

Modeling the Mashup Ecosystem: Structure and Growth

Michael Weiss¹, and G R Gangadharan²

¹weiss@sce.carleton.ca

Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada

²gr@novay.nl

Novay, Enschede, The Netherlands

Mashups combine data and services provided by third parties through open APIs (such as Google Maps and Flickr), as well as internal data sources owned by users. The creation of mashups is supported by a complex ecosystem of interconnected data providers, mashup platforms, and users. In this paper, we examine the structure of the mashup ecosystem and its growth over time. Several observations follow from our analysis. First, we can conclude that while the number of new APIs and mashups over time follows a linear growth pattern, the distribution of mashups over APIs is not uniform, but follows a power law. This implies that a small number of APIs provides the basis for the majority of mashups, and the other APIs are only used in certain application niches. Second, our analysis suggests that mashup platforms were introduced in response to the increasing complexity of mashups, as mashups evolved from one-feature mashups (widgets). Third, we observe that complementary relationships between open APIs are formed based on the position of the APIs in the ecosystem. The propensity of two APIs to be used together in the same mashup depends on the existing number of mashups to which they both contribute. The growth of the mashup ecosystem follows a pattern where keystone data providers or “powerful hubs” attract niche data providers as complementors, and the positions of keystones in the ecosystem are mutually reinforcing.

1. Introduction

Since late 2005, there has been a rapid proliferation of applications, referred as mashups, that combine data and services provided by third parties through open APIs (such as Google Maps and Flickr), as well as internal data sources owned by users. Open APIs give users access to the data or services of an IT platform. A well-known example is the Google Maps API: it generates maps for a given location, and its output can be combined with other data and services into mashups. Mashups allow the quick creation of custom applications. Since they often only have a short life span and are created for a specific group of users and needs, they are also known as situational applications. In many cases, they are developed by users. In the following, we will thus examine the creation of mashups through the lens of user innovation.

Open APIs and mashup platforms (such as mashup composers, which facilitate the construction of mashups) provide users with an innovation toolkit in the sense of (Thomke and von Hippel 2002). In this paper, we examine how mashups enable the user-driven design of applic-

ations based on open APIs, and how their creation is supported by an ecosystem of data providers, mashup platforms, and users. Its results are relevant to providers of open APIs, developers of mashup platforms, and users who want to combine open APIs with their own data.

The remainder of this paper is organized as follows. In Section 2, we review the related literature on user innovation, ecosystems, and recombinant innovation. In Section 3, we describe our research method. It uses network analysis to create models of the mashup ecosystem, which allow us to gain a deeper understanding of its structure and dynamics. Subsequently (in Section 4), we present our findings on the structure and growth of the mashup ecosystem. In Section 5, we present the managerial implications of our research and identify future work.

2. Background

2.1 User innovation

Traditionally, product development has been company-centric (Sawhney et al. 2005). In this model, the interface to the customer is the product prototype, and feedback on how well customer needs are met is only obtained late in the product development cycle. In user innovation, the locus of innovation shifts from the company to the customer (Thomke and von Hippel 2002). Sawhney et al. (2005) refer to this type of innovation as customer-centric.

The new interface to the customer is now a solution platform that customers can adapt to their needs using innovation toolkits (Thomke and von Hippel 2002). The iterative experimentation needed to develop new products is now carried out by the customer. This results in significantly faster cycles and reduces cost. In customer-centric innovation, customers can more quickly obtain feedback from the experiments, and, most importantly, create a solution that closely meets their needs. As Sawhney et al. (2005) note, rather than being a passive receiver of innovation, customers are now actively involved. The increased control over the IP created within the customer-specific aspects of the solution provides an added incentive for customers to take on more responsibility during product development (Thomke and von Hippel 2002).

The creation of mashups is an example of modding (Nambisan and Sawhney 2008). Modding is a familiar concept from the world of computer games. A “mod” is a modification of a computer game to perform new functions, e.g. creating a new game map. More radical changes to the game behavior are also not uncommon. In the modding approach, a community of innovators is created around a technology or platform. In the case of mashups, a data provider, by releasing an open API to its data or services, acts as an innovation catalyst.

2.2 Ecosystems

Innovations have become increasingly systemic (Maula et al. 2006). Focal companies define the architecture of the systemic innovation. Yet, they depend on external parties to provide necessary components that complement the core innovation by the focal company. These external parties are not under the direct control of the focal company. To tap into external resources, the focal company must demonstrate a credible commitment to the systemic innovation. Data providers try to attract complementors by freely providing access to their data (Weinberger 2007), and reducing their control of ownership over data and data formats used by the APIs as suggested by the observations on collaborative webs in (Sawyer 2007).

As stated by West (2006), the success of an ecosystem requires providing access to information on the innovation architecture, participating in standardization efforts, as well as investing in the providers of complements. These activities, which a focal company or keystone (Iansiti and Levien 2004) performs, facilitate cumulative

innovation. West provides a value network perspective on ecosystems. A value network consists of technology and component providers, system integrators, providers of complements, and users. Components are integrated with other components into systems. A system, in turn, becomes the more attractive and valuable to users the more complementary products are customized to work with it.

