Supporting Users Tasks with Personal Information Management and Web Forms Augmentation

Sergio Firmenich[1,2], Vincent Gaits[2], Silvia Gordillo[1,3],
Gustavo Rossi[1,2], and Marco Winckler[4]

[1] LIFIA, Facultad de Informática
[2] Universidad Nacional de La Plata and Conicet Argentina
[3] CiCPBA
[4] IRIT, Université Paul Sabatier, France
{sergio.firmenich,vgaits,gordillo,gustavo}@lifia.info.unlp.edu.ar,
winckler@irit.fr

Abstract. Currently, many tasks performed on the Web prompt users to provide personal information through forms. Despite the fact that most users are familiarized with this kind of interaction technique, the use of Web forms is not always straightforward. Indeed, some users might need assistance to understand labels and complex data format required to fill in form fields that, quite often, vary from a Web site to another even when requesting similar data. Filling in forms can be tedious and repetitive as many Web sites request similar information. In this work we analyze user's interactions with Web forms and propose an approach for enhancing Web forms using client-side adaptation techniques in order to assist users to fill in Web forms. As the use of Web forms is closely related to the management of personal information our approach includes the support for data exchange between user's personal information management systems (PIMs) and third-party Web forms. The approach is illustrated by a set of client-side adaptation tools and a pervasive Personal Information Management Systems called PIMI.

1 Introduction

For many Web applications, forms are essential components that allow users to provide data and interact with the system. Despite the fact that forms can be very effective for gathering information, the tasks users engage when filling in forms can be complex: at first, users must to *understand how to fill in the forms*; users should also be able to *recall information* that should be entered into form fields; only then users can start typing to *fill in the forms;* it is worth noting that quite often users need to *record data used in forms* in their personal information management systems (PIMs) for later use. If these users' tasks are not supported properly they might raise several usability problems. For example, users can make mistakes if they don't understand labels or if they do not know how to enter data in the expected data format [3]. Users can also left form fields blank if mandatory fields are not clearly indicated. Recalling information is cognitively demanding [14] and particularly painful if users should recall the information provided to every Web form. Whilst typing in form fields, users can introduce typos. Filling in Web forms can also be tedious and repetitive [5]; this is particularly true because, quite often, different Web forms will request similar personal information from users [25].

M. Brambilla, T. Tokuda, and R. Tolksdorf (Eds.): ICWE 2012, LNCS 7387, pp. 268–282, 2012.

In the last years, several strategies have been developed to assist users to fill in forms. On one hand there are techniques focused on the system design for example, guidelines aimed to provide advices for building usable Web forms [14]. Another example is third-party applications that include a user personal space on the server-side for providing pre-filled forms on the client-side.

On the other hand, some enhanced Web browsers and plug-ins implement techniques for assisting users to fill in forms such as auto-filling techniques [10] and auto-completion [22]. The auto-filling techniques [10] can "remember" which values were entered by the users in a given form in a previous visit to the Web site. Some Web browsers like Safari[1] implement by default an auto fill form mechanism that reuses previously filled form for automatically filling out different forms with data. However this will ultimately require users to login and record Web forms requesting personal data, which implies that users should remember their login information.

Despite the fact that all these solutions provide undeniable help, they suffer of at least one of the following limitations:

- We depend on the diligence of Web-browsers designers and developers of third-party Web applications;
- These solutions focus on a single problem, quite often limited to automating filling in forms and neglecting overall user tasks with Web forms;
- There is no integration between data available on users' personal information management systems and Web forms. Even if many Web applications can keep updated records of personal information in remote servers, there are legal and technical issues that prevent them from sharing personal data among different applications. Moreover, users must keep multiple accounts which increase the complexity of personal data management [15].

In order to assist users in all the tasks they engage whilst interacting with Web forms this paper proposes an approach based on Web forms augmentation to support a straightforward interaction between third-party Web forms and users' personal information management systems.

The underlying premise for the current work is that users will be more efficient whilst filling in forms if they were allowed to reuse data from their personal information management systems (PIMs). This work also assumes that some of the usability problems occur because users have to interact with different Web forms and it is virtually impossible to create a uniform presentation for all third-part Web forms available on the Web. In the following sections we describe the general approach and the tools that have been developed to solve these problems.

2 Task Analysis of User Interaction with Web Forms

2.1 Motivating Scenario

Some of the inner difficulties for interacting with Web forms only become evident when users have to accomplish the same tasks across different Web forms. To illustrate this, we present below a trip planning scenario which ultimately requires users to provide the same data on different third-party Web forms, as follows:

[1] Safari - Auto Fill: Personal Information, `http://www.apple.com/safari`

"John wants to bring his wife Judy to Berlin for her birthday, for that he is booking a flight, a hotel and a car at the following Web sites: expedia.com, booking.com and hertz.com. By doing so, John fills in three Web forms with information about the travel (i.e. dates, city of origin and destination) and the people travelling (i.e. name, personal address, billing address, credit card information, frequent flyer number, driver license, and passport number). Interesting enough, the three Web sites propose different names for form fields requesting similar information. John knows by heart his personal information but it does not go the same for Judy's passport information and frequent flyer number. Whilst filling in the forms, John is puzzled to see that the Web site could recall his credit card number and passport (he does not remember to have asked the web site to record such information in the past…). Moreover, the Web site recalls John's old passport number which John didn't realize immediately but he was very precautious to crosscheck it and change the information in time. After completing the booking, John has a last task to collect information for his travel (e.g. addresses of hotels, car rental…). John also takes note one of his many credit cards he used with the different Web sites."

Whilst performing the scenario above, it is possible to observe some issues that make the following users tasks difficult. As illustrated at Fig. 1, these tasks have several implications on the user side (concerning user requirements for accomplish the task), on the system side (impact on the design of Web forms), or both (the system implementation has an impact on users tasks).

