G.R. Gangadharan
Telematica Institute, Enschede, The Netherlands, gr@telin.nl
Michael Weiss
Carleton University, Ottawa, Canada, weiss@sce.carleton.ca
Vincenzo D'Andrea
University of Trento, Trento, Italy, dandrea@disi.unitn.it
Renato Iannella
National ICT Australia, Brisbane, Australia, renato@nicta.com.au

EXPLORING THE INTELLECTUAL RIGHTS IN THE MASHUP ECOSYSTEM

Abstract: Over the last three years there has been a rapid proliferation of mashups as an emerging paradigm of Web 2.0. Mashups are applications that combine data and services provided through several open APIs, allowing the quick creation of custom applications by users. However, the intellectual rights associated with services and data associated with mashups are not focused intensively. In this paper, we explore the actors and roles involved in a mashup ecosystem and analyze the intellectual rights associated with mashups.

Keywords: Mashups, Web services, Intellectual rights, Licenses, Rights expression

1. Introduction

Mashups are the new hybrids of interactive applications that are created from combining the various service components and data sources to provide new value or adding value in some way. Mashups enable developers to "mix and match" data and user interface elements from different online information sources to create new value offers. Since they often only have a short lifespan and are created for a specific group of users (often identical to the developers), they are also known as situational applications [1].

A mashup is an application that combines data or services obtained from multiple open APIs. The first mashup ever was a combination of a service that scraped a housing website with the Google Maps API. Interestingly, at that point in time, the Google Maps API was still considered a closed API, and, technically, the mashup was in violation of its license agreements. The subsequent opening of the API by Google set a precedent for other APIs to become open in a similar vein. Mashups are relatively simple to create, which

enables the rapid exploration of different combinations of APIs by developers and users alike. Thus, one would expect the rate at which applications that are well-aligned with user needs are created in the form of mashups to be rather high in comparison to more traditional ways of building software. As mashups are proliferating seamlessly, it becomes significant to understand the relationship among the actors of a mashup ecosystem and the permissions and the prohibitions imposed by the licenses of open APIs that governs a mashup.

The remainder of this paper is organized as follows. In Section 2, we describe actors and roles involved in a mashup ecosystem. In Section 3, we present a scenario that illustrates the creation of a mashup involving open APIs. We describe possible intellectual rights that can arise in combing open APIs as a mashup in Section 4. In Section 5, we describe open API licensing clauses in a machine interpretable form that can be perceived as a step towards finding the compatibility between licensing clauses of several open APIs involved in a mashup and present our conclusions in Section 6.

2. An Ecosystem of Mashups: Actors and Roles

A mashup is crated by combing several open APIs integrating services and data sources. A mashup ecosystem constitutes of following actors [2]: ***Consumers*** are the end users of a mashup. ***Mashup Developers*** develop mashups by combing data sources and services using mashup technologies or mashup platforms. Mashup developers compose several open APIs into mashups. Developers choose between competing providers of a required service. ***Data Providers*** provide services and data exposed through open APIs to developers to build mashups. Service offerings from different data providers that provide the same type of service (for example, a mapping service) compete. Figure 1 illustrates a mashup ecosystem that constitutes mashup developers, data providers, and consumers.

As the number of APIs, and thus the complexity of selecting mashups and the value perceived by businesses of creating mashups increased, tool providers entered the ecosystem to fill the void. Initially, these were graphical tools (such as QEDWiki (http://services.alphaworks.ibm.com/qedwiki/) from IBM) to simplify the composition of APIs into mashups, but tool providers quickly also started to offer marketplaces for APIs and mashups. At present, there is as yet no leading tool provider, nor a leading marketplace that could serve all user needs.

