OSBR.ca

The Open Source Business Resource

FEBRUARY 2011

FEBRUARY 2011

PUBLISHER

The Open Source Business Resource is a monthly publication of the Talent First Network. Archives are available at: http://www.osbr.ca

EDITOR

Chris McPhee
chris.mcphee@osbr.ca

ISSN

1913-6102

Editorial

Chris McPhee

From the Editor-in-Chief

The editorial theme for this issue of the OSBR is Recent Research. In this issue, the authors report on the findings and relevance of their recent research into open source and application ecosystems.

Carlo Daffara, head of research at Conecta, discusses the factors to consider when choosing open source licenses and business models, including recommendations for selecting a license to suit both business objectives and licensing constraints.

Monique Bardawil from Carleton University's Technology Innovation Management program outlines her recent research into the identification of key players within the mashup ecosystem. The results of network analysis techniques can help incumbents and entrepreneurs develop business strategies as API providers in the mashup ecosystem.

Amanda Shiga, CMS Practice Lead at non~linear creations, studied the competitive actions taken by API providers in the mashup ecosystem. She presents her findings, which yielded insights for API providers to consider when tailoring their competitive strategies to suit this environment.

Islam Balbaa, Technical Business Analyst at Kinaxis, describes his recent research into the Force.com AppExchange, which examined the fit between software-as-a-service products and the requirements of particular business units.

Chulaka Ailapperuma, Senthilkumar Mukunda, and Shruti Satsangi from Carleton University's Technology Innovation Management program describe recent research that illustrates how social network analysis can be used to study online communities, including free/libre open source software developer teams.

We encourage readers to share articles of interest with their colleagues, and to provide their comments either online or directly to the authors.

The editorial theme for the upcoming March issue is Co-creation. We have invited authors from the Research Forum to Understand Business in Knowledge Society (http://ebrf.fi) to contribute to this special issue. The Guest Editors will be Stoyan Tanev from the University of Southern Denmark and Marko Seppä from the University of Jyväskylä.

For subsequent issues, we welcome general submissions on the topic of open source business or the growth of early-stage technology companies. Please contact me if you are interested in submitting an article (chris.mcphee@osbr.ca).

Chris McPhee

Editor-in-Chief

Chris McPhee is in the Technology Innovation Management program at Carleton University in Ottawa. Chris received his BScH and MSc degrees in Biology from Queen's University in Kingston, following which he worked in a variety of management, design, and content development roles on science education software projects in Canada and Scotland.

Open Source License Selection in Relation to Business Models

Carlo Daffara

"The roads we take are more important than the goals we announce. Decisions determine destiny."
Frederick Speakman

This article provides recent research results from the European Union's FLOSSMetrics project (http://flossmetrics.org). The results focus on the business and practical aspects of the adoption of open source within software products or as a basis of service offerings. Research into free/libre open source software (F/LOSS) is usually conducted with a software engineering focus or with an emphasis on F/LOSS as a spontaneous or directed collaboration effort. The FLOSSMetrics project expanded that research with an investigation on how licenses, business models, and project choices affect development and productization. This article provides a summary of common licensing issues and business models choices in F/LOSS, and it provides a list of recommendations for selecting a license for a software project to suit both business objectives and licensing constraints.

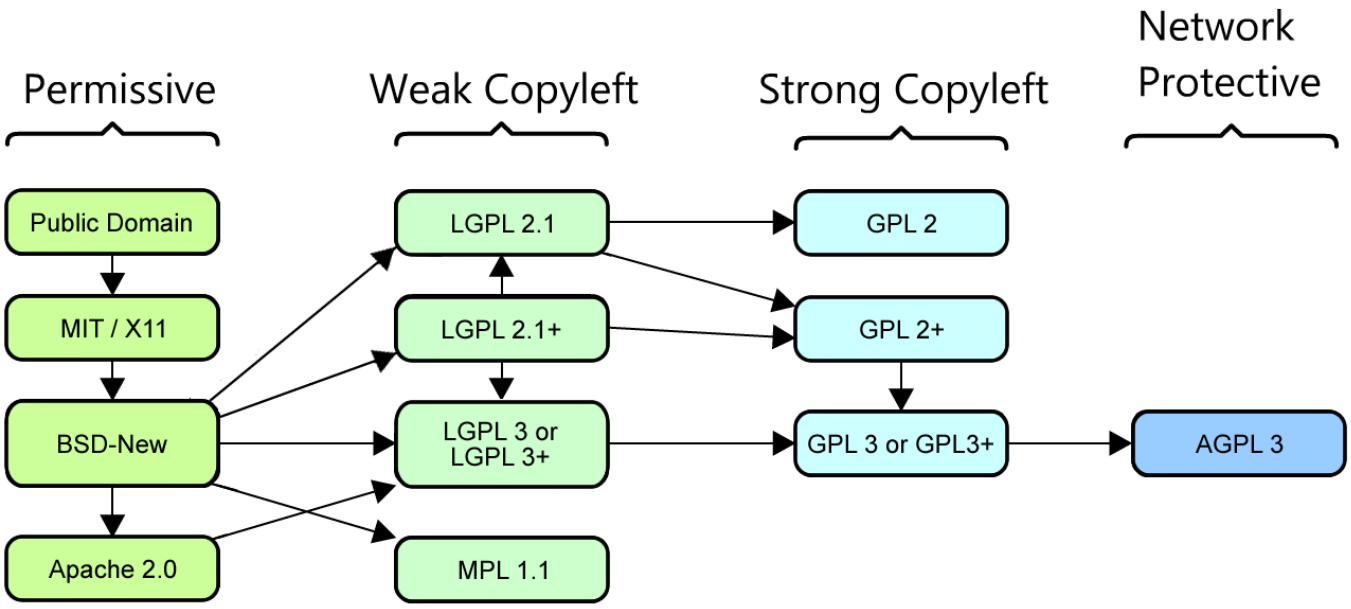
Introduction

There are literally hundreds of different licenses for free/libre open source software (F/LOSS), the majority of which are used for only a single software application. As of early January 2011, the top 20 most commonly used licenses are used in 96% of all projects, as listed in the Black Duck Software Knowledgebase (http://blackducksoftware.com/oss/licenses#top20). The GNU General Public License (GPL; http://gnu.org/licenses/gpl.html) family of licenses remains the most widely used license group for F/LOSS projects, with over 60% of all projects using one of the GPL licenses. This skewed distribution of license usage has prompted a community call for standardization on a limited set of known and recognized F/LOSS licenses, both to ensure a clear understanding of mutual obligations in case of mixing of code from different projects and to facilitate the process of managing contributions.

Components from different license groups sometimes can be combined together to create an aggregated object. Most licenses allow for such recombination freely, while some others introduce various constraints that may limit the potential reuse of a project in different conditions. Figure 1 illustrates how popular licenses may be combined. An arrow from one box to another indicates that those two licenses can be combined and that the combined result effectively has the result of the license at the arrow's destination. To determine whether two licenses can be combined, find a common license that can be reached by pathways leading from each license. For example, an Apache 2.0 license and a GPL2+ license can be combined using GPL3 or GPL3+.

