Using Microformats to Personalize Web Experience

Michael Mrissa[1], Mohanad Al-Jabari[1], Philippe Thiran[2]
[1]PReCISE Research Center, University of Namur, Belgium
{michael.mrissa, mohanad.al-jabari}@fundp.ac.be
[2]PReCISE Research Center, University of Namur and Louvain School of Management, Belgium
philippe.thiran@fundp.ac.be

Abstract

As envisioned by its creator, the WorldWideWeb gathers billions of users from different communities all over the world. A recent evolution of the Web has been witnessed with microformats, which allow authors to semantically annotate the contents of Web documents (webpages, blog posts, news articles, RSS feeds, etc.), and enable inter-software interactions by exporting this annotated content to external applications (calendars, address books, etc.). However, Web users still originate from different communities, and thus follow their own local semantics (referred to as context in this paper) for data interpretation and representation. Hence, there is a need to transform Web content created according to the author's context into the different contexts of its readers. We refer to such transformation process as personalization. In this paper, we identify users' requirements for Web content personalization and we present a solution that takes advantage of microformats in order to enhance users' experience on the Web with contextualized information. We show how microformats offer a great opportunity to adapt the contents of Web documents to different users' contexts.

1 Introduction

During the last few years, the emergence of the Web 2.0 has revolutionized the way information is designed and accessed over the Internet. On the client side, manual browsing of websites has given place to automatic aggregation of RSS feeds into client applications. User-friendly interfaces propelled with Asynchronous Javascript and XML (AJAX) facilitate user interactions while reducing bandwidth [11].

On the server side, the content of websites now tends to a better structuring, thus adapting more easily to heterogeneous platforms with the use of XHTML and CSS. On the client side, the user interaction paradigm is switching from passive (i.e. surfing on the Web) to active (i.e. authoring/editing information on the Web) via weblogs, wikis, and user-driven contents in general.

Also, webpages now tend to integrate semantic information coming from the user. Weblogs and user pages but also official websites massively introduce semantic information via "tags", or keywords. A tag is associated to a particular piece of information (i.e. a post in a blog, an article in a magazine) and provides some insight on the subject this piece of information is about. Web 2.0 sites such as $del.icio.us$ or $flickr$ take advantage of such users' tags to proposing sets of tag-related links as answers to users' queries. Semantic wikis are flourishing [2]. New tools are proposed that link tags to semantic Web applications, thus linking the Web 2.0 to the Semantic Web [10].

1.1 Microformats

Another big change that participates in this Web evolution is the birth of microformats [9], which are tiny pieces of information inserted into the XHTML code of a webpage. Microformats are developed according to a set of open standards called microformat specifications [1, 4, 8]. With the help of microformats, semantic information is directly attached to the contents of webpages. While the objective of microformats is to enhance user experience, microformats are first detected by XML parsers, and provide explicit, non-ambiguous, machine-interpretable semantic information about the content they are attached to.

Among the most famous Web 2.0 sites such as Twitter, Flickr, LinkedIn, Upcoming and Yahoo[1] have already adopted microformats. Indeed, the -mostly unexploited- potential benefits offered by microformats are numerous:

- automatic analysis of Web information,
- export of microformatted information to external applications,

[1]http://www.yr-bcn.es/demos/microsearch/

- no need for complex ontologies to add information,
- human readability with the help of browser plugins.

Microformats are typically utilized as a tool to enable inter-application interactions. For instance, an event described in a webpage that is annotated with a microformat enables (via the browser's plugin) one-click export of the event description into the user's calendar application. Tools have already been developed that export contact information (hCard microformat) and event information (hCalendar microformat) into address book and calendar applications.

1.2 Challenges & motivation

Users typically encounter data interpretation difficulties while browsing the Web. These difficulties are due to several discrepancies between the semantics of the webpage author and those of the webpage reader. Most of these discrepancies originate from these persons' *local contexts* that promote different interpretations of the same contents. A local context is a set of common knowledge (or common cultural conventions) that is shared between a group of community members, like language, measurement units, and date/time formats [6, 5]. Although the common local conventions of group of members are often implicit and can be viewed from different perspectives, [12] argue that local community members not only share a common language, but also common culture conventions, such as measurement units, keyboard configurations, character sets and notational standards for writing time, dates, addresses, numbers, currency, etc. In the following, we present an example motivated by the belongings of a webpage author and reader to French and English communities.