Iyer and Davenport (2008) describe the ecosystem Google has built around its proprietary platform. At the core of Google's ecosystem is its vast computing infrastructure. It enables Google to leverage third-party innovation while maintaining architectural control. The infrastructure comprises services such as Google Ajax Search, AdSense, and Maps. Third parties can create mashups that incorporate these services, and have them hosted by Google. They also get exposure to Google's huge user base. In exchange, the Google gains access to more ideas for new applications and improvements to its services than it could not have discovered internally.

2.3 Recombinant innovation

Hargadon (2002) highlights the recombinant nature of the innovation process. From this perspective, innovation can be described as the construction of new ideas from existing ones. Benefits of recombination include shortening the learning curve by combining known elements in new ways, sharing of past experience across organizational boundaries, and the diversity of problem solving frames. Recombinant innovation emphasizes the highly collaborative nature of innovation, and the role of knowledge brokers to bridge between knowledge domains and reinterpret existing ideas in new contexts.

The concept of recombinant innovation is closely linked to the concept of modularity, which works to accelerate innovation (Baldwin and Clark 2000). Modularity allows relatively independent innovation within components, or localized adaptation, and the creation of new products by mixing and matching components, or recombination (Ethiraj and Levinthal 2004). The increased modularity implied by open APIs is of great influence on the development of mashups. Open APIs are the modules that can be combined and recombined into mashups. Modularity is also the basis for imitating the design of a mashup, when a user “clones” an existing mashup.

Shuen (2008) interprets mashups as a form of capability syndication. This interpretation requires us to view companies as sets of capabilities that can be shared with and sold to other companies. These include content and services that used to be tightly guarded by companies. This creates an opportunity for companies to sell capabilities at which they excel, as well as for new entrants to define value propositions that did not exist before by syndicating capabilities offered by others. However, as emphasized by Iyer and Davenport (2008), creating value by recombining capabilities requires a more open approach to innovation in which companies providing the capabilities relinquish some of their control over them.

3. Research method

We tracked the growth of open APIs, mashups, and associated platforms since late 2005 based on public data sources. One of these sources is the ProgrammableWeb site.ⁱ It lists APIs and mashups by date of introduction, and provides a profile of each. It also categorizes APIs and mashups through a provided taxonomy and through tags that users can associate with the entries. In addition, the site offers information on mashup tools. Since the contents of the site are user-contributed, not all APIs and mashups in existence are indexed. However, the ProgrammableWeb is probably the most widely recognized mashup directory, and its contents can be considered representative of the state of the mashup ecosystem. Thus, while our analysis is likely to underestimate the total size of the mashup ecosystem, it can be expected to accurately represent the relations between ecosystem members.

At first, we extracted time-stamped information on when APIs were introduced, and when mashups were created over the course of the 598 days between September 14, 2005 and May 3, 2007. This information included the names of APIs and mashups, and the list of APIs each mashup combined. In the case of inconsistencies, for example, when the reported date of API introduction fell on a later date than the creation of the mashup, we moved the introduction date of the API up to the creation date of the mashup. Such inconsistencies can occur, because the data are user-reported, and the dates attached to the entries, in a strict sense, reflect when the entry was posted.

Next, we captured the relationships between mashups and APIs in an affiliation network. Originally developed for representing teams and their membership (see Uzzi et al. (2007) for a current review), the links in the affiliation network for the mashup ecosystem indicate which APIs are used in which mashups. Figure 1 shows some of the relationships from the actual data we collected. For example, Steampad and Flitter use the Amazon API.

An affiliation network can also be projected to a unipartite network. A projection of the affiliation network allows us to examine relationships between nodes of the same type. Thus, APIs that are used together in the same mashup are linked in a projection of the affiliation network to APIs. For example, since the Amazon, del.icio.us, and Flickr APIs are used in the Flitter mashup, they are linked in a projected network of APIs. Figure 2 shows the projection of the affiliation network above. We could similarly project the affiliation network to a network of mashups, and thus cluster mashups according to API.

Affiliation networks lends themselves to a visual analysis of the network data. Many relationships only become apparent by visualization. These observations then direct the further analysis as to what aspects of the net-

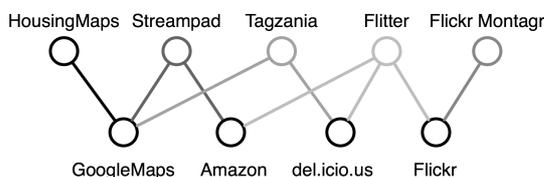


Figure 1: Affiliation network (the top row shows mashups, and the bottom row shows the APIs used by those mashups).

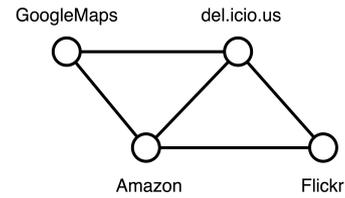


Figure 2: Projection of an affiliation network to APIs

work to study. For visualizing networks we used the Network Workbench developed at Indiana University.ⁱⁱ This is an open source platform for network visualization, and it incorporates a rich library of layout algorithms. Specific network properties were analyzed with the open source statistical analysis environment R,ⁱⁱⁱ and a set of custom Perl scripts were created for pre- and post-processing network data, as well as for format conversion. Specifically, we created scripts for binning data into histograms (raw and logarithmic), for filtering network nodes based on their degree and degree product, and for converting to the format required by the Network Workbench.