A license of particular importance that is still not represented within the top 20 licenses listed above is the EUPL, the European Union Public License (http://osor.eu/eupl). This license was

Figure 1. Compatibility Relationships Between Popular F/LOSS Licences*



*Adapted from David A. Wheeler (2007; http://dwheeler.com/essays/floss-license-slide.html)

originally intended to be used for the distribution of software developed in the framework of the European Union's IDABC programme (http://ec.europa.eu/idabc/). This license is designed to be consistent with the copyright law in the 27 Member States of the European Union, while retaining compatibility with popular F/LOSS licenses such as the GPL. Version 1.1 of the EUPL was published by the European Commission in January, 2007 and is available in all official languages of the European Union. All 22 linguistic versions have identical value, which gives the EUPL a distinct advantage compared with the GPL, for which only the official, English edition is considered valid.

Intellectual Property Rights

The debate on software patents is still not entirely settled. On one side, most F/LOSS companies are vigorously fighting the process of patenting software-based innovations; on the other side, large software companies, for example SAP (http://sap.com), are defending the practice. An especially important point of F/LOSS licenses relates to "embedded intellectual property rights (IPR)." Embedded IPR is released code that relates to software patents held by the releasing authority. Most open source licenses explicitly mention that software patents held by the releasing authority are implicitly licensed for use with the code. This means that business practices that rely on separate patent licensing may be incompatible with some specific F/LOSS licenses, in particular the Apache License (http://tinyurl.com/6b8kb7h) and the GPL family of licenses. The Eclipse Public License (http://tinyurl.com/bg6frp) gives patent grants to the original work and to enhanced versions based on the original work but not to code that is not directly derived from the release. In contrast, permissive licenses like BSD (http://tinyurl.com/4bg6gb) and MIT (http://tinyurl.com/jye8e) give no patent rights at all.

If a license that explicitly gives IPR rights must be selected for purposes of compatibility or derivation, and the company or research organiza-

tion wants to maintain the rights to use IPR in a manner that is not compatible with the license, a possible solution may be the use of an "intermediate releaser." An intermediate releaser is an entity that has no IPR on its own, to which the releasing organization gives a copy of the source code for further publication. Since the intermediate release has no IPR, the license clauses that require patent grants are not activated, while the code is published with the required license. This approach has been used by Microsoft for some of its contributions to the Apache POI project (http://tinyurl.com/35mu3).

License Selection

The choice of an open source license for a project's code base is not clear-cut and depends on several factors. In general, when reusing code that comes from external projects, license compatibility is the major consideration in selecting a license. Red Hat has provided a compatibility matrix for its Fedora project to enable contributors to clarify compatibility issues they might encounter when mixing and integrating different components into this free Linux distribution (see http://fedoraproject.org/wiki/Licensing).

Licenses have an impact on development activity, depending on the kind of project and who controls the project's evolution. Some studies have shown that restrictive, copyleft licenses have a negative impact on contribution (e.g., Fershtman and Gandal, 2007; http://tinyurl.com/4vvu349). However, Stewart and colleagues (2006; http://tinyurl.com/4ndj4ju) found that restrictive licenses are associated with lower development activity in projects with non-market sponsors, such as foundations, than is seen in projects that are coordinated by a company. Generally, this effect is related to the higher percentage of “infrastructure” projects (such as libraries, development tools, and enabling technologies) undertaken by foundations.

Business Models

License selection is also impacted by the expected (or potential) business models underlying an open source project. F/LOSS business models can be analyzed by examining the two possible sources of value:

1. Intellectual property: a right that can be transferred. With F/LOSS, property is usually non-exclusive, with the exception of the open core business model where part of the code is not open at all. (For an overview of open source business models, including open core, see: http://slideshare.net/cdaffara/linuxtag-daffara.) Examples of intellectual property are trademarks, patents, and licenses – anything that may be transferred to another entity through a contract or legal transaction.

2. Efficiency: the ability to perform an action with a lower cost (both tangible and intangible). It is inherent in what the company does and how they do it, and it follows the specialization in a particular work area or appears following the creation of a new technology or process. For example, it could be the decrease in time necessary to perform an action associated with an increase in expertise and experience in performing this action. Another example is the introduction of a tool that simplifies a process and introduces a substantial improvement in efficiency.

These two sources of value are the basis of all open source business models, which can be represented along a continuum between property and efficiency (Figure 2). Among the results of our recent research, we found that property-based projects tend to have lower contributions from the outside because this requires a legal transaction for a contribution to become part of the company’s properties. Consider dual licensing: for contributions to become part of the

product source code, external contributors need to sign off their rights to the code so that the company can sell the enterprise version alongside the open version. Note that dual licensing also requires at least one of the licenses to be a strong copyleft license, like the GPL.

In contrast, models based purely on efficiency tend to have higher contributions and visibility, but lower monetization rates. It is important to recognize that there is no single ideal business model, but a spectrum of possible models, and companies should evolve according to changing market conditions and adapt their model as required. Some companies start with purely efficiency-based business models and build internal property value with time; others may start with property-based models and move to the other side to reduce engineering effort though increased contributions or to enlarge the user base and create alternative ways of monetizing users.

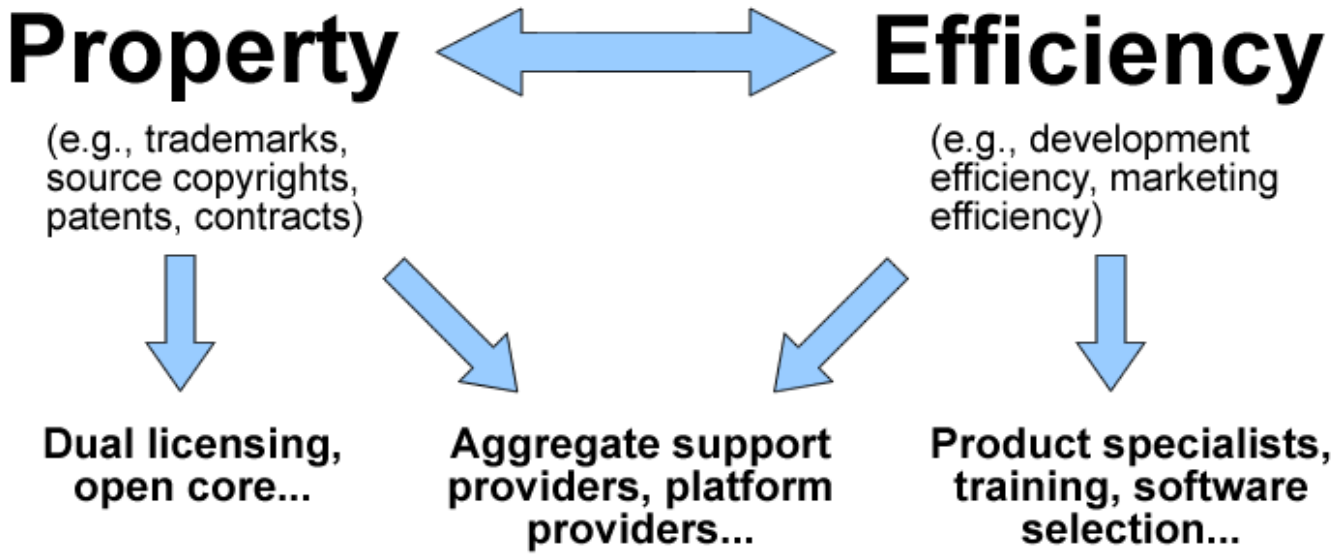
Recommendations

We have already identified some of the possible constraints in selecting a F/LOSS license for a project; among them, compatibility with an upstream project from which code has been reused, different contribution rates for non-market sponsors, and constraints related to the business model. In general, the recommended approaches follow from the main licensing and business model constraints:

1. When the project is derived from an external F/LOSS project, then the main constraint is the original license. In this case, the basic approach is to find a suitable license from those compatible with the original license, and select a business model that is consistent with the selected exploitation strategy.

2. When one of the partners has an IPR licensing policy that is in conflict with a F/LOSS license,

Figure 2. Open Source Business Models Along the Property-Efficiency Continuum



the project can select an MIT or BSD license (if it is compatible with an eventual upstream release) or use an intermediate release; in the latter case there are no constraints on license selection. If an MIT or BSD license is selected, some business models are difficult to apply. For example, open core and dual licensing are difficult to implement because the licenses lack the reciprocity of copyleft.

3. When there are no external licensing constraints, and external contributions are important, a license can be more or less freely selected, but models that reduce contributions (such as open core and dual licenses) should be avoided. When the software produced is related to infrastructure or when the future project releases are expected from a non-market entity (such as a consortia), a copyleft license may be more effective in stimulating developer participation.

Conclusion

Research into F/LOSS commonly focuses on community, participation, or contributions; licensing and business models are often overlooked. However, licensing and IPR are substantial factors in deciding whether or not a software project can be used in a specific environment. These factors also influence the degree of adoption by commercial companies as an embedded element. It is hoped that this summary of important license selection issues in relation to business models may help others decide upon the best approach to suit their circumstances.

Carlo Daffara is head of research at Conecta, an open source consulting company. He is the Italian member of the European Working Group on Libre Software, chairs several other working groups, including the Open Source Middleware Group of the IEEE Technical Committee on Scalable Computing and the Internet Society Working Group on Public Software, and contributed to the article presented by ISOC to UNESCO on global trends for universal access to information resources. His current research activity is centered on the sustainability of business models for open source software.

Key Player Identification in the Mashup Ecosystem

Monique Bardawil

"A leader needs enough understanding to fashion an intelligent strategy."

John Kotter

By combining multiples sources of data to create a new application, mashups represent a powerful source of innovation. Together, the various data providers, developers, mashup platforms, and users constitute an ecosystem that depends on innovation from these various players for its growth and success. This article summarizes recent research into the network structure of the mashup ecosystem, along with the positions and roles of entities within it. This research illustrates analytical methods for identifying key players in an ecosystem, while delivering new insights into the structure of the mashup ecosystem. Finally, the implications of these findings for entrepreneurs and incumbents are discussed.

Introduction

To succeed in innovative markets, companies need to make decisions based on an accurate understanding of the structure and organization of their business environment. Mapping and characterizing a company's business environment will help companies understand their position and the position of other companies in that environment. For incumbents, this information allows them to adjust their business strategy in response to changes in the environment. For new entrants, characterizing a business environment is essential when developing innovations to suit that environment. The objective of this paper is to describe recent research into the structure of the mashup ecosystem and methods of identifying key players within it.

The Mashup Ecosystem

Facilitated by the evolution of Web 2.0, the mashup ecosystem grew and evolved through interactions between data providers, developers, mashup platforms, and users (Weiss and Gangadharan, 2010; http://tinyurl.com/4dbfpft). A mashup is a custom web application that uses various data sources to create and deliver new services. Data providers offer developers access to their data via application programming interfaces (APIs). Using these APIs, developers are able to query the data and build mashups that combine data from multiple APIs (Yu and Woodword, 2009; http://tinyurl.com/49sq8w5). Examples of API providers include Google and Yahoo, who release data through the Google Maps and Yahoo Search APIs, which can be used by independent developers to create mashups.

The mashup ecosystem relies on innovation for its growth and success. The ecosystem is supplied with raw ingredients through data and APIs, and the Internet provides the environment to support the creation of new applications by developers (Fichter, 2009; http://tinyurl.com/

5v9mqxj). The mashup ecosystem is unique because its growth is strongly influenced by independent players. Even though API providers supply the data and the tools to access the data, they are not the major driving force for mashup development. That process lies in the hands of developers.

Research Design

As part of the author's recent Master's degree in the Technology Innovation Management program (http://carleton.ca/tim) at Carleton University in Ottawa, the mashup ecosystem was studied to identify its key players.

The first step in analyzing the mashup ecosystem was to represent two networks within it: i) the API affiliate network, which captures the relationships between mashups and APIs; and ii) the API provider network, which captures the relationships between mashups and API providers. The API affiliate network represents linkages at the API level within the ecosystem, while the API provider network reflects these linkages at the provider level and therefore represents the ecosystem at the firm level.

The API and mashup data used to establish the structure of these networks were manually extracted from the ProgrammableWeb (http://programmableweb.com), one of the largest online repositories of mashups and APIs. The extracted data included mashups created between September 2005 and August 2010.

Once the API affiliate network and API provider networks were established, four distinct methods of network analysis were applied to these networks. These techniques examined different structural and positional properties in the two networks and identified key players in each. Descriptions of each type of analysis and their key findings are provided in the sections that follow.

Network Centrality Measurements

These techniques identify core and central entities in an ecosystem. Centrally located nodes have a more strategic and important position in the network and have faster access to information and resources. Thus, centrality indicates the extent to which a node's strategic position is defined by its strategic ties to other nodes in the network (Gnyawali and Madhavan, 2001; http://tinyurl.com/6y8z942). The types of centrality measurement used in this research were degree centrality, betweenness centrality, closeness centrality, and eigenvector centrality (http://wikipedia.org/wiki/Centrality).

The results confirmed that a provider's status is dependent on the specific centrality measurement being considered. Each of the different centrality measurements will identify central nodes in the network based on the network position examined by the measurement. The centrality measurements revealed a similar set of API providers that hold prominent position and influence in that network. However, the rank and importance of those providers shift dependent on the centrality measurement, as shown in Table 1.

Community Detection Algorithm

This algorithm identifies communities of nodes within a network (Girvan & Newman, 2002; http://tinyurl.com/4jpbzq9). Communities are sub-networks of nodes that are more densely linked internally than they are with the remainder of the network. In the mashup ecosystem, these communities can be described as collectives, because the connectivity of nodes is driven by user innovation rather than company alliances and associations.

