Innovation in the Programmable Web: Characterizing the Mashup Ecosystem

Shuli Yu and C. Jason Woodard

School of Information Systems
Singapore Management University
{shuli.yu.2004,jwoodard}@smu.edu.sg

Abstract. This paper investigates the structure and dynamics of the Web 2.0 software ecosystem by analyzing empirical data on web service APIs and mashups. Using network analysis tools to visualize the growth of the ecosystem from December 2005 to 2007, we find that the APIs are organized into three tiers, and that mashups are often formed by combining APIs across tiers. Plotting the cumulative distribution of mashups to APIs reveals a power-law relationship, although the tail is short compared to previously reported distributions of book and movie sales. While this finding highlights the dominant role played by the most popular APIs in the mashup ecosystem, additional evidence reveals the importance of less popular APIs in weaving the ecosystem's rich network structure.

Keywords: API, mashup, social network, power law, long tail, small world.

1 Introduction

The emergence of a new generation of web-based technologies, collectively dubbed "Web 2.0" [1], has fueled the growth of applications such as wikis and blogs that make it easier for users to publish their own content. A subset of these technologies have set the stage for an even more profound transformation by enabling users to go beyond static publishing and create their own web applications using powerful building blocks provided by third parties. This paper explores the recent explosion of these personalized applications, called *mashups*, and the application programming interfaces (APIs) they build upon.

The term "mashup" is borrowed from pop music, where it denotes remixing songs (or parts of songs) to create new derivative works. Similarly, web-based mashups are created by integrating data from one or more sources to create a new application, typically in a way that hides the details of the source applications to provide a seamless experience for the user [2]. Major companies like Google, Amazon and eBay have provided interfaces to many of their services at little or no cost, allowing individuals and other businesses to create composite applications with novel functionality. As more firms choose to provide APIs for public use, the number of opportunities to combine these APIs in new ways increases exponentially. Each new mashup may then attract its own base of users, further extending the market reach of the API providers.

G. Feuerlicht and W. Lamersdorf (Eds.): ICSOC 2008, LNCS 5472, pp. 136–147, 2009.

Despite the potential importance of this trend, few empirical studies have attempted to characterize the ecosystem of Web 2.0 mashups and APIs in a general way. Most of the existing literature is focused on the concerns of stakeholders in a particular domain (e.g., health librarians [3] or digital journal publishers [4]), the legal and policy issues associated with remixing content [5, 6], or the underlying technologies [7, 8]. A number of classification schemes for mashups have been proposed [1, 9], but we are unaware of any studies that apply these schemes in an empirical setting.

The task of characterizing the mashup ecosystem is made more complicated—and more interesting—by the fact that the web of relationships among mashups and APIs has evolved along with the populations of each. To understand the dynamics of these relationships, we need to investigate the process by which mashup creators choose APIs to build on, which in turn depends on the decisions of API providers (to expose their APIs in the first place, as well as the terms under which they do so) and the expectations of the mashup creators' own target audiences. The importance of these interlinked decisions suggests viewing mashups and APIs as a single evolving network rather than as independent populations of discrete entities. This network-based approach allows us to explore how APIs become connected through mashups, and how these connections influence the overall network structure and the popularity of individual APIs.

This paper presents the results of a preliminary effort to study the API–mashup network using well-established concepts and techniques. Section 2 describes our data set, which was obtained from publicly available sources. We examined the following characteristics of the network, and report the results in sections 3–5 respectively:

- *Graphical network structure.* Visual snapshots of the network illustrate its rapid growth and reveal qualitative structural patterns, most notably a partition of APIs into three tiers, with Google Maps at the center. APIs in Tier 1 and 2 tend to serve as platforms, while those in Tier 2 and 3 often serve as data sources.
- *Degree distributions.* Plotting the cumulative distribution of mashups to APIs reveals a power-law relationship, which is commonly generated by processes in which popular network nodes attract new links at a higher rate than less popular ones [10]. We also assess the extent to which the mashup ecosystem exhibits the "long-tail" property found in studies of books, movies, music and other information goods [11]. Perhaps surprisingly, the distribution of API popularity has a relatively short tail compared to other types of goods.
- *Social network statistics.* Analysis of the API affiliation network provides additional information on the evolving network topology. We find that the links between APIs exhibit the properties of a small-world network [12], suggesting a high level of novelty in mashup designs. While the most popular APIs are responsible for the vast majority of links, the small-world structure is due mainly to the less popular APIs.