Figure 3 shows a snapshot of the mashup ecosystem using data based on the first month of records on the ProgrammableWeb site and using a radial layout suitable for showing hierarchical relationships. Only the names of APIs are shown to keep the diagram readable; mashups are shown as unnamed nodes. The graph accounts for 246 mashups. Even at this early stage, some of the most well-known open APIs are already prominently positioned in the network. We find Google Maps at the center and other prominent APIs such as Flickr, Amazon, Yahoo Maps and del.icio.us along the first ring around the center. Named nodes at the periphery of the graph represent APIs that have been used less frequently. Similarly, there are many one-feature mashups (also known as widgets or badges) that combine only one external API with internal data.

A variety of network properties can be obtained from this data that allow us to characterize the network and identify significant ecosystem members and their relationships. These include the degree of a network node, and the degree distribution. The degree k of a node representing an API is a measure of the number of mashups using the API (see Section 4.1). The degree distribution of these nodes measures the probability $P(k)$ and is an indicator of the popularity of an API (see Section 4.1). A projection of the affiliation network allows us to examine inter-node (API or mashup) relationships. As part of this analysis we will calculate the degree product k_1k_2 of two nodes in the projected network, which can be interpreted as a measure of the complementarity of two APIs, and the degree product distribution $P(k_1k_2)$ (see Section 4.3). In the projected network, the degree of a node is an indicator of its centrality (prestige). While there are more sophisticated centrality metrics (such as betweenness centrality), degree centrality is easy to compute and interpret.

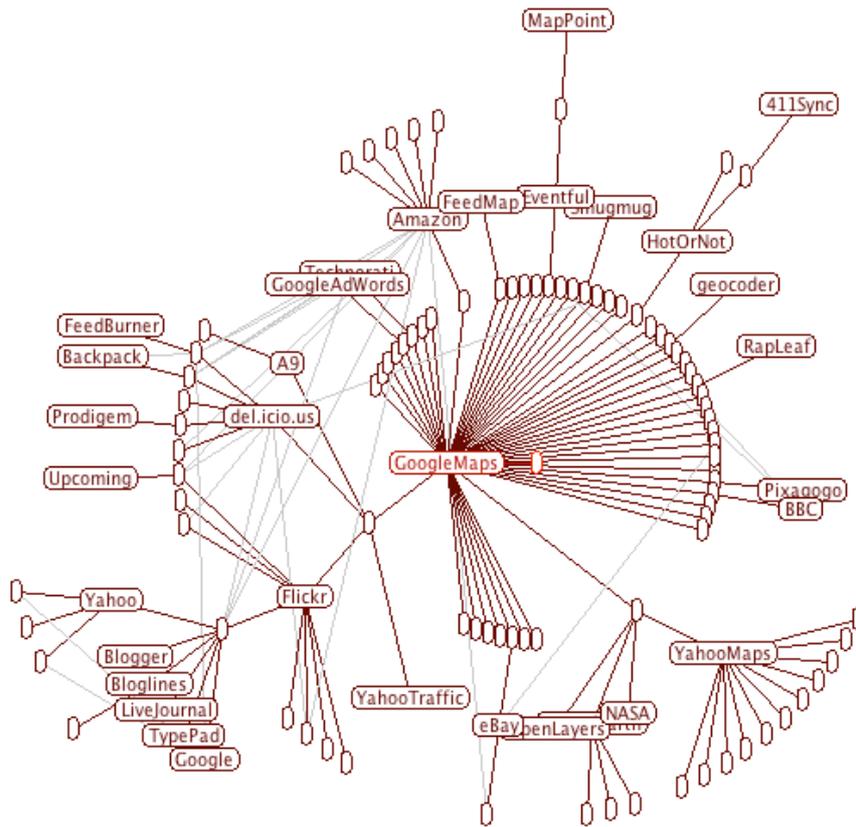


Figure 3: Snapshot of the mashup ecosystem using the first month of records from the ProgrammableWeb (named nodes represent APIs, unnamed nodes represent mashups, a link between a mashup and an API indicate that the mashup uses the API).

4. Findings

In this section, we report on our findings from the data analysis. Mashups enable users to “mix and match” data and user interface elements from different online information sources to create new applications (Yu et al. 2008). Initially, there were two types of players in the mashup ecosystem: data providers that release open APIs (such as Flickr or Google) and users/developers creating mashups. The only way for users to create mashups was by manually combining open APIs exposed by data providers. In some cases, data providers, in turn, aggregate the data offered by other providers. For example, Google Maps obtains its raw map data from a number of geographic data services. Access to those data providers is often not directly accessible through open APIs, so an API such as Google Maps is not itself a mashup. Somewhat surprisingly, there are only few mashups that offer their own APIs. We believe that the rationale for this is a combination of licensing issues and business reasons.

4.1. Open APIs and mashups

Throughout the observation period, the data shows a consistent growth in the number of open APIs and mashups. Figure 4a demonstrates that, over the course of the study period, 422 new open APIs were defined, or on average 0.70 APIs a day. From Figure 4b we see that over the same time frame, users created 1865 new mashups, for an average of 3.10 mashups each day. On average, there are 4.41 mashups to each API. Such linear growth was also observed in other types of networks, for example, collab-

oration networks (Barabasi et al. 2002).