The results of this analysis revealed that the mashup ecosystem is centered on a core collect-

Table 1. Centrality Measurement Results for API Providers

API Provider	Degree Centrality		Betweenness Centrality		Closeness Centrality		Eigenvector Centrality	
	Value	Rank	Value	Rank	Value	Rank	Value	Rank
Google	497	1	51237	1	0.14	1	1.00	1
Yahoo	409	2	25355	2	0.13	2	0.94	2
Twitter	210	3	11241	3	0.13	2	0.52	4
Amazon	196	4	7887	4	0.13	2	0.47	5
Microsoft	159	5	2375	10	0.13	2	0.58	3
Facebook	147	6	4154	5	0.13	2	0.43	9
eBay	134	7	2833	8	0.13	2	0.44	8
Last.fm	134	8	2618	9	0.13	2	0.45	6
Digg	127	9	4131	6	0.13	2	0.45	6
Technorati	111	10	757	16	0.13	2	0.41	12
AOL	108	11	1052	11	0.13	2	0.41	11

ive of 45 API providers that contribute to the ecosystem through the release of APIs that are frequently used together in mashup development. The formation of this core collective of providers is driven by developers that deploy those APIs in mashups. Prominent API providers in the core collective are 12seconds.tv, 43Things, Amazon, AOL, eBay, Google, Microsoft, Technorati, Wikipedia, and Yahoo. In addition, the community analysis revealed that there are multiple smaller collectives of API providers, consisting of members that do not belong to the core collective.

Key Player Problem

This methodology attempts to solve the problem of finding the key player in a network by focusing on two related sub-problems (Borgatti, 2003; http://tinyurl.com/6xlvqaz). First, it focuses on identifying the set of nodes that, if removed, would maximally disrupt communication among the remaining nodes in the network. Second, it focuses on identifying the set of nodes that are maximally connected within the network. In this way, Borgatti's method identifies a set of key players in a network, rather than a single key player.

Analyses of the key player problem revealed two similar sets of API providers, as shown in Table 2. It is not surprising that Google and Yahoo belong to both sets of key players identified by the two algorithms because they release a large number of APIs into the mashup ecosystem. However, API providers like Twitter, Facebook, Digg, and Box.net release one API each yet they rank as one of the key players. The popularity of these specialiaed APIs is likely due to the nature of the data they provide and their general appeal to mashup developers.

Table 2. Results of Key Player Problem Analyses

Type	Group Size	Key Players
Maximal disruption	5	Google Yahoo Amazon Twitter Facebook
Maximal reach	5	Google Yahoo Digg Twitter Box.net

Topological Importance Index

This technique measures a node's importance on its direct and indirect neighbours. The effect of a node outside its immediate neighbourhood can be determined by measuring its influence on other nodes with a pre-defined distance.

The results revealed that API providers exert different levels of influence on their immediate neighbourhood. Also, some providers (such as 23, 12seconds.tv, and 43Things) show stronger levels of influence on more distant neighbours relative to other providers (such as Google, Amazon, and Yahoo). AOL maintains its positional importance, suggesting that they occupy a unique position in the network. One can speculate that companies like Google and Yahoo have many direct connections to other providers on the periphery of the network but have fewer non-direct connections to other providers. And hence, their influence is highest on their direct neighbours. Similar to what was observed in the API affiliate network, these providers gain importance due to their high eigenvector centrality measurements and their connections to more central providers in the network.

Implications for Entrepreneurs and Incumbents

Entrepreneurs and incumbents in the mashup ecosystem need to develop appropriate business strategies when releasing APIs. The success of these APIs is dependent on their appeal to mashup developers as they drive the development and growth of the mashup ecosystem. However, API providers should also consider the following implications of this research when developing their business strategy within the mashup ecosystem:

1. New entrants should consider releasing APIs that are compatible with the key player APIs. This may encourage developers to use the entrant's API when developing mashups. The community detection algorithm identified a core collective of APIs that are closely linked due to their frequent usage in mashups. Developing APIs that are complementary to the APIs in the core collective may increase the chance of success in the mashup ecosystem.

2. This research can help incumbents and entrepreneurs when developing business strategies in the mashup ecosystem. As suggested by Weiss and Gangadharan (2010), new entrants can follow or complement the strategies of key players to achieve successful entry into the ecosystem. This is also true for incumbent providers who are looking to improve their position and their role within the ecosystem.

3. An API's influence and reach in a network is dependent on the position and centrality of its neighbours. Certain APIs and API providers have strong influence based on their connectivity to high-ranking providers. Entrepreneurs and incumbents should develop APIs that are compatible with high-ranking providers to improve their position and influence within the mashup ecosystem.

Key Player Identification in the Mashup Ecosystem

Monique Bardawil

Conclusion

The mashup ecosystem is a unique business ecosystem as it is dependent on innovation from users (independent developers) for its growth and success. The research described in this article examined this business ecosystem to identify its key players. The findings from this research can help incumbents and entrepreneurs develop business strategies as API providers in the mashup ecosystem.

Monique Bardawil recently completed her Master's degree at Carleton University with a thesis entitled "Identifying key players in the mashup ecosystem." Her research interests are social network analysis, product architecture and design, and Web 2.0. She holds a Bachelor of Engineering degree in Systems and Computer Engineering from Carleton University and a Bachelor of Science degree in Biochemistry from McGill University.

Competition in the Mashup Ecosystem

Amanda Shiga

"It is the recombinant nature of revolutionary innovations that contribute to their dramatic effects."
Andrew Hargadon

Mashups combine data from multiple sources to create innovative web applications. Data providers gain compelling advantages in offering an open application programming interface (API), but face a competitive environment where growth occurs by virtue of developers' independent choices and where competitors are also complementors.

This article explores the nature of competition within the mashup ecosystem by focusing on competitive actions taken by API providers and their link to mashup network structure. The resulting insights help entrants and incumbents refine their competitive strategies within this complex and unique environment.

Introduction

Over the last decade, the Internet has played host to an extraordinary explosion of developer innovation in the form of mashups. Built using web services exposed by third-party data providers via APIs, mashups combine data from multiple APIs into innovative web applications that often meet a long-tail need and have been hailed as "the next major new software development model" (http://tinyurl.com/6aah4je).

The mashup ecosystem is defined as the combined mashup platforms, data providers, and users that support the creation of mashups (Weiss & Gangadharan, 2010; http://tinyurl.com/4dbfpft). ProgrammableWeb.com (http://programmableweb.com), an online database tracking the mashup ecosystem, lists 33 APIs within the "mapping" category alone, showing healthy competition amongst API providers and representing a rich opportunity to explore it through research.

While mashup developers are motivated by the enjoyment of niche, long-tail problem solving and the opportunity to create novel applications with powerful, highly developed technologies (Floyd, Jones, Rathi, & Twidale, 2007; http://tinyurl.com/5vbu7tv), data providers gain significant benefits by offering an API, including free advertising and exposure, a new source of revenue, and free research and development. This article focuses on exploring competition in this space and providing insight to new entrants and incumbents offering an open API.

