

Personalised Mashups: Opportunities and Challenges

Minh Thang Dang, Vania Dimitrova, and Karim Djemame

School of Computing, University of Leeds,
Leeds LS2 9JT, UK

dang.minh.thang@gmail.com, {karim, vania}@comp.leeds.ac.uk
<http://www.comp.leeds.ac.uk>

Abstract. Web 2.0 has emerged as the business ideology and development paradigm for the next generation of web applications. Mashups are one of the most popular types of Web 2.0 applications. This paper proposes the use of personalisation techniques to enhance the functionality of web mashups. The design, implementation, and evaluation of personalised mashups is illustrated with a case study of a travel assistant that combines interactive maps with access to data from both a public resource directory and a social database. This work contributes to user modelling by (a) presenting a robust and flexible service-oriented architecture of personalised mashups, and (b) identifying opportunities and challenges mashups bring to personalisation research.

Key words: Web 2.0, mashups, application of user modelling and personalisation, service-oriented architectures, evaluation

1 Introduction

Coined in 2004 [16] as an attempt to describe a new trend of innovation-driven and user-centric web development, Web 2.0 receives nowadays strong industrial support and is influencing the way Internet technologies are being produced and deployed [1]. Key principles, which advocate harnessing collective intelligence for the creation and distribution of web content, mass aggregation of data sources, and lightweight programming models for rapid development of creative applications that enrich the users' experiences on the Web, have already made a major impact on the development of successful Web 2.0 applications. They can be categorised into: *Mashups* (applications that collect and integrate information and services from more than one source ¹), *Social community* (where people can share thoughts, exchange ideas, and make friends ²), *Social media* (user-contributed

¹ For example, My Yahoo (my.yahoo.com), Google News (news.google.com)

² For example, MySpace.com, Blogger.com, and 360.Yahoo.com

multi-media content ³), and *Desktop-like web applications* (applications on the web that provide desktop-like experience ⁴).

The time appears ripe for the user modelling community to start looking at how the Web 2.0 paradigm will affect research on personalisation and adaptation. From the four categories of Web 2.0 applications, social community systems have been given most attention by user modelling researchers. Examples include integrating tags in recommender systems [19], providing social navigation support in course recommender systems [9], generating adaptation mechanisms to maintain sustainability in online communities [5], providing community-aware mobile learning content adaptation [12], modelling users' behaviour in social systems [8]. Recently, approaches for personalisation in desktop-like applications are being developed [6]. To the best of our knowledge, user modelling research has not examined the application of personalisation techniques to mashups.

With the Web becoming more open, mashups have emerged as new compelling applications whose numbers grow steadily ⁵. Recognising the emerging potential of mashups, various initiatives have been established to facilitate mashup development [15] and to standardise their architectures in certain domains [11]. However, none of the aforementioned systems and initiatives considers the use of personalisation functionality to address information overload - one of the key challenges in the current information age [21], particularly critical for mashups because they integrate a vast amount of diverse resources in novel applications that the resource providers may have never thought of.

This paper will illustrate the use of personalisation techniques to enhance the functionality of web mashups. We will present the design, implementation, and evaluation of a mashup application which provides user-tailored access to geographic information on the web. A service-oriented architecture for personalised mashups will be proposed and exemplified in a prototype of a travel assistant which combines interactive maps with access to data from both a public resource directory and dynamically created content from a social community application. Based on the results of an evaluation study with the travel assistant prototype, we will discuss opportunities and challenges for personalised mashups.

2 Case Study: Personalised Travel Assistant

This section will illustrate the design and implementation process of personalised mashups by using a case study of a travel assistant that helps users find places to visit and events to attend.

³ For example, Wikipedia.org (encyclopaedia), Flickr.com (photos), Youtube.com (video), Digg.com (news), and Eventful.com (events)

⁴ For example, [Google Docs \(docs.google.com\)](http://docs.google.com) and [Google Calendar \(calendar.google.com\)](http://calendar.google.com)

⁵ <http://www.programmableweb.com> maintains a comprehensive list of mashups.