Section 6 comments on the implications of our findings for stakeholders in the mashup ecosystem, and concludes with a call for further research.

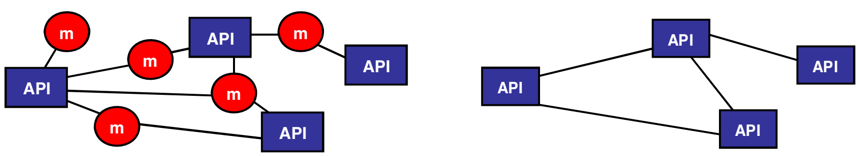
2 Data: The Programmable Web

To construct a network view of the mashup ecosystem, we turned to the largest online repository of information about Web 2.0 mashups and APIs, ProgrammableWeb.com. This aggregator site provided the most comprehensive listing of mashups and APIs available, including information on which mashups use which APIs.

Our data set consisted of 2664 mashups and 590 APIs that were registered between September 2005 and December 2007. We used this data to create snapshots of the mashup ecosystem at 9 quarterly intervals, beginning in December 2005.

For each snapshot, we created a rectangular matrix with mashups on the rows and APIs on the columns. Each cell in the matrix was assigned a binary value (0 or 1) indicating whether the mashup corresponding to the given row uses the API corresponding to the given column. These relationships can also be represented graphically. In Figure 1a, APIs are shown as boxes and mashups as circles. The line segments connecting them represent instances in which a mashup uses (i.e., builds on) a particular API.

While the API–mashup network is useful for visualizing relationships among APIs and mashups, many network analysis techniques are limited to a single type of entity. For the analysis in Section 5, we therefore followed standard practice in the social network literature [13] and transformed the rectangular (two-mode) API–mashup matrices into square (one-mode) affiliation matrices to study the relationships between APIs. These API affiliations can also be represented graphically (Figure 1b). Two APIs are linked by a line segment if they have been used together in a mashup.



(a) (b)

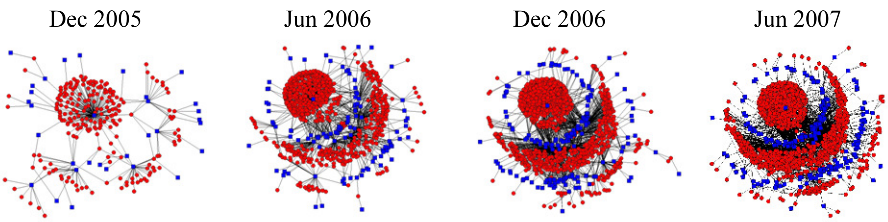
Fig. 1. (a) API–mashup network (b) API affiliation network