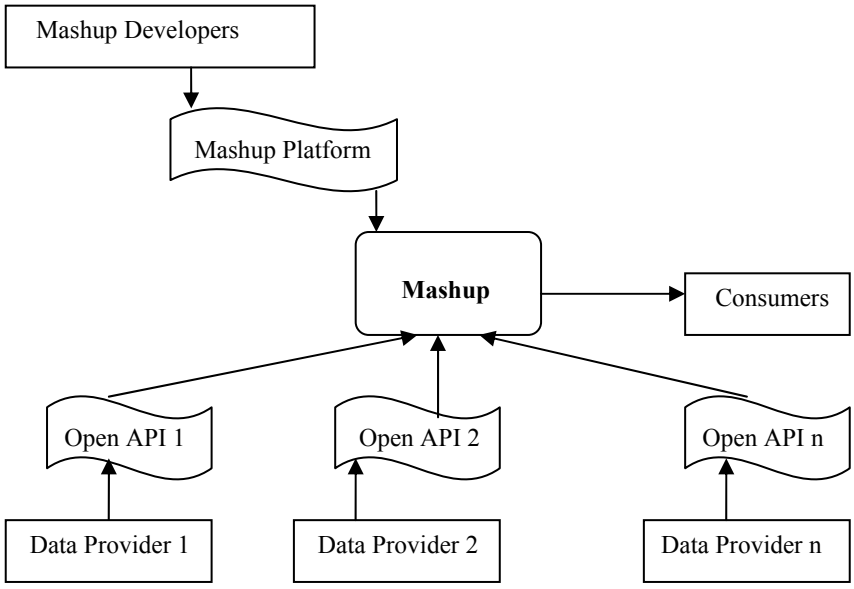

Figure 1. A mashup ecosystem

In a closed API access is only granted to select developers, typically after their applications built on top of the API have undergone some kind of certification process. Conversely, anyone is allowed to access an open API and build any kind of application on top of the API [3]. A mashup is an application that combines data or services obtained from multiple open APIs.

In our earlier work [4], we have examined the structure of a mashup ecosystem and its growth over time. Using network analysis we created models of a mashup ecosystem that allowed us to gain a deeper understanding of its structure and dynamics.

3. A Motivating Scenario

We present a scenario in which a mashup can be created by combining a *birdwatch* service API and a mapping service API. This proposed mashup, in fact, uses a *birdwatch* service offered by a service provider and a geographical map data source offered by another service provider. The combination of the two information sources would create a map where places are marked with the species of birds that are likely to be present in that location.

A *birdwatch* service (provided by International Birdwatching Network (IBN) [1]) provides two main contexts of use: data input, a free operation allowing a birdwatcher to record their observations, and data output, a pay-per-use operation providing a list of birds related to a given location. These operations (and parameters) are described as follows.

- *birdin(Location place, String birdname, Integer MemberID):* Any member of a IBN can invoke this operation for recording his/her observations for a bird on a given location. Given inputs by a birdwatcher are geographical location, name of the bird, an optional description about the bird (where *Location* is a complex data type consisting geographical coordinates *(x,y)* and Θ (an angle indicating the direction where a bird is observed)). The inputs are stored in a database owned by IBN with the date and time of entry. The member transfers data ownership to the IBN database.
- *birdout((Zone viewpoint)→ [Location place, String bird]):* An operation that receives a viewpoint from a birdwatcher and retrieves an array of locations and watchable birds in that surrounding location. The *Zone* is a complex data type consisting geographical coordinates *(x,y)* and *radius* (a distance specified by a birdwatcher wishing to watch birds within the circumference)). The array of locations (with geographical coordinates *(x,y)* and Θ) and bird details (bird name and bird description with a timestamp) are retrieved from the IBN database.

Assume that the *birdwatch* service is licensed as follows.

1. *The service can be composed with other services.*
2. *The service costs 0.10 Euros per use of the service for the birdout operation while the birdin operation is offered as free of cost.*
3. *The service provider describes the following warranties for the service:*
    - *Mean response time : 10-20 milliseconds*
    - *Mean availability rate: 99%*
4. *The service provider will defend the consumer from any action based on a claim that the use of the service in accordance with the given service license infringes intellectual property rights of any third party.*
5. *The service provider is not liable for any kinds of functional or non-functional errors in the infrastructure and networks of the service.*

Geographical map images are usually provided by certain third party providers. These images are used by map service providers through APIs. In our

[1] International Birdwatching Network is a fictitious name of a service provider for our illustrated *birdwatch* service.

proposed mashup, we use Google maps as a geographical data source. Geocoding data for map content in Google Maps is provided under license by Navteq North America LLC (NAVTEQ) and/or Tele Atlas North America, Inc. (TANA) and/or other third parties, and subject to copyright protection and other intellectual property rights owned by or licensed to NAVTEQ, TANA and/or such other third parties.

Some of the significant terms of Google maps API license are as follows.

- *Google Maps, including local search results, maps, and photographic imagery, is made available for your personal, non-commercial use only.*
- *Google makes no representations or warranties regarding the accuracy or completeness of the information provided by these third parties.*

In our proposed mashup, we use the *birdout* operation of the *birdwatch* service. Given viewpoint coordinates and radius by a birdwatcher, she retrieves an array of locations and watchable birds from the IBN data source. This array of locations is marked on the Google maps as various colored balloon markers depending on bird details.

4. Intellectual Rights Issues in Mashups

Combining information (services and data) from several sources raises several issues related to intellectual rights in mashups. As the concept of mashups is currently being in its nascent stage, service and data providers often underestimate the relevance of these issues [5].

Today, the main techniques used by developers for gathering the data and information required for building mashups are by screen scraping or through APIs exposed by service/data providers. Screen scraping refers to a process of extracting data from the display of another program, intended primarily for final display to a human user [6]. This practice clearly violates the vast majority of web site user agreements and thus generally constitutes an infringement. Accessing the database information through an exposed services interface will not infringe any copyrights as long as the terms and conditions of the given API are abided.