Currently, the data authored on the Web are written according to the author's semantics. For example, a French user browsing an English website on the Web has to translate an English-formatted date (mm/dd/yyyy) to its own format (dd/mm/yyyy) in order to interpret it correctly. While there are some exceptions (the 6^{th} of June, the 12^{th} of December), most of the time these differences in the semantic organization of data require additional work for correct data interpretation. A similar situation occurs with prices, lengths, weights, in general unit measures, and probably many other pieces of information related to local semantics.

At first sight, microformats do not offer very much to users in terms of personalization: while the final goal of microformats is to enhance human experience, the semantic information they offer is not meant to be directly read by users but machines first. However, they have the characteristic to be machine-interpretable, thus allowing programs to "understand" them. In this paper, we take advantage of the possibilities offered with microformats to enhance users' Web experience with a personalized display of information. We propose a Web document personalizer that provides users with a representation of microformatted information in webpages that is adapted to their local contexts.

1.3 Paper organization

This paper is structured as follows. Section 2 explores the needs for personalization of information from a user's point of view. Section 3 introduces microformats and presents the most advanced propositions. Section 4 discusses the relation between microformats and users' personalization requirements, before presenting our proposal for Web contents personalization. Section 5 discusses the results obtained and gives some insights for future work.

2 Users' personalization requirements

In this section, we identify users' requirements in terms of personalization. By no means we claim to propose an exhaustive list of personalizable concepts, but we try to address the main concerns that rose up from our own experience surfing the Web. Hence, we focus on the following personalizable concepts:

- *Date/time* are organized in different ways according to the user's language and country[2].
- *Prices* are expressed in different formats, (currencies, VAT rate included, etc).
- *Addresses* are structured differently. Postcode formats are different from country to country, sometimes street number is before street name, (like in France), sometimes after (like in Belgium).
- *Measure units* also depend on the country (mainly English and Metric systems are used).
- *Telephone numbers* depend on the country too.

According to these notions, we identify a set of user characteristics that currently form our user context. Here also, we do not aim at building an exhaustive list of required context parameters but we gathered the parameters that are required to answer the personalization needs of the notions listed previously.

One could argue that these personalizable concepts depend on the user's country, which can be obtained from the IP address contained in HTTP requests. However, we assume that users connected from a foreign country do not want the webpage information to be personalized according

[2] http://en.wikipedia.org/wiki/Calendar_date

to the local context of the host country. Furthermore, one country could have several communities, e.g: Belgium.

As a consequence, we establish a combination of language and country as the main parameter for context, together with timezone, optional date style and currency parameters to distinguish users' local contexts. The $language(country)$ parameter is used to adapt the formatting of the original webpage information, and is combined with a $datestyle$ parameter to format the dates according to the user's context. $timezone$ and $currency$ parameters respectively identify the time zone and local currency of the user and enable correct conversion of time and price information displayed on webpages.

3 Microformat specifications

Several microformats have been designed in order to describe the semantics of the most typical elements users can encounter on web documents. The most well-known microformats are $hCard$, $hCalendar$ and $hReview$. We detail microformats below according to two categories: *accepted standards* that have been validated by the community and thus that should be used as described in the specification, and *emerging proposals* that are already advanced specification drafts but could be subject to further modifications.

3.1 Accepted standards

hCard. The $hCard$ microformat describes people and organizations. It is identified with $vcard$ as a class name. It requires at least the fn or $n*$ subclass that identifies an individual with a fullname or another type of name (given name, family name, etc.). Then, several other classes are optional together with their subclasses ($nickname$, url, $email$, tel, adr, org, etc.). This microformat is based on the vCard specification described in RFC 2426[3].

hCalendar. This microformat describes events and calendar information. It is identified with a *vcalendar* or *vevent* class name. Mandatory subclasses are *dtstart* and *summary*, they respectively describe the starting time and summary of an event. Optional subclasses are possible based on the vCalendar specification described in RFC 2445[4].