However, the distribution of mashups over APIs is far from uniform, as shown by Figure 5. The graph shows the number of APIs with a degree k , which is proportional to the probability $P(k)$ of finding an API with this degree. Some APIs enjoy significantly greater popularity than others. The distribution follows a power law with an exponent λ of 0.76 and adjusted R^2 of 0.913. A distribution is said to follow a power law, if it adheres to the form $P(k) \sim k^{-\lambda}$. Since the distribution has a characteristic “long tail”, there are less values of $P(k)$ for higher values of k . To determine the exponent of the distribution we, therefore, need to apply logarithmic binning. The rightmost entry in the graph includes the Google Maps API, which is used by 937 mashups. The leftmost entry includes 63 APIs, which have been used only once. The left side of the graph lumps together more recently introduced APIs (which had not yet had a chance to be used in mashups) and APIs that have not found widespread adoption.

One explanation for this result is that there will be competition between APIs that offer the same type of service. For example, multiple APIs provide map services such as Google Maps, Yahoo Maps and Microsoft’s Virtual Earth. Choice requires users to select, and users will initially prefer some APIs over others. The more they select one API, the more likely it will be selected in the future. The result is that, eventually, one API will be significantly more popular than the others. However, it also implies that a small number of APIs – the keystones of the ecosystem (Iansiti and Levien 2004) – provide the basis for the majority of mashups, and all other APIs are

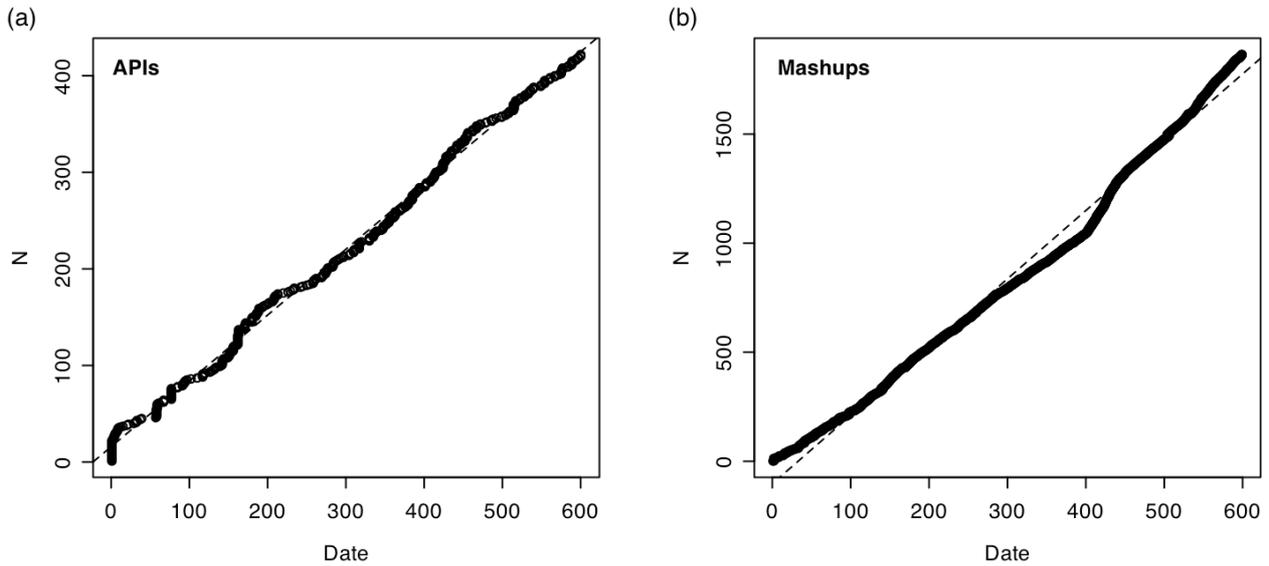


Figure 4: Growth of the mashup ecosystem. (a) Number of APIs. (b) Number of mashups.

only used in certain application niches. Using a 2/3:1/3 split as the basis for distinguishing between keystone APIs and niche APIs, we find that the top 12 (2.8%) APIs account for 2/3 of the contributions to mashups. The 2/3 cut-off is chosen based on Bradford's law (Bradford 1950), which describes the distribution of the top journals in a discipline, but making the same adaptation as Crowston et al. (2006), who used Bradford's law to estimate the size of the core group of an open source project.

In summary, we see that while the number of new APIs and mashups grows in a linear fashion, the distribution of mashups over APIs is not uniform, but follows a power law. This implies that a small number of APIs provides the basis for the majority of mashups, and the other APIs are only used in certain application niches.

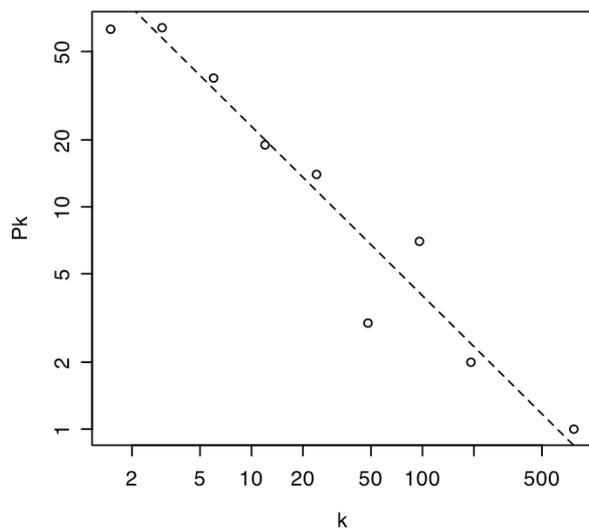


Figure 5: Distribution of mashups over APIs.

