A Web Agent for Automating E-Commerce Operations

Juan Raposo, Manuel Álvarez, Ángel Viña
Department of Information and Communications Technologies - University of A Coruña. 15071 A Coruña - Spain
{jrs,mad,avc}@udc.es

Paula Montoto, Justo Hidalgo, Alberto Pan
Denodo Technologies Inc. Real 22, 3º- 15003 A Coruña - Spain
{pmontoto,jhidalgo,apan}@denodo.com

Abstract

The great amount of information that a user handles in their typical transactions on the Internet –identifiers, addresses, credit card numbers, among others-, makes it necessary to have a mechanism which facilitates the secure management of that information, and its use in a convenient way on a daily basis. The main goal is to favour the B2C –Business to Consumer- e-commerce activity by creating a friendly and secure environment. This paper introduces an Internet navigation assistant, implemented as a generic, extensible and dynamically-updatable personal agent. This agent is included into a distributed architecture which is composed by a set of secure web servers and on the client side, one agent per user machine, developed as a Microsoft Internet Explorer toolbar. The agent includes, among others services, an automatic-filler which learns from users' activity, and secure payments services, like a virtual-card generator or an Europay’s SPA/UCAF wallet.

1. Introducción

The great amount of information that a user handles in a typical Internet transaction usually converts the experience of B2C purchasing into a tedious, messy and error-prone process. The typical information that a user must deal with includes the following:

– Personal information for registration in web sites: name, surname, home and email address, etc.
– User identifiers and passwords to access authentication-enabled sites.
– Information related to making orders, like credit card numbers, address for invoice, delivery addresses, etc.

An application which facilitates the automatic management of such information will indeed be of great help for the user in the purchasing process, translating it into a much friendlier experience.

This paper presents a personal agent for B2C purchase-process assistance. The system is installed in the client’s browser (Microsoft Internet Explorer [1], MSIE from now on), as a toolbar, and is responsible of managing relevant user data for the purchase, automating several tasks such as filling out order forms, or electronic payment by credit cards. This application contributes to the state of the art of this field with important advantages over previous systems like Gator [4] or RoboForm [5].

The automatic filling of web forms is achieved by a collaborative mechanism, by means of which the system learns how to fill out the forms of a particular shop by analysing the experience of previous users in such site.

The agent also integrates different mechanisms for secure electronic payment when using credit cards, such as generation of virtual card numbers, or the SPA/UCAF (Secure Payment Application / Universal Cardholder Authentication) protocol [2][3].

It is also important to remark that this system is in operation in several Spanish banks since the start of year 2003.

This paper is structured as follows. Section 2 shows the system architecture. Section 3 describes in detail the form filler service. In section 4 other available services are commented. Finally, section 5 establishes the conclusions and outlines our future work.

2. System architecture

To manage the necessary information of the users, the system makes use of a distributed architecture, based on a group of secure web servers. This offers two main advantages: first, it allows for a complete user mobility, and secondly, it minimizes the exposure of sensible data, thus avoiding several security risks associated to elements such as local file storage and taking into account others like browser cached information.

In particular the system consists of the following components (see Figure 1):

– User profile server. The services provided by the agent are strongly based on user-specific information. The system utilizes XML-encoded user profiles which contain a list of information categories for each user –e.g addresses, credit card information, etc- and also the list of visible services for the user. Each
information category contains a list of attributes with a value assigned to each of them. The user profile server stores this information for all the users.

- Form meta-information server. It stores meta-information used to automatically fill forms on behalf of the user (see section 3).
- Authentication server. Performs user authentication and system logout.
- Toolbar profile server, which stores different versions of the agent configuration files for different software distributions (see section 2.1). Each configuration file contains the list of all available services for a particular distribution and the list of those that will be visible before user authentication.
- The agent, installed on the client side, is currently implemented as a toolbar plug-in for MSIE.

These components interact as follows:

a) When a user starts an instance of the MSIE browser, the agent reads the local profile and creates a communication protocol with the toolbar profile server, indicating the versions of both the software and agent profile. If the server detects that there is a newer version, it sends an update back to the client.

b) The agent loads the profile and shows all the services described by it.

c) System authentication is accomplished through a web form—to do this the agent has a built-in service which opens a predefined URL on the authentication server.

d) After authentication, the browser receives a cookie which enables it to access the user and form profile servers. Besides, the authentication server sends a hidden frame to the browser that is detected by a special built-in service which causes the agent to refresh user’s personal data plus the personalized configuration of the agent. When data are received and stored in the browser’s session, the services indicated by the profile will be shown in the toolbar.

e) If the user profile includes the form filler service, the toolbar will ask the forms server about the filling meta-information of that user. Then the agent also will monitor the user actions to detect whenever they fill out a new form or modify the assignments to one that had been filled before; when this situation occurs the agent sends to the forms server the new meta-data.

f) The system logout is accomplished through a web service which deletes the browser cookie; the toolbar will then destroy the session information, and recover its initial look.

2.1. Agent architecture

The agent installed must have access to the objects which have been loaded into the user browser—pages, forms, form fields, and so on-, monitoring certain events produced on them, accessing and modifying some of their properties. Therefore the agent must be coupled to the browser, suggesting it should have a different implementation depending on the particular browser to be considered. In this first implementation we have chosen MSIE, due to its broad spread among the potential users of the system presented here. The type of plug-in selected has been a toolbar, which accomplishes both technical and interface requirements.

The agent is composed of a set of services which are parameterized instances of service types that can be developed independently.

