

Handling Users Local Contexts in Web 2.0: Use Cases and Challenges

Mohanad Al-Jabari^{1*}, Michael Mrissa², and Philippe Thiran¹

¹ PRECISE Research Center, University of Namur, Belgium

² SOC Research Team, LIRIS, University of Lyon 1, France

Abstract. Creating, updating, and aggregating Web contents from different Web users and sites form the heart idea of Web 2.0. However, Web users originate from different communities, and follow their own semantics (referred to as local contexts in this paper) to represent and interpret Web contents. Therefore, several discrepancies could rise up between the semantics of Web authors and readers. In this paper, we present several Web 2.0 use cases, and illustrate the possible challenges and trends to handle the local contexts of Web users in these use cases.

1 Introduction

During the last years, the emergence of the Web 2.0 has revolutionized the way information is designed and accessed over the internet. The term *Web 2.0* was officially coined by Tim O'Reilly in [11] as a set of design principles and exemplified by sites such as *Wikipedia*³, *MySpace*⁴, *Upcoming*⁵, etc. However, several researchers including Tim O'Reilly himself argue that there is no clear-cut definition of this term [6, 2, 3].

The heart idea of Web 2.0, in addition of using Web technologies as a platform, lies into the sharing of Web contents from different sources. *Community collaborations and contents mashups* are the most common Web 2.0 features [3]. To illustrate these features, let us distinguish them from the classical Web (called "Web 1.0") features as follows:

- **Community collaboration.** In Web 1.0, a few *Web authors* create and update Web contents for relatively passive *Web readers*. However, Web 2.0 sites enable Web users not only to browse the Web but also to create, update, and share Web contents in usually self-organizing manner. Hence, Web users now can act as active *Web authors*.
- **Contents mashups.** In Web 1.0, Web contents (information and services) on a single web page are usually belong to one Web site. In Web 2.0, contents from several sites can be aggregated, mixed, and displayed together.

* Supported in part by the Programme for Palestinian European Academic Cooperation in Education (PEACE).

³ <http://wikipedia.org>

⁴ <http://www.myspace.com>

⁵ <http://upcoming.yahoo.com>

The emerging results of community collaboration and contents mashups could not be achieved by individual users and individual Web sites, respectively. Each user gains more from the systems than he puts into it. Also, one Web site can not satisfy all the users' needs. Contents from different sites are to be aggregated and mixed together to satisfy complex users' requests.

1.1 Users Local Contexts

The Web gathers billions of Web users from all over the world. These users originate from different communities, and follow their local contexts for interacting with Web contents. By *local context*, we mean a set of common knowledge such as a common language and common cultural conventions such as measure units, keyboard configurations, character sets, notational standards of writing times, dates, numbers, currency [14, 5].

Since different communities usually have different local contexts, a same concept (a Web concept) could be represented differently by different *Web authors*. Also, the same Web content (the representation of a Web concept) could be interpreted in different ways by different *Web readers*. Hence, several discrepancies could be arisen between the semantics of Web authors and readers. For example, assume a French reader who wants to interpret a *price* Web content which is authored by a British author. In this context, the price is represented in British Pound and follows the British currency format (e.g., 1,234.50). As the French currency is Euro and different format is used (e.g., 1 234,50), the price must be converted from British Pound to French Euro by the reader. Note that the situation can be even worse if the reader wants to interpret a *date* content. The reader could misinterpret the date content (e.g., 07/08/2008) as the 7th of August 2008 (following the French format) instead of the 8th of July 2008 (following the British format). Similar situations may occur with other pieces of Web contents that are related to users' local contexts.

1.2 Web 2.0 and Users Local Contexts

The emergence of the Web 2.0 raises new challenges. Web contents in a single page can be authored (created and updated) by several authors who have different local contexts. Moreover, contents authored from several authors on several Web sites could be dynamically aggregated, mixed, and displayed together in a single Web page. This paper presents several possible Web 2.0 use cases and explores some possible challenges and trends for handling users' local contexts in these use cases.

This paper is organized as follows. Section 2 presents several Web 2.0 use cases. Section 3 introduces a set of concepts that could be represented and interpreted according to users' local contexts and the challenges of handling them in the Web 2.0 use cases. Section 4 introduces semantic annotation as a possible solution. Finally, Section 5 concludes the paper.

