Really Simple Mash-Ups

Yvonne Dittrich, Peter Madsen, and Rune Rasmussen

IT-University of Copenhagen
Rued Langaards Vej 7
2300 Copenhagen S, Denmark

Abstract. Mash-ups are applications – typically web applications – designed by combining data from several web services into a new tool or expression. New mash-ups emerge every day. Different End-User Development environments for mash-ups are available. However, the identification of mashable web-services, their exploration and the definition of data aggregation from heterogeneous web services still requires familiarity with programming representations and environments. Can the provision and usage of mashable web-services become as easy as the provision and usage of RSS feeds? What is needed is (1) a web server that allows providers of mashable web services to define data-formats describing their interface and users of the same web services to identify mashable data sources and (2) a simple standard to annotate the result of web services so that mash-ups editors can support even non professional developers in pruning and aggregating the data according to their needs. The article presents our bid on how this vision can be realized.

Keywords: Mash-ups, Cooperative End-User Development, Web-services.

1 Introduction

Mash-ups composed from existing web services into new functionality, are becoming a more and more appreciated way of personalizing home pages, creating picture galleries of your own photos and providing different ways of presenting e.g. search results. Several end-user environments, like Yahoo Pipes, allow users to easily combine known building blocks.

However, the discovery, exploration and aggregation of new mashable web services is still a bottleneck. This is especially relevant for mash-ups combining heterogeneous data sources, e.g. in data source, consumer, or enterprise mashups [8]. And maybe this is the reason why only few services are used in mash-ups. Existing standards – like the Dublin Core Metadata – that could support the data aggregation are cumbersome both for suppliers and users of mashable data sources. Is there a possibility to make the provision usage of mashable web services as easy as the provision and presentation of an RSS-feed?

This short paper presents the result of our exploration of this question. It does not address the End-User interface of mashing-up services but focuses on an infrastructure that supports the sharing and combination of mashable services through Mashup editors like Marmite [9]. The following section explores the existing possibilities for

M.F. Costabile et al. (Eds.): IS-EUD 2011, LNCS 6654, pp. 227–232, 2011.

sharing mashable web services and support for aggregation of results. Thereafter we present our concept, consisting of (1) a web server supporting a community process defining data formats for provision and consumption of web services, the publication of mashable web services, and the possibility for developers to search for suitable web services to use for their mash-up ideas; and (2) a standard for annotation of web services so that the results of different web services can easily be aggregated. Our proposal of a RSM standard and infrastructure is supported by two prototypes; the RSM web server allowing for definition of data formats and a query by example based editor demonstrating how the information in the RSM annotations to web services can be used for aggregating data.

2 Existing Support for Sharing of Mashables and Data Aggregation

Mash-ups are not a new phenomenon, and data exchange through web services and other techniques has been done and supported for a long time. Below, we shortly discuss the three initiatives closest to our vision.

programmableweb.com
The programmableweb.com [1] is a web site allowing to upload and download both mashable services and mash-ups. The site lists at the time of writing 2151 mashable services, 'APIs'. Each API is described with a short comment. The statistics indicate that only a fraction of the uploaded mashables is used by the mash-ups uploaded on the same site. The activity on the web site shows that it provides a well-received location service for APIs and for mash-ups. However, there is no support for potential users of mashable services to explore the interface prior to accessing the service.

The Microformat Community
The microformats community [7] defines and shares xhtml-based formats for marking web content in order for automatically being able to identify and collect information e.g. about events from webpages and blogs. Part of the idea presented here is doing the same for mashable web services. On top of the sharing of formats, functionality to locate mashable services and a common way to describe the results of web-services to allow for data aggregation is needed.

The Resource Description Framework and the Dublin Core Metadata
The Resource Description Framework (RDF) is a metadata format developed by the World Wide Web Consortium (W3C). [2] The RDF provides an ontology and knowledge management inspired notation allowing to describe the contents of web documents in the form of 'subject-predicate-object' statements. A net of such statements can be depicted as a tree.

The Dublin Core Metadata Initiative develops and maintains an ISO standard for data exchange that is formulated using the RDF notation. [3] On first glance, it sounds like a good idea to use these standards: RDF schemas and descriptions can be formulated in XML. There exist a XML version of the Dublin Core Metadata (DCM). Why not just annotate the result of web service calls with the suitable RDF description and use the description to merge and aggregate data from different web services? There

already exists a standard for 'Semantically Annotated' web services: SA-REST. [4] The output from web-services is translated, 'lifted', to a common RDF schema, merged, pruned, aggregated, and translated back, 'lowered', to serve as input for another service. A more lightweight version, WSMO-Lite, is presented in [10]: web services can be annotated with among other information interface, functional and behavioral specifications.