3 Graphical Network Structure

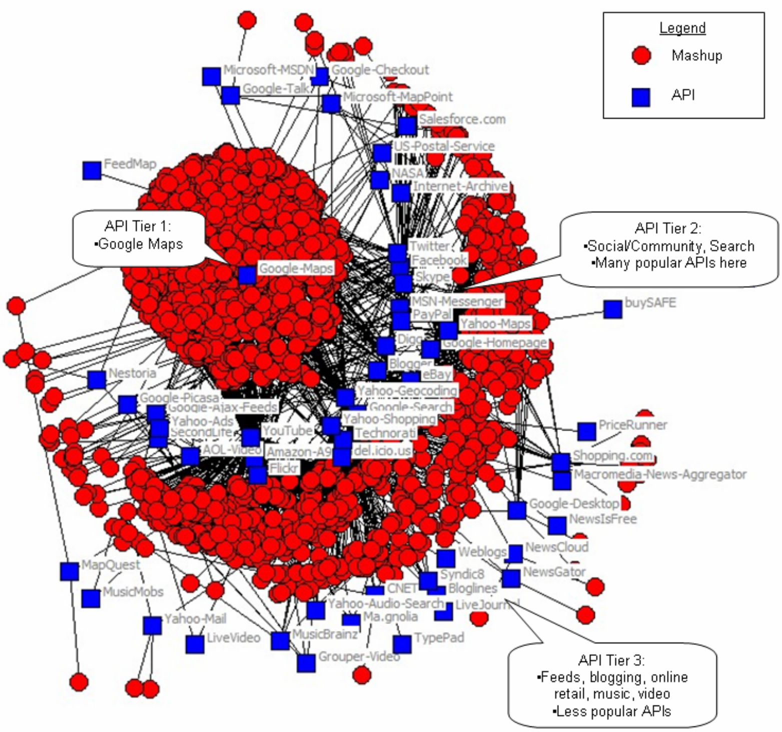
We used NetDraw [14], a popular social network visualization tool, to study the growth of the mashup ecosystem. Figure 2a shows four snapshots of the API–mashup network rendered using NetDraw's node repulsion algorithm. The population of both APIs and mashups grew roughly linearly over time. API growth tended to be slower and more consistent than mashup growth, with APIs growing at a mean rate of 20.1 per month (SD = 6.2) and mashups at a rate of 93.8 per month (SD = 25.4).

While network visualization is as much an art as a science—and the choice of layout algorithm is to some extent arbitrary—it is striking that each snapshot exhibits a distinctive three-tiered structure, with a layer of mashups between each API tier. These layers appeared in all time periods, even as the total number of APIs and mashups increased tenfold.

The enlarged snapshot from December 2007 (Figure 2b) shows a more detailed view of the key APIs and their corresponding mashups. The network clearly centers around Google Maps (Tier 1), with a ring of popular but less central APIs around it (Tier 2) and a constellation of other APIs (Tier 3) on the periphery. This layered structure suggests that Google Maps and at least some of the Tier 2 APIs play the role of *platforms* in the mashup ecosystem. Google Maps, in particular, adds value to the multitude of other APIs that provide spatial data by providing a powerful and convenient way to display this data in a web application. The fact that it is freely available and uses common protocols and data standards makes it an especially attractive choice for mashup developers.



(a)



(b)

Fig. 2. (a) Evolution of the Web 2.0 mashup ecosystem. (b) The API–mashup network in December 2007.

Although the relationships between Tier 2 and Tier 3 APIs are less clear-cut, we observe similar patterns of complementarity between platform-type services and services that supply raw data. APIs for mapping (Google Maps, MS MapPoint), search (Google Search, Yahoo Search), community (Facebook), payment (PayPal) and telephony (Skype) are often combined with APIs that provide data like images (Flickr), video (YouTube, LiveVideo), product details (eBay, Shopping.com, PriceRunner) and news feeds (Technorati, CNET, LiveJournal). Most of the APIs that provide platform services reside in Tier 2, while most of data providers (except the most popular ones) are in Tier 3.

4 Degree Distributions

Before exploring the network structure of the mashup ecosystem in greater depth, we pause to consider a more aggregate phenomenon, namely, the relative frequency with which APIs are used in mashups. This analysis will shed light on how API "market share" is distributed, and provide clues about how some APIs become vastly more popular than others.

4.1 A Power Law: The Rich Get Richer

Figure 3 plots the cumulative distribution of mashups over APIs at two points in time, December 2005 and December 2007. For each distribution, the vertical coordinate of the top-left data point indicates the number of APIs that were used by a single mashup between September 2005 and the given date. The horizontal coordinate of the bottom-right point indicates the number of mashups that used the single most popular API during the same time period. Plotting the distributions on log-log axes reveals a linear relationship characteristic of a power-law distribution.

In a power-law distribution, small events are extremely common and large events are extremely rare [15]. In the context of the mashup ecosystem, this would mean that a large number of APIs are used by few mashups and a small number of APIs are used by many mashups, compared to a bell-shaped pattern in which more and less popular APIs are distributed symmetrically. Such a phenomenon occurs in many markets that are dominated by a few popular products, for example a bookstore that sells a few blockbuster novels in large quantities along with many obscure texts in small quantities.

