An Approach and Tool Support for Assisting Users to Fill-in Web Forms with Personal Information

Marco Winckler, Vicent Gaits
ICS-IRIT, University Paul Sabatier
118 route de Narbonne
31062 Toulouse, France
winckler@irit.fr, gaits@irit.fr

Dong-Bach Vo
Telecom ParisTech
46, rue Barrault
75013 Paris, France
dong-bach.vo@telecom-paristech.fr

Sergio Firmenich, Gustavo Rossi
Universidad Nacional de La Plata
La Plata, Argentina
sergio.firmenich@lifia.info.unlp.edu.ar
gustavo@lifia.info.unlp.edu.ar

ABSTRACT

Web forms are massively used as a very effective way for user interaction with information systems. Notwithstanding, filling in forms with personal data can be tedious and repetitive. Due to legal and technical constraints, full interoperability of information systems is not a straightforward solution. So that several client-side techniques have been developed in the last years to automate the task of filling in forms; for example, auto-filling and auto-complete are very well-known techniques that employ contextual information to fill in automatically Web forms. However, the accuracy of these techniques is limited by the contextual information available on the Web browser. Some information systems can record users' personal information on the server side and use them to provide pre-filled forms to returning users. The problem with such as an approach is that users must keep updated records of personal information in remote servers; legal and technical issues prevent from sharing personal data among different applications, thus users must maintain multiple accounts. Interestingly enough, the analysis of data requested in forms reveal a pattern in the set of pieces of personal information that are often required (e.g. names, affiliations, billing address, home address, bank account, etc). In this paper we propose a new approach for automating filling in form that relies on these patterns of personal information. Our ultimate goal is to provide means for supporting the exchange of data between user's Personal Information Management Systems (PIMS) and Web forms. The approach is supported by a tool called PIAFF (which stands for Personal Information Assistant for Filling Forms) and illustrated by a case study concerning forms used for student applications.

Categories and Subject Descriptors

H.5.3 [**Information Systems**]: Group and Organization Interfaces - Web-based interaction. H.5.2 [**Information Systems**]: User Interfaces - Graphical user interfaces (GUI), Interaction styles.

General Terms

Design, Documentation, Human Factors, Standardization.

Keywords

Web forms, personal information systems, Web-based interaction, form filling in interaction techniques, *Microformats*.

SIGDOC'11, October 3–5, 2011, Pisa, Italy.

1. INTRODUCTION

Currently, a large amount of data users provided to Web application is supplied through forms. Notwithstanding, filling in forms with personal data can be tedious and repetitive [4][6]. This problem becomes even more evident when they occur in a same application domain. For example, in the context of the project ANR PIMI we have carried out the analysis of 40 administrative Web forms that students have to fill in to accomplish 7 administrative applications along their undergraduate courses. The procedures are related to the student life including registration at undergraduate/graduate programs, residence booking inside the campus, scholarships requests, internships applications, student exchange programs such as ERASMUS. We have found out that the most frequent pieces information requested thought Web forms concern personal information (e.g. name, contact information, home address, bank account, etc).

Jones and Teevan (2007) [18] argue that grouping related information is a central personal information management systems (PIM) activity currently hindered by the artificial separation imposed by the different applications. Indeed, from the users' point of view it is not clear why they should provide the same information several times. The reason is that each form is used by an independent administrative unit (e.g. university, bank, student associations) that due to legal and/or technical constraints cannot share user's personal records. The problem is not easily solved even when administrative units are allowed to make their databases interoperable because internal organization will ultimately occur over time (e.g. data structure is updated, roles and hierarchy might change into the organization, access rights can be added/suppressed by new policies, etc) and constraint previous agreements for data exchange.

The example above is revealing of the complexity of making different information system interoperable. One might argue that the barriers inside of an application domain (e.g. university and interdependent organizations) could be removed at some point. However, even so, users will face similar problems with other Web applications (e.g. e-commerce, e-banking, e-government, social networks, etc.) that might also request personal information. Thus, in a general case, it is not possible to carry out a standardization of all exchanges between existing information systems [21][27].

In more recent years several techniques such as auto-completion and auto-filling emerged to assist users to fill in forms [14]. These techniques are implemented on client-site (i.e. Web browsers) and work independently of the back-end information system. Auto-complete [30] is very well-known technique implemented nowadays by many applications (not only Web browsers) that involves a program predicting a word or phrase that the user wants to type in without the user actually typing it in completely. Whilst useful in some situations, auto-complete is error-prone (prediction is not 100% accurate) and it is not directly connected with user's personal information. Auto-filling is driven by contextual information made available by the Web browser [7]; usually the browser can "remember" which values were entered by the users in a given form in a previous visit to the Web site. Several tools implement the Auto-filling strategy. For example, the plug-ins Google Toolbar Auto-Fill [12] and Auto-fill Forms Mozilla Firefox [2] are able to auto-fill forms with information previously recorded by users; however it requires users to login and record Web forms requesting personal data. Safari Web browser [26] implement by default an auto fill form mechanism that reuses previously filled form for automatically filling out different forms with data; however the matching is based on the field labels which may differ across Web applications.

All these tools provide undeniable help to users but they suffer of at least one of the following drawbacks:

- Limited data mapping: data exchange requires exact string matching of form fields (e.g. Safari auto fill);
- Lack of support for collecting data: users have to create their personal data record before filling in forms (e.g. Google ToolBar and Autofill forms);
- Dependency of a particular browser and/or plug-in: this is particularly tricky when users are not in front of the computer that contains their personal data;
- Little user's control on the automation process of filling in forms: currently available tool will automatically fill-in form fields with pre-defined information (e.g. address) but they do not allow users to chose a data set among those available in their information space (ex. home address, professional address, school addressee, billing address, …);
- No clear connection with personal information systems. Another particularly pervasive problem of personal information management is the information fragmentation, i.e., the fact that information related to a single task is often scattered across several different applications and environments.