XHTML Friend Networks (XFN). XFN describes relationships between people. It allows one to specify other persons as friends, colleague, etc. using the rel attribute[5].

3.2 Emerging proposals

hReview. The $hReview$ microformat allows describing online reviews and ratings. It is a composite format that has only one mandatory subclass $itemInfo$ which contains either a fn fullname (with url or $photo$ subclasses), or a $hCard$ or a $hCalendar$ subclass (events can be reviewed too, like concerts for example). Several optional elements complete the microformat (*reviewer* (*hCard*), *dtreviewed*, *rating*, *description*, *tags*, *permalink*, *license*).

hListing. The $hListing$ Microformat provides listings format suitable for embedding in (X)HTML, Atom, RSS, and arbitrary XML. it is identified with a $hListing$ class name. Mandatory subclasses are *listingAction*, *lister(hCard)*, and *description*. Several optional subclasses includes *dtlisted*, *dtexpired*, *price*, etc.

hAtom. The $hAtom$ microformat is intended to describe web contents that can be syndicated, e.g: weblog postings. It is identified with *hentry* and optional *hfeed* class names. Mandatory subclasses are *entry-title*, *updated*, and *author*. They describe Atom entry title, updated date, and the author name, respectively. Optional subclasses like *entry-content*, *entry-summary*, *published*, and *bookmark* are also possible based on the Atom syndication format described in RFC 4287[6].

hMeasure. The $hMeasure$ microformat describes physical quantities measured according to specific units. Mandatory subclasses are *value* and *unit* that respectively specify numeric value and measurement unit of the physical quantity. Optional subclasses include *item*, *type* and *tolerance* to specify which item or product is being measured, the dimension being measured (e.g. height or width of length quantity), and the error rate (percentage or nested *hMeasure*).

hMoney. The $hMoney$ microformat describes money information. It is identified with the $money$ class name. It requires at least the $amount$ subclass that specifies the numerical value of money, together with *currency*, *unit*, and *date* optional subclasses, which respectively specify ISO 4217[7] currency code, currency unit (e.g: Euro, cent), and the date associated to the value.

adr. The adr microformat is utilized as an optional subclass in several microformats (e.g.: hCard(adr), hCalendar(location(adr)), hListing(item info(adr)), etc.) that specifies the address information. It is identified by the adr class name, and *post-office-box*, *extended-address*, *street-address*, *locality*, *region*, *postal-code*, and *country-name* subclasses.

geo. The geo microformat is also an optional subclass of several microformats (e.g.: hCard(geo), hCalendar(location(geo)), hListing(item info(geo)), etc.) that specify geographic coordinates. It is identified by the geo class name, together with *latitude* and *longitude* subclass names.

[3] Available on `http://www.ietf.org/rfc/rfc2426.txt`
[4] Available on `http://www.ietf.org/rfc/rfc2445.txt`
[5] More information on `http://www.gmpg.org/xfn/11`.
[6] Available on `http://www.ietf.org/rfc/rfc4287`
[7] Available on `http://www.iso.org/iso/support/faqs/faqs_widely_used_standards/widely_used_standards_other/currency_codes/currency_codes_list-1.htm`.

μ-formats	Date/Time	Price	Measurements Units	Address	Tel Number
hCard	bday,tz		geo	adr	tel
hCalendar	dtstart, dtend dtstamp, duration rdate, (via hCard)		geo description (hMeasure)	location (adr) (via hCard)	(via hCard)
hReview	dtreviewed (via hCard, hCalendar)	price	description (hMeasure)	(via hCard) (via hCalendar)	(via hCard)
hListing	dtlisted, dtexpired (via hCard, hCalendar)	price	item info(geo) description (hMeasure) (via hCard,hCalendar)	item info (adr) (via hCard) (via hCalendar)	(via hCard)
hAtom	published, updated, (via hCard, hCalendar) (via hReview, hListing)	entry-content (via hReview) (via hListing)	entry-content (hMeasure) (via hCard, hCalendar) (via hReview, hListing)	(via hCard) (via hCalendar) (via hListing)	(via hCard)
hMoney	date	money			

Table 1. Correspondences between users' personalization requirements and μ-format specifications.