4.2. Complementary nature of APIs

Open APIs are the components of mashups, and as such provide value to users on their own. However, their value increases when other data providers offer complementary APIs that extends their functionality or allow them to be used in new contexts. For example, Flickr complements Google Maps, because it allows photos of a given location to be shown on a map. In fact, the combination of these services was so compelling that both Flickr and Google decided to offer new services to show photos on a map. The more complementary APIs and API has, the more attractive it becomes to users to users, and it will, hence, be selected more often for inclusion in a new mashup.

Figure 6 explores the complementary nature of open APIs. This network was obtained as projection of the affiliation network on APIs. A link between two APIs indicates that the APIs are used together in a mashup. The graph also shows the degree centrality of each API (an indicator of prestige) through the size of the nodes^{iv}, and the strength of the interaction between APIs (number of times they occur together in a mashup) by the width of the edges. The network has a core that consists of a small number of highly connected APIs that are used by many mashups, and more specialized APIs that are linked to the core (they are only used within a specific niche).

In order to make the network core more visible, we limited the graph to those links where both nodes have a high degree, and hence a high propensity to form a link. As a cut-off we used a product of 2500, which equates to two nodes of degree 50 forming a link, which appears reasonable given that only the top 12 of the 422 APIs have more than 50 links, and the maximum degree of an API is 937. Note this is only a heuristic. This approach is consistent with the finding of Barabasi et al. (2001) in their study of collaboration networks, where it is shown that the probability of two nodes forming a link can be approximated as a function of the product of their degrees.

The distribution in Figure 7 clearly indicates that APIs with higher degree centrality are more likely to interact.

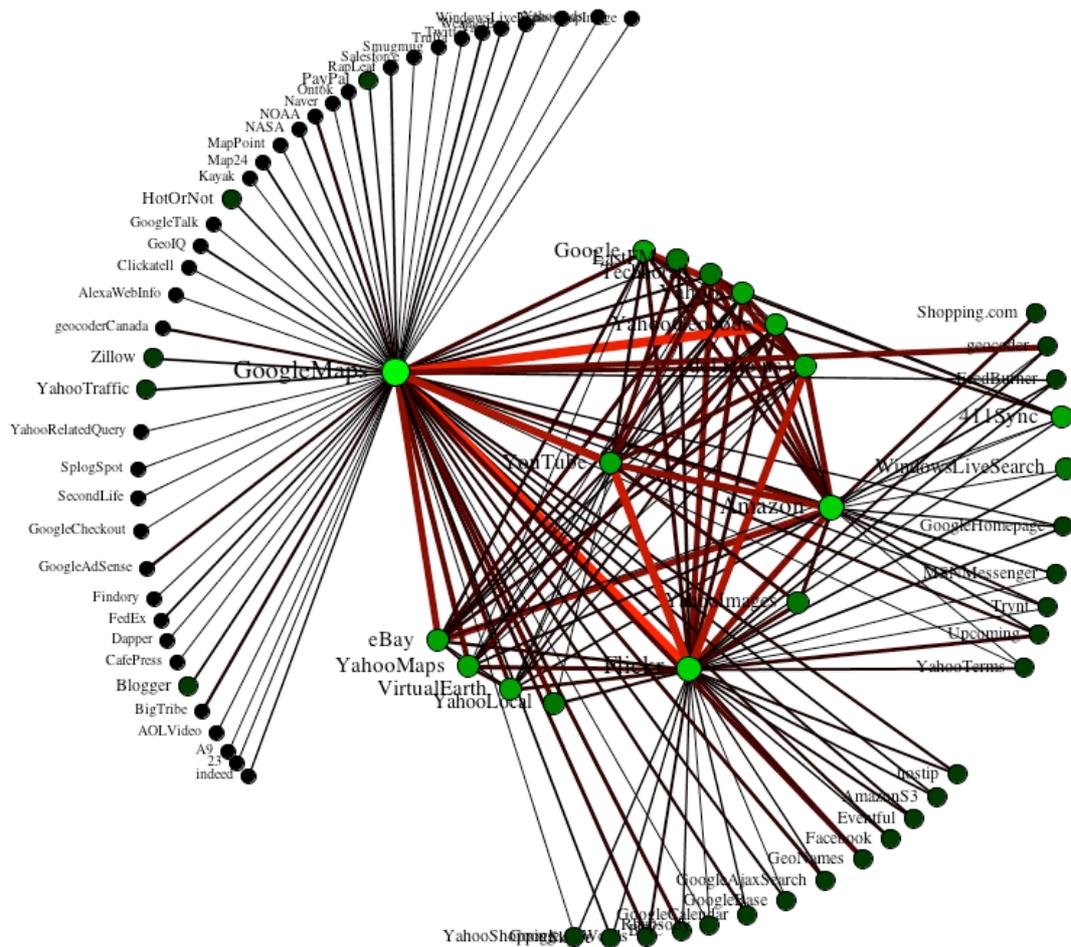


Figure 6: Projection of the affiliation network that shows the complementary nature of open APIs (nodes are APIs, links indicate that APIs are used together in a mashup; size of nodes and width of edges: large nodes have the highest degree, thick edges the highest strength).