The power-law pattern is sometimes expressed by the "Pareto principle," which states that 20% of the causes often yield 80% of the effects. In the mashup ecosystem, the distribution is even more lopsided: by December 2007, the top 20% of APIs (121 out of 590) had captured 95% of the market (2535 mashups out of 2664).

Prior research identifies *preferential attachment* as a mechanism that can give rise to power-law distributions [10]. If preferential attachment were operating in the mashup ecosystem, APIs that became popular early in the ecosystem's development would continue to gain mashups at a higher rate than less popular APIs, reinforcing their popularity. We did not test this hypothesis quantitatively, but it seems like a plausible explanation.

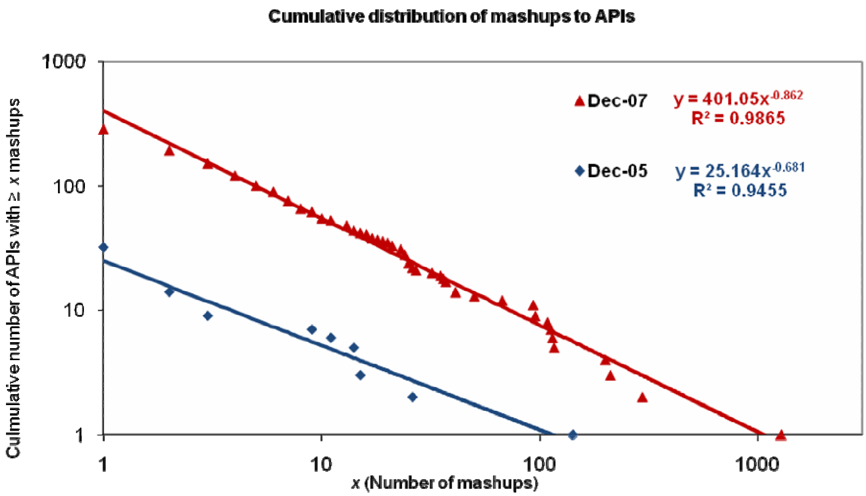


Fig. 3. Cumulative distribution of mashups to APIs for December 2005 and 2007

Although the data for both time periods is consistent with preferential attachment, the strength of the effect appears to have weakened over the period of our study. We computed best-fit lines for the December 2005 and December 2007 distributions, yielding the equations shown in Figure 3. The absolute value of the exponent increased in magnitude from 0.68 to 0.86, which implies a decrease in the fraction of mashups that use the top APIs relative to the fraction of APIs with a small number of mashups. The impact of this change can be seen by looking at the top and bottom ends of the distribution. At the bottom, the number of APIs with only one mashup increased by a factor of 16, from 25 in 2005 to 401 in 2007. At the top, the API with the largest number of mashups (Google Maps) also increased its number of mashups, but only by 9 times, from 114 in 2005 to 1048 in 2007.

4.2 A Long Tail? Yes, But a Short One

The concept of a power-law distribution is frequently associated with the idea of a "long tail." Long-tailed distributions are characterized by a large number of low-frequency occurrences that can cumulatively outweigh the high-frequency ones [11]. Such distributions are common in online retailing, where product selection is not limited by physical storage restrictions or holding costs, and consumers can easily find specific products by searching online or acting on recommendations, resulting in an overall high volume of sales from niche products [16]. Since the mashup ecosystem is similarly unencumbered by physical constraints, it is plausible that the number of APIs with few mashups could be so numerous that they form a long tail and capture the lion's share of the market.

In 2007, Kilkki [17] proposed a mathematical formula to model long-tailed distributions:

$$F(x) = \frac{\beta}{(\frac{N_{50}}{x})^{\alpha} + 1}$$

When products (in this context, APIs) are ranked according to their volume or share, $F(x)$ represents the share of total volume covered by products up to rank x. In this model, three parameters determine the size of the tail: (1) N_{50} is the number of products that cover half of the total volume; (2) α is the factor that defines the form of the function by describing the steepness of slope in the middle part of the function; and (3) β is the total volume of the distribution, including latent demand suppressed by the current structure of the product market. Kilkki applied the model to a wide range of data sets, including book sales and movie viewership in the United States. He found that the distribution of book sales has a much longer tail (N_{50} = 30714, α = 0.49 and β = 1.38) than movie viewership (N_{50} = 56, α = 0.82 and β = 1.60).