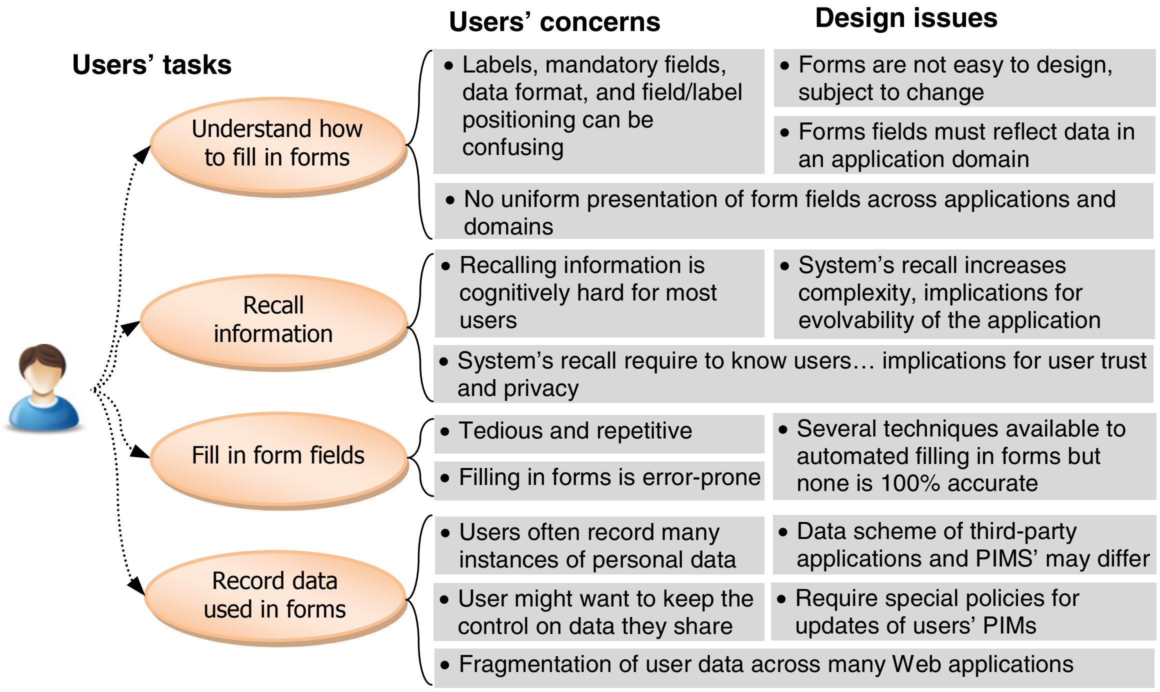

Users' concerns
Design issues
Users' tasks
Understand how to fill in forms
Labels, mandatory fields, data format, and field/label positioning can be confusing
Forms are not easy to design, subject to change
Forms fields must reflect data in an application domain
No uniform presentation of form fields across applications and domains
Recall information
Recalling information is cognitively hard for most users
System's recall increases complexity, implications for evolvability of the application
System's recall require to know users… implications for user trust and privacy
Fill in form fields
Tedious and repetitive
Filling in forms is error-prone
Several techniques available to automated filling in forms but none is 100% accurate
Record data used in forms
Users often record many instances of personal data
Data scheme of third-party applications and PIMS' may differ
User might want to keep the control on data they share
Require special policies for updates of users' PIMs
Fragmentation of user data across many Web applications

Fig. 1. Main issues for supporting common users' tasks with Web forms

2.2 Rational for Improving User Interaction with Web Forms

An overview in the current literature will point out several recommendations for solving each individual issue described at Fig. 1. Hereafter we present possible solutions for each task:

- Understand how to fill in forms
  - *Provide the same form structure through different Web sites;*
  - *Always describe the information demanded with standardized labels;*
  - *Provide a help page to explain the form;*
  - *Provide contextual help for filling every form field;*

- Recall information
  - *Store information in the browser/server-side to recall users of previously entered data on Web forms;*
  - *Provide users with a pervasive access to their personal space*
- Filling in forms
  - *Support user profiles to complete some data automatically;*
  - *Give to users full control over data used for filling in forms;*
  - *Manage a PIM and establishing semantic correspondence with forms.*
- Record data used in forms
  - *Allow users to record into their personal space the data provide to Web forms;*
  - *Allow users to collect information in Web sites and make it available across the Web at client-side.*

It is interesting to notice that existing solutions will focus on a single issue at a time. Moreover, some solutions such as labels standardization among the Web seem technically possible but unrealistic in practice as application domains might vary and Web developers are usually "creative" in their designs. Indeed, existing solutions will present two main drawbacks: dependency on the diligence of designers of Web applications and/or lack of support for integrating personal information space and Web forms. In order to solve these problems we propose below some alternative solutions:

- *Allow users community to describe the form entries and show these descriptions in the form page.*
- *Provide a pervasive personal information space with seamless integration with Web forms. This solution could help users to recall their personal information that could also be used for filling in forms.*
- *Allow users to collect information in Web sites and make it available across the Web at client-side.*

3 Outline of the Approach

Our approach relies in a distributed architecture encompassing a set of independent Web applications that can be combined as shown by Fig. 2. The architecture is conceived to support user interaction with Web forms provided by third-party applications. The personal information space is a piece of software whose main task is to store users' personal data (e.g. address, bank account…). The personal information space is deployed into a Web server, rather than on the Web browser, to provide users with a pervasive access to their personal data. The third element in this architecture is the personal assistant for filling forms, which is a piece of software implementing client-side adaptation techniques for supporting Web form augmentation and ensuring data interoperability between the personal information space and third-party Web applications. The main principles that underlying the approach are: i) third-party Web form augmentation through the use of client-side adaptation techniques; ii) the availability of a pervasive personal information management system; iii) the development of an annotation service for supporting data interoperability between third-party Web forms and user personal information systems. The rest of this section describes the rationale behind every element of this architecture.