Uniqueness of the Mashup Ecosystem

The mashup ecosystem is a unique and complex competitive environment, as exemplified by the following characteristics:

1. Growth by independent choices: the mashup ecosystem grows by virtue of mashup developers' independent choices instead of purposeful strategic alliances in inter-firm networks.

2. A small world where the rich get richer: the majority of mashups use a small number of APIs and API popularity is self-reinforcing (Weiss & Gangadharan, 2010; Yu & Woodard, 2009: http://tinyurl.com/49sq8w5).

3. API providers as complementors: API providers function as both competitors and complementors; the combination of two or more datasets into a mashup may be more powerful than a lone dataset.

4. User innovation: mashup development can be considered part of the democratization of innovation, a trend driven by the steadily improving innovation toolkits made available to users, and the increasing ability for users to combine and coordinate their innovative efforts over the Web (von Hippel, 2004; http://tinyurl.com/5wceo3m).

5. Unique resource flows: in an inter-firm network, three types of resource flow occur between partners: information flow, asset flow, and status flow (Gnyawali & Madhavan, 2001; http://tinyurl.com/6govdvn). A firm's ability to access and use these resources varies based on its structural position in the network. The mashup ecosystem may have unique flows; suggested variations are listed in Table 1.

Competitive Actions in the Mashup Ecosystem

As part of the author's recent research, 1277 blog entries spanning five years and pertaining to eight mapping APIs offering similar functionality were categorized into 16 competitive action categories. For example, a blog entry from Google Maps announcing that reverse geocoding was added to the API would be classified as "product development." This process assumed that ac-

Table 1. Resource Flows in the Mashup Ecosystem

Typical Resource Flows Between Firms	Variations Within Mashup Ecosystem
Information flow: information and knowledge about competitive intent, strategies, and resources flows between connected firms.	***Innovation flow:*** the observed integration of two APIs may inspire imitation and innovation by both API providers (in the form of future enhancements) and mashup developers.
Asset flow: resources such as money, equipment, technology, and organizational skills flow between connected firms.	***Experience flow:*** accumulated developer experience in integrating two APIs influences the development of new mashups using those APIs.
Status flow: legitimacy, power, and recognition flow from high-status firms to lower-status firms.	***Status flow:*** a high-status API bestows that status on a lower-status API with which it is combined in a mashup (by association).

tions published on an API provider's blog comprise a competitive action. Table 2 outlines the 16 categories and their subsequent refinement to six final categories specific to the mashup ecosystem.

Examining the distributions of these competitive actions across the eight mapping APIs yielded the following insights:

1. Within the distributions of competitive actions, two action categories were universally dominant: feature enhancement and mashup spotlight. While feature enhancement can be considered an established category in competitive strategy theory (Ferrier, 2001; http://tinyurl.com/64xxret) – that is, corresponding to the product category – the mashup spotlight category is unique to the context of this research. This indicates that API providers are making efforts to promote the work of developers using their API, which is a testament to the increasing importance of product and service development by users.

2. The three oldest and most popular APIs in the research population are Yahoo Maps, Google Maps, and Bing Maps, based on their date of entry into the ProgrammableWeb database and subsequent activity. These three APIs were consistently in their top four categories within the distribution: marketing, mashup spotlight, feature enhancement, and instructional. This demonstrates consistency of behaviour in the major players in the mashup ecosystem, but may also indicate the relatively vast resources

Table 2. Categorizing Competitive Actions in the Mashup Ecosystem

Original 16 Categories	Refined Categories and Definitions
Feature enhancement New product Increased geographical coverage New platform support	***1. Product development:*** feature enhancement, new product, geographical expansion, or new platform support
Marketing Engaging the community Earning opportunity	***2. Marketing:*** any marketing activity (e.g., events, promotions) or engagement of the developer community
Mashup spotlight Mashup spotlight (tied to a current event)	***3. Success:*** highlight or promotion of a mashup using provider's API
Service disruption announcement Instructional Certification program launch Developer tool launch	***4. Service:*** tools, programs, announcements, and documentation for developers
Legal (terms of service update)	***5. Legal:*** any litigation activities or updates to terms of service
New partnership Acquisition	***6. Capacity:*** partnership or acquisition to increase output or service offerings

available to the industry-heavyweights that are providing these three APIs: Yahoo, Google, and Microsoft.

3. While marketing and feature enhancement correspond to established categories, the mashup spotlight and instructional categories emphasize service to the developer community. This further demonstrates a commitment from the major players in the mashup ecosystem to support user innovation.

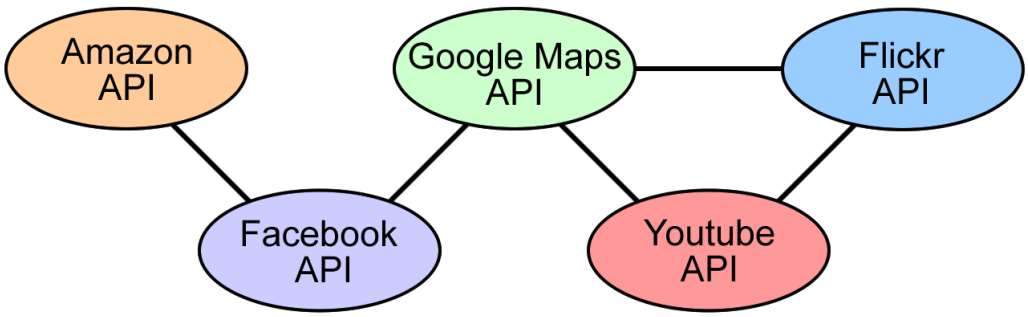
4. Lastly, the newest and least popular APIs within the research population were Cloudmade, Multimap, and Maponics. The providers of these APIs focused more on engaging the community and less on providing instructional services to developers. This may indicate that resources are more focused on promotion and growth for new entrants to the mashup ecosystem, and may also imply a lifecycle model where entrant behaviour evolves into incumbent behaviour. The providers of these less-popular APIs may also benefit from emulating the major players by improving service to the developer community.

A Network Within the Mashup Ecosystem

Approaching the mashup ecosystem as a network invites the use of social network analysis (http://en.wikipedia.org/wiki/Social_network#Social_network_analysis) to explore how APIs are connected via mashups and the overall characteristics of the network. Analysis of this network reveals the different ways in which mashup developers combine APIs to create original applications – specifically, an API's access to resources, its popularity, the number of times it has been combined with other APIs, and the diversity of those combinations. The network can be visualized as nodes and links, as illustrated in Figure 1, which shows a subset of the API network used in this research. Here, the nodes represent APIs and the links indicate APIs that have been used together in a mashup.