In this paper we propose a new approach for automating fill in form with personal information. Our ultimate goal is to investigate data schemas and interaction techniques for supporting data exchanges between user's personal information space and applications based on Web forms. Section 2 presents our approach. Section 3 presents the tools we have developed to support the approach. These tools are duly illustrated by the means of user scenarios. Section 4 presents related work and section 5 presents conclusion and future work.

2. OUTLINE OF THE APPROACH

Our approach relies on following premises:

- A single pervasive information space for storing user's personal data is needed to prevent information fragmentation and for making information fully accessible to the users;
- Data interoperability is supported by recognized standards allowing data sharing between user's personal information space and Web forms;
- User should have full control on the diffusion of personal data space to Web forms. This means that users can choose which personal data forms will be filled with.

2.1 Distribution of the Information Space

In order to create a pervasive information space, our approach relies on a set of independent Web applications that are able to cooperate and execute independently of the Web browser as shown by Figure 1:

- The **Personal Information Space** is a piece of software whose main task is to store users' personal data. It also allows users to organize personal information in meaningful structures (e.g. address, bank account…). Such as personal information is deployed into a cloud computing platform thus ensuring that users' data is ubiquitously available;
- There is a set of **third-party Web applications** whose main interaction technique with user is based on Web forms; These application are not (necessarily) connected to each other but require similar information from the user's personal space;
- A **Personal Assistant for filling forms** is a piece of software that intermediates the communication with the users and ensure the interoperability of data between the personal information space and the Web applications.

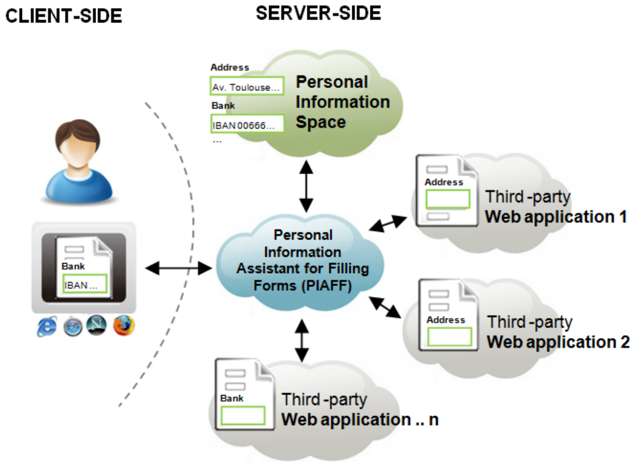

CLIENT-SIDE
SERVER-SIDE
Address
Av. Toulouse...
Bank
IBAN 00666...
...
Personal Information Space
Address
Third -party Web application 1
Bank
IBAN...
Personal Information Assistant for Filling Forms (PIAFF)
Address
Third -party Web application 2
Bank
Third -party Web application .. n

Figure 1. Overview of the approach for ubiquitous assistant for filling in Web forms.

2.2 Data interoperability

The Web forms provided by third-party applications may have different structure and inner organization. For example, form fields can have diverse labels such as city, town, locality, etc; an address can include (or not) the state and mailbox; and so on. So that a mapping process is required to ensure that data can be exchanged between applications. In our approach data operability is ensured by complying form fields with an emerging standard called *Microformats*, which can be defined as 'a set of simple, open data formats built upon existing and widely adopted standards' [19]. *Microformats* are used to describe data with semantic by the means of a markup language (e.g. semantic (X)HTML). This makes them easily readable by humans.

The Figure 2 shows the structure of the *Microformat* hcard. The tag *vcard* indicates the class of the *Microformat*; the *hcard* was originally proposed upon the standard vCard RFC 2426 (Card MIME Directory Profile) to identify individuals. The tag *fn* is used for full name and it is the only mandatory element. The tags *org, adr, street-address, locality, region, postal-code*, and *country-name* are some of other 29 optional tags can be used to identify a person.

```
<div class="vcard">
<span class="fn">Marilyn Monroe</span>
<div class="adr">
<div class="street-address"> Pennsylvania Avenue</div>
<span class="locality"> Brentwood</span>,
<span class="region">CA</span>,
<span class="postal-code"> 90049</span>
<span class="country-name">United States</span>
</div>
```

Figure 2. Excerpt of the *Microformat hcard*.

Microformats are defined in a similar way as design patterns, which mean that they have to be discovered rather than created. Moreover, *Microformats* must be agreed by a community of users. Currently there are only 9 stable, widely recognized *Microformats* such as *hCalendar* (for calendar-based events), *hCard* (for people contact) and *XFN* (for expressing human relationships using links). However, the community is very active and other 17 *Microformats* such as *tgeo* (for geographic coordinates), *hRecipe* (for cooking recipes) and *hResume* (for publishing CVs) are currently available as drafts. The interested reader can access the full list at http://Microformats.org/.

Data encoded using *Microformats* are suitable to be embedded into other data formats such as (X)HTML, Atom, RSS, and arbitrary XML. Currently, there are several online tools for encoding user data according to *Microformat* structures. There is also some plug-ins that can detect automatically the presence of data into Web pages encoded accordingly to *Microformats* [19]. The use of *Microformats* is spreading fast on the Web so that is not difficult to field Web page that are built over this pattern.

Current use of *Microformats* is often limited to flat text that is going to be published on Web pages. Our approach requires that *Microformats* should also be used to encoded form fields. In this context, we should consider two possible scenarios for third-party Web applications:

a) Forms fields are created according to *Microformats* structures;

b) Forms fields do not embed *Microformats*.

In the first case (a) Web forms already integrate *Microformats* so that forms can be used as such with the personal assistant for filling forms. In the second case (b) the original Web forms have to be annotated with *Microformats*. These scenarios are illustrated by Figure 3. The annotation is done by using an annotation tool and stored as external files into the personal assistant for filling forms. Therefore, the annotation process needs to be done only once. If known annotations for a Web form exist, they are added to the original Web form to produce a modified form featuring *Microformats*. Notice that Web forms are not modified on the third-party Web server making the solution independent of the Web form provider.