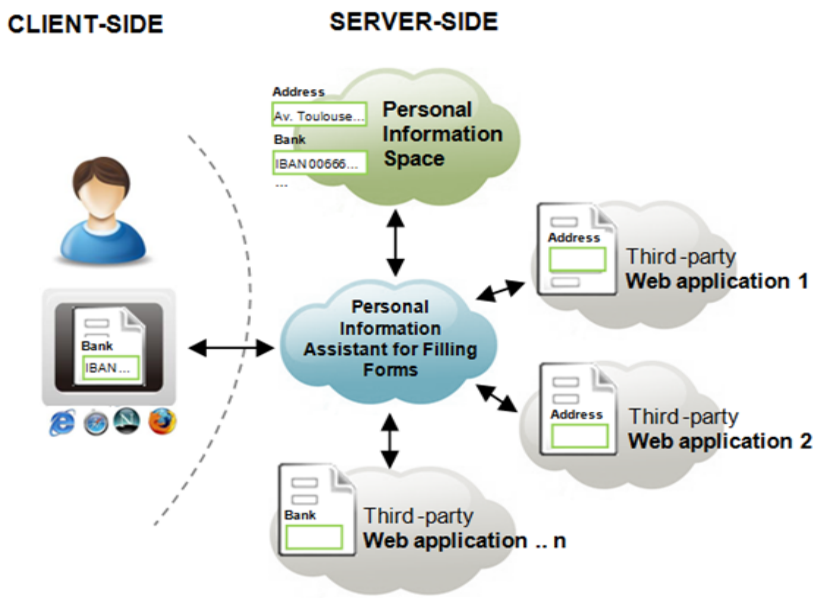

CLIENT-SIDE
SERVER-SIDE
Address
Av. Toulouse...
Bank
IBAN 00666...
...
Personal Information Space
Bank
IBAN...
Personal Information Assistant for Filling Forms
Address
Third-party Web application 1
Address
Third-party Web application 2
Bank
Third-party Web application .. n

Fig. 2. Overview of architecture of our approach for Web form augmentation

3.1 Web Form Augmentation

Previous works [4][7] have demonstrated that client-side adaptation techniques are an effective way to empower users in order to adapt Web sites according with their concern. For example, client-side adaptation techniques may be used to create small software packages called *augmenters* [7] that can modify, in the Web browser, the content of third-party Web applications. *Augmenters* are built as generic adaptations featuring behaviors such as adding user defined links to Web pages, highlighting text, etc. Technically speaking, an *augmenter* is just a script (written in JavaScript) that is able to modify the DOM. *Augmenters* are developed using the CSN framework [7] and then installed in the browser as a plug-in. Whilst navigating the Web, end-users can use the framework to trigger *augmenters* whenever they are needed to improve the user interaction. In our previous work [7] we have explained how to use the CSN framework to build not only individual *augmenters* for supporting opportunistic user tasks but also how to combine different *augmenters* to supporting complex scenarios.

The present work borrows the basic infrastructure provided by the CSN framework to build client-side adaptations and it extends it by describing a set of useful *augmenters* supporting Web form augmentation (i.e. *augmenters* aimed to help users to interact with Web forms). These augmenters can be triggered either by the users (i.e. manually) or programmed to execute automatically under certain circumstances. In this way, users may perform adaptations under demand, but the same adaptations may be executed automatically by the tool (a Web browser plug-in).

3.2 A Pervasive Personal Information Management System

The second main element in our approach is a pervasive information space for storing user's personal data. Managing large sets of information is strongly related to the domain of Personal Information Management systems (PIMs), which corresponds [15] to the research field addressing the way people manage their physical documents (books, notebooks, sheets, etc.), as well as their electronic documents (files, emails, Web pages, etc.), with the aim of designing tools that support the management of electronic documents (PIM tools). While the PIMs area usually covers many contexts and activities, in this paper we look at PIMs in a more specific way: the individual information items people keep on various notes, cards, forms, agendas, etc., the ones that user are mostly have to use when interacting with Web forms.

Our approach relies on a Personal Information Management system (PIMs) as a tool allowing users to persist information which may be used when they it is needed. In this way, users may have always relevant information for performing tasks, specifically those which require forms. The pervasive aspect for the PIMs is aimed to reduce information fragmentation and for making information fully accessible to the users.

3.3 Data Interoperability through Web Forms Annotation

The Web forms provided by third-party applications might have different structure and inner organization. For example, form fields can have diverse names such as city, town, locality, etc; an address can include (or not) the state and mailbox; and so on. Therefore, 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 such as Microformats [17] and Microdata [12]. Our approach can support diverse data formats, but for the purposes of the illustration, the current implementation is built upon Microformats.