2 Web 2.0 Use Cases

In this section, we describe several possible use cases that users could perform when they use Web 2.0 systems. By no means we aim at covering all Web 2.0 use cases, but we attempt to classify the aforementioned Web 2.0 features (i.e., community collaboration and contents mashups) into three use cases: Web contents creation, Web contents update, and Web contents aggregation.

2.1 Web Contents Creation

Several Web 2.0 systems enable Web authors to create Web contents, without giving the opportunities to update the published contents or parts of them. We focus on the Web contents creation in this use case. To illustrate this, let us consider the following Web 2.0 services:

- *Weblog (also called blog)*. Web 2.0 systems such as *WordPress*⁶ allow a single author to create Web contents (e.g., scientific articles, privacy issues, etc.) called *post*, whereas other secondary users can add comments to contents created by the original author as new html nodes.
- *Bulletins Section*. Web 2.0 social systems such as *Facebook*⁷ and *MySpace* provide a service to a group of users called “bulletin board”. Bulletin board allows a user to add a piece of Web content (e.g., text message), whereas other users on the group list can see this content. Bulletins can be useful to contact an entire friends list without resorting to messaging users individually.
- *Group Section*. Social systems also provide a service called “group section”. One or more users can create a common page (i.e., group section). The group creator(s) can invite any one to join, deny user’s join request, delete or update users’ contents, etc. Joined users, in addition to the group creator(s), usually can browse and create contents on the group section.

2.2 Web Contents Update

Several Web 2.0 systems enable Web authors to update Web contents after publishing. In this use case, a Web author could update the Web contents that she/he creates (referred to as a personal contents update) or the Web contents that other Web authors create (referred to as a community contents update). The following Web 2.0 services illustrate this use case:

- *Personal contents update*. Web 2.0 commerce systems such as *eBay*⁸ allow a Web user to update the contents about the items she/he wants to sell. A user can update the contents concerning these items like the price, the photos, the selling location, etc. Other users can not change these pieces of contents. In addition, social systems allow a user to update his own profile such as login name and password, preferred language, interests, etc.

⁶ <http://wordpress.org/>

⁷ <http://www.facebook.com/>

⁸ Available on <http://www.ebay.com/>.

- *Community contents update.* Wiki systems such as *Wikipedia* allow one or more users (usually authorized users) to create Web contents as a set of interlinked Web pages and update these contents using creating and editing services. For example, a Web user can define the term *local context* or update the existing definition authored from other author(s). In addition, collaborative editing systems such as *Google Docs* allow a group of users (might be from different locations) to collaboratively create and update documents (e.g., word document) online. Finally, the group creator(s) of the group section presented above can update the contents created by joined users.

2.3 Web Contents Aggregation

Several Web 2.0 systems and technologies provide Web contents aggregation and mixing services. In this sense, the aggregation and mixing services could be performed on client-side (referred to as a client-side aggregation) or on a specific server-side application (referred to as a server-side aggregation). The following Web 2.0 services illustrate this use case:

- *Client-side aggregation.* RSS feed reader (aggregator) is the most known technology that allows client-side applications (e.g., Web browser) to find out and collect Web contents from RSS-enabled Web sites⁹. In addition, *Piggy bank* [7] and *Kalpana* [4] provide client-side aggregation services. These services aim at enabling Web readers to extract and aggregate personal information from different Web sites, and to store them locally in RDF formats.
- *Server-side aggregation.* Several Web 2.0 systems mix Web contents from different sites. For example, Google provides an advertisement service called *adSense*¹⁰ which enables Web site to add text, image, or video advertisement from other Web sites. In addition, several Web 2.0 systems provide aggregation services for specific types of Web contents. For example, *Technorati*¹¹ aggregates and indexes different types of contents such Weblogs, photos, news, DVDs, etc. Also, Technorati allows readers to search these contents in different ways (e.g., readers can search Weblogs according to Weblogs' language). *Upcomming* is another system that aggregates events from users communities and commercial sites. Users can indicate their plans by marking that they are “going” to or “interested” in events that are occurred in a location, date, future periods, etc. Also, users can choose which events who are interested in such as education, music, sports, etc.

Finally, several E-commerce systems compose Web services together (e.g., airplane ticket reservation, car rental reservation, and hotel reservation) from different service providers (i.e., Web sites) to satisfy a complex user request. In this sense, we can assume these systems as server-side aggregators.

⁹ Any website that offers RSS feeds for its content.