In addition to directly violating the license of the information sources, mashup developers can infringe the copyright of data by using the database in whole or combining more several information sources having incompatible licenses. The development of mashups can even make patent and trademarks infringements [7].

As a mashup can be created by using several open APIs which are provided by several data providers, a mashup developer agrees to comply with all the

individual licenses provided by these open APIs. The resulting mashup should comply with these license clauses. The complexity of licensing compatibility increases directly with the number of open APIs that a mashup mashes with itself.

Figure 2 depicts the mashup ecosystem with the licensing interconnections among the actors. *L(MD,DPn)* indicates a license provided by a data provider to a mashup developer. *L(MD,Con)* indicates a license provided by a mashup developer to consumers. However, *L(MD,Con)* is not focused for creation of mashups.

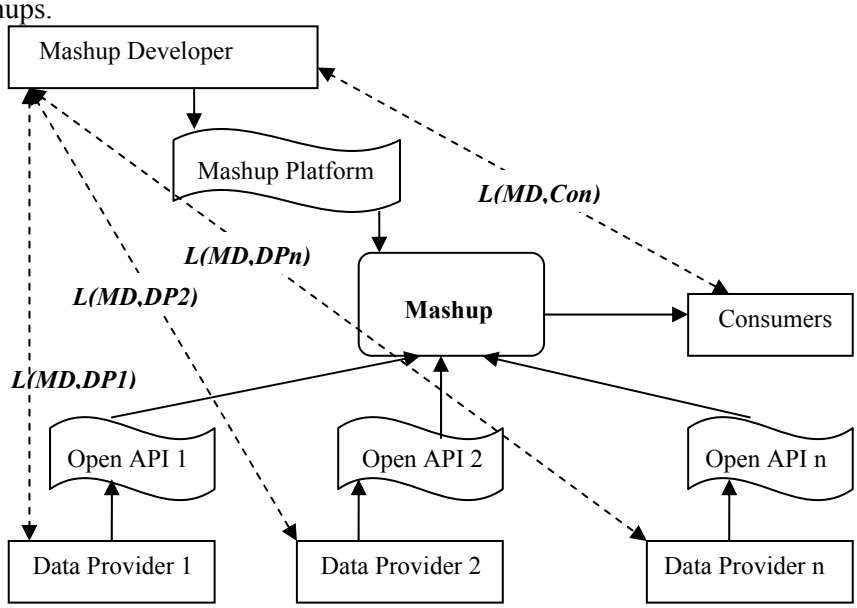

Figure 2. Licensing interconnections among the actors of a mashup ecosystem (dashed lines show the licenses between mashup developers and data providers/consumers)

5. On Resolving the Intellectual Issues of Mashups

As open APIs expose services, the concept of service licensing [8] applies to open API licenses. A service license includes all transactions between the licensor and the licensee, in which the licensor agrees to grant the licensee the right to use and access the service under predefined terms and conditions. We can express a service license using ODRL-S[2] [9].

[2] ODRL-S is a language profile extending the Open Digital Rights Language (ODRL) to implement the clauses of service licensing in machine interpretable form. ODRL-S is nominated by the ODRL initiative for approval as a standard (http://www.odrl.net/Profiles/Services/).

Following is an ODRL-S representation of the given ***birdwatch*** service license.

```
<!-- Namespace declarations go here -->
1   <o-ex:offer>
2     <o-ex:permission>
3        <sl:composition/>
4     </o-ex:permission>
5     <o-ex:requirement>
6      <o-dd:peruse>
7       <o-dd:payment>
8        <o-dd:amount o-dd:currency=EUR> 0.10
9        </o-dd:amount>
10      </o-dd:payment>
11     </o-dd:peruse>
12    </o-ex:requirement>
13    <o-ex:requirement>
14     <sl:warranty>
15      <sl:performance>
16        <sl:responsetime>
           <o-ex:constraint> <o-dd:range>
              <o-dd:min>10.0</o-dd:min>
              <o-dd:max>20.0</o-dd:max>
           </o-dd:range> </o-ex:constraint>
17        </sl:responsetime>
18       </sl:performance>
19       <sl:reliability>
20        <sl:availabilityrate>
           <o-ex:constraint> <o-dd:range>
              <o-dd:min>99.0</o-dd:min>
           </o-dd:range> </o-ex:constraint>
21        </sl:availabilityrate>
22       </sl:reliability>
23      </sl:warranty>
24       <sl:indemnity>
25       <sl:thirdpartyinfringementsclaims/>
26       </sl:indemnity>
27    </o-ex:requirement>
28  </o-ex:offer>
```

Following is a representation of some of the terms of Google Maps API in ODRL based form[3].