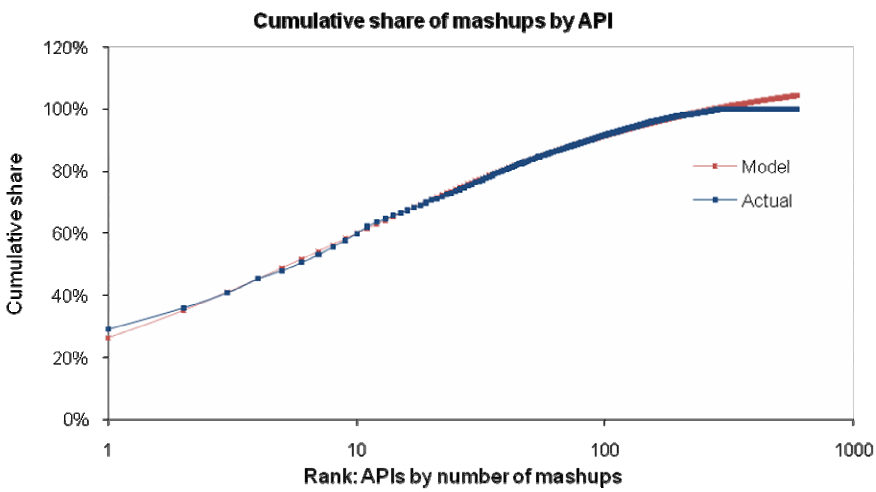


Fig. 4. Cumulative share of mashups by API in December 2007

Fitting the December 2007 API and mashup data to the model using nonlinear regression yields N_{50} = 8.24, α = 0.570 and β = 1.14. Figure 4 plots the actual data along with fitted data from the model. Although the model fits the data very well, it would be misleading to conclude that the mashup distribution has a long tail in the same sense as books and movies do. On the contrary, the parameters indicate that API use by mashups has a shorter tail than book sales on all three dimensions, as well as a shorter tail than movie viewership on two dimensions, N_{50} and β.

This result is consistent with the frequency data reported earlier. Recall from Section 4.1 that the head of the mashup distribution is top-heavy, with the top 20% of the APIs capturing 95% of the mashups, a far greater share than the 80% suggested by the Pareto rule. Moreover, the tail flattens out quickly at the bottom end, with 51% of the APIs excluded because they were not used by any mashups at all.

5 Social Network Statistics

The impression conveyed by the analysis so far is that the most popular APIs are vastly more important than the rest in terms of their contribution to the mashup ecosystem. While it is true that a small set of APIs account for the majority of mashups (with the top 10 responsible for well over 50% of the observed mashup links), a closer look at the network structure reveals a subtler but perhaps equally important role for the less popular APIs in the ecosystem.

In this section, we focus on the API affiliation network. Investigating the structure of API affiliations enables us to explore patterns of innovation by mashup creators. Since each link in the affiliation network represents a different pair of APIs used by one or more mashups, this analysis sheds light on the ways in which mashup creators combine APIs to create applications with novel functionality.

We computed a set of network metrics commonly employed in the social network literature: mean degree, normalized degree, network density, characteristic path length and clustering coefficient. These metrics were calculated for each of the 9 quarterly network snapshots, and are summarized in Table 1.