This graph plots the distribution of interaction strengths over the product of the degree centralities ($k_1 k_2$). The fat tail of the distribution for low strength values (5 or lower) can be explained in terms of highly central APIs (such as

Google Maps or Flickr, which have degree centrality of 937 and 211, respectively) interacting with APIs of low centrality. Figure 6 and 7 suggest that complementary relationships between open APIs are formed based on the position of these APIs in the ecosystem.

In summary, we observe that the growth of the mashup ecosystem follows a pattern where keystone data providers attract niche data providers as complementors, and the positions of providers in the ecosystem are mutually reinforcing. Chakravorti (2004) talks of these keystones as “powerful hubs”. Similarly, Valverde et al. (2006), in their study of insect and open source communities, have observed that network positions reinforce each other. In part, this is certainly due to an accumulation of coordination knowledge – as APIs are used together, users build up an experience base on how to integrate them.

4.3. Mashup platforms

As the number of APIs, and thus the complexity of selecting mashups and the value perceived by businesses of creating mashups increased, platform providers entered the ecosystem to fill the void. Initially, these were graphical tools (such as QEDWiki from IBM) to simplify the composition of APIs into mashups, but platform providers quickly also started to offer marketplaces for APIs and mashups. At present, there is as yet no leading platform

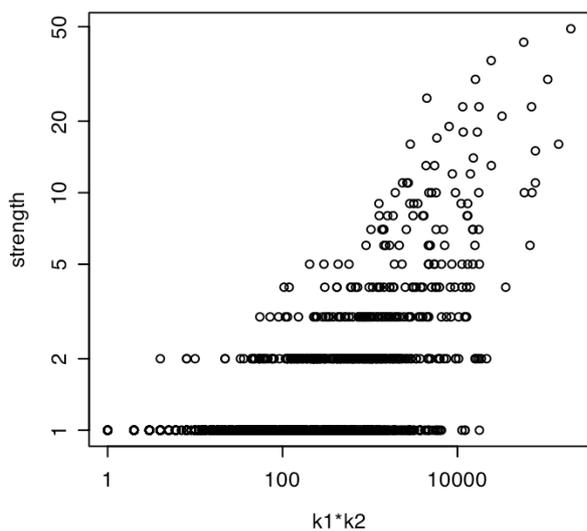


Figure 7: Distribution of the interaction strength of APIs plotted against the product of their respective degree centralities.

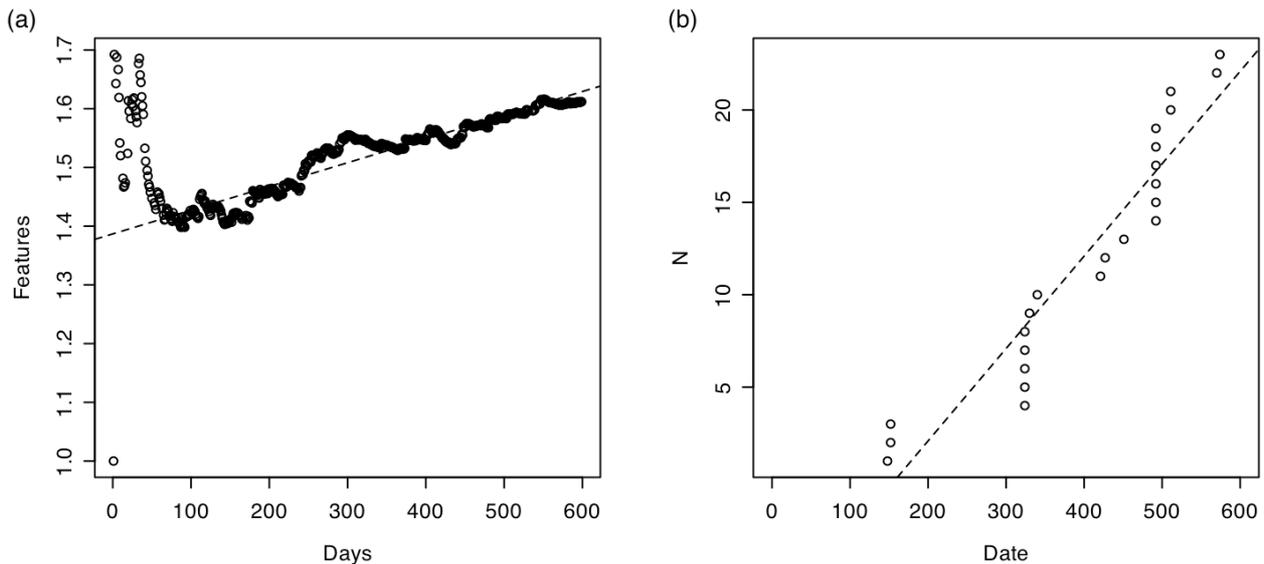


Figure 8: Growth in mashup complexity. (a) Increase in the complexity of mashups. (b) Timing of the introduction of mashup platforms.

provider, nor a leading marketplace that could serve all user needs. Figure 8a shows the increase of complexity with time. For each day we calculated the cumulative average of the number of APIs (or features) used in a mashup. By day 100 the average number of features was 1.42. By day 598, this number had increased to 1.61. Another perspective is to look at the extreme values. By day 598, the maximum number of features was 22, up from 10 by day 100. The average is significantly determined by the number of “one-feature” mashups or widgets.