The annotation process can be done in many ways either manually coding the annotations into a text file or by using several annotation tools such as Greasemonkey scripts [11] and open annotation services [28][29]. We consider that annotation should be done by web developers, but the use of annotation tools require little training and can be mastered by experienced Web users. Moreover, the efforts of annotating are reduced by the collective effort. So that most of users will not need to annotate forms before using them. The process for modifying Web pages using external annotations is supported some tools such as [15] [16].

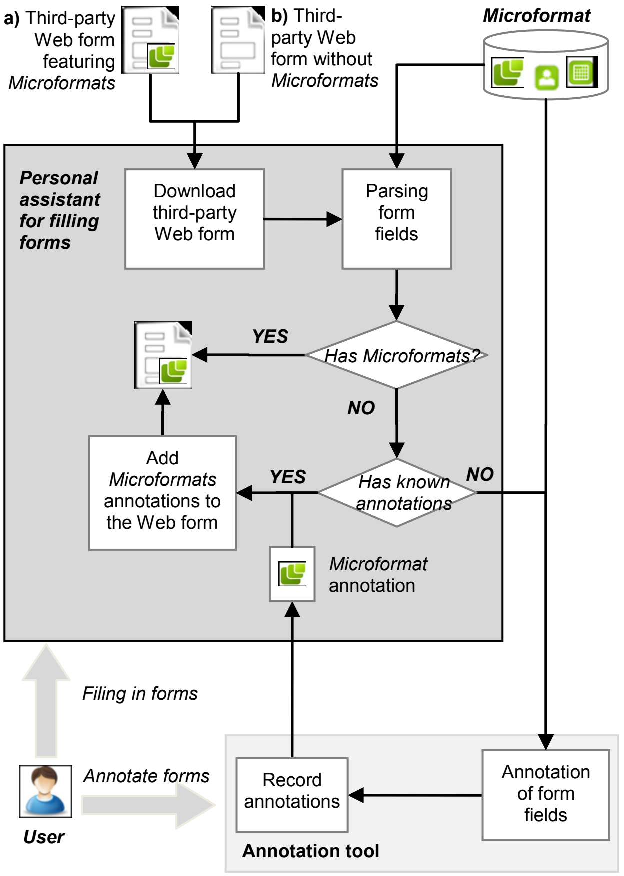

a) Third-party Web form featuring Microformats
b) Third-party Web form without Microformats
Microformat
Personal assistant for filling forms
Download third-party Web form
Parsing form fields
YES
Has Microformats?
NO
Add Microformats annotations to the Web form
YES
Has known annotations
NO
Microformat annotation
Filing in forms
Annotate forms
User
Record annotations
Annotation of form fields
Annotation tool

Figure 3. Embedding *Microformats into Web forms.*

2.3 User's Control

Most of available techniques will fill in forms with data without prompting users. As users do not know beforehand which data will be automatically entered, they have to cross-check all form fields. Wrong predictions of data put into form fields might cause frustration and ultimately reduce user performance. Moreover, users' confidence on the system may decrease if they do not fell in control of the data transfer.

In other to prevent these drawbacks, our approach defines user's control on automatic form fill in as follow:

- It should be possible to create as many records of personal information as needed, for example multiple personal addresses (e.g. home address, secondary house, etc);
- Users have to select the data going from their personal information space to Web forms;
- The personal information space can be update at any time;
- Users must be allowed to modify partially/totally the content of form fields even after the personal assistant has previously filled them in with data chose by users;
- Personal data have to be accessible from everywhere so that users can keep control of their personal data regardless the browser and/or the computer they are using.

3. TOOL SUPPORT AND CASE STUDY

In this section we present the tools we have developed to support our approach. Section 3.1 and 3.2 are concerned by the tools for filling in forms; these tools are aimed to be available for any Web users. Section 3.3 presents a tool for annotating Web forms; such as tool was develop to help developers (or at least experienced Web users) to annotated third-party Web forms that do not originally include *Microformats*.

3.1 Tools support for filling forms

We have developed two applications for supporting automatic fill in Web forms, as follows:

- A personal data server application, which is responsible for providing secure access to users' data;
- A client application tool called PIAFF which stands for Personal Information Assistant for Filling Forms.

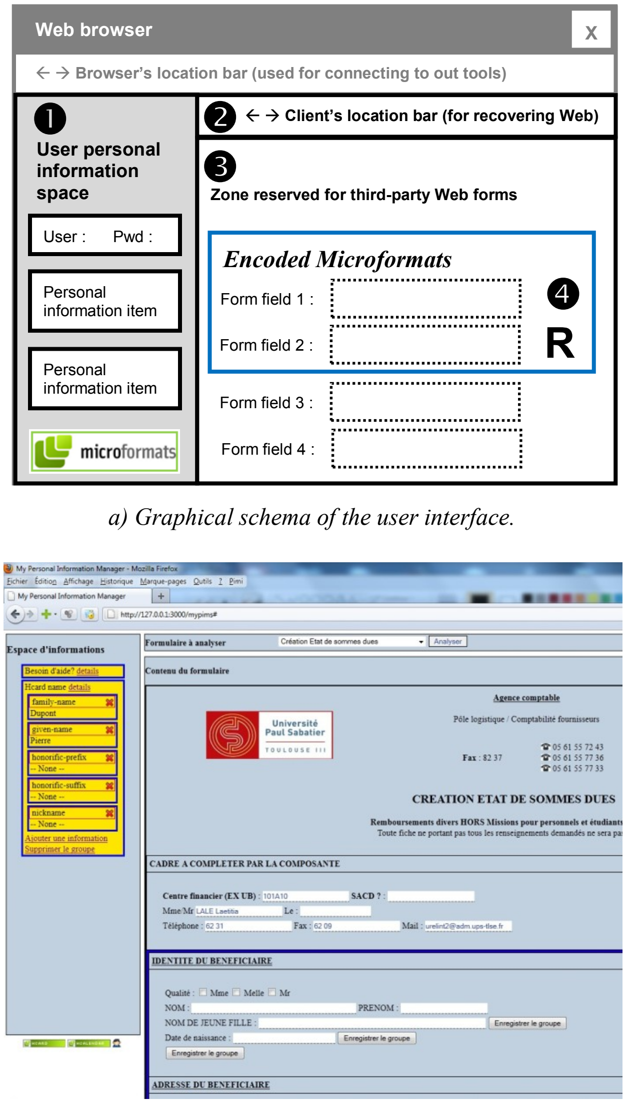
In order to ensure a pervasive access to users' personal data, the data server is hosted into a clouding computing network from where users can manage their personal records. Users can create their own information space, manage their personal records and make these records available on a Web page. Personal records are organized in the form of cards that can be pilled up and managed as digital posts-it. The inner structure of these cards is based on existing *Microformats*. The personal information server is implemented as a simple database secured by a login. So far, the personal data server is only accessible via PIAFF. However, in the future we are planning to deploy the personal data server as a standalone application.