There are other microformats that describe licenses (*rel-license*), tags, keywords, categories (*rel-tag*), and also lists and outlines ($XOXO$). For brevity purpose, we do not give details on these microformats in this paper, and we refer the reader to `http://microformats.org` for additional information. Note that the specifications of microformat proposals could still be subject to major changes as they are not yet accepted as standards.

4 Personalizing Web documents

In this section, we examine to which extent microformats are useful for the personalization purpose, before presenting our personalization approach and detailing its implementation and deployment.

4.1 Microformats and users' personalization requirements

Microformats can be atomic, i.e. self-contained like adr or geo, or they can be composite, like $hCard$ or $hCalendar$. Table 1 summarizes the correspondences between the main composite microformats and users' personalization requirements. Each cell of Table 1 describes the particular microformat utilized by the composite microformat in order to represent the semantic information. For brevity purpose, we exclude atomic microformats, which have straightforward correspondences (i.e. adr corresponds to the address requirement, geo and $hMeasure$ correspond to the Measurement units requirement).

Table 1 shows that the personalizable concepts aforementioned are present in most existing microformats. Furthermore, it is possible for a webpage author to mix/nest several microformats that contain different pieces of information, as for $hReview$, which may host $hCard$ and $hCalendar$ microformats. Therefore, personalizable microformats $class$ attributes should be directly extracted from webpages independently of the container microformat and personalized according to the user's preferences.

4.2 General approach

Our personalization approach focuses on adapting the contents of webpages based on a set of parameters that help setup the user's context. We devised a personalizer engine shown in Fig. 1 as the core component of our approach. Our personalizer engine parses a URL-identified web document and user context parameters as inputs and produces a personalized web document that can be viewed according to the user's context.

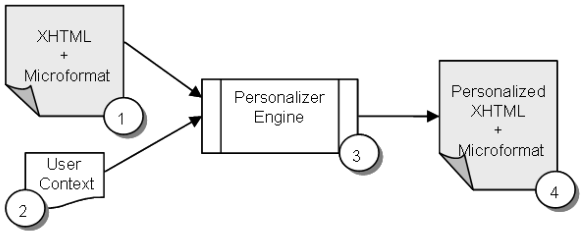

Figure 1. Personalizer overview.

The main idea developed in this work consists in parsing the XHTML Web document and identifying elements with $class$ attributes that have for values the names of our personalizable elements ($dtstart$, $dtend$, $bday$, $dtreviewed$, tel, etc.). Then, the personalized information obtained the web document is added to the original contents. In order to ensure good user understanding, the original version is kept

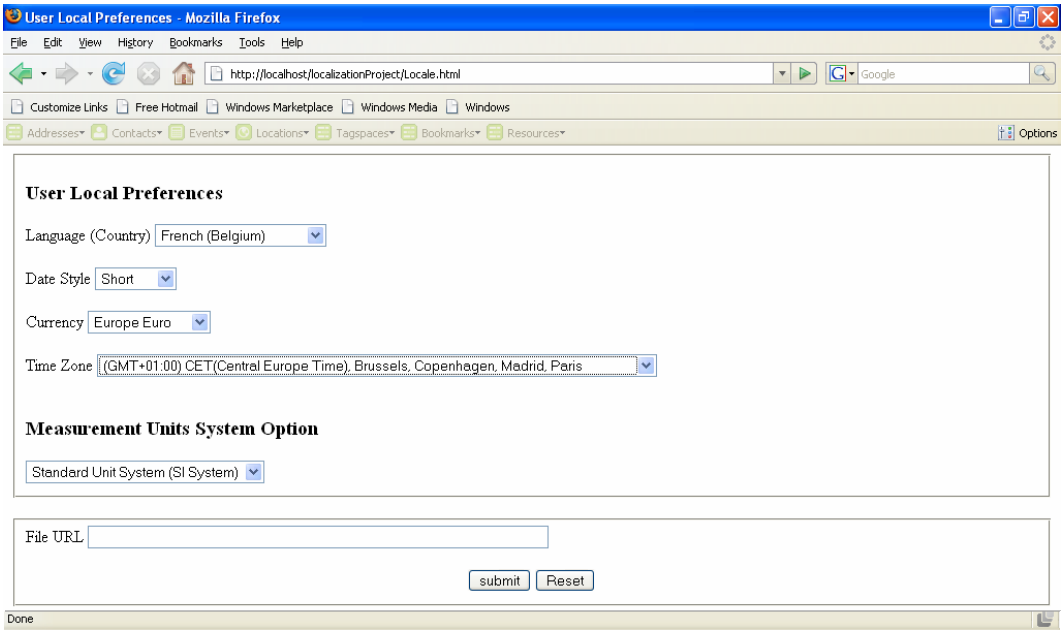

Figure 2. Screenshot of the servlet preference page.