Discussion
The RDF schemes have the reputation of being unnecessarily verbose. [5] The logic of subject-predicate-object triples might provide an additional hinderance for providing a for programmers of mashable services easily accessible representation for data merging, pruning and aggregation.

For the exchange of business critical data, like e.g. electronic invoicing or the like, this approach is a necessary one. However, for a developer who contemplates to make his recent fun development available for others, the effort to understand the logic of RDF schemes and find a fitting ontology would provide an overhead he might not want to take. Even if the developer does not need define and maintain 'lifting' and 'lowering' schemes this might provide a hinder to annotate his service.

An additional problem is that the DCM cannot easily be extended. The DCM has the status of an ISO standard. That means, there is a committee that is steering the development of the standard. New schemes undergo review and revisions before they are finally included in the standard and published for usage. If no fitting RDF scheme is available, one would need to submit an addition to the right standardization committee. Furthermore, the usage of RDF, SA-REST, and maybe the DCM standards does not help with sharing and locating mashable services.

Is there a possibility to combine the advantages the three above presented approaches? Can we have a lightweight definition of formats combined with a *really* lightweight annotation mechanism and a location service for mashable services?

3 Really Simple Mash-Ups

The discussion above indicates that the existing solutions are not satisfying for a simple way to share mash-ups and support the aggregation of data from heterogeneous services.

What would be needed is the possibility to (a) define common formats in a community process without too much overhead and organizational red tape, (b) easily publish mashable web services; (c) to match the existing interfaces of web services with the common format in case that the interface cannot be adjusted; (d) locate services and information about the format they use; and (e) access to the meta-data necessary to aggregate the output.

The Really Simple Mash-up concept consists of a web server, the global RSM dictionary, solving (a), (b) and (d), and a format for annotations of web services solving (c) and (e). Both the RSM dictionary and the annotation format depend on the concept of an RSM-key, a way to specify the format of an atomic piece of data. All three ingredients are presented below. Feedback is welcome.

3.1 RSM Keys

RSM keys are defined in the global RSM dictionary and are used to annotate the result of web services. RSM keys are defined slightly different in the annotated documents than in the dictionary:

Attribute	Used in RSM dictionary	Used in annotated document
Name	Required	Required
Tag	Required	Required
Format	Required	Optional
Descriptor	Required	Not applicable
Value	Not applicable	Required
Local Name	Not applicable	Optional
Id	Required	Required

Name. The name should refer to the concept the data describes, e.g. "GPSposition". The RSM Dictionary guarantees the uniqueness of the name.

Tag. Tags are a non-rigid way of grouping and relating RSM tags. Its tags will relate a RSM key to one or more real world domains.

Format. This element contains the regular expression used to validate whether a specific value complies with the format of a RSM key. In document annotations it supports validation of values without access to the RSM dictionary.

Descriptor. Descriptors can be used to assign descriptive information to a RSM key, like date of creation, author, date of modification, or description. A key can contain an arbitrary amount of Descriptor elements.

Value. This element contains the actual key value. It is used to annotate a specific data set.

Local Name. A local name can be assigned for an annotation to describe a local appliance of a key concept, e.g. a RSM key with the Name "GPSposition" could have a local name of "InterestPoint" either to distinguish it from another usage of "GPSposition" in the same document or simply to help mash-up creators.

Id. Unique identification of a RSM key, required in both annotation and definition.

3.2 A Global RSM Dictionary

The global RSM dictionary needs to support three basic needs or use cases: The definition of RSM keys and their formats, the location of keys for annotation of mashable web service results, the sharing and the location of mashable web services.Tags denoting application domains are used to group and relate RSM keys in a non-rigid way. Rather than applying a screening or review process it supports reaching a core set of commonly used RSM keys by inviting users to reuse concepts already defined. The only restriction is that the RSM key names should be unique.

The RSM dictionary will encourage data providers to register their services and link them to the relevant domains as well as the RSM-keys they use. It might expose information about the usage to help data providers choose the most accepted RSM-keys when annotating their data services. The RSM dictionary should hold additional

searchable information to assist the process of identifying RSM enabled mashable web services. For a more thorough presentation of the design see [6]. A prototype for the RSM dictionary is available from the authors.