The main goal for PIAFF is to provide users with means for transferring personal data from the personal information space to third-party Web forms. In order to accomplish this task, PIAFF implements a parser that is able to detect form fields encoded using *Microformats*. These features are implemented using AJAX [8] technology which is necessary to ensure that he can run in most currently available Web browsers. The communication between the client and the server application is implemented under an architecture REST [9] that allows the client to fetch data on the server using the protocol HTTP without having to refresh the page in the client. Under this architecture users can access and modify their personal records using any Web browser including those embedded into Smartphones.

PIAFF is organized in four graphical zones inside the Web browser as shown by Figure 4.a. The numbers indicate the ordering on which users are supposed to start using the tool. The zone ❶ provides support to login and it also is used to display the items of personal information, which are depicted as small cards. Then, the zone ❷ is used to provide the URL of the Web form that have to be filled in with personal data. The zone ❸ presents the Web form after client application has finished parsing it. The zone ❹ is reserved for user interaction with form fields encoded with *Microformats*. Figure 4.b features a screen shoot of PIAFF.

Our parser was conceived to modify the structure of original Web forms using a client-side Web adaptation technique [10]. When the parser detects a *Microformat* in a Web page, the corresponding fields are highlighted using a blue box (see zone ❹ at Figure 4); other fields are presented without change (e.g. form fields 3 and 4). Inside the blue box, i.e. zone ❹, a button **R** is added to allow used to *record* filled in data in Web forms into the personal information space.

The main purpose of the tool PIAFF is to allow users to automatically fill in the form with personal information. In order to accomplish this tasks, users have to select items of personal information available at the zone ❶ and move them to the zone ❸. PIAFF implements a drag & drop interaction techniques for moving items of personal information from a zone to another. This is only possible because the technology of Rich Internet Applications (RIAs) we are using, i.e. AJAX [8] and REST [9]. Using Drag & Drop, users manipulate entire blocks of personal information (e.g. home address encoded using *hcard*) instead of individual fields (e.g. street address). When the user releases the bloc of personal information item into a zone ❹ containing the corresponding *microformat*, the form fields are automatically filled in. The process is flexible enough to fill in forms with data available leaving the other form fields empty. After PIAFF has filled in the form fields, users can decide to modify these fields locally without changing the data stored in the personal information space.


Web browser
X
← → Browser's location bar (used for connecting to out tools)
1
User personal information space
User : Pwd :
Personal information item
Personal information item
microformats
2 ← → Client's location bar (for recovering Web)
3
Zone reserved for third-party Web forms
Encoded Microformats
Form field 1 :
Form field 2 :
4
R
Form field 3 :
Form field 4 :

a) Graphical schema of the user interface.

My Personal Information Manager - Mozilla Firefox

Fichier Édition Affichage Historique Marque-pages Outils ? Pimi

My Personal Information Manager

http://127.0.0.1:3000/mypims#

Espace d'informations

Besoin d'aide? details

Hcard name details

family-name
Dupont

given-name
Pierre

honorific-prefix
-- None --

honorific-suffix
-- None --

nickname
-- None --

Ajouter une information

Supprimer le groupe

Formulaire à analyser: Création Etat de sommes dues [Analyser]

Contenu du formulaire

Agence comptable

Pôle logistique / Comptabilité fournisseurs

Fax : 82 37 ☎ 05 61 55 72 43 ☎ 05 61 55 77 36 ☎ 05 61 55 77 33

CREATION ETAT DE SOMMES DUES

Remboursements divers HORS Missions pour personnels et étudiants

Toute fiche ne portant pas tous les renseignements demandés ne sera pa

CADRE A COMPLETER PAR LA COMPOSANTE

Centre financier (EX UB) : 101A10 SACD ? :

Mme/Mr LALE Laetitia Le :

Téléphone : 62 31 Fax : 62 09 Mail : urelint2@adm.ups-tlse.fr

IDENTITE DU BENEFICIAIRE

Qualité : □ Mme □ Melle □ Mr

NOM : PRENOM :

NOM DE JEUNE FILLE : [Enregistrer le groupe]

Date de naissance : [Enregistrer le groupe]

[Enregistrer le groupe]

ADRESSE DU BENEFICIAIRE

b) Screenshot of the application in a Web browser.

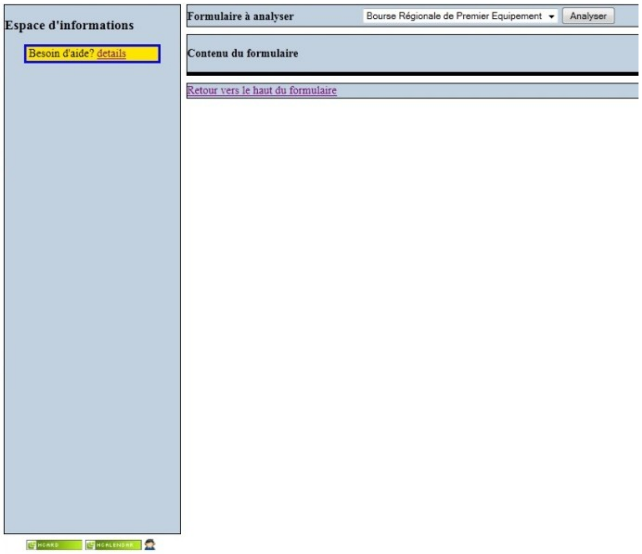
Figure 4. The organization of PIAFF inside of a Web browser.