Table 1. API affiliation network metrics by quarter

Date	*Mean Degree*	*Normalized Degree*	*Network Density*	*Characteristic Path Length*	*Clustering Coefficient*
Dec-05	1.152	0.0127	0.0162	2.355	0.320
Mar-06	2.971	0.0175	0.0272	2.284	0.500
Jun-06	4.901	0.0231	0.0415	2.228	0.399
Sep-06	5.279	0.0189	0.0344	2.206	0.395
Dec-06	6.171	0.0176	0.0329	2.243	0.418
Mar-07	9.170	0.0230	0.0395	2.223	0.458
Jun-07	9.155	0.0200	0.0350	2.237	0.448
Sep-07	8.929	0.0172	0.0344	2.240	0.428
Dec-07	8.488	0.0144	0.0281	2.282	0.414

5.1 API Affiliation Metrics

The degree of a network node is the number of connections the node has with others. In the API affiliation network, the degree indicates the number of other APIs that are used in common with a given API by one or more mashups. Examining the *mean degree* of the network over time, we see a steady increase from 1.152 mashups per API in December 2005 to a maximum of 9.170 in March 2007, followed by a plateau and a slight downward trend for the remainder of the year.

The *normalized degree* is the mean degree divided by the maximum possible degree (the total number of nodes minus one), providing a measure of network connectivity that controls for the growth of the network. Over the period of the study, the normalized degree varied much less than the mean degree, ranging from about 0.013 to 0.023.

The *network density* is the ratio of the number of actual links in the network to the total number of possible links that would exist if all nodes were directly connected to each other. Hence, it is another measure of network connectivity, i.e., the extent to which APIs are connected to each other by mashups. The network density fluctuated substantially over the study period, from about 0.016 to 0.045, but without a clear trend.

The *characteristic path length* (CPL) is the expected distance along the shortest path between any two nodes in the network. After a slight drop between December 2006 and March 2006, it remained fairly stable (between about 2.2 and 2.3).

The *clustering coefficient* (CC) measures the extent to which network nodes tend to form groups with many internal connections but few connections leading out of the group. Like the CPL, it remains nearly constant after March 2006 (between about 0.40 and 0.45).

5.2 Small-World Analysis

The characteristic path length and clustering coefficient are the key statistics used to identify networks with the *small-world property* coined by Watts and Strogatz [12]. The concept of a small-world network was inspired by Stanley Milgram's famous experiment in which randomly selected individuals in Nebraska and Kansas were able to forward letters to a target in Boston through an average of only six intermediaries.

In the formalization of the concept proposed by Watts and Strogatz, a network has the small-world property if its characteristic path length is similar to that of a random network with the same density despite having a much larger clustering coefficient. In a random network, the expected CPL is approximately $\ln n / \ln k$ and the expected CC is approximately k, where n is the number of nodes and k is the mean degree of the network. For the API affiliation network in December 2007, $CPL_{rand} = 2.98$ and $CC_{rand} = 0.0144$, while the actual CPL was 2.28 (even lower than CPL_{rand}) and the actual CC was 0.414 (almost 30 times higher than CC_{rand}). The API affiliation network thus easily qualifies as a small-world network. Further investigation shows this to be true throughout the two-year sample period.

What does it mean for the API affiliation network to have the small-world property? Loosely speaking, nodes in a small-world network are more closely connected than one would expect based on their density and clustering. In the context of the mashup ecosystem, this suggests that APIs with very different functionality (e.g., mapping, audio search, and news feeds) are more likely to be connected through mashups (e.g., an application that shows famous places from popular songs, and one that finds news stories on popular artists) than one might otherwise expect. Although we caution against reading too much into this finding, it is an encouraging sign that the mashup ecosystem has generated a substantial level of novelty and surprise.

5.3 Importance of Peripheral APIs

To better understand why the API affiliation network has the small-world property, we repeated the analysis on the affiliation network formed by a subset of the most popular APIs. To construct the subset, we selected APIs that were used by at least 5 mashups in December 2007. These 152 APIs comprised 57% of the APIs with at least one mashup, or about half of the APIs that played an active role in the mashup ecosystem.