2.1 Domain Choice

The first step in designing a mashup is to choose an appropriate domain, for which two factors are critical: (a) availability of free access to appropriate data, and (b) potential for personalisation to add value to the application. Based on this, we identified access to geographic data as a suitable domain.

Availability of data. Geographic mashups are among the most popular mashup categories. Commonly, these applications use interactive maps for access to public data, e.g. *Chicagocrime* plots data from Chicago Police Department on Google Maps ⁶ or *Housingmaps* offers interactive search for houses to rent ⁷. Scientists have also taken advantage of map access to present their findings, e.g. using Google Maps to show data about zoology species ⁸ or employing Google Earth to illustrate distribution of avian-flu epidemic ⁹. At the heart of any geographic information system is a database [22], which within the Web 2.0 paradigm will be an open and distributed collection of heterogeneous data that can include geographic data (e.g. US zip codes), cartographic resources (e.g. *Google Maps*, *Yahoo! Maps*, *MapQuest*), and geocoded data (e.g. *Yahoo! Local Search*). In recent years, free access to such data has been provided via appropriate APIs. The following data and APIs were chosen for our case study:

- *Google Maps* ¹⁰ which is a Javascript library enabling web developers to integrate interactive maps in their applications.
- *Yahoo! Local Search* ¹¹ which allows search through the *Yahoo!* directory for places (e.g. businesses, parks, shops) by specifying a set of criteria, e.g. category, country, city, street.
- *Upcoming.org* ¹² which enables search through a social database with descriptions of various events, e.g. concerts, exhibitions, and sport activities.

These sources cover the three main types of data in geographic mashups - maps, public directories, and social content.

Potential for personalisation to add value. There are strong reasons to believe that personalisation can add value to geographic mashups where users come from the general public (hence, differ significantly in their preferences, tasks, interests, and capabilities), and finding information in vast repositories of data can be a tedious task. Jameson has shown that in such cases, personalisation techniques, such as filtering and recommendations, can be very beneficial [10].

⁶ <http://www.chicagocrime.org>

⁷ <http://www.housingmaps.com>

⁸ <http://darwin.zoology.gla.ac.uk/~rpage/ispecies>

⁹ <http://www.nature.com/nature/googleearth/avianflu1.kml>

¹⁰ <http://www.google.com/apis/maps/>

¹¹ <http://developer.yahoo.com/search/local/>

¹² <http://www.upcoming.org>

A number of user-adaptive applications for access to geographic data have been implemented, see recent reviews in [13] for examples of location-aware guides and [4] for examples of adaptive presentation of geographic information. The case study presented here was inspired by work on adaptive tourist guides which provide user-tailored information about weather, points of interest, accommodation, dining, travel, etc. [14, 17, 18]. However, a distinctive characteristics of the travel assistant presented here is that instead of accessing a centrally maintained (and often proprietary) database, we will pull data from open databases and dynamic community-created repositories.

2.2 System Architecture

A Service Oriented Architecture (SOA) has been used for the implementation of a personalised mashup in our case study. The advantages of SOAs include decentralised middleware, a strong emphasis on loose coupling, and adherence to well-supported standards [2]. The loosely-coupled nature of SOAs, together with the use of standard interfaces which hide system heterogeneity, allows for services to be invoked in defined sequences to form business processes. This is a powerful aspect of SOAs, especially they are coupled with Web 2.0 applications and mashups. In line with relevant approaches for adaptive web applications, e.g. [3], we have used Web services to implement SOA.

The travel assistant has three main components (Fig. 1): external data sources, a set of web services which maintain access to external databases and implement the system's personalisation functionality, and a client web page manager which integrates interactive maps with data provided from the web services.

