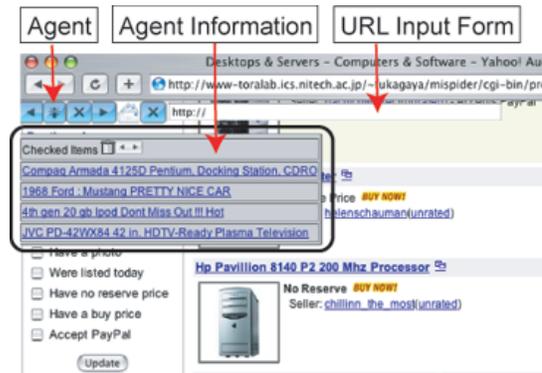


Figure 4: providing information based on user's access

The system transforms JavaScript object to the XML source and send the XML source to Save CGI. In the figure3 shows an example of the JavaScript object to the XML source transformation.

## 3. APPLICATION

### 3.1 Providing Information Based On User's Access

MiSpider turns a display mode by clicking the agent. Agent information and URL form are displayed in On-mode. MiSpider provides the interface which doesn't prevent user's browsing. If user scrolls the browsing page, relative agent position doesn't change.

In figure 4 shows the providing information based on user's access. An Agent appends the link of the user's access page as agent information. MiSpider extracted the keyword from the page users have visited. MiSpider append the keyword to the query in using the search engine.

## 4. CONCLUSIONS

In this paper, we proposed an agent system on web browsers called MiSpider, and we presented the system characteristics, the implementation method of system functions, and the applications using our system. The advantages of MiSpider are following. First, MiSpider enables to use an agent system generally by developing an agent system on existing web browsers. Second, users don't need special softwares and complicate configuration by using the web browsers as interface. Third, MiSpider provides interactive services.

## 5. REFERENCES

- [1] D. Kotz and R. S. Gray, "Mobile agents and the future of the internet.", ACM Operating Systems Review, pp. 7-13, August 1999.
- [2] N. Fukuta, T. Ito, and T. Shintani, "MiLog: A Mobile Agent Framework for Implementing Intelligent Information Agents with Logic Programming, " PRIIA2000, pp.113-123, 2000.
- [3] D. Lange and M. Oshima, "Programming and deploying Java mobile agents with Aglets.", Addison-Wesley, 1998.
- [4] Using the XML HTTP Request object <http://jibbering.com/2002/4/httprequest.html>.