3.2 Scenarios of filling in Web forms

The sequence of scenarios below concern a *personas* named *Pierre Dupont* who is a student applying for two scholarships. The main goal of Pierre is to fill in the Web forms necessary for scholarship applications. Our aim is to show how the tool PIAFF can assist Pierre to record personal information filled in Web forms, how to manage such as personal records and then reuse previously recorded information to fill in other Web forms.

3.2.1 Initializing PIAFF

We assume that our user Pierre never used PIAFF before. So at first he starts by visiting the PIAFF web site and create an account for his personal information space. After that, Pierre is presented with an empty information space as shown by Figure 5. Notice that on the top-right site the combo box which offer the list of forms available for scholarship applications. From this point user can populate their personal information space.



Espace d'informations

Besoin d'aide? details

Formulaire à analyser: Bourse Régionale de Premier Equipement

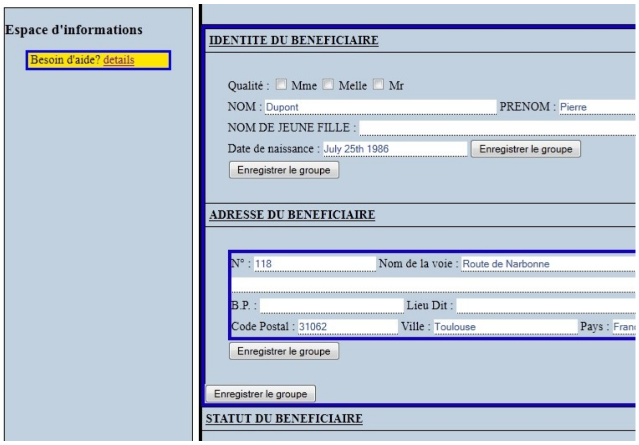
Contenu du formulaire

Figure 5. PIAFF after the creation of a user account.

3.2.2 Populating users' information space with data from Web forms

We assume that users would prefer to populate his personal space whilst filling in forms rather than to setup all information before starting using the tool. This scenario describes therefore how PIAFF assists users to record information into their personal information space with data from the filled in Web forms. Using a Web browser Pierre open the Web site of PIAFF, create an account and then he gets logged into it for his very first time. Once he is connected to PIAFF, he can open the Web sites containing the Web form required for scholarships applications. PIAFF parses the application form and identifies form fields that were encoded as *Microformats*.

As shown at Figure 6, these form fields are surrounded by a blue box; data inside the box can be manipulated as blocs of information rather than individual form fields. Inside each box there is a button named "*record information*" (i.e. **R**, see Figure 4.a) that can be used to transfer data from forms to the personal information space. Notice that several buttons (**R**) have been added to allow users to interact with different groups of information (i.e. contact, address, etc). An addition button allows users to record all information at once.



Espace d'informations

Besoin d'aide? details

IDENTITE DU BENEFICIAIRE

Qualité : □ Mme □ Melle □ Mr

NOM : Dupont PRENOM : Pierre

NOM DE JEUNE FILLE :

Date de naissance : July 25th 1986

ADRESSE DU BENEFICIAIRE

N° : 118 Nom de la voie : Route de Narbonne

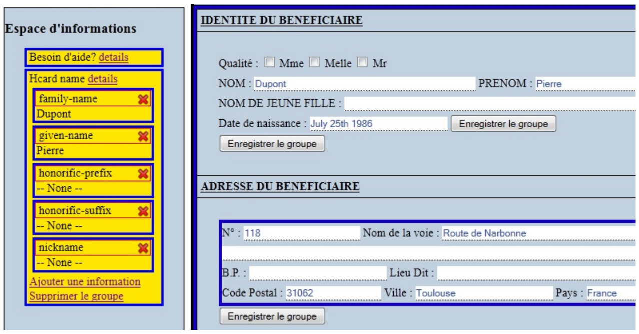
B.P. : Lieu Dit :

Code Postal : 31062 Ville : Toulouse Pays : Franc

STATUT DU BENEFICIAIRE

Figure 6. Presentation of the user's personal information space (at left) and the parsed Web form (right) highlighting form fields containing personal data by blue box.

As his personal information space is empty, Pierre should fill in the form manually. However once fields form have been populated, Pierre can click on the button next to the blue box to record this information into his information space. Recorded information appears in the information space as shown by Figure 7. By repeating this task, Pierre can gradually populate his information space. Notice that only fields containing valid values are recorded (i.e. *name/Nom, firstname/Prénom)*. These blocs of information can be completed and revised latter on. It is noteworthy that blocs of information are recorded as independent entities and they are not specific to a particular Web form. So that Pierre will be able to reuse these blocs of information in other Web forms.



Espace d'informations

Besoin d'aide? details

Hcard name details

family-name: Dupont

given-name: Pierre

honorific-prefix: -- None --

honorific-suffix: -- None --

nickname: -- None --

IDENTITE DU BENEFICIAIRE

Qualité : □ Mme □ Melle □ Mr

NOM : Dupont PRENOM : Pierre

NOM DE JEUNE FILLE :

Date de naissance : July 25th 1986

ADRESSE DU BENEFICIAIRE

N° : 118 Nom de la voie : Route de Narbonne

B.P. : Lieu Dit :

Code Postal : 31062 Ville : Toulouse Pays : France

Figure 7. User's personal information space presenting data recorded by the user (i.e. yellow box at left).