External data sources are accessed by sending queries to the corresponding APIs. For example, the following queries get theme parks in New York from **Yahoo! Local Search** and rock-related events (e.g. concerts, exhibitions, festivals) from **Upcoming.org**:

```
http://api.local.yahoo.com/LocalSearchService/V3/localSearch?appid=<API_Key>
&query=theme parks&location=New York
```

```
http://upcoming.org/services/rest/?api_key=<API_KEY>&method=event.search
&search_text=rocks&location=New York
```

Web services implement the main personalisation components of the system, i.e. user modelling and adaptation. The separation of these components into

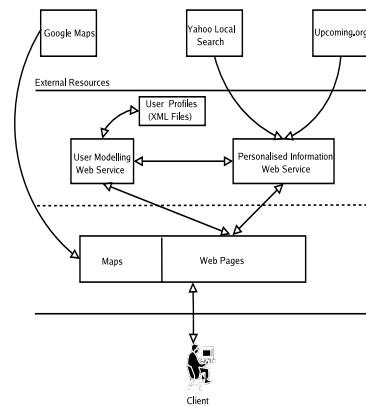


Fig. 1. The architecture of the personalised travel assistant mashup.

different Web services enables reuse and integration of personalisation functionality across applications. A *user modelling web service* (UM-WS) is responsible for maintaining a model of each user registered with the system, while a *personalised information web service* (PI-WS) adapts the information from external resources to the user model by querying UM-WS. The web services have been implemented with JAX-RPC¹³ using SOAP protocol for message passing to and from web services and Apache Axis for web services deployment. Following the Web 2.0 principles of quick prototyping and lightweight programming, we have employed simple content-based filtering and presentation adaptation techniques.

Client Web page manager integrates Google Maps with information received from PI-WS.



Fig. 2. Results after searching for places in New York. Data, which in this case is taken from Yahoo! Local Search, is filtered and re-ordered according to the user's profile.

Upon registration, users provide information about general interests (e.g. music, theatre, sports) and places of interests (e.g. bars, restaurants, museums, galleries). This information is passed to UM-WS which creates an XML-based user profile. The user can make queries by typing free text (e.g. the location they wish to visit). The client application passes each query to PI-WS which contacts UM-WS to get the user's profile and uses it to compose appropriate queries to the APIs of external resources. The results passed by the external resources are then filtered and ordered by PI-WS according to their relevance to the user's profile. For this, simple heuristic rules are used, e.g. the most recently added interest are given higher priority, ranking from external sources is taken into account, if provided. Fig. 2 shows a screen shot with places to visit tailored to a user. The user can click on a link or interact with the map to get information about a place or an event (see Fig. 3).

¹³ A Java library which enables the development of web services by using XML-based remote procedural calls.



Fig. 3. Details of a place in New York. The user can see this information either by clicking on a link from search results or on a location marker on the map.

3 Evaluation

One of the advantages of mashups is their quick development which allows potential and drawbacks of applications to be examined with users at early stages. We conducted an experimental study to: (a) test the robustness of the architecture and examine the system's performance; (b) assess the users' satisfaction with the system; (c) determine what benefits, if any, can be added by personalisation; and (d) identify future improvements.

3.1 Participants and Procedure

Two versions of the travel assistant mashup were prepared - personalised (as explained above) and non-personalised (where UM-US and PI-WS were replaced with a simple web information web service that pulled data from the external resources based only on the user's query and passed the results to the client application without any filtering or re-ordering). 60 people (including undergraduate and postgraduate students, university researchers and lecturers, and IT professionals) were invited to take part in the evaluation. They were assigned a version of the system to work with and were sent an email with the corresponding URL. 32 people chose to take part in the study, 18 of them interacted with the personalised version (Group1) and 14 interacted with the non-personalised version (Group2). Both groups had a fairly similar distribution with regard to gender and profession/age¹⁴. The users were asked to imagine planning a short

¹⁴ Male/female ration: 1.57 for Group1 and 1.8 for Group2;

Profession/age: Group1 - 33% undergraduate students younger than 24, 61% postgraduate students and researchers age 25-34, 6% faculty and professionals age 35-44; Group2 - 29% undergraduate students younger than 24, 57% postgraduate students and researchers age 25-34, 14% faculty and professionals age 35-44.

trip to New York and to use the travel assistant to find five interesting places or events. The users had to identify also two other cities to visit with three interesting places or events for each city. The participants were asked to record their time for each task and filled in an electronic questionnaire with their feedback. Next we sketch the main findings of the evaluation study, see [7] for more details.