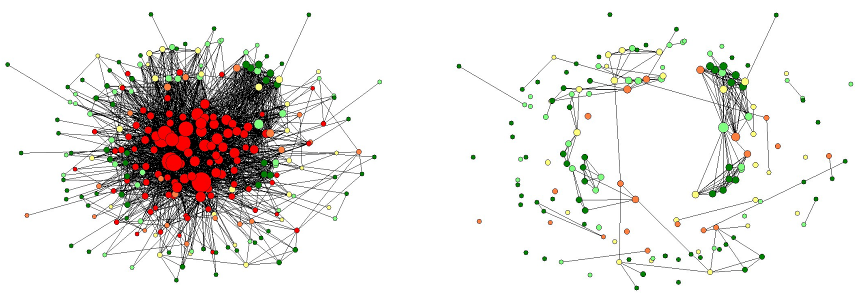
Omitting the less popular APIs from the network reveals their important role in giving rise to the small-world property of the mashup ecosystem as a whole. As Table 2 indicates, the affiliations among most popular APIs barely qualify as small-world. Their clustering coefficient (0.446) is similar to the full set, but only 3.02 times that of a comparable random network, compared to 28.78 for the full set. Despite this comparatively low level of clustering (which intuitively ought to reduce path lengths), the CPL of the subset affiliation network is longer than CPL_{rand}, indicating a relative absence of the distinctive short paths associated with the small-world property.

Table 2. Small-world metrics for a subset of the most popular APIs (December 2007)

APIs	*n*	*k*	*CC*	CC_{rand}	$\frac{CC}{CC_{rand}}$	*CPL*	CPL_{rand}	$\frac{CPL}{CPL_{rand}}$
Full set	590	8.49	0.414	0.0144	28.78	2.28	2.98	0.765
Popular	152	22.42	0.446	0.1475	3.02	1.99	1.62	1.233

The role of the less popular APIs in forming these short paths can be seen in Figures 5a and 5b. These visualizations were generated using NetDraw's spring embedding algorithm, which locates pairs of nodes with the shortest path lengths closest to each other. The full network is shown in Figure 5a; the network with the most popular APIs omitted is shown in Figure 5b. The nodes are colored according to the number of mashups for each API: red nodes represent the most popular 152 APIs with 5 or more mashups, while orange, yellow, light green and dark green represent APIs with 4, 3, 2 and 1 mashups respectively. APIs with no mashups were omitted. The size of each node corresponds to the degree of each API, with larger nodes indicating APIs with higher degrees.

In Figure 5a, the most popular APIs form the core of the network, and are highly interconnected through the mass of black lines. The less popular APIs in orange, yellow, light and dark green, form rings around the core, indicating that they are further away in terms of path length. These peripheral APIs mostly have connections to the APIs in the core, forming clusters around them. Figure 5b shows clearly that there are few direct connections between less popular APIs.



(a) (b)

Fig. 5. (a) Full set of API affiliations. (b) Affiliations of peripheral APIs.

This additional analysis highlights the structural differences between the more and less popular APIs in the network. This is consistent with the qualitative findings of Section 3, in particular that APIs can be roughly segmented into two types: core platform APIs, which tend to form hubs since they can be used with many other APIs, and peripheral APIs, which tend to be more specialized in function (e.g., supplying a particular type of raw data) and less attractive for reuse. The structure of the affiliation network suggests that innovation (i.e., the appearance of mashups that create new affiliations) is concentrated in two areas: within the core set of platform APIs, and between the platform APIs and the periphery.

6 Conclusion

The growth of the mashup ecosystem has attracted media attention and fueled high expectations. In 2006, commentators called it "incredibly ripe for innovation" [18] and saw "numerous forces combining to make the mashup ecosystem 'explode'" [19]. However, the ProgrammableWeb.com data revealed a more modest growth pattern. (In fact, the growth rate declined toward the end of our sample period and into 2008.) Moreover, as the highly concentrated distribution of API popularity makes clear, simply releasing an API is no guarantee that mashup creators will build on it. These factors should encourage ecosystem participants to think carefully about the roles they want to play, and how to succeed in these roles.

This paper was motivated by the desire to help inform such thinking with empirical data from the mashup ecosystem. Our analysis was intended to be exploratory rather than conclusive. It shows that mashup developers do not simply mix and match APIs arbitrarily to create new mashups. Instead, they tend to build on a small subset of platform-like APIs (which become very popular and densely interconnected) while drawing less frequently from a much wider range of APIs that perform more specialized functions. By far the most popular of the platform APIs is Google Maps, which was used by 48% of the mashups in our sample.