In the author's recent research, applying social network analysis techniques to the API network and testing relationships between network structure and competitive actions revealed several insights. Overall, network position was not observed to have a significant influence on competitive action patterns; that is, API providers do not appear to gain advantage from resource asymmetries in the API-API network. They may also simply give little regard to developer activities and existing mashups in their competitive strategies. However, an isolated result suggested a possible relationship between the volume of competitive actions taken and the diversity of

Figure 1. A Subset of the API Network



combinations with other APIs. This may indicate that a broader popularity within the mashup ecosystem may enable or drive increased competitive actions, and it suggests API providers may benefit from nurturing mashup combinations with many different other APIs.

Furthermore, APIs pairs that were frequently combined showed very different competitive action patterns over time. This may be attributed to API providers' efforts to differentiate themselves in competitive strategy, possibly to deter imitation, which is a threat in the context of modularization (Ethiraj, Levinthal, & Roy, 2008; http://tinyurl.com/6kcvv84). Other research indicates that gains in market share increase when top firms seek a unique approach to their diversity of product offerings, technological leadership, and branding (Ferrier, Smith, & Grimm, 1999; http://tinyurl.com/6b6awb4), and API providers may be well-served to diversify their competitive strategies, especially when faced with close competition for mashup market share.

Conclusion

API providers have much to gain in offering an open API, including revenue, exposure, and free research and development. However, the mashup ecosystem is a complex and unique competitive environment. API providers take specific actions in their competitive strategies, and they place a strong emphasis on frequent feature enhancements and promotion of third-party developer mashups. This demonstrates their commitment to user innovation. New entrants to the mashup ecosystem may gain an advantage in focusing on service to developers.

Competitive action is not strongly embedded in the API network structure. However, entrants and incumbents would be well served by examining this network structure and their position within it to observe the volume, variation, and innovation of mashups formed with other APIs and adjusting their competitive strategies accordingly. The mashup ecosystem remains rich fodder for competitive dynamics research and a cutting-edge playing field for data providers looking to disrupt their markets and gain a new competitive edge.

Amanda Shiga recently completed the requirements for the Technology Innovation Management Master's program at Carleton University. Her thesis, entitled "Mashup network ecosystem structure: A driving force of competitive actions?" examined competition and network structure in the mashup ecosystem. Prior to her work at Carleton, Amanda received her B.Sc. in Computer Science at the University of Ottawa. Amanda has over 10 years' experience delivering web-based business solutions and currently leads the CMS Practice Area at non~linear creations.

Software-as-a-Service Offer Differentiation by Business Unit

Islam Balbaa

"There is a great satisfaction in building good tools for other people to use."

Freeman Dyson

This article summarizes the author's recent research into the fit between software-as-a-service (SaaS; http://wikipedia.org/wiki/Software_as_a_service) tools and the requirements of particular business units. First, an overview of SaaS is provided, including a summary of its benefits to users and software vendors. Next, the approach used to gather and analyze data about the SaaS solutions offered on the Force.com AppExchange is outlined. Finally, the article describes the managerial implications of this research.

Introduction

SaaS is software that is deployed over the Internet to be run on local machines. This is a cloud-based software distribution model in which vendors host applications and manage their infrastructure and online delivery to customers. SaaS has become popular recently, in part due to advances in software and hardware that make this approach feasible, but also because of reduced cost and increased availability of bandwidth. As a result, it is now affordable for companies to acquire the level of connectivity required to allow online applications to perform well.

SaaS provides a number of benefits to both consumers and vendors (http://tinyurl.com/32bpwx). For customers, SaaS relieves them of the frustration of high up-front costs and vendor lock-in associated with the traditional software buying cycle, in which the purchase of software licenses is followed by time-consuming and expensive upgrades. Instead, SaaS give customers greater control over their software expenditures through flexible payment models, such as monthly subscriptions, "pay per use" or "pay per transaction" models, or payments linked to the achievement of business goals. SaaS provides greater accessibility and mobility. It also makes IT administration considerably easier and cheaper since the service typically includes maintenance and perhaps support as well. A company can rapidly scale up or down to meet changing demands. There is also no client/server installation or maintenance, even for upgrades, which happen frequently and automatically at the host side. It also provides companies with greater freedom to switch to another provider.

For vendors, SaaS provides a predictable and steady revenue stream that can be reliably forecasted. The ability to closely monitor a customer's usage also provides insights into further development and sales opportunities. Also, since the software is hosted by the vendor, it is

easier to deploy small, incremental upgrades and defect fixes compared with rolling out patches to on-premise software.

Today, SaaS provides alternatives to on-premise software in most industries and there are many applications available to support business activities across various business units, including sales, marketing, support, project management, finance, human resources, and information technology. However, it can be difficult for users to find the tools that are best suited to the requirements of their specific department and industry. While the wide availability of SaaS solutions means that users benefit from a variety of options to choose from, there is an accordingly large increase in the time required to research the right tool for a specific business requirement.

This article summarizes recent research by the author to analyze the factors that differentiate various SaaS supplier offerings so that potential customers can save time finding suitable tools to meet their needs. This research is relevant to business system managers and IT managers who are responsible for providing their organization with high-value products and applications that are adaptable and cost-effective. It will help them identify the applications that will promote efficiency and productivity within their organization. This research is also relevant to SaaS vendors because it will help them identify areas of saturation and opportunity within the market, as well as informing their sales strategy.

Research Approach and Findings

As part of the author's Master's thesis in the Technology Innovation Management program (http://carleton.ca/tim) in Ottawa, a study was conducted of 431 SaaS firms active within the Force.com (http://salesforce.com/platform/) cloud-based platform for SaaS business applications. Force.com allows external developers to create add-on applications that integrate into the main SalesForce application and are hosted on SalesForce.com's infrastructure. The directory of applications built for SalesForce by third-party developers is known as the AppExchange. At the time the data were gathered, the AppExchange offered more than 1000 SaaS applications.

The study used a data-mining (http://wikipedia.org/wiki/Data_mining) technique to extract patterns from the data, which in this case were keywords relating to the different types of SaaS offers available and their relevance to the functions of different business units. The business units studied were sales, marketing, product management, support and maintenance, project management, human resources, finance, and information technology.

The data-mining technique began with the selection of keywords related to the functions of a firm's different business units. The extensive search process for selecting keywords included both functional and non-functional criteria. Functional criteria cover the major functions for which the department would use the software application, including keywords such as "sales pipeline" and "lead scoring" for sales software. On average, 14 functional criteria keywords were selected for each business unit in the study. Non-functional criteria include the add-on benefits of the software application, including its price, level of support, popularity, and user rating.

Next, data were gathered from each of the 431 webpages of businesses within the AppExchange section of Force.com. The data included information relating to the non-functional criteria, such as the product name, the product or firm's website, and the application's price, popularity, support options, and user rating.

After all the product names and their corresponding websites were gathered and documented, a keyword search tool was used to

assess the content of each website against the selected keywords (functional criteria). The results indicate the level of fit between a particular offer and a business unit's requirements. Table 1 provides a sample of the results by listing the top products for each cluster, or product area.