3.2 Results

System performance. In general, both versions of the travel assistant performed well and none of the participants reported any significant problems. The users completed their tasks and provided lists with their selections. The study confirmed the robustness of the architecture described in Section 2. All participants except one ¹⁵ found the system’s speed quick or acceptable. Several users pointed out that the mashup overcame problems they had experienced with travel planning earlier, such as combining information from several places (e.g. *‘One has to go through many sites to find information about places and events, not many cities have unified portals’*) and relating search results to locations (e.g. *‘I usually have to filter lots of information found with google and then to find out where the places are located which is sometimes very difficult and time consuming’*). The seamless integration of interactive maps and data from several places was pointed by almost all users as major advantage of the system.

User satisfaction 78% of the participants in each group found the travel assistant intuitive. While only 1 participant from Group2 found the system frustrating, 4 participants from Group1 were frustrated with the personalised version. Only 5 participants (28%) would use the personalised version to plan their holiday trips and 7 users (39%) would recommend it to friends. The corresponding percentages are slightly higher for the non-personalised version - 48% (6 users) felt they would use and recommend the system. Due to the small sample, the quantitative results are inconclusive. Therefore, we analysed the qualitative questionnaire to identify what problems the users had faced with the system.

Several problems were **common for both groups**: (a) Quality of information: insufficient detail, lack of pictures, unclear descriptions (as summarised by one participant *‘It was frustrating to click on a link and find very little useful information’*); (b) Incomplete data: missing information about cities (e.g. *‘I didnt get the Gugenheim museum, which I expected and would like to see to be able to trust the system’*, *‘Searching for New York did not give me the information I expected, e.g. the Statue of Liberty and Central Park were missing.’*), there were blank pages at times when no results were found; (c) Peculiarities: wrong locations (e.g. *‘I searched for London, it automatically assumed I meant London, Kentucky but I did not want to go there’*), strange places/events (e.g. users were recommended New York police department or San Gennaro Festival held

¹⁵ This participant did not specify any problems with the system but mentioned that they hardly ever used Internet before for travel planning

in 2004); (d) Interface problems: map window was too small, map navigation was cumbersome, items should be grouped in categories.

These problems affect the users' trust in the system and should not be ignored by the developers of personalised mashups. While (d) can be fixed easily, (a)-(b) are typical for Web 2.0 applications which provide access to dynamic collections with user-created web content, and (c) directs to the need of semantic approaches to enhance geographic search.

Certain problems were experienced **only by Group1**: (a) Recommendations limited the choices (e.g. *'I would have liked to receive suggestions for popular places, e.g. the Statue of Liberty or Empire State Building, even though they may not match my profile'*); (b) Recommendations did not correspond to the user's profile (e.g. *'The system did not provide me events that matched with my interests'*); (c) Lack of control (e.g. *'I couldn't update my profile after I registered', 'I would like to extend/limit the amount of recommendations I would be given'*); (d) Limited visibility (e.g. *'I was not sure why I was given some recommendations'*); (e) Need for group recommendations (e.g. *'In my profile, I say my interests are sleeping and eating. However, in case I go to NY with my friend and she wants to go to cinema, how can we find places for both of us'*).

These problems were specific for the personalised version. While (a) points at typical limitations of content-based filtering which can be addressed with collaborative filtering, (b) directs to the need of enhanced algorithms for comparing similarities between user profiles and open, community-created categories in Web 2.0 applications, (c)-(d) refer to usability treats brought by personalisation [10], and (e) points out that group recommendations can be beneficial in mashups.

Finally, we identified problems experienced **only by Group2**: (a) Irrelevant recommendations (e.g. *'Some of the suggestions were completely outrageous: a charity organisation, a parking authority'*); (b) Hard to find relevant data (e.g. *'I could not find my favourite topics', 'I would have liked to know whether there was a classical concert in Washington between 2nd and 5th of September'*). These problems point at opportunities for personalisation to enhanced the users' experiences with intelligent mashups.

Benefits of personalisation. Most participants from Group1 (78%) found the recommended information relevant to their interests, as opposed to less than half of the participants from Group2 (43%) with the same opinion. These results point at potential of personalisation to add value to mashups.