This way, different software distributions of the agent can be generated declaratively by simply specifying the service types to include in each distribution, and the desired parameterized instances of them.

Besides, as we have already seen, the user profile can be used to personalize the set of services visible for a particular user, among those available in the distribution.
To add a new service type some interfaces must be implemented, and then the service type can be subscribed as observer of one or more event groups:

- The events which indicate the activation of a service.
- The event which indicates that the window where the toolbar resides, changes its size.
- The events defined when a user starts a drag&drop operation.
- The events representing navigation actions on the browser -start of navigation, document’s end of download, form submit, and so on.
- Etc.

2.2. Internationalization

Taking into account the need for internationalization, the installation process enables language selection, and also, allows for the user to change that selection after installation. Because of that, every single text or image is included in the toolbar profile in different languages.

3. Form filler

As we have already said, once the user has been authenticated, the agent will request the fill-out meta-information to the form server. This meta-information tells the agent how to automatically fill the list of forms known by the system in behalf of the user. The meta-information for each form assigns each field to one of the following elements:

- A constant value.
- An attribute from an information category of the user profile.
- A pattern to combine several attributes or portions of attributes from the user profile into one single composite value.

This way, when the user arrives to a page containing a form, they can ask the agent to automatically fill it on their behalf. If the form is included in the current meta-information then the agent fills it out. In other case, the agent makes visible an emergent second toolbar containing the user profile data, so they can drag the individual attributes into the form fields. In this case, the agent monitors the drag&drop actions to build new assignments between the form fields and the user profile attributes.

The agent also monitors the form submitting to detect whether the filled form is new, or it is a known form containing new or modified assignments. In any of these cases it is necessary to send updated information to the forms server.

3.1. Learning process

The system is able to use the form meta-information received from the agents to learn common ways of filling forms and offer them to other users, even if they had never accessed that particular page or form.

The forms server stores the meta-information for each user which is updated as shown in previous section.

The server has a process which runs periodically and analyses the meta-information of every user, to extract general assignments which could be considered valid for all the users of the system. In particular, when it is detected that a predefined number of users has filled out the same form, and a given percentage of those users has done it the same way, that particular set of assignments is added as general meta-information –valid for all every user. The assignment’s learning process has been implemented according to a set of interfaces, so that it can be redefined easily. More complicated techniques could be used, like, for example, taking into account advanced classification comparisons or granting different punctuation to the way different users fill their forms.

When the user is authenticated in the system, the form server sends to the agent the information of their particular assignments as generated in previous sessions, and the information about validated -global- associations derived by the server process explained above.

4. Other service types

This section briefly describes other service types currently available in the agent. Some of them are necessary to accomplish the functionality described in previous sections.

4.1. URL Service

It enables the visualization of a web address in the browser or in another window.

4.2. Virtual Card Service

Using this service, a client can ask for a virtual card, different from their real one, and different for each purchase, so that user’s real card information will not be exposed to the Internet, thus assuring security and confidentiality. Later on, the virtual card service manager will charge the purchase amount to the real card.

The Virtual Card service integrates into the toolbar with the external application in charge of virtual card generation. The service’s mission is to listen to every loaded page to detect a special frame with a specific name and content.
This service is integrated with the form filler in such a way that when filling payment information, it allows the user to create a virtual card and to use it.

4.3. Drag&Drop Service

Drag&Drop service allows for the user to manually fill out a web form in an easy and intuitive way, and it also enables the agent to identify the mapping relations between the user profile information and the web form elements.

4.4. SPA/UCAF Service

It implements a SPA/UCAF [2] wallet, that is, the element which connects the purchasing client with a SPA server, allowing for the agent to act as a wallet which supports that protocol. If the service identifies a UCAF page, it establishes new specific values over a particular form of the page, even submitting it automatically, if necessary. Later on, it will check that the user has been authenticated, and will open a modal window with relevant information about the trade about to happen.

If the user accepts the action, the service will send the AAV –Account Holder Authentication Value- to an external application, it will fill out automatically the UCAF fields with the response and the user selected data, and will submit the shopping form to the trader.

4.5. Selection List Service

This service allows for the selection of one of the elements which belongs to a concrete information category.

4.6. Compound Service

It is a container service represented as a special button, which, when clicked, unfolds a selection menu, where each option represents a concrete service.

4.7. Second Bar Service

It is a container service which is shown in the main toolbar as a button which, when clicked, shows/hides a second toolbar with a list of services.

5. Related Work, Conclusions and Future Work

This paper has introduced a shopping personal agent which turns the B2C purchase process into a friendlier, more secure and less error-prone experience. This system is currently in production phase, and is being used by real e-banking users of several Spanish saving banks.

The most similar systems to the one presented here are automatic form fillers such as Gator [4] or RoboForm [5]. Current work here provides a more general framework, thus achieving the following advantages:

- Secure and centralized storage of user-sensible information -the system architecture facilitates user’s mobility.
- The system incorporates learning capabilities, in such a way that a user who has never performed a trade in a particular e-shop can take advantage as well, because of the experience obtained from previous users.
- It is integrated in a generic platform which facilitates the inclusion of additional value-added services related to electronic payment, such as the already integrated SPA/UCAF and virtual card services.
- Due to its generality and extensibility, the system can contain new services, added in an efficient and convenient way, allowing to change its composition dynamically.

New lines of future work include:

- Migration of the agent to other containers –browsers, such as Netscape/Mozilla.
- The intrinsic capability of the system for monitoring all the user’s browsing actions, allows the application to obtain relevant information about a user’s behaviour that could make the addition of very powerful personalized agents a feasible task.

6. References