The choice for Microformats is motivated by the fact that they can embedded into other data formats such as (X)HTML, Atom, RSS, and arbitrary XML. Moreover, there are several plug-ins that can detect automatically the presence of data into Web pages encoded accordingly to Microformats. Fig. 3 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>
```

Fig. 3. Excerpt of the Microformats *hcard*

In our approach, Microformats are used as a kind of *lingua franca* that supports data exchange between the different elements described in Fig. 2. The use of Microformats is spreading fast on the Web, but not all Web pages are built using Microformats. Thus, we have to face two possible scenarios: a) forms fields were built using 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 must be annotated with Microformats.

Fig. 4 illustrates how the approach supports the detection of Microformats in Web pages and the corresponding annotation of third-party Web forms with Microformats. It is important to notice that annotations are necessary to make Web forms compatible with the users' data stored into their personal information. Annotations are stored as external files in a dedicated database. 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.

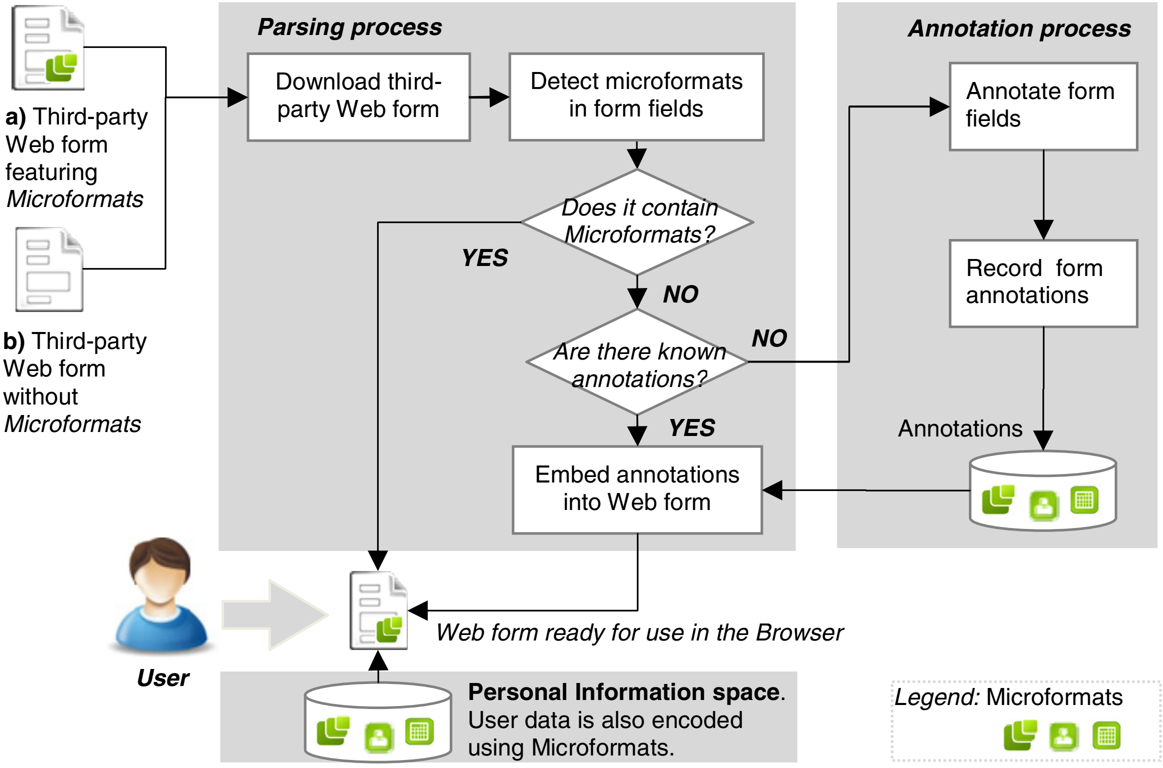

a) Third-party Web form featuring *Microformats*
b) Third-party Web form without *Microformats*
Parsing process
Download third-party Web form
Detect microformats in form fields
Does it contain Microformats?
YES
NO
Are there known annotations?
NO
YES
Embed annotations into Web form
Annotation process
Annotate form fields
Record form annotations
Annotations
User
Web form ready for use in the Browser
Personal Information space. User data is also encoded using Microformats.
Legend: Microformats

Fig. 4. Annotation process for supporting interoperability between third-party Web forms and user's personal information space using Microformats

Annotations 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[2] and open annotation services [21]. We consider that annotation should be done by web developers, but since the use of annotation tools require little training they can be also mastered by experienced Web users. The efforts of annotating are reduced by the collective effort. Therefore most users will not need to annotate forms before using them. The process for modifying Web pages using external annotations is supported by some tools such as [11][13]. Despite the fact that we assume that annotation of forms could be easily performed by advanced Web users, we do not exclude the possibility that the annotation process could be automated. Indeed, tools such as Carbon [1] and Opal [9] are able to analyze Web forms with respect to their fields and labels. For example, Opal [9] interprets form labels and classifies the fields according to a given domain Ontology. This classification mechanism could be exploited to support automated annotation of Web forms.

[2] Greasemonkey, `http://www.greasespot.net/`

3.4 User Control on Data Transfer

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 must 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 might decrease if they do not fell in control of the data transfer. This aspect of the user interaction has become an essential aspect for the design of new applications [16]. In other to prevent these drawbacks, our approach defines user's control on automatic form fill in as follows:

- Users should be able to create as many records of personal information as needed, (e.g. home address, secondary house, etc);
- Users should be able to control data flowing from/to the personal information space from/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 must be accessible from everywhere so that users can keep control of their personal data regardless the browser and/or the computer they are using.

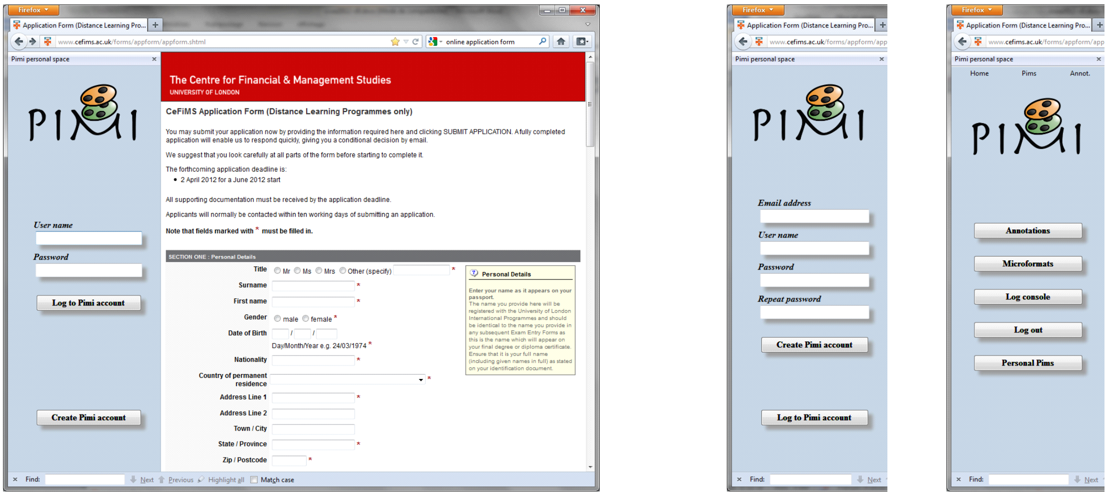
4 Case Study and Tool Support

In order to demonstrate the feasibility of the approach we have developed a set of tools including a ***personal information space***, a ***personal assistant for filling forms*** that provides support for Web form augmentation and an ***annotation service***. Actually these tools have been combined in a single user interface called PIMI, which was built upon the CSN framework [7]. As shown in Fig. 5, PIMI is delivered as a Firefox extension that, when activated, appears at the left-hand side (see Fig. 5.a) of the Web browser whilst users are navigating on the Web at the right-hand side (see Fig. 5.b). Fig. 5.a, Fig. 5.c and Fig. 5.d shows the corresponding screenshots when the user can log into PIMI (a), create a new account (b) and access to the main menu that give access to the tools for supporting users in filling forms while they perform their tasks.

The rest of this section is organized as follow: section 4.1 presents how user can use PIMI to manage their personal information; section 4.2 illustrates how PIMI supports automated filling in forms; section 4.3 describes the annotation service for making new forms interoperable with the user personal information space.

4.1 Personal Information Space

One of the main features of our approach is that users should be able to connect their personal information system with third-party Web forms. In order to ensure a pervasive access to users' personal data, PIMI includes a data server from where users can manage their personal records. Such personal information space can be seen as a standalone application. However, PIMI integrates data from user's personal spaces for allowing them to reuse such data whilst they are also navigating third-party applications as shown by Fig. 6.