Contrary to our initial expectations, the users with the personalised system did not perform the tasks quicker. The average time for task 1 was 7.5 min (STDEV=4.7) for Group1 and 8.9 min (STDEV=5.9) for Group2, while the average time for task 2 was 10.9 min (STDEV=4.9) for Group1 and 5.9 min (STDEV=5) for Group2. This can be explained with problems related to the personalised version, as described above. We expect that the benefits of personalisation will be stronger when more data resources are integrated and advanced personalisation techniques are employed.

Future improvements. Based on their experience with the system, the users pointed at future improvements which could be grouped in three categories: (a) Searching through more data sources: include information about places and events from other data repositories and add relevant information about flights and public transport; (b) Expanding the functionality: add route planning and driving directions, suggest optimal path between locations in terms of distance and time, compare place/event location with accommodation address, search for locations within convenient distance; (c) Improving the personalisation algorithms: integrate collaborative filtering, add web services that dynamically refine the user's profile, employ semantic-enhances services to find relevant information (e.g. some users pointed that an ontology would be helpful).

4 Discussion

4.1 Opportunities for Personalised Mashups

The personalised travel assistant is an initial prototype which illustrates the application of user modelling techniques to filter relevant information from large repositories of data accessed via intelligent mashups. Our study showed evidence that personalisation is needed and that it can improve the users' experiences with mashups. There is a strong reason to believe that SOAs provide robust and flexible way to integrate personalisation functionality in mashups. The architecture proposed here can be extended to provide access to additional external resources or to enhance UM-WS and PI-WS with advanced personalisation techniques.

Although the case study was limited to a particular domain, the domain selection criteria followed can be applied to other subject areas where free data is available and personalisation can add value to the applications. We believe that at present personalised mashups should easily be implemented for news and online shopping applications. With more and more data made available, we will see personalised mashups combining several types of media (e.g. text, pictures, videos) in applications that provide web, as well as mobile access.

Our case study illustrated the employment of simple user modelling techniques, such as user profiles and content-based filtering, for rapid development of personalised mashups. With more user modelling services being available, we expect to see creative applications which take advantage of advanced personalisation techniques. Opportunities include integrating collaborative filtering services (such as those offered by Amazon) in a shopping assistant mashup, adding group recommendations for trip planning, or using presentation adaptation techniques to combine data from several media. We believe that personalised mashups will offer exciting opportunities for the deployment of user modelling techniques.

4.2 Challenges for Personalised Mashups

Although our case study was fairly simple, it pointed at several challenges that mashups bring to personalisation research. The moving from central databases

with relatively stable data fields and controlled information quality to open, dynamic, and community-driven heterogeneous collections imposes major challenges to web applications in general, and to personalisation research, in particular. One of the most critical problems is the diversity of semantics and XML Schemas across data repositories accessed in a mashup application. [20] suggests building a conceptual model (ontology) of the domain in question and using that ontology to annotate the XML Schema of the service inputs. If the approach is followed, user modelling and adaptation web services have to consult the ontology (or several ontologies) and mappings when deciding whether information is relevant to a user's profile. The openness of the web requires tolerance of different viewpoints, especially when community-driven content is employed. Although collaborative filtering provides techniques for accommodating community opinions, it requires availability of rankings, which may come from several sources and can have different range and semantics.

Another fundamental challenge to personalisation, particularly relevant for mashups, is dealing with incomplete and noisy data. As shown in our study, this can lead to offering inappropriate or insufficient information and can reduce the users' satisfaction and trust in the system. There can be quick partial solutions, some of which were pointed by the evaluation of the travel assistant, e.g. including *default* or *popular* data, and integrating descriptions from several places by employing some techniques from adaptive hypermedia. However, in order to fully address this challenge, more sophisticated approaches would be needed. For example, taking into account the user's opinions about quality and relevance of data and incorporating an appropriate trust model.

Service-oriented personalisation architectures face challenges related to service availability and reliability. Our study did not encounter such problems but many cases are reported within SOA research. For example, how will an application proceed if a service, e.g. user modelling or adaptation service, is unavailable when a request is sent to it. This may require some *caching* of earlier information over a threshold period, or adding some degree of uncertainty in the adaptation decisions taken by every web service.