3.3 Annotating Web Service Results

We propose to use an annotation approach rather than ask providers to change the interface of their services. The web services might have been designed based on in-house policies or to fit with specific local requirements. The overhead for providing the services to other users should be kept as low as possible. We therefore follow the SA-REST and WSMO-lite strategy to enrich the services with RSM keys.

```
<?xml version="1.0" encoding="utf-8"?>
<books xmlns:rsm=http://www.reallysimplemashup.org/schema/rsm/v1.0/
       xmlns="http://www.mydomain.com">
  <book>
    <title>Title A</title>
    <author>John Doe</author>
    <isbn>1111111111;2222222222</isbn>
    <rsm:key name="isbn" tag="book|library" value="1111111111"
             id="A5A45C15-9D52-46D6-A2B2-5972FD34427C" />
    <rsm:key name="isbn" tag="book|library" value="2222222222"
             id="A5A45C15-9D52-46D6-A2B2-5972FD34427C" />
    <rsm:key name="title" tag="book" value="Title A"
             id="C73C2A21-1060-48A9-8EDE-B70EA852A227" />
  </book>
  <book>
    <title>Title B</title>
    <author>Jane Doe</author>
    <isbn>3333333333</isbn>
    <rsm:key name="isbn" tag="book|library" value="3333333333"
             id="A5A45C15-9D52-46D6-A2B2-5972FD34427C" />
    <rsm:key name="title" tag="book" value="Title A"
             id="C73C2A21-1060-48A9-8EDE-B70EA852A227" />
  </book>
</books>
```

Example of a XML document after RSM annotation.

Web service results can come in different notations. By defining the RSM keys notation independent of the format, we open up for defining annotations for different notations. Above we present our annotation format for XML. A similar one can be designed for JSON or any other API format.

4 Discussion and Conclusion

The Really Simple Mash-Up infrastructure proposed here provides a low overhead and low bureaucracy approach to sharing formats for mashable web services. It supports acquisition and aggregation of data from heterogeneous web services when designing new mash-ups. The definition of RSM keys and formats is left to the users of the RSM dictionary in a true web 2.0 spirit. By proposing an annotation of the web

service results rather than asking providers of mashable services to change the interface, we minimise the additional effort for providers of services.

As the focus was on data acquisition and aggregation, we did not address how to use the RSM keys for specifying input parameters. Another extension would be to provide the functionality of the RSM dictionary as a mashable service, described by RSM keys. Mash-Up editors could then seamlessly include the RSM functionality.

Any design, however, has to be tested with real users and real applications. We did only develop a very basic proof of concept prototype (e.g. there is presently no support for registration and location of RSM-annotated services). If the RSM dictionary and annotation format would be adopted and used by a broader community the design would for sure be refined. The concrete design of the web site, the keys, and the annotation format might be up to discussion. The basic ideas, however, we believe will hold beyond our proof of concept prototypes.

Acknowledgements

Thanks to the anonymous reviewers for their valuable feedback.

References

1. The Programmable Web, http://www.programmableweb.com/ (last accessed January 2011)
2. Ressource Description Framework (RDF), W3C Semantic Web Activity, http://www.w3.org/RDF/ (last accessed January 2011)
3. Dublin Core MetaData Initiative, http://www.dublincore.org/ (last visited January 2011)
4. Sheth, A.P., Gomadam, K., Lathem, J.: SA-REST: Semantically Interoperable and Easier-to-Use Services and Mashups. IEEE Internet Computing 11(6), 91–94 (2007)
5. Horrocks, I., Schneider, P.P., van Harmelen, F.: From SHIQ and RDF to OWL: The making of a web ontology language. J. of Web Semantics 1, 7–26 (2003)
6. Madsen, P., Rasmussen, R.: Really Simple Mash-Ups. Final project on the Diploma in IT program. Supervised by Y. Dittrich. IT University of Copenhagen (2009)
7. Microformats, http://microformats.org/ (last accessed March 2011)
8. Wong, J., Hong, J.I.: Making mashups with marmite: towards end-user programming for the web. In: Proc. of CHI 2007, pp. 1435–1444. ACM, New York (2007)
9. Lee, C.-J., Tang, S.-M., Tsai, C.-C., Chen, Y.-C.: Toward a new paradigm: Mashup patterns in web 2.0. WSEAS Trans. Info. Sci. and App. 6(10), 1675–1686 (2009)
10. Vitvar, T., Kopecky, J., Zaremba, M., Fensel, D.: WSMO-Lite: Lieghtweight Semantic Descriptions for Services on the Web. In: Proc. of the 5th ECOWS, pp. 77–86. IEEE, Los Alamitos (2007)