Despite the apparent dominance of the most popular APIs, we find a significant role for the ecosystem's less popular APIs as well. Many of these peripheral APIs are involved in mashups that bring together novel combinations of functionality, thus creating new links in the API affiliation network. We view the structural richness of this network as a sign of innovative activity in the mashup ecosystem.

There are many limitations in our analysis. In particular, although ProgrammableWeb.com was the most comprehensive repository of API and mashup data available to us, it relies extensively on data reported by API providers and mashup creators. Both groups have incentives to participate (there is no cost to list an API or mashup, and both stand to benefit from the visibility provided by the site), but there are undoubtedly APIs and mashups in existence that are not registered.

There is also much more analysis that could be done. This paper took a first cut at describing the characteristics of the mashup ecosystem through graphical visualization and quantitative investigation of its network structure. Future research can build on these findings to develop a more rigorous theoretical framework for explaining the patterns we observed and predicting how the ecosystem will evolve. We hope this and subsequent work will help Web 2.0 participants make sound strategic choices about designing and releasing APIs, and enable mashup creators to innovate more effectively by recombining these APIs in compelling new ways.

Acknowledgments. Many thanks to John Musser of ProgrammableWeb.com for providing API access to the site, and to Darshan Santani for his generous help with data extraction and cleanup.

References

1. O'Reilly, T.: Web 2.0: Compact Definition? (2005), http://radar.oreilly.com/archives/2005/10/web-20-compact-definition.html
2. Wikipedia: Mashup, http://en.wikipedia.org/wiki/Mashup_web_application_hybrid
3. Cho, A.: An Introduction to Mashups for Health Librarians. JCHLA 28, 19–22 (2007)
4. Kulathuramaiyer, N.: Mashups: Emerging Application Development Paradigm for a Digital Journal. JUCS 13, 531–542 (2007)
5. O'Brien, D.S., Fitzgerald, B.F.: Mashups, Remixes and Copyright Law. INTLB 9(2), 17–19 (2006)
6. Goodman, E., Moed, A.: Community in Mashups: The Case of Personal Geodata (2006), http://mashworks.net/images/5/59/Goodman_Moed_2006.pdf
7. Jackson, C., Wang, H.J.: Subspace: Secure Cross-domain Communication for Web Mashups. In: 16th International World Wide Web Conference, pp. 611–620 (2007)
8. Liu, X., Hui, Y., Sun, W., Liang, H.: Towards Service Composition Based on Mashup. In: 2007 IEEE Congress on Services, pp. 332–339 (2007)
9. Hinchcliffe, D.: Is IBM Making Enterprise Mashups Respectable? (2006), http://blogs.zdnet.com/Hinchcliffe/?p=49
10. Barabási, A.-L., Albert, R.: Emergence of Scaling in Random Networks. Science 286, 509–512 (1999)
11. Anderson, C.: The Long Tail (2004), http://www.wired.com/wired/archive/12.10/tail.html
12. Watts, D.J., Strogatz, S.H.: Collective Dynamics of "Small-World" Networks. Nature 393, 409–410 (1998)
13. Wasserman, S., Faust, K.: Social Network Analysis: Methods and Applications. Cambridge Univ. Press, Cambridge (1994)
14. Borgatti, S.P.: NetDraw: Graph Visualization Software. Analytic Technologies (2002)
15. Adamic, L.A.: Zipf, Power-Laws, and Pareto: A Ranking Tutorial, http://www.hpl.hp.com/research/idl/papers/ranking/ranking.html
16. Brynjolfsson, E., Hu, Y.J., Smith, M.D.: From Niches to Riches: Anatomy of the Long Tail. Sloan Mgmt. Rev. 47(4), 67–71 (2006)
17. Kilkki, K.: A Practical Model for Analyzing Long Tails. First Monday 12(5) (2007)
18. Berlind, D.: Mashup Ecosystem Poised to Explode (2006), http://blogs.zdnet.com/BTL/?p=2484
19. Hinchcliffe, D.: The Web 2.0 Mashup Ecosystem Ramps Up (2006), http://web2.socialcomputingmagazine.com/the_web_20_mashup_ecosystem_ramps_up.htm