Finally, personalised mashups can expose usability treats linked to user modelling and automatic adaptation [10]. For example, our study pointed at problems with predictability and transparency. Other problems may include obtrusiveness or privacy. It can be expected that relevant user modelling research in these areas will be applicable to personalised mashups.

References

1. Web 2.0 summit: <http://www.web2con.com/web2006/>.
2. G. Alonso, F. Casati, and H. Kuno. *Web Services: Concepts, Architectures and Applications*. Springer, 2004.
3. L. Ardissono, A. Goy, G. Petrone, and M. Segnan. Interaction with web services in the adaptive web. In *Proc. of AH04*, pages 14-23. Springer, 2004.

4. A. Bunt, G. Carenini, and C. Conati. Adaptive content presentation for the web. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The adaptive web: Methods and strategies of web personalization*. Springer, 2006.
5. R. Cheng and J. Vassileva. Design and evaluation of an adaptive incentive mechanism for sustained educational online communities. *User Modelling and User-Adapted Interaction*, 16(2/3):321–348, 2006.
6. A. Cristea, M. Hendrix, and W. Nejdl. Automatic and manual annotation using flexible schemas for adaptation on the semantic desktop. In *Proc. of the 1st European Conference on Technology Enhanced Learning ECTEL06*, pages 88–102. Springer, LNCS, 2006.
7. M. T. Dang. Personalisation through web services meets the next paradigm of the web - a case study in geographical information systems, Sept. 2006.
8. T. Erickson. Inhabited models: Supporting coherent behavior in online systems. In *Proc. of UM05*, pages 5–8. Springer, LNCS, 2005.
9. R. Farzan and P. Brusilovsky. Social navigation support in a course recommendation system. In *Proc. of AH06*, pages 91–100. Springer, LNCS, 2006.
10. A. Jameson. Adaptive interfaces and agents. In J. A. Jacko and A. Sears, editors, *The Human-Computer Interaction Handbook. Fundamentals, Evolving Technologies and Emerging Applications*. LEA, 2003.
11. A. Jhingran. Enterprise information mashups: integrating information, simply. In *Proc. of the 32nd international conference on Very large data bases, VLDB06*, pages 3–4. VLDB Endowment, 2006.
12. R. Klamma, M. Spaniol, and Y. Cao. Community aware content adaptation for mobile technology enhanced learning. In *Proc. of the 1st European Conference on Technology Enhanced Learning ECTEL06*, pages 227–241. Springer, LNCS, 2006.
13. A. Krueger, J. Baus, D. Heckmann, M. Kruppa, and R. Wasinger. Web-based mobile guides. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The adaptive web: Methods and strategies of web personalization*. Springer, 2006.
14. R. Malaka and A. Zipf. DEEP MAP challenging IT research in the framework of a tourist information system. In *Proc. of the 7th International Congress on Tourism and Communication*, 2000.
15. S. Murthy, D. Maier, and L. Delcambre. Mash-o-matic. In *Proc. of the 2006 ACM symposium on Document engineering, DocEng06*, pages 205–214, New York, NY, USA, 2006. ACM Press.
16. Tim O'Reilly. What is web 2.0:
<http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html> [last accessed: 26/10/2006].
17. C. Rinner and M. Raubal. Personalized multi-criteria decision strategies in location-based decision support. *Journal of Geographic Information Sciences*, 2:149–156, 2004.
18. B. Schmidt-Belz, S. Poslad, A. Nick, and A. Zipf. Personalized and location-based mobile tourism services. In *Proc. of Mobile HCI02, Workshop on Mobile Tourism Support Systems*, 2002.
19. S. Sen, S. Lam, A. M. Rashid, D. Cosley, D. Frankowski, J. Osterhouse, M. Harper, and J. Riedl. Tagging, communities, vocabulary, evolution. In *Proc. of CSCW06*, 2006.
20. A. Sheth, K. Verma, and K. Gomadam. Semantics to energize the full services spectrum. *Communications of the ACM*, 49(7):55–61, 2006.
21. P. Waddington. Dying for information? a report on the effects of information overload in the uk and worldwide. Technical report, Reuters, UK, 1998.
22. M. Worboys and M. Duckham. *GIS - A Computing Perspective*. CRC Press, 2004.