as is and the personalized data is put next to it into brackets. Our prototype currently detects dates, currencies and time zones.

4.3 Implementation and deployment

Our personalizer has been developed under the Eclipse™ environment and Java™ platform. In order to make our solution embeddable into the largest number of existing architectures, we developed it and tested its deployment in three different fashions: server-side, client-side and as a library. The deployment of our personalizer on the client-side gives users the opportunity to personalize the contents of all web pages directly on the user's computer. Also, users' parameters are kept locally, thus favorizing privacy and security concerns. On the other hand, the deployment of our personalizer on the server-side in a proxy-like fashion gives control to the Web server and allows exploiting the information entered by users and performing statistics on users' preferences, number of users, etc. However this deployment method is less reliable when it comes to the security and privacy concerns.

Server-side deployment. Our personalizer is deployed on the server-side as a Web servlet that gets the Universal Resource Location (URL) of a Web document in addition to user's personalization parameters, and returns the same webpage with additional personalized contents (Fig. 2). Our Web interface acts as a proxy that performs on-the-fly personalization of Web contents.

Client-side deployment. Client-side deployment is performed via a Java program that is made accessible via a Firefox extension (Fig. 3). In order to link our Java program to the Firefox extension, XPCOM components are utilized. The Firefox extension integrates seamlessly into the user's browser and adds personalization capabilities to Firefox. Our extension prototype is available at `http://perso.fundp.ac.be/~pthiran/microformats/`.

For the purpose of client-side deployment, we integrate our personalizer engine as an extension to the Firefox browser (Fig. 2). In order to embed our java-based personalizer engine; XUL (XML User Interface Language), JavaScript, and XPCOM technologies are utilized. XUL is used for implementing the user context interface, while JavaScript and XPCOM used as glue, where JavaScript code gets the URL of webpage and the users' preferences and send them to java code using XPCOM components.

Java library. Our personalizer is also available as a Java library (available at the same url address than the extension prototype), as we believe it could be adapted to many (any) other Java-based application dealing with microformats: browser (Firefox/IE plugin), RSS feed readers, email application, calendar application, etc.

5 Conclusion

In this paper, we identify users' needs for personalization of webpage contents and we take advantage of microfor-