¹⁰ <http://www.google.com/adsense>

¹¹ <http://technorati.com/>

3 Web 2.0 Use Cases and Users Local Contexts

As we mentioned, several discrepancies could be arisen between the semantics of Web authors and readers, since they could have different local contexts. In this section, we initially presents a set of concepts that could be represented and interpreted according to users' local contexts. Then, we discuss the challenges of handling the local contexts of these concepts in the above Web 2.0 use cases.

3.1 Context-Sensitive Web Concepts

Based on local context, we aim at classifying Web concepts into context¹²-sensitive and non-context-sensitive concepts. *Context-sensitive concepts* refer to the concepts which could be represented in different ways by different authors. The following list identifies a set of context-sensitive concepts. By no means we claim that this list covers all context-sensitive concepts, but we try to address the main concerns that are rose up in the aforementioned use cases [10, 9].

- **Date/time.** Date refers to a particular day of a month or a year within a calendar system (e.g., Gregorian, Islamic, Japanese, etc.). In addition, different communities represent Date in different ways. The day, month, and year are ordered differently, and different separators are used. Also, text representation of Date depends on user's local language and country. Finally, Time could be represented in 12-hour AM/PM or 24-hour style, and with different time zone.
- **Number.** In mathematics, Numbers are mainly used for counting and measuring amounts or quantities of objects based on a number system. Different local symbols are used to represent numbers (also called numerals such as English and arabic numerals¹³). Also, different decimal and thousands separators (i.e., dot and comma) are used in different countries.
- **Price.** Price refers to a numerical monetary value assigned to a good, service or asset. Prices are expressed in different formats, currencies¹⁴, and Tax systems (Tax rates, included/excluded, etc.).
- **Physical quantities.** Physical quantities such as weight, length, temperature, etc. are measured using a set of units called measure units. Countries are used different measure systems (mainly Imperial and Metric systems), different unit prefixes, and different error percentage¹⁵.
- **Telephone number** refers to a unique sequence of numbers used to identify a telephone endpoint. Based on ITU¹⁶ numbering plan *E.164*, each country has a different *international call prefix* and *country calling code*. Furthermore, each country uses a specific telephone number's format.

¹² Context here refers to the local context.

¹³ See numeral systems on http://en.wikipedia.org/wiki/Numeral_system

¹⁴ See ISO 4217 for used currency list.

¹⁵ More information available on http://en.wikipedia.org/wiki/Units_of_measure

¹⁶ International Telecommunication Union: <http://www.itu.int/>

3.2 Challenges of Handling Users Local Contexts

We can conclude that the local context represents a part of the semantic for the above Web concepts. Also, the semantic discrepancies that could arise do not relate to these concepts themselves, but rather to the local contexts of Web authors and readers that are implicitly used when they represent and interpret these concepts.

In order to address this issue, several approaches have been proposed to adapt Web contents to be suitable to readers' local contexts [12, 14, 8]. These approaches are mostly based on two assumptions: (1) the semantics of target Web contents to be adapted are known in advance; (2) Web contents are represented according to a single local context.

However, the use cases presented above illustrate that these assumptions are not valid anymore. Web contents are shared (created, updated, and aggregated) from different sources (i.e., Web users and Web sites). Hence, they are represented according to different local contexts and have heterogeneous semantics. Therefore, the following challenging issues should be tackled:

1. *Semantic identification*. What is the information that required to identify the semantics of Web contents and the local contexts of Web users?
2. *Semantic information management*. How can the contents' semantics and the users' local contexts information be managed in terms of acquiring, representing, and storing this information? Also, what is the local context that used for representing each piece of Web content?

Semantic Identification

As mentioned before, Web contents could be created, updated, and aggregated from different sources. In this sense, different Web contents from different sources could refer to the same context-sensitive concept. For example, different authors could use *cost*, *price*, and *amount* contents to refer to the *price* concept. In addition, the value of the price concept could be represented in different ways, according to the authors' contexts.

Moreover, Web contents can be *stored, aggregated and hosted* on the server-side and can be *aggregated and presented* on the client-side. Server-side and client-side applications can not interpret Web contents if they are represented only using XHTML. Hence, a server-side application can not be aware if Web contents such as *cost*, *price*, and *amount* refer to the *price* concept or not, and it can not know which local context was used for representing them.

In this sense, several questions could be raised here. Firstly, what is the information that required to identify the semantics of Web contents, so that server-side and/or client-side applications can interpret that Web contents from different sources refer to one context-sensitive concept? Secondly, what is the (minimum) information required to identify the users' (authors and readers) local contexts, so that server-side and/or client-side applications can adapt Web contents from authors' contexts to readers' contexts. One could argue that the local context depends on users' countries, which can be obtained from the IP

address contained in the HTTP header. However, this assumption is not valid as one country can have several communities (e.g., Belgium). Another question: how can we identify the local contexts of cross-sites aggregated contents?