The Centre for Financial & Management Studies
UNIVERSITY OF LONDON

CeFiMS Application Form (Distance Learning Programmes only)

You may submit your application now by providing the information required here and clicking SUBMIT APPLICATION. A fully completed application will enable us to respond quickly, giving you a conditional decision by email.

We suggest that you look carefully at all parts of the form before starting to complete it.

The forthcoming application deadline is:

- 2 April 2012 for a June 2012 start

All supporting documentation must be received by the application deadline.

Applicants will normally be contacted within ten working days of submitting an application.

Note that fields marked with * must be filled in.

SECTION ONE : Personal Details

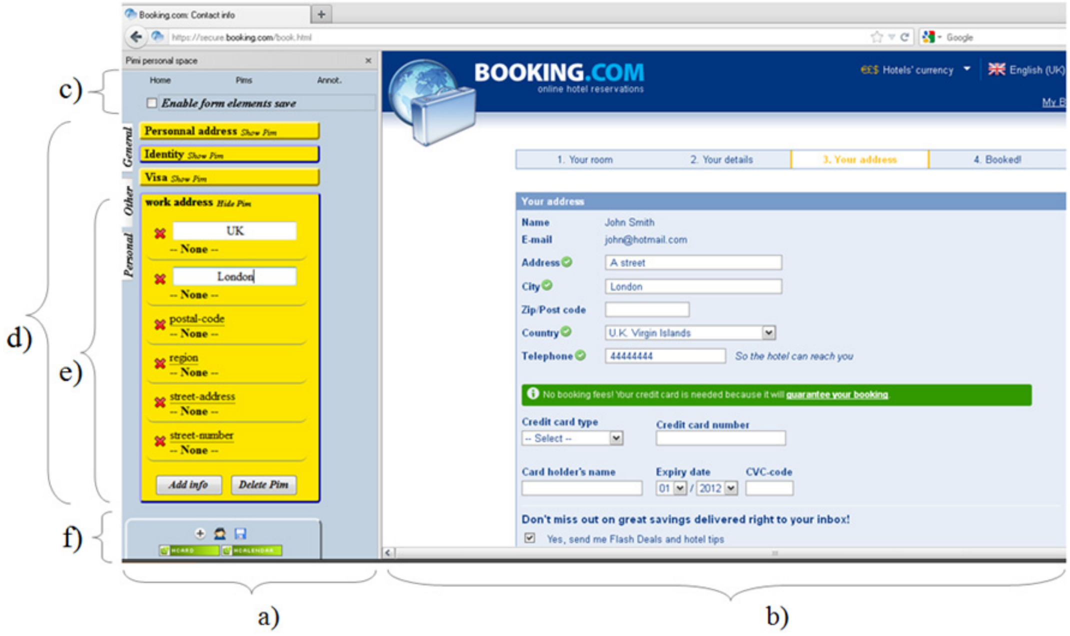
Title ○ Mr ○ Ms ○ Mrs ○ Other (specify) *
Surname *
First name *
Gender ○ male ○ female *
Date of Birth / / Day/Month/Year e.g. 24/03/1974 *
Nationality *
Country of permanent residence *
Address Line 1 *
Address Line 2
Town / City
State / Province *
Zip / Postcode *

Personal Details

Enter your name as it appears on your passport. The name you provide here will be registered with the University of London International Programmes and should be identical to the name you provide in any subsequent Exam Entry Forms as this is the name which will appear on your final degree or diploma certificate. Ensure that it is your full name (including given names in full) as stated on your identification document.

| a) PIMI login |… b) third-party web forms …………| |c) new account| |d) main menu |

Fig. 5. Views PIMI inside the Web browser whilst navigating the Web



□ Enable form elements save

work address: UK -- None --; London -- None --; postal-code -- None --; region -- None --; street-address -- None --; street-number -- None --

Your address

Name John Smith
E-mail john@hotmail.com
Address A street
City London
Zip/Post code
Country U.K. Virgin Islands
Telephone 44444444 So the hotel can reach you

No booking fees! Your credit card is needed because it will guarantee your booking.

Credit card type -- Select --
Credit card number
Card holder's name
Expiry date 01 / 2012
CVC-code

Don't miss out on great savings delivered right to your inbox!

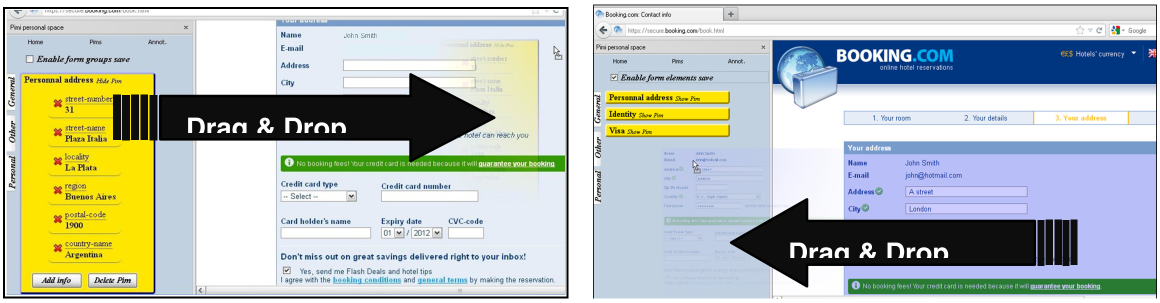
☑ Yes, send me Flash Deals and hotel tips

Fig. 6. Overview of the personal information space at PIMI: *a) PIMI; b) third-party application; c) tools menu; d) personal records items; e) expanded items; f) items menu*

Once connected to his account, the user can see his personal information space at the left-hand side of the Web browser (Fig. 6.a) and the third-party web form at the right-hand side (Fig. 6.b). The personal records are organized in the form of cards that can be piled up and managed as electronic posts-it containing a simple label (see Fig. 6.d). Each personal record can be expanded to show all detailed information; for instance, Fig. 6.e shows all the fields associated to the label *work address*. The current structure of these cards is based on existing *Microformats* and only accessible from PIMI. However, it would be possible to extend it to use other data format. Fig. 6.f shows the menu for creating new personal records using existing Microformats elements. On the top (Fig. 6.c) there is a menu for allowing navigation to the other tools delivered with PIMI including the navigation to the main menu (as shown at Fig. 5d), the personal assistant to filling forms (see section 4.4) and the annotation service (see section 4.5).

4.2 Personal Assistant for Filling Forms

The personal assistant for filling forms is a tool that implements a set of client-side adaptation techniques that can modify third-party Web applications and thus making them compatible with the user's personal information space. One of the most interesting features of this tool is the automated fill in of Web forms that is supported by a simple "drag and drop" (D&D) of information items as shown by Fig. 7.a. The tool is able to detect if the Web page embeds Microformats that can be used to support the exact mapping between personal user data and the form fields. Such D&D is possible because both input and property have the same Microformat. Notice that users do not need to move every field (e.g. *street number*), but instead they manipulate the whole block of information (i.e. *personal address*). The entire set of data is copied into form fields whenever there is a correspondence between the data type. If form fields already contain data, the D&D action will replace it with the data from the personal information space. The opposite operation is also possible so that users can also populate their personal information space by performing a D&D from a Web form towards the personal information space. For that, the user just needs to enable the option "*Enabling form elements save*" on the top menu before performing the D&D.


Pimi personal space

Home Pims Annot.

□ *Enable form groups save*

Personnal address *Hide Pim*

street-number 31

street-name Plaza Italia

locality La Plata

region Buenos Aires

postal-code 1900

country-name Argentina

Name John Smith

E-mail

Address

City

No booking fees! Your credit card is needed because it will guarantee your booking.

Credit card type -- Select --

Credit card number

Card holder's name

Expiry date 01 / 2012

CVC-code

Don't miss out on great savings delivered right to your inbox!

☑ Yes, send me Flash Deals and hotel tips

I agree with the booking conditions and general terms by making the reservation.

Booking.com: Contact info

https://secure.booking.com/book.html