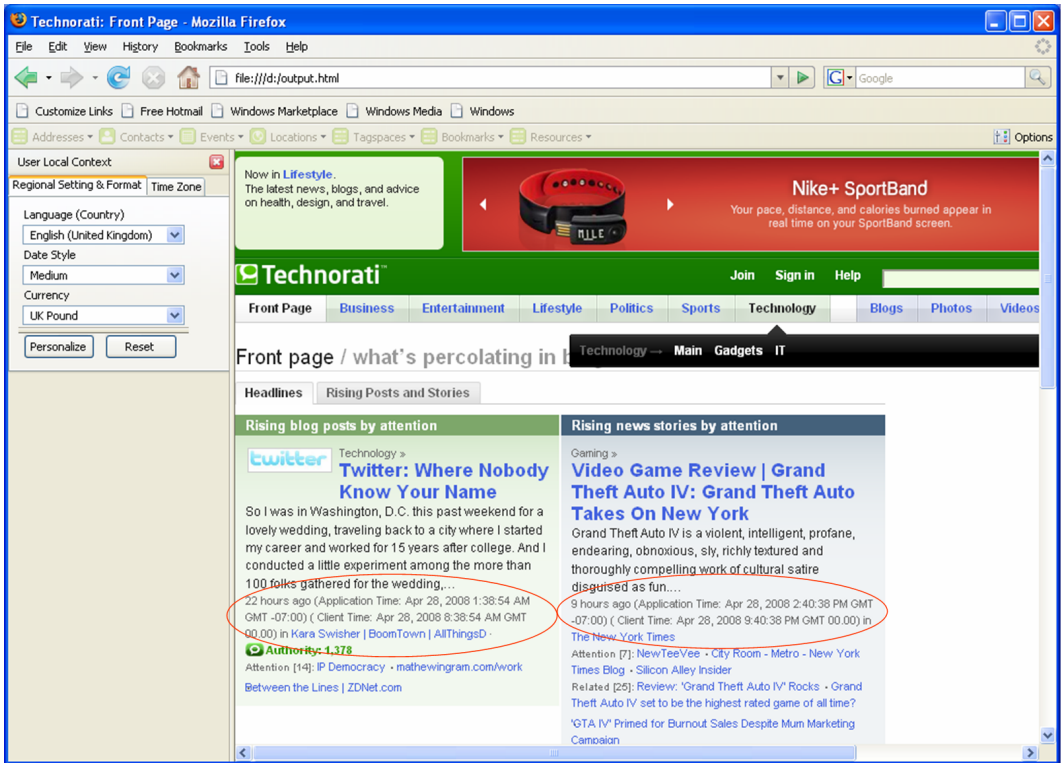

Figure 3. Screenshot of the Firefox preference extension.

mat annotations in order to personalize the contents of Web documents. Our proposal relies on a limited set of user parameters in order to enable personalization of webpage contents. We implemented and validated our proposal both on the client-side with a Firefox plugin and on the server-side with a servlet application.

This work illustrates one of the advantages microformats can bring to the Web. However, as microformats propose a finite set of specifications, they remain rather limited. As a future work, we believe it could be interesting to evaluate to which extent our personalization approach could be adapted to emerging semantic annotation proposals such as RDFa [3] or eRDF [7], which do not restrict semantic annotations to a set of specifications.

References

[1] Microformat homepage (wiki). `http://www.microformats.org/wiki/` (last accessed: 20 Apr. 2008).

[2] Semantic MediaWiki Homepage (wiki). `http://www.semanticweb.org/wiki/Semantic_MediaWiki` (last viewed April 29, 2008).

[3] B. Adida and M. Birbeck. Rdfa primer 1.0 embedding rdf in xhtml. W3c working draft, W3C, October 2007.

[4] J. Allsopp. *Microformats: Empowering Your Markup for Web 2.0*. Apress, 2007.

[5] W. Barber and A. Badre. Culturability: The merging of culture and usability. In *The 4th conference on human factors and the Web*, 1998.

[6] D. Cyr and H. Trevor-Smith. Localization of web design: An empirical comparison of german, japanese, and united states web site characteristics. *JASIST*, 55(13):1199–1208, 2004.

[7] I. Davis. RDF in HTML (eRDF). `http://research.talis.com/2005/erdf/wiki/Main/RdfInHtml` (last viewed April 29, 2008).

[8] R. Khare. Microformats: The next (small) thing on the semantic web? *IEEE Internet Computing*, 10(1):68–75, 2006.

[9] R. Khare and T. Çelik. Microformats: a pragmatic path to the semantic web. In L. Carr, D. D. Roure, A. Iyengar, C. A. Goble, and M. Dahlin, editors, *WWW*, pages 865–866. ACM, 2006.

[10] A. Passant. MOAT: Meaning Of A Tag - Project Homepage. `http://moat-project.org/` (last viewed April 29, 2008).

[11] K.-U. Schmidt, L. Stojanovic, N. Stojanovic, and S. Thomas. On enriching ajax with semantics: The web personalization use case. In E. Franconi, M. Kifer, and W. May, editors, *ESWC*, volume 4519 of *Lecture Notes in Computer Science*, pages 686–700. Springer, 2007.

[12] O. D. Troyer and S. Casteleyn. Designing localized web sites. In X. Zhou, S. Y. W. Su, M. P. Papazoglou, M. E. Orlowska, and K. G. Jeffery, editors, *WISE*, volume 3306 of *Lecture Notes in Computer Science*, pages 547–558. Springer, 2004.