Semantic Information Management

In addition to the aforementioned issues, the information required to identify the semantics of Web contents and the local contexts of Web users needs to be acquired, represented, and stored.

In this sense, several questions have to be tackled. Firstly, how can the required information be acquired from different sources (i.e., users and sites). Assume the Web contents creation and update use cases. Does the required information be acquired directly from the authors or be acquired (predicted) from the server-side applications? Also, when the required information be acquired? (i.e., before contents creation or update, during contents creation or update). Assume the Web contents aggregation use case. How can this information be acquired from different sites.

Secondly, how should the required information be represented and where it should be stored (i.e., on the server-side or on the client-side), so that the local contexts of context-sensitive concepts can be handled in the above use cases. For example, the required information should be accessible from the client-side applications in order to handle the client-side aggregated contents. Also, it should be accessible from the server-side applications in order to handle the server-side aggregated contents.

Finally, how to specify the local context that used for representing each piece of Web content? Assume the community update use case where one Web author can update the contents created by other authors (e.g., Wiki contents). The question here: are the updated contents related to the context of the original author or the context(s) of the author(s) who update these contents? Moreover, assume, in contents aggregation use case, the case where the authors' local contexts for parts of the aggregated contents are not specified. How can this case be handled?

4 Possible Solution

One possible solution to handle the aforementioned challenges is to directly rely on the authors for annotating Web contents with semantic metadata, so that the former become machine interpretable [15]. Semantic metadata are used to describe contents' semantics and users' local contexts explicitly. In this sense, Client-side and server-side applications can interpret a Web content (e.g., cost) that is related to a specific context-sensitive concept (e.g., *price*). Also, they can interpret that this content is represented according to a specific local context. Therefore, Web contents can be adapted from authors' local contexts to different readers' local contexts.

In addition, semantic metadata are accessible from server-side and client-side applications, as they are combined with Web contents. In the content aggrega-

tion use case, Server-side and/or client-side applications aggregate Web contents together with the corresponding semantic metadata. Finally, the Web authors, in contents update use case, should update Web contents and also the corresponding semantic metadata.

In this field, there are two alternative approaches. The first approach aims at standardizing the representation of Web contents and their semantics for all sources. For example, representing the Date/Time concepts according to the ISO 8601 specification¹⁷. The second approach aims at allowing authors to represent Web contents in different ways, but explicitly annotate them with semantic metadata (i.e., contents' semantics and authors' local contexts). Microformats technology¹⁸ follows the first approach and RDFa¹⁹ technology follows the second one [13, 10].

4.1 Microformats

Microformats propose a set of standards, or *specifications*, and reuse XHTML attributes such as *id* and *class* to embed those specifications into XHTML documents. For example, the *hCard* specification identifies vocabularies based on the vCard²⁰ specification that provide semantic information about people and organization. Microformats specifications standardize the representation of Web contents and their semantics at different three levels as follows:

- *Schema level*. Identifying a specific schema for each Microformats specification in terms of concepts and sub-concepts (called classes and subclasses) that can appear and their cardinalities (e.g., required, optional, etc.), the ordering of schema classes, etc. For example, hCard should have *vcard* class, *fn* and *n* subclasses at minimum.
- *Concept level*. Identifying a specific semantic vocabulary (Semantic label) for every class and subclass in each Microformats specification. Therefore, standardizing contents' semantics.
- *Representation level*. Identifying a specific representation for each class's and subclass's values. The authors should follow these representations as much as possible, so that Microformats parsers can interpret these representations. Therefore, standardizing authors' local contexts.

Server-side and/or client-side applications can interpret Web contents annotated with Microformats (i.e., exchange, aggregate, adapt, etc.) without significant loss of meanings. However, Microformats are not extensible and do not fulfill all authors' use cases. In our previous work, we conclude that Microformats remain rather limited as they propose a finite set of specifications [10]. Technically, Web authors can create new specifications, but it is not recommended without extensive discussion with the Microformats community for a general (i.e.

¹⁷ http://en.wikipedia.org/wiki/ISO_8601

¹⁸ More information available on <http://microformats.org/>

¹⁹ More information available on RDFa wiki: <http://rdfa.info/wiki/RDFa>.