General Managerial Insights

The detailed research results identified the firms that offer SaaS solutions best suited to the needs of relevant business units. Here, the general managerial insights from this research are summarized:

1. Marketing solutions in the AppExchange are plentiful. Solutions offering marketing automation and integrated marketing communication are highly rated, supported, and popular. Relationship marketing solutions are highly rated but are not popular.

2. There are not many SaaS products designed specifically for sales in the AppExchange; most products in the AppExchange are marketing solutions. However, quote management solutions are popular. Together, quote generation and tracking form one cluster because they share related functions within quote management software.

3. Support and maintenance SaaS tools have average ratings and not many reviews, perhaps because most support and maintenance software

Table 1. Top Products by Cluster

Product Area	Product (*Provider*)
Marketing automation services	1. Marketing Automation, Lead Management, and Email Marketing (*Marketo*) 2. Marketing Automation, Demand Generation, Email Marketing (*Genius Inc.)* 3. Eloqua Express (*Eloqua*) 4. SEO for SalesForce (*DemandResults*)
Product management software service	1. DreamTeam Project Management (*Dreamfactory*) 2. Fulfillment Manager (*@task*) 3. Projector Professional Services Automation (*Projector*) 4. Resource Management (*@task*)
Support software services	1. Ticket Management (*TeamSupport*) 2. AMC Multi-Channel Integration Server (*AMC Technology*) 3. Style Intelligence - Reporting & Dashboard App (*InetSoftware*) 4. OnDemand Call Center Platform (*LiveOps*)
Finance operations services	1. Z-Billing 2.0 (*Zuora*) 2. Managed Payment 2.0 (*eWay*) 3. Intacct Financial Management (*InetSoftware*) 4. Expensify (*Expensify*)

is built in-house. Also, users may not be ready to take advantage of the support and maintenance SaaS tools available

4. Most support services, helpdesk, and customer services tools share the same feature sets, but are used in different ways. Support services tools are mainly used by customers and include ticket management and workflow features. Helpdesk tools are used by both customers and internal users; the user interface is very important in this context. Customer service tools can be used by many business units other than support and maintenance and related products are accordingly flexible.

5. There is room for SaaS vendors to provide product management software tools for comparing pricing, functionality, and benefits.

6. The popularity of tools for support services and marketing automation may be related to the maturity of this development area. In contrast, most products for human resources are poorly rated or unpopular. Relative to the other offer types, this is a new development area for SaaS tools on Force.com

Conclusion

Users and software providers are realizing the benefits of SaaS, as shown by the popularity of applications in the Force.com AppExchange. The research summarized in this article examined the fit between SaaS tools and the requirements of particular business units. The results will save time for potential customers looking for solutions to fit their needs, and they suggest areas of opportunity where SaaS providers may wish to focus product development efforts.

Islam Balbaa is a Technical Business Analyst at Kinaxis. He recently completed his Master's thesis on "Software as a Service Offer Differentiation based on Suitability for Particular Business Units" in the Technology Innovation Management Program at Carleton University in Ottawa. He also holds a Communications Engineering degree from Carleton University and has worked as an Application Specialist at Montera Corporation.

Q&A

Chulaka Ailapperuma, Senthilkumar Mukunda, Shruti Satsangi

Q. How is social network analysis used in studies of open source?

A. Social network analysis (SNA) can be used to study online communities, including free/libre open source software (F/LOSS) developer teams. SNA techniques provide insight into these communities and enable researchers to make predictions based on these insights. They can be used to model the nature and patterns of interactions that can be used as a predictor of group behaviour, trust, knowledge generation, and information diffusion (Crowston et al., 2010; http://tinyurl.com/4hw4ssv). SNA can also be used make predictions about other kinds of networks other than pure social networks, such as networks based on relationships between code artifacts.

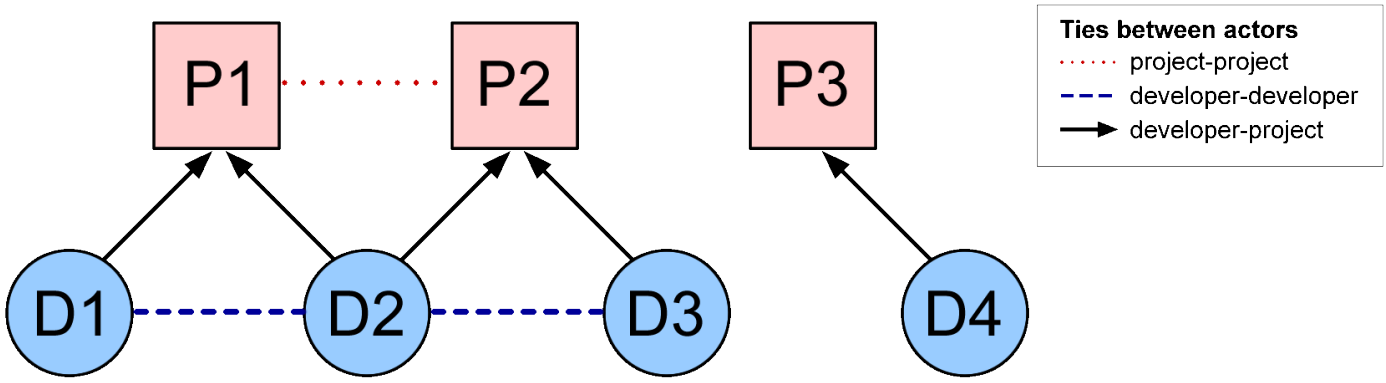
In this article, we answer the question of how SNA has been used to study open source. We begin by describing social networks and how they can be deconstructed to examine the relationships between entities within them. Next, we discuss social networks within F/LOSS communities and describe how SNA gives insights into the various actors and groups acting within networks. Finally, we provide an overview of common SNA measures used to study open source, including examples of how they have been used to provide insights about F/LOSS communities.

Social Networks

A social network is made up of individuals or organizations who are linked. It can be viewed as a network of nodes and links, where the nodes are actors (such as individuals or organizations) and links represent some kind of connection between actors. This connection could represent a variety of ties, such as affiliation or membership in an organization, dependency, social relationships, information flow, or interactions (Crowston et al., 2010). All these types of ties can be represented in a single network, which can illustrated graphically. For example, Figure 1 shows developers and their relationship with multiple projects, where actors such as projects and developers are represented as nodes and developer interactions and affiliations with projects are represented as ties.