Pimi personal space

Home Pims Annot.

☑ *Enable form elements save*

Personnal address *Show Pim*

Identity *Show Pim*

Visa *Show Pim*

1. Your room 2. Your details 3. Your address

Your address

Name John Smith

E-mail john@hotmail.com

Address A street

City London

No booking fees! Your credit card is needed because it will guarantee your booking.

a) from user personal space towards Web form b) Web form towards personal information space

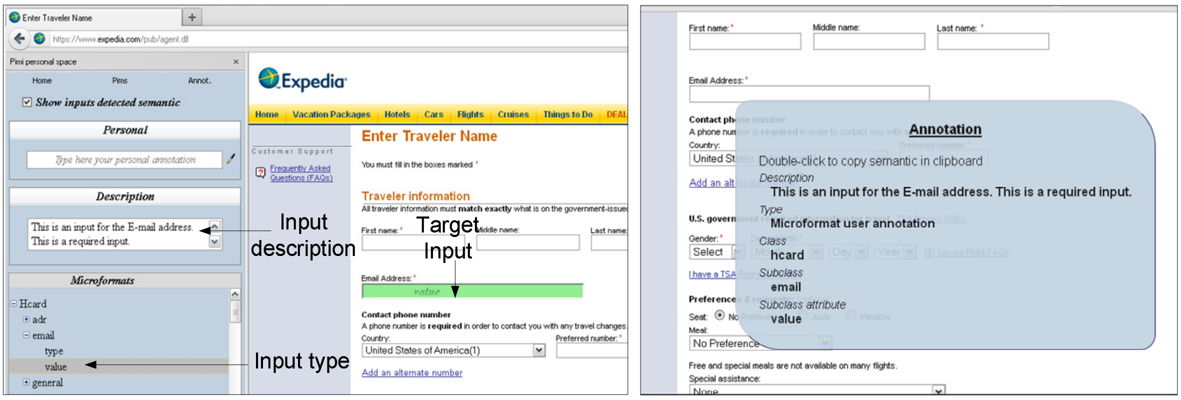
Fig. 7. Personal assistant for filling forms using D&D

The personal assistant for filling form implements that *parsing* process already described in Fig. 4. At first it tries to detect if the third-party Web forms have been built with Microformats, or if there are known annotations for the Web form that would make the Web form compatible with Microformats. Otherwise the users should annotate the Web form before being able to perform D&D operations.

4.3 Tool Support for Annotation of Web Forms

In our approach, annotations are a key step for implementing Web forms augmentation. On one hand, annotations are needed for providing users with contextual help to labels and thus help them to fill in forms. On the other hand, annotations can be used to make third-party Web forms interoperable with the user's personal information. Our tools support two kinds of annotations: a) semantic annotation of form fields to make them compliant with Microformats; b) textual annotations that can be used as contextual help.

Fig. 8 illustrates the annotation process of the Web form available at the Web site Expedia.com. By navigating on the top menu it is possible to reach the annotation tab that basically shows a list of existing Microformats. So far the current Microformats are supported: *Hcalendar, Hcard, Hreview, Xoxo, Haddress, Hbank, Hcontct, Hidentity,* *Hlog*. To annotate a form field, the only thing the user should do is to select an ***input type*** of a Microformats (for example the property *value* of the field *email* embedded into Microformats *Hcard*; Fig. 8.a) and then perform a D&D to the ***target input*** in the Web form. When doing so, the target form field becomes green to show that it has been annotated (see at Fig. 8.a, ***target input***). Whilst the form field is selected, users can also add a description to it using the field ***input description***. This operation should be repeated in all fields that require annotations. The tool will automatically record these annotations in a dedicated database. Once annotated, the event *mouse over* will trigger the contextual help embedding the user-defined annotations as a virtual post-it as shown in Fig. 8.b. Web forms annotated in this way will became semantically compatible with Microformats and thus enabling integration with data from the user personal space (as described in section 4.2).



This is an input for the E-mail address. This is a required input.

Double-click to copy semantic in clipboard
Description
This is an input for the E-mail address. This is a required input.
Type
Microformat user annotation
Class
hcard
Subclass
email
Subclass attribute
value

a) Annotation of form fields using Microformats *b) Contextual help provided by annotations*

Fig. 8. Overview of form annotations in Expedia.com

In the current implementation, annotations performed by a user will automatically become available to users visiting the same Web forms, provided that other users also have a PIMI account. This feature is aimed to support a crowdsourcing approach for annotations of Web forms so that users create new annotations that are shared with the community and then any other user can profit from the new annotations.

4.4 Evaluation of Tool Support

The tools presented above are full operational and have been used with a large set of third-party Web forms.

In order to highlight the contributions of these tools, we present in this section a quantitative assessment of user tasks with Web forms accordingly to the GOMS-Keystroke (KLM) model [6]. GOMS ("Goals, Operators, Methods, Selection rules") is a cognitive formal model used to rigorously evaluate how efficiently a trained person can interact with a given software system or program. GOMS is a human information processing model that is built upon a detailed sequence of users operations with a system. The GOMS-Keystroke (KLM) is a variant of that model which includes values to known users' actions; so that it is possible to predict the time skilled users will spend in seemingly unpredictable situations. For example, the average time to perform the action *reach for mouse is of 0.40 seconds, click on a field is of 0.20 seconds, etc.* Thus by providing a detailed scenario of user actions with tools including low-level user actions, it is possible to use GOMS-KLM to predict performance of computers (e.g., speed).

Due to the limited space, Table 1 presents a summary of the results obtained by applying the GOMS-KLM over some tasks identified in the scenario "planning a travel to Berlin" described in section 2. The last line in the table provides the estimated time for the full scenario. We compare tasks performed using PIMI and without PIMI. As we shall see, the estimated time for some tasks in the scenario such as "*Search Flights"* and *"Selecting Flights"* are the same. However users will save time when reusing data that is already on the personal information space as the D&D is faster than typing every word in the form field. At Table 1 we can also see that some new tasks will appear when using PIMI as users will be requested to *create an account* and eventually populate the information space with data such as *personal contact* and *credit card*. It also includes the estimated time for annotating third-party form fields using PIMI; ex. annotating fields related to credit cards is estimated to 15 sec. Whilst users spend some time populating the personal information space, they will save time reusing that information in future operations.