3.2.3 Using the information space to fill in forms

In this scenario Pierre applies for a complementary scholarship and for that he needs to fill in another application form. Some of the information requested in the application form is already available from his personal information space so that he starts by connecting to PIAFF and then chose corresponding application form. The Web form is parsed by PIAFF that highlights the form fields encoded with *Microformats*. Such form fields can be populated by performing a *Drag & Drop* operation from the information space (at left) the Web form (at right) as shown by Figure 8. During the *moving* action (preceding the *drop*) form fields that match to the *Microformat* are colored in green, so when the users *release* the data the form fields are automatically filled in with the corresponding information. Notice that users can modify the entries to form fields afterwards if needed.

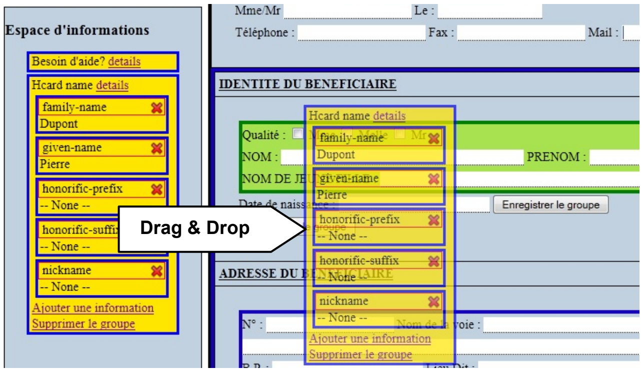
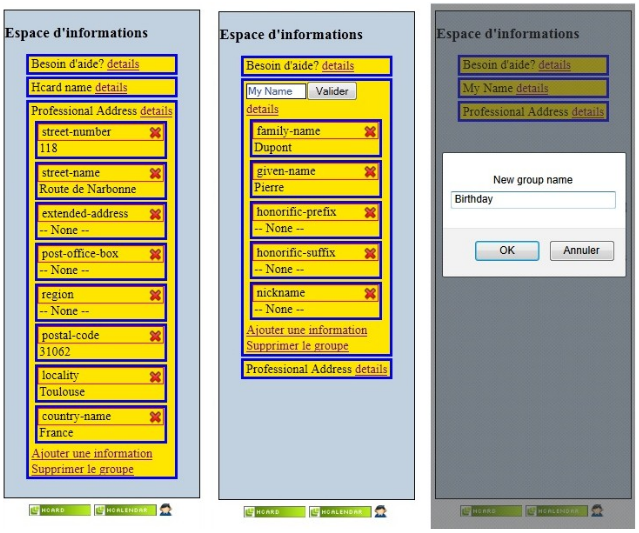


Figure 8. Use of *Drag & Drop* to populate Web forms (at right) from personal information space (at left).

3.2.4 *Managing the information space*

PIAFF allows the manipulation of personal information records as either in condensed or detailed view modes as shown Figure 9a. Condensed view presents just the label of the information (see *Hcard name* at Figure 9a) whilst detailed view show details and enable the edition mode (Figure 9b). Users can modify pre-filled information, add/remove new fields in the bloc, delete the entire bloc of information or just change its label. They can also create new records by selecting the green boxes of *Microformat* in the bottom part of the information space; after selecting the creation of a new record our user Pierre is prompted to give a name to it (see Figure 9c). In real world, users may have different records for similar information. So that, Pierre is allowed to create as many record of a same type as needed (ex. mama's birthday, dad's birthday …).



a) Visualization mode b) Edition mode c) creation mode

Figure 9. Visualization, edition and creation modes in PIAFF.

3.2.5 *Publishing the information space*

Most of information stored in personal information spaces is for private use only. However, users could decide to share non-sensible information in their own Web sites or personal networks. For that, PIAFF implements an additional feature that allows users to publish selected pieces of information. Once the user has selected their personal data, the tool can export personal either in the format of a publically available Web page or a piece of (X)HTML code that can be embedded by the users in own Web pages. It is noteworthy that the publication of the information space follows the structure of *Microformats*, which may facilitate its integration in order personal information space. For example, *Pierre* could visit his father's public personal space and the collect there the home address as a *Microformat* (e.g. *hcard)* that can be immediately integrated to Pierre's information space for filling in applications forms.

3.3 Annotation of Web forms

In order to allow the annotation of third-party Web forms that do not originally include *Microformats* we have developed a plug-in shown by Figure 10. Once activated, this tool highlight in red form fields that do not use *Microformats* and places next these fields a button (see the green cross). By clicking of this button, a pop-up dialog opens guiding the user along the annotation process. Annotations are stored in the PIAFF Web site and are made available for whoever requests the annotated third-party forms as shown by Figure 3.

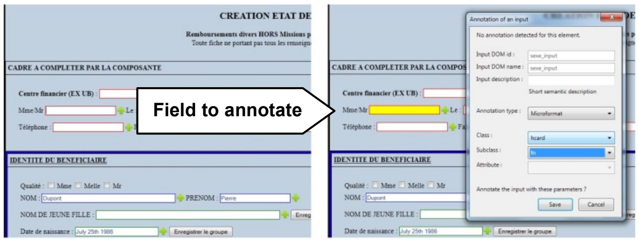


Figure 10. Annotation of Web forms with *Microformats*.

4. RELATED WORK

As presented in the introduction, currently available techniques implemented on the clients (i.e. for *Auto-complete* [30] and *Auto-filling* [7]) for automating that task of filling in Web forms are not perfect solutions. The diversity of structure and organization of Web forms is major constraint that still prevents the development of a seamless solution for automating filling in Web forms. The lack of standardization in data entry forms is not a new problem as pointed out by Sankar (1984) [27], however, it is still standing one. Some recent work [1][31] try to tackle the problem by using similarity functions to predict which personal information is expected for each form fields. Their results are very promising [1] but it is not always accurate. Moreover, such as prediction techniques fail to provide users with full control of their own data exchange; this issue might have an impact on trust and potential of user adoption of the final solution.

Other authors [4][33] investigate the use of Semantic Web technology for developing data bindings schemas. Data binding patterns are established techniques that help to connect user interface elements and data objects of applications [15]. The main drawback with such as techniques is that one needs an Ontology describing third-party applications before to perform the data integration. The creation of Ontology is time consuming and only can be afforded by large companies.