Figure 8b shows the timing of the introduction of mashup platforms.^v The first set of platforms was introduced around day 150, and included libraries (for example, Yahoo! User Interface Library) and templates (for example, the templates provided by the Ning social networking site). Around day 325, the first hosted sites for mashups were introduced (for example, Coghead). The first platform that can be considered a mashup composer (DataMashups.com) was also released by day 325, as was the first platform (Dapper) for extracting implicit APIs from web sites. However, around 500 days after the publication of the first mashups, there was a flurry of releases of mashup composers (these included Teglo, QEDwiki, and Yahoo! Pipes). Many of these also integrated interfaces to search for known APIs and to integrate them into a mashup. The evolution of more sophisticated mashup platforms can also be seen as linked to an increased interest in creating enterprise mashups.

In summary, we find that mashup platforms have increased in sophistication (from early hosting for mashups and screen scraping tools to more recent graphical mashup composers) in response to the increasing complexity of mashups and the needs from enterprise users. One of the major shifts has been in the types of mashups created: from one-feature mashups, e.g. mashups that showed store locations on a map, to mashups that combine multiple open APIs and internal data sources. The latter type of application requires more advanced tools.

5. Conclusion

In this paper, we examined the structure of the mashup ecosystem and its growth over time. The main contribution of our paper is a research method for the analysis of mashup ecosystems. Its novelty lies in the development of techniques for mapping the mashup ecosystem, and the use of network analysis to obtain key characteristics of the ecosystem and identify significant ecosystem members and their relationships. These include the degree of a network node representing an API as a measure of its popularity and the degree product of two network nodes as a measure of the complementarity of two APIs. In this section, we summarize the managerial implications of our analysis, and discuss opportunities for future research.

5.1 Managerial implications

The managerial implications of this paper are in three areas related to the creation of mashups and the development of open APIs: selection of APIs, introduction of new APIs, and composition of APIs into mashups.

First, the research suggests that the position of a data provider in the mashup ecosystem affects the likelihood of their API to be incorporated into a mashup. The number of mashups using a given API is a first indicator of how likely an API will be selected as the basis of a new mashup. For users, the popularity of an API is a signal of its quality. When users select an API, they will give preference to more widely used APIs. But popularity alone does not fully explain how APIs are selected, except where a mashup consists of exactly one API. In all other cases, the number of interactions with other APIs also plays into the decision to select a given API. So does likely the category to which the API belongs.^{vi} However, this factor was not explored by the current research.

The frequency with which APIs are combined in a mashup is an indicator of how likely they will be combined in future mashups. We observed that the positions

of data providers in the mashup ecosystem are mutually reinforcing. One factor we would like to offer as an explanation is that when APIs are used together, significant experience on how to integrate these APIs is obtained. This, in turn, will lead developers to prefer proven combinations of APIs when developing new mashups. Another likely factor is that mashups – as the literature on the role of imitation in innovation such as (Ethiraj et al. 2008) leads us to conclude – are developed by emulating existing mashups. In our research to date, however, we have not studied the impact of copying or cloning of mashups on the structure and growth of the mashup ecosystem.

This has implications for users of mashups and data providers. Users will select APIs based on how many other mashups use a given API, as well as the collective experience in using a given API with other APIs to be selected for the mashup. Data providers, when introducing a new API, will benefit from ensuring that their API integrates well with existing APIs that are strongly positioned in the mashup ecosystem. Therefore, data providers should look for opportunities to complement existing APIs. By complementing the existing API, the new API will also benefit the provider of the existing API by providing additional contexts of use for the API and increasing its potential share of mashups that use it. In order to identify potential niches to enter, data providers need to gain a good understanding of the structure of the current ecosystem. A map of the ecosystem can be produced following the research method described in this paper.

Second, our analysis suggests that complexity of mashups drives the development of mashup platforms. The design of more complex mashups requires more sophisticated mashup platforms. This coincides with the increasing interest in enterprise applications of mashups, which may itself, in turn, be a major contributor to higher complexity. Platform providers need to introduce tools that help manage this complexity. This comes with challenges, as we look at some of the components of the complexity of mashup development: searching for APIs, enforcing design rules during the composition of APIs, and certification of APIs. The selection of APIs turns into a combinatorial problem of finding the right combination of APIs for a given purpose. Enforcing design rules requires a codification of integration experience so it can (at least partially) be automated by a tool. Finally, APIs need to be certified in terms of meeting quality standards.

5.2 Future work

Several open questions remain, including what laws underlie the growth of the mashup ecosystem. One question is whether the preferential attachment (“rich get richer”) hypothesis formulated by Barabasi et al. (2002) applies to mashup ecosystems, and in what form. Probing further, we would like to understand the mechanisms that underlie the creation of mashups. We also have an intuitive understanding that mashups are often created by emulating and adapting existing mashups. The question then becomes one of how to detect and measure the degree of imitation in the creation of mashups vs. other growth mechanisms. As noted above, the selection of APIs can be expected to be influenced by the category to which the API belongs.