[3] The given license in ODRL based representation does not represent the complete Google Maps terms of use. Furthermore, the machine interpretable form does not represent the views of Google and/or associated third party sources and the views of ODRL initiative.

```
<!-- Namespace declarations go here -->
1 <o-ex:offer>
2   <o-ex:requirement>
3     <o-cc:noncommercialuse/>
4   </o-ex:requirement>
5 </o-ex:offer>
```

Compatibility between services is one of the active research areas in service oriented computing [10]. The present researches on the compatibility of services have been focused on the matching of functional properties of services. An interesting approach for matching non-functional properties of Web services represented using WS-Policy is described in [11]. The most comprehensive work on automated compatibility analysis of WSLA service level objectives is elaborated in [12]. However, license clauses are not simple as in the case of service level objectives of WSLA or policies of WS-Policy. The problem of licensing compatibility is difficult to resolve automatically as license clauses are generally written in a natural language (like English) and contains highly legalized terms, sometimes even difficult for the end users to understand. In our earlier work [13], we have analyzed the compatibility of service licenses by describing a matchmaking algorithm.

Our proposed mashup combines Google Maps API with ***birdwatch*** service. When we analyze the license terms of these services based on [12], there is no conflicts arise in mashing up these services. It is interesting to note that though a mashup developer needs to pay for the use of ***birdwatch*** service in the mashup, the mashup should be noncommercial as one of the license terms of Google Maps API is to be noncommercial.

6. Concluding Remarks

The issue of intellectual rights in a mashup remains open and seeks a careful attention from the research community. In this paper, we have explored the ecosystem of mashup containing mashup developers, data providers, and consumers and their interactions on the intellectual rights issues. We have analyzed the issues by a scenario and presented a preliminary approach towards finding a solution for mashing up rights in a mashup.

In the context of mashups that involve services and data, it is rarely to see a mashup with a license that governs the consumers how to use the given

mashup[4]. Representing license terms in a machine interpretable way is a first step towards resolving the intellectual rights in mashups. As mashups are interwoven with data, the representation of licensing clauses related to data has a central role. In this paper we have not yet touched this issue, addressed in our current work. We are, for instance, considering the mapping of the Open Data Commons Database License clauses to the semantics of ODRL, ODRL/S and ODRL/CC. In addition, our future work includes analyzing the compatibility of licenses between multiple Open APIs: a complex and currently unresolved task.

References

[1] PALFREY, J., GASSER, U., Mashup Interoperability and eInnovation, Berkman Publications Series (2007)

[2] WATT, S., Mashups – The Evolution of SOA, IBM Technical Report (2007)

[3] MUSSER, J., O'REILLY, T., Web 2.0 Principles and Best Practices, O'Reilly Radar Report (2006)

[4] WEISS, M., GANGADHARAN, G.R., Innovation in the Mashup Ecosystem, Proceedings of the R & D Management Conference (2008)

[5] ZOU, J., PAVLOVSKI, C., Towards Accountable Enterprise Mashup Services. Proceedings of the IEEE International Conference on e-Business Engineering (2007)

[6] HEMENWAY, K., CALIHAIN, T., Spidering Hacks. O'Reilly (2003)

[7] GERBER, R., Mixing It up on the Web: Legal Issues Arising from Internet Mashups, Intellectual Property and Technology Law Journal 18(8), 11-14 (2006)

[8] GANGADHARAN, G.R., D'ANDREA, V., Licensing Services: Formal Analysis and Implementation, Proceedings of the Fourth International Conference on Service Oriented Computing (2006)

[4] API licenses are becoming more common nowadays, but mashup licenses are still rare. Some individual's mashups are licensed under Creative Commons license or General Public License. The most interesting one that we found was Frappr (http://www.frappr.com). This is clearly a mashup licensed with commercial intent (http://platial.com/terms).

[9] GANGADHARAN, G.R., D'ANDREA, V., IANNELLA, R., WEISS, M., ODRL Service Licensing Profile (ODRL-S). Proceedings of the 5th International Workshop for Technical, Economic and Legal Aspects of Business Models for Virtual Goods (2007)

[10] ALONSO, G., CASATI, F., KUNO, H., MACHIRAJU, V., Web Services Concepts, Architectures, and Applications. Springer Verlag (2004)

[11] VERMA, K., AKKIRAJ, R., GOODWIN, R., Semantic Matching of Web Service Policies. Proceedings of the Second International Workshop on Semantic and Dynamic Web Processes (2005)

[12] YANG, W., LUDWIG, H., DAN, A., Compatibility Analysis of WSLA Service Level Objectives. Technical Report RC22800 (W0305-082), IBM Research Division (2003)

[13] GANGADHARAN, G.R., WEISS, M., D'ANDREA, V., IANNELLA, R., Service Licensing Composition and Compatibility Analysis, Proceedings of the Fifth International Conference on Service Oriented Computing (2007)