Table 1. Estimated performance of some tasks accordingly to GOMS-KLM model

Tasks	Without PIMI	With PIMI
Search Flights in Expedia.com	6.3 sec	6.3 sec
Selecting Flights in Expedia.com	32 sec	32 sec
Filling in form with passenger information in Expedia.com	26 sec	9.9 sec
Filling in form with credit card information in Expedia.com	67 sec	11.9 sec
Create an account in PIMI	-	27.6 sec
Add personal contact into personal information space	-	27.5 sec
Add credit card information into personal information space	-	41 sec
Annotated form fields related to credit card (bank, account number…)	-	15 sec
Total task: travel planning buying flights tickets (expedia.com), booking accommodation (Booking.com)	**240.6 sec**	**116.8 sec**

Note that, besides these quantitative time estimations, the approach and the tool support help to add overcome limitations of existing forms with respect to the lack of contextual help and seamless management of personal information.

5 Discussion and Related Work

The approach presented in this paper opens questions of practical and theoretical significance including: What makes the task of filling forms so difficult for users? Which functions could be implemented to provide better support for helping users to fill in Web forms? How to support a better integration between Web forms and personal information management? Which is the best standard for describing personal user data? How to support data interoperability between different third-party web forms? How to make third-part Web forms interoperable?

The study of interaction techniques for improving the user interaction with Web forms is not a new research theme. Specifically, the lack of standardization in data entry forms is not a new problem [8], however, it is still a standing one. The diversity of structure and organization of Web forms is a major constraint that still prevents the development of a seamless solution for automating filling in Web forms. Most of currently available techniques focus on the automation of the tasks of filling in forms (i.e. for Auto-complete [22] and Auto-filling [2]). Some recent work [1][9][23] try to tackle the problem by using similarity functions to predict which personal information is expected for each form fields. However, 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 [5][24] 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 [11]. The main drawback with such techniques is that one must have an Ontology describing third-party applications before performing the data integration. OpenID [20] 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 approaches is that it requires the agreement of third-party Web sites to operate. Despite the fact that OpenID has been around for some years, its use is still limited to a few specialized Web sites. Moreover, the reinforced user identification promoted by OpenID is not always a mandatory requirement for user interaction with most Web forms. Instead of focusing on a custom Ontology for particular Web applications, some binding schemas relies on the emergence of open standard data types such as Microformats [17] and Microdata [12]. 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 are pretty similar. However, Microformats have the advantage of an open community and already existing tools to support it.

This work is also closely related to the emergence of Personal Information Management Systems (PIMs) [15]. 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. However, in more recent years some authors started investigating the management of personal information over the Web [18][19]. For example, [19] 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 [26].

The approach introduced in this paper also made use of client-side adaptation techniques for modifying third-party applications. Indeed, the tool support delivered with the approach is able to add new interactors on third-party Web forms (i.e. highlight, new buttons and D&D 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 needs to be changed, and so our approach has virtually no impact on the server-side. However, client-side adaptation is not yet widely known by users so that more research is required to investigate the effect of such technology on the user experience.

Another challenge is to manage possible inconsistencies in the annotations made by the community. A possible solution for that is to rank the annotations in order to determine which are the more reliable annotations.

6 Conclusions and Future Work