One area of future work is to extend our analysis to the growth of categories of APIs and mashups. Finally, some data providers offer multiple APIs and they may hold positions of different strength in the ecosystem through each of their APIs. An area of future research is then to examine how data providers can leverage existing positions in APIs for the introduction of additional APIs. Exploring these issues will lead to further managerial insights and to refinements of the research method introduced in this paper. In conclusion, the mashup ecosystem presents a rich environment for research on the nature of innovation and ecosystems that bears many of the characteristics of what Fine (1999) calls “clockspeed” industries.

Acknowledgements

The authors wish to thank Jeff Butler, the editor, and Tony Bailetti for reading earlier drafts of this paper and providing us with insightful comments, as well as the participants at the R&D Management 2008 conference, where an earlier version of this paper was first presented.

References

- Baldwin, C.; Clark, K. (2000): *Design Rules: The Power of Modularity*, MIT Press.
- Barabasi, A.L.; Jeong, H.; Neda, Z.; Ravasz, E.; Schubert, A.; Vicsek, T. (2002): Evolution of the Social Network of Scientific Collaborations, in: *Physica A*, 311(3-4), 590-614.
- Bradford, S. (1950): *Documentation*, Public Affairs Press.
- Chakravorti, B. (2004): The New Rules for Bringing Innovations to Market, in: *Harvard Business Review*, 82(3): 58-67.
- Crowston, K.; Wei, K.; Li, Q.; Howison, J. (2006): Core and Periphery in Free/Libre and Open Source Software Team Communications, in: *Hawaii International Conference on System Sciences*, 6, 118-124.
- Ethiraj, S.; Levinthal, D. (2004): Modularity and Innovation in Complex Systems, in: *Management Science*, 50(2), 159-173.
- Ethiraj, S.; Levinthal, D.; Roy, R. (2008): The Dual Role of Modularity: Innovation and Imitation, in: *Management Science*, 54(4), 939-955.
- Fine, C. (1999): *Clockspeed: Winning Industry Control in the Age of Temporal Advantage*, Basic Books.
- Hargadon, A. (2002): Brokering Knowledge: Linking Learning and Innovation, in: *Research in Organizational Behavior*, 24, 41-85.
- Iansiti, M.; Levien, R. (2004): Strategy as ecology, in: *Harvard Business Review*, 82(3), 68-78.
- Iyer, B.; Davenport, T.H. (2008): Reverse Engineering Google’s Innovation Machine, in: *Harvard Business Review*, 86(4), 58-69.
- Maula, M.; Keil, T.; Salmenkaita, J.-P. (2006): Open Innovation in System Innovation Contexts, in: Chesbrough, H.; Vanhaverbeke, W.; West, J. (2006): *Open Innovation: Researching a New Paradigm*, Chapter 12, 249-257.
- Nambisan, S.; Sawhney, M. (2008): *Global Brain: Your Roadmap for Innovating Faster and Smarter in a Networked World*, Wharton School Publishing.
- Sawhney, M.; Verona, G.; Prandelli, E. (2005): The Internet as a Platform for Customer Engagement in Product Innovation, in: *Journal of Interactive Marketing*, 19(4), 4-17.

- Sawyer, K. (2007): *Group Genius: The Creative Power of Collaboration*, Basic Books.
- Shuen, A. (2008): *Web 2.0: A Strategy Guide*, O'Reilly.
- Thomke, S.; von Hippel, E. (2002): Customers as Innovators: A New Way to Create Value, in: *Harvard Business Review*, 80(4), 74-81.
- Uzzi, B.; Amaral, L.; Reed-Tsochar, F. (2007): Small-World Networks and Management Science Research: A Review, in: *European Management Review*, 4, 77-91.
- Valverde, S.; Theraulaz, G.; Gautrais, J.; Fourcassié, V.; Solé, R. V., Self-organization Patterns in Wasp and Open Source Communities, in: *IEEE Intelligent Systems*, 21(2):36-40, 2006.
- Weinberger, D. (2007): If you Love Your Information, Set It Free, in: *Harvard Business Review*, 85(6), 20-21.
- West, J. (2006): Does Appropriability Enable or Retard Open Innovation, in: Chesbrough, H.; Vanhaverbeke, W.; West, J. (2006): *Open Innovation: Researching a New Paradigm*, Chapter 6, 109-133.
- Yu, J.; Benatallah, B.; Casati, F.; Daniel, F. (2008): Understanding Mashup Development, in: *IEEE Internet Computing*, 12(5), 44-52.

ⁱwww.programmableweb.com

ⁱⁱ<http://nwb.slis.indiana.edu>

ⁱⁱⁱ<http://www.r-project.org>

^{iv}To ensure that the nodes can be displayed in the diagram, their size is determined as $1 + a \log(k)$, where k is the degree of the node, and a a suitable multiplier.

^vThis data was obtained from news releases on the ProgrammableWeb, blogs, and platform provider sites.

^{vi}Some of the most popular API categories listed on the ProgrammableWeb site include: mapping, Internet, reference, community, shopping, music, search, telephony, and enterprise.