OpenID [24] technology allows users to provide certified identification and share information with trusted Web sites. Personal records can then be used to automatically fill in Web forms of trusted Web sites. One of the inconvenient of such as approaches is that it requires the agreement of third-party web sites to operate. Despite the fact OpenID has been around for some years, his use is still limited to a few specialized Web sites. Moreover, the reinforced user identification promoted to OpenID is not always a mandatory requirement for user interaction with most Web forms.

Instead on focusing on custom Ontology for particular Web applications, some binding schemas relies on the emergence of open standard data types such as *Microformats* [19] and *Microdata* [13]. *Microdata* is an under development standard of the World Wide Web Consortium whose aim is to integrate complex data as native types in XML-like technologies. The structure and underlying approach of *Microformats* and *Microdata* is pretty similar. However, *Microformats* have the advantage of an open community and already existing tools to support it.

We strongly believe on the emergence of open, standard, data formats giving more semantics to content available over the Web [17], which may ultimately include content provided via Web forms. The entire approach discussed in this paper relies, indeed, on the premise that *Microformats* can be used in personal data exchanges with Web forms. *Microformats* [19] are indeed a promising solution to the problem of data interoperability. The number of existing stable and widely agreed *Microformats* is still limited (9 so far), but several drafts are in discussion. More and more Web sites are using *Microformats* to enrich their data with semantics. We should say that due to the need of large agreement to define the characteristic of *Microformats,* they are more likely to cover only common persona data types (e.g. contacts, geographical position, addresses, etc). Notwithstanding, it is worth notice that most repetitive data exchange through Web forms concerns personal data for which *Microformats* are indeed available. So that, the universal use of *Microformats* to support interaction with all data possibly available in Web forms can be dismissed.

This work is also closely related to the emergence of Personal Information Management Systems (PIMS) [18]. PIMs studies have mostly focused on very large data sets, such has the whole content of a user hard drive, and therefore has mainly concentrated on search/ retrieval issues, with some findings about the great variability in which people search their own information (see for instance [3]). However, in more recent years some authors start to investigate the management of personal information over the Web [20][23][25]. For example, Norrie (2008) [23] proposes a complete architecture based on Web 2.0 technology enabling users to manage their personal records on the Web and synchronize them with other Web applications, in particular social networks. Notwithstanding, these efforts are mainly related to textual flat data and do not take into account interactive users tasks such as filling in forms. Our approach is another motivating example for promoting the development of pervasive PIMS [34].

The approach introduced in this paper also made use of client-side adaptation techniques for modifying third-party applications [10]. Indeed, the tool PIAFF add new *interactors* on third-party Web forms (i.e. highlight, new buttons and drag & drop interaction techniques) for supporting users tasks. The adaptation on the client of Web pages is an emerging topic of research. Our tool demonstrates that client-side adaptation is feasible from a technical point of view. As far as the adaptation occurs in the client-side, neither the information system hosted in the server-side or the Web forms it provides need to be changed to that our approach has virtually no impact on the sever-side. However, client-side adaption is not very known of the large public so that more research is required to investigate the effect of such as technology on the user experience.

5. CONCLUSIONS AND FUTURE WORK

More than a definitive solution to the problem of filling forms, this paper aims at raising questions of both scientific and practical significance such as the integration of information spaces in Web applications, the automation of common user's tasks with the Web applications, composition of distributed applications, the evolution of the Web technology allowing use of more advanced interaction techniques (e.g. drag & drop), the development of possible uses of emerging data patterns such as *Microformats*.

In this paper we introduce a new approach for assisting users to filling in Web forms with personal information. Our solution combines different existing technologies and proposes a new usage for them. The approach is validated by a prototype tool that proves the concept. The tool is fully operational and can be used as a demonstrator. However, despite a few beta-users that demonstrate a vivid interest on the tool [32], no proper usability testing has been organized yet. As part of our future work, we are planning to test the prototype with real users to investigate several dimensions that may have in impact on the efficiency, usability, trust, security on data transfer, felling of user control, and overall satisfaction.

We have found that currently available tools for annotation satisfy the basic technical requirement of encoding *Microformats* into Web forms. Notwithstanding, they still required a user with some basic programming skills. Currently we are working on the development of tools for making the annotation process suitable available for end-users. Indeed, the annotation process is the bottleneck of our approach as it is very unlikely that all organizations will create their Web forms using *Microformats*. We believe, however that some end-users would be keen to annotate the Web forms they use the most and they would like to share their annotations with other users.

6. ACKNOWLEDGMENTS

This work is supported by the ANR project MyCitizSpace and ANR PIMI.

7. REFERENCES

[1] Araujo, S., Gao, Q., Leonardi, E., Houben, G. J. 2010. Carbon: domain-independent automatic Web form filling. In Proceedings of the 10th international conference on Web engineering (ICWE'10), Boualem Benatallah, Fabio Casati, Gerti Kappel, and Gustavo Rossi (Eds.). Springer-Verlag, Berlin, Heidelberg, 292-306.

[2] Autofill Forms - Mozilla Firefox add-on. Available at: http://autofillforms.mozdev.org/

[3] Blanc-Brude, T., Scapin, D. L. : What do people recall about their documents? Implications for desktop search tools, in : IUI (10th Intelligent User Interfaces Conference), Honololu, HI, USA, January 28-31 2007. pp. 102-111) (2007).

[4] Bownik, L., Gorka, W., Piasecki, A. Assisted Form Filling. Engineering the Computer Science and IT. Safeeullah Soomro (ed.) Publisher: InTech, October 2009. ISBN 978-953-307-012-4

[5] Brabrand, C., Møller, A., Ricky, M., and Schwartzbach, M. I. 2000. PowerForms: Declarative clientside form field validation. World Wide Web 3, 4 (Dec. 2000), 205-214.