One of the contributions of this paper is to highlight the user tasks while interacting with Web forms. Despite of some progress in terms of new interaction techniques for filling in forms, most existing approaches do not provide a big picture of user tasks with Web forms. In this respect, the task analysis presented in this paper can provide new insights. This paper also envisages a possible solution to these problems. For that we have presented an approach based on Web form augmentation for supporting users' tasks when interacting with Web forms. The approach is driven by the fact that users need a better integration of third-party Web forms and their personal information space. For that purpose we combine several techniques including client-side adaptation for form augmentation, annotation of Web pages, and personal information management systems. One of the originalities of the contributions is to combine all these techniques into a single approach. The approach is fully supported by a set of tools named PIMI that were built upon client-side adaptation techniques. Despite the fact that only a few augmenters have been implemented, they provided undeniable support for several user tasks with Web forms. We are currently extending the set of these augmenters for supporting user interaction with Web forms. Anyway, the web site http://www.vincent.gaits.fr/piaff.php contains a set of videos illustrating the use of PIMI and it provides the link for downloading and installing the tool. One of the next steps in this research will be to deploy these tools for investigating the crowdsourcing potential of the approach. Other aspects include the integration of other personal information space that could become available via the tool PIMI.

References

1. Araujo, S., Gao, Q., Leonardi, E., Houben, G.-J.: Carbon: Domain-Independent Automatic Web Form Filling. In: Benatallah, B., Casati, F., Kappel, G., Rossi, G. (eds.) ICWE 2010. LNCS, vol. 6189, pp. 292–306. Springer, Heidelberg (2010)
2. Autofill Forms - Mozilla Firefox add-on, `http://autofillforms.mozdev.org/`
3. Bargas-Avila, J.A., Orsini, S., Piosczyk, H., Urwyler, D., Opwis, K.: Enhancing online forms: Use format specifications for fields with format restrictions to help respondents. Interacting with Computers 23(1), 33–39 (2011), `http://dx.doi.org/10.1016/j.intcom.2010.08.001`
4. Bouvin, N.O.: Unifying Strategies for Web Augmentation. In: Proc. of the 10th ACM Conference on Hypertext and Hypermedia (1999)
5. Bownik, L., Gorka, W., Piasecki, A.: Assisted Form Filling. In: Soomro, S. (ed.) Engineering the Computer Science and IT. InTech (October 2009) ISBN 978-953-307-012-4
6. Card, S., Moran, T., Newell, A.: The psychology of human-computer interaction, 448 p. Lawrence Erlbaum Associates, Hillsdale (1983)
7. Firmenich, S., Winckler, M., Rossi, G., Gordillo, S.: A Framework for Concern-Sensitive, Client-Side Adaptation. In: Auer, S., Díaz, O., Papadopoulos, G.A. (eds.) ICWE 2011. LNCS, vol. 6757, pp. 198–213. Springer, Heidelberg (2011)
8. Girgensohn, A., Leeb, A.: Seamless integration of interactive forms into the Web. Computer Networks and ISDN Systems 29(8-13), 1531–1542 (1997)

9. Guo, X., Kranzdorf, J., Furche, T., Grasso, G., Orsi, G., Schallhart, C.: OPAL: A Passe-partout for Web Forms. In: Proc. 21st Int. Conf. World Wide Web (WWW 2012 Companion), pp. 353–356. ACM, New York (2012)
10. Hartmann, M., Muhlhauser, M.: Context-Aware Form Filling for Web Applications. In: Proceedings of the 2009 IEEE International Conference on Semantic Computing (ICSC '09), pp. 221–228. IEEE Computer Society, Washington, DC (2009)
11. Heinrich, M., Gaedke, M.: WebSoDa: A Tailored Data Binding Framework for Web Programmers Leveraging the WebSocket Protocol and HTML5 Microdata. In: Auer, S., Díaz, O., Papadopoulos, G.A. (eds.) ICWE 2011. LNCS, vol. 6757, pp. 387–390. Springer, Heidelberg (2011), doi:10.1007/978-3-642-22233-7_32
12. Hickson, I.: HTML Microdata (2011), `http://www.w3.org/TR/microdata/`
13. Hori, M., Kondoh, G., Ono, K.: Annotation-based Web content transcoding. In: Proc. of the 9th Int. World Wide Web Conference, pp. 197–211. North-Holland Publishing Co., Amsterdam (2000)
14. Jarrett, C., Gaffney, G.: Forms that Work: Designing Web Forms for Usability, 288 Pages. Morgan Kaufmann (November 2008) ISBN 1-55860-710-2
15. Jones, W., Teevan, J.: Personal Information Management, p. 334. University of Washington Press, Seattle (2007)
16. Olsen, K.A., Malizia, A.: Interfaces for the ordinary user: can we hide too much? Commun. ACM 55(1), 38–40 (2012)
17. Khare, R.: Microformats: The Next (Small) Thing on the Semantic Web? IEEE Internet Computing 10(1), 68–75 (2006)
18. Leone, S., Grossniklaus, M., de Spindler, A., Norrie, M.C.: Synchronising Personal Data with Web 2.0 Data Sources. In: Chen, L., Triantafillou, P., Suel, T. (eds.) WISE 2010. LNCS, vol. 6488, pp. 411–418. Springer, Heidelberg (2010)
19. Norrie, M.C.: PIM Meets Web 2.0. In: Li, Q., Spaccapietra, S., Yu, E., Olivé, A. (eds.) ER 2008. LNCS, vol. 5231, pp. 15–25. Springer, Heidelberg (2008)
20. Recordon, D., Reed, D.: OpenID 2.0: a platform for user-centric identity management. In: Proceedings of the Second ACM Workshop on Digital Identity Management (DIM 2006), pp. 11–16. ACM, New York (2006)
21. Signer, B., Norrie, M.C.: A Model and Architecture for Open Cross-Media Annotation and Link Services. Information Systems 36(3) (May 2011)
22. Stocky, T., Faaborg, A., Lieberman, H.: A commonsense approach to predictive text entry. In: CHI 2004 Extended Abstracts of CHI 2004, Vienna, Austria, April 24 -29, pp. 1163–1166. ACM, New York (2004)
23. Toda, G.A., Cortez, E., da Silva, A.S., de Moura, E.: A probabilistic approach for automatically filling form-based Web interfaces. Proc. VLDB Endow. 4(3), 151–160 (2010)
24. Wang, Y., Peng, T., Zuo, W., Li, R.: Automatic Filling Forms of Deep Web Entries Based on Ontology. In: Proceedings of the, International Conference on Web Information Systems and Mining (WISM 2009), pp. 376–380. IEEE Computer Society, Washington, DC (2009)
25. Winckler, M., Gaits, V., Vo, D.-B., Firmenich, S., Rossi, G.: An Approach and Tool Support for Assisting Users to Fill-in Web Forms with Personal Information. In: Proc. of the ACM SIGDOC 2011, Pisa, Italy, October 3-5 (2011)
26. Zhou, D., Chander, A., Inamura, H.: Optimizing user interaction for Web-based mobile tasks. In: Proceedings of the 19th International Conference on World wide Web (WWW 2010), pp. 1333–1336. ACM, New York (2010)