²⁰ More information available on <http://www.isi.edu/in-notes/rfc2426.txt>

worldwide) adoption. Until this point is reached, Microformats parsers could not interpret what are considered as “exotic” Microformats specifications.

4.2 RDFa

RDFa provides a more abstract solution that aims at expressing RDF statements in XHTML documents. More precisely, RDFa provides a collection of XHTML attributes (reuses existing attributes such as *content* and *rel* and introduces new ones such as *about* and *property*) to embed RDF statements in XHTML, whereas it provides processing rules for extracting RDF statements from XHTML.

Web authors can reuse existing RDF-based semantic metadata (e.g., Dublin Core and FOAF metadata) and create their own semantic metadata. Therefore, RDFa is fully extensible. However, since Web contents and semantic metadata from different sources are represented in different ways; the interpretation of these contents (i.e., exchange, aggregation, adaptation, etc.) require a prior semantic reconciliation between server-side and client-side applications [3].

In [1], we propose an approach that uses RDFa to annotate context-sensitive concepts with authors’ local contexts, so that these concepts can be adapted into different readers’ local contexts.

5 Conclusion

The main strength of the Web lies in its capacity to interconnect billions of users from all around the world. However, this gathering of communities can lead to the misunderstanding of Web contents as each community of users uses its own context for interacting with Web contents. In this paper, we identified new challenges in improving the context interpretation of Web contents in some typical Web 2.0 use cases. We also explained how existing technologies such as RDFa and Microformats can help people to better understand each other on the Web. Based on [1], our future work aims at providing an intuitive way for helping authors to annotate context-sensitive concepts with contextual attributes.

References

1. M. Al-Jabari, M. Mrissa, and P. Thiran. Towards web usability: Providing web contents according to the readers contexts. In *Proceedings of the First and Seventeenth International Conference on User Modeling, Adaptation, and Personalization (UMAP'09)*, Lecture Notes in Computer Science, 2009.
2. S. Amer-Yahia, V. Markl, A. Halevy, A. Doan, G. Alonso, D. Kossmann, and G. Weikum. Databases and web 2.0 panel at vldb 2007. *SIGMOD Rec.*, 37(1):49–52, 2008.
3. A. Ankolekar, M. Krötzsch, T. Tran, and D. Vrandečić. The two cultures: Mashing up web 2.0 and the semantic web. *J. Web Sem.*, 6(1):70–75, 2008.
4. A. Ankolekar and D. Vrandečić. Kalpana - enabling client-side web personalization. In *Hypertext*. ACM, 2008.

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. G. Cormode and B. Krishnamurthy. Key differences between web 1.0 and web 2.0. *First Monday*, 13(6), 2008.
7. D. Huynh, S. Mazzocchi, and D. R. Karger. Piggy bank: Experience the semantic web inside your web browser. *J. Web Sem.*, 5(1):16–27, 2007.
8. P. S. (Innsbruck). Website localization and translation. In H. G.-A. S. S. N. (Saarbrücken), editor, *MuTra 2005 - EU-High-Level Scientific Conference: Challenges of Multidimensional Translation*, May 2005.
9. T. Jevsikova. Localization and internationalization of web-based learning environment. In R. Mittermeir, editor, *ISSEP*, volume 4226 of *Lecture Notes in Computer Science*, pages 310–318. Springer, 2006.
10. M. Mrissa, M. Al-Jabari, and P. Thiran. Using microformats to personalize web experience. In *Proceedings of the 7th International Workshop on Web-Oriented Software Technologies (IWWOST08)*, 2008.
11. T. O'Reilly. What is web 2.0? design patterns and business models for the next generation of software. 2005.
12. K. Reinecke, G. Reif, and A. Bernstein. Cultural user modeling with cumo: An approach to overcome the personalization bootstrapping problem. In *Workshop on Cultural Heritage Systems in the Semantic Web 2007*, Lecture Notes in Computer Science, 2007.
13. E. Torres. Open data in html. XTECH CONFERENCE, 2007.
14. O. D. Troyer and S. Casteleyn. Designing localized web sites. In *WISE*, volume 3306 of *Lecture Notes in Computer Science*, pages 547–558. Springer, 2004.
15. Ü. Yoldas and G. Nagypál. Ontology supported automatic generation of high-quality semantic metadata. In R. Meersman and Z. Tari, editors, *OTM Conferences (1)*, volume 4275 of *Lecture Notes in Computer Science*, pages 791–806. Springer, 2006.