[6] Camenisch, J., Shelat, a., Sommer, D., and Zimmermann, R. 2006. Securing user inputs for the Web. In Proceedings of the Second ACM Workshop on Digital Identity Management (Alexandria, Virginia, USA, November 03 - 03, 2006).

[7] Chusho, T., Fujiwara, K., Minamitani, K. 2002. Automatic Filling in a Form by an Agent for Web Applications. In *Proceedings of the Ninth Asia-Pacific Software Engineering Conference* (APSEC '02). IEEE Computer Society, Washington, DC, USA, 239-.

[8] Crane, D., Pascarello, E., James, D. (2005) Ajax in Action. Manning Publications, 680 pages. ISBN-10: 1932394613

[9] Fielding, R. T. 2000 Architectural Styles and the Design of Network-Based Software Architectures. Doctoral Thesis, University of California, Irvine.

[10] Firmenich, S., Winckler, M., Rossi, G. A Framework for Concern-Sensitive, Client-Side Adaptation. In Proc. of International Conference on Web Engineering (ICWE 2011), Paphos, Cyprus, June 20-24, 2011. Springer, LNCS 6757, pages 198-213.

[11] Greasemonkey, At: http://www.greasespot.net/

[12] Google Toolbar Autofill. At: http://toolbar.google.com

[13] Hickson, I. HTML Microdata – W3C Working drafy (2011), http://www.w3.org/TR/microdata/

[14] Hartmann, M., Muhlhauser, M. 2009. Context-Aware Form Filling for Web Applications. In Proceedings of the 2009 IEEE International Conference on Semantic Computing (ICSC '09). IEEE Computer Society, Washington, DC, USA, 221-228.

[15] Heinrich, M., Gaedke, M. WebSoDa: A Tailored Data Binding Framework for Web Programmers Leveraging the WebSocket Protocol and HTML5 Microdata. In proc. of International Conference on Web Engineering (ICWE 2011), Paphos, Cyprus, June 2011, Springer, LNCS Vol. 6757, pp. 387-390, DOI: 10.1007/978-3-642-22233-7_32

[16] Hori, M., kondoh, G., Ono, K. 2000. Annotation-based Web content transcoding. In Proc. of the 9th int. World Wide Web conference. North-Holland Publishing Co., Amsterdam, The Netherlands, 197-211.

[17] Holzinger, W., Kruepl, B., Baumgartner, R. 2008. Exploiting semantic Web technologies to model Web form interactions. In Proceeding of the 17th international conference on World Wide Web (WWW '08). ACM, New York, NY, USA, 1145-1146.

[18] Jones, W., Teevan, J. Personal Information Management. University of Washington Press. Seattle, WA, USA. 2007. 334 p.

[19] Khare R., "Microformats: The Next (Small) Thing on the Semantic Web?," IEEE Internet Computing, vol. 10, no. 1, pp. 68-75, January/February, 2006.

[20] Leone, S., Grossniklaus, M., de Spindler, A., Norrie, M. C.- Synchronising Personal Data with Web 2.0 Data Sources. WISE 2010 411-418.

[21] McGuinness, D. L.,van Harmelen, F. (eds.) OWL Web Ontology Language. W3C Recommendation 10 February 2004. http://www.w3.org/TR/owl-features/

[22] Maret, P., Rubel, P., Beney, J. 1999. Multimedia Information Interchange: Web Forms Meet Data Servers. In Proceedings of the 1999 IEEE International Conference on Multimedia Computing and Systems - Volume 02 (ICMCS '99), Vol. 2. IEEE Computer Society, Washington, DC, USA.

[23] Norrie, M. C. PIM Meets Web 2.0. ER 2008 15-25.

[24] Recordon, D., Reed, D. 2006. OpenID 2.0: a platform for user-centric identity management. In *Proceedings of the second ACM workshop on Digital identity management* (DIM '06). ACM, New York, NY, USA, 11-16.

[25] Rukzio, E., Noda, C., De Luca, A., Hamard, J., Coskun, F. 2008. Automatic form filling on mobile devices. *Pervasive Mob. Comput.* 4, 2 (April 2008).

[26] Safari - Auto Fill: Personal Information. Available at: http://www.apple.com/safari

[27] Sankar, C. S. 1984. A method to simplify filling data entry forms. *SIGDOC Asterisk J. Comput. Doc.* 10, 4 (December 1984), 15-21.

[28] Signer, B. and Norrie, M.C. A Model and Architecture for Open Cross-Media Annotation and Link Services, Information Systems 36(3), Elsevier, May 2011

[29] Signer, B., Norrie, M. C. An Architecture for Open Cross-Media Annotation Services. WISE 2009. p. 387-400.

[30] Stocky, T., Faaborg, A., and Lieberman, H. 2004. A commonsense approach to predictive text entry. In CHI '04 Extended Abstracts on Human Factors in Computing Systems (Vienna, Austria, April 24 - 29, 2004). CHI '04. ACM, New York, NY, 1163-1166.

[31] Toda, G. A., Cortez, E., da Silva, A. S., de Moura, E. 2010. A probabilistic approach for automatically filling form-based Web interfaces. *Proc. VLDB Endow.* 4, 3 (December 2010), 151-160.

[32] Vo, D.-B., Winckler, M. 2009. PIAFF: un outil d'aide à la saisie d'informations personnelles pour les formulaires éléctroniques. In Proc. of the 21st International Conference on Association Francophone d'Interaction Homme-Machine (IHM '09). ACM, New York, NY, USA, 355-358.

[33] Wang, Y., Peng, T., Zuo, W., Li, R. 2009. Automatic Filling Forms of Deep Web Entries Based on Ontology. In *Proceedings of the 2009 International Conference on Web Information Systems and Mining* (WISM '09). IEEE Computer Society, Washington, DC, USA, 376-380.

[34] Zhou, D., Chander, A., Inamura, H. 2010. Optimizing user interaction for Web-based mobile tasks. In Proceedings of the 19th international conference on World wide Web (WWW '10). ACM, New York, NY, USA, 1333-1336.