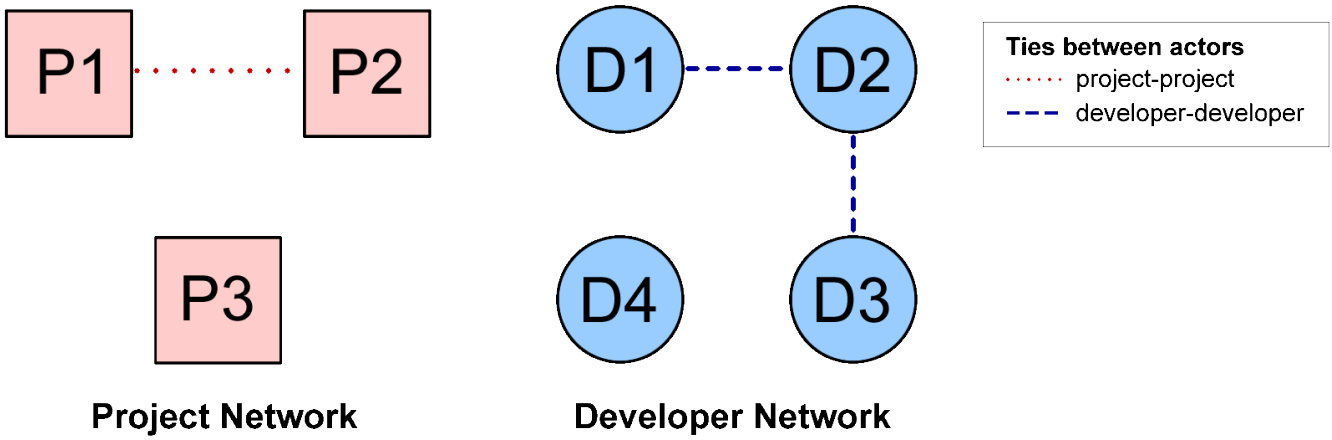
In this example, developers are linked if they belong to the same project, and projects are linked if a developer works on both projects. Even with just two types of ties as shown in Figure 1 (i.e, developer-project relationships and project-project relationships), the network can quickly become difficult to analyze. The analysis is improved by modeling developer-developer ties and project-project ties as two different social networks, as shown in Figure 2. Projects P1 and P2 are related because developer D2 (from Figure 1) is involved with both projects. Similarly, developer D1 is related to developer D2 because they both work on project P1. Also, developer D2 and D3 are related because they work together on project P2.

Figure 1. A Social Network*



*Adapted from Michael Weiss (2010, "SYSC5801: Open Source Business," Carleton University).

Figure 2. Deconstructing a Social Network*



*Adapted from Michael Weiss (2010, "SYSC5801: Open Source Business," Carleton University).

SNA examines the relationships between these actors, the characteristics of these relationships, and their impact on the actors. It provides a means to formalize social properties and processes by providing testable models of social concepts. SNA has been used for studying relationships between people, groups, organizations, and other social actors, including relationships within F/LOSS communities.

F/LOSS Communities and Social Networks

F/LOSS communities exhibit properties of social networks in that they consist of actors who are linked by some interdependency. SNA techniques have been used by researchers to understand the dynamics of such communities. For example, Madey and colleagues (2004; http://tinyurl.com/4sxu8y9) studied almost 60,000 F/LOSS projects hosted by SourceForge (http://sourceforge.net/) and applied SNA measures to detect the presence of certain properties of social networks in the SourceForge developer community. They found that the SourceForge community showed properties of being a social network in that: i) it has hub actors, who are key to information flow within the network and also tie separate parts of the network together; and ii)

it is a self-organizing system that forms "patterns of connectivity, that emerge from bottom up process based on local interactions."

The use of SNA in open source is not limited to using people or projects as the actors in a network. Nguyen and colleagues (2010; http://tinyurl.com/4kmsqhr) modeled the Eclipse project (http://eclipse.org) as a dependency network of software packages and used various network analysis measures to predict post-release failures in Eclipse projects.

Contexts for SNA

Social network analysis gives us insight into the various roles and groupings in a network. Most research asks the following types of questions:

1. Who are the information hubs within the network and who bridges different groups of clusters together?

2. Who is important in the network and who has influence over the network?

3. What is the level of activity in the network?

4. Where in the network is there a need for improved communication?

To answer these questions, identifying the types of actors is particularly important. Certain actors hold privileged positions within the network, which enables them to have greater influence over the network or earlier awareness of new information relative to other members of the network. For example, in a study of the spread of H1N1 virus, Christakis and Fowler (2010; http://tinyurl.com/3x4ueml), found that, by monitoring the health of central actors (rather than the usual approach of monitoring a random sample from the population), health professionals could detect the spread of the virus up to 16 days earlier in central actors than in the general population. Identifying central actors will enable organizations involved in F/LOSS projects to react to changes within the community faster and more aptly.

Another area where insights from SNA are important is organizational mergers. When organizations merge, challenges arise when combining the formal structures of operations. There is also an issue of merging distinct corporate cultures. Cultures are created, maintained, and shared through interactions between people in networks. Just after the merger, the new organization consists of two virtually separate social networks. If the social networks of the organization remain separate, so will their culture and the flow of communication between the people. Thus, efforts early on should be directed toward identifying central actors and combining the networks. To track the progress of the merger, snapshots of the organization-wide network should be taken at different points in time to measure the connectedness of the network and where gaps remain.

SNA Measures

The following SNA measures have been used to study F/LOSS communities:

1. Betweenness centrality: this measure identifies information hubs within a network, which act to bridge or "glue together" different parts of a network that would otherwise be apart (Martinez-Romo et al., 2008; http://tinyurl.com/4m5f7qy).

Madey and colleagues (2004) used betweenness centrality to study F/LOSS projects hosted on SourceForge. The study modeled the developer community as a collaborative network. The study demonstrated that “linchpin” or hub developers play a central role in linking fragmented developer communities in a F/LOSS community.

Martinez-Romo and colleagues (2008; http://tinyurl.com/4m5f7qy) used betweenness centrality to measure positions of developer leadership in a study of company involvement in an open source project. They showed that actors with high values of betweenness centrality are on paths that provide opportunities to others, even if they are not directly connected to those benefiting from the opportunities. By identifying the leaders and information controllers in the network, the study was able to show that company employees held leadership positions with low degree of turnover.

2. Eigenvector centrality: this measure identifies positions of importance and influence within a network. In the study of company involvement in an open source project, Martinez-Romo and colleagues (2008) used this measure to identify developers of high influence. Nguyen and colleagues (2010) used eigenvector centrality as a component measure to identify post-release failures in the Eclipse project.

The betweenness centrality and eigvenvector centrality identify different forms of leadership within a network. Betweenness centrality identifies information hubs; eigenvector centrality identifies nodes that have influence over the network. Martinez-Romo and colleagues (2008) showed that it is harder to gain positions of influence than become an information hub.

3. Coordination degree: this is a measure of the ability of a vertex to interchange information. It shows the ability of a node to receive information from the network and capture information about activity in a project (Martinez-Romo et al., 2008).

Martinez-Romo and colleagues (2008) used coordination degree to measure the role of a company in an open source project. They found that periodic, time-based releases of code increased developer activity more than feature-based code releases. Using a slightly different measure, the average coordination degree, the study found phases in which the network structure was efficient and when it was not. Comparing that with levels of corporate involvement, the study showed that corporate involvement in F/LOSS projects lead to more efficient development, but only if both the company and the F/LOSS community cooperate in the development efforts. There was less activity when there was no corporate involvement or when the company choose not to engage the F/LOSS community.

Conclusions

SNA provides a set of measures well suited to analyzing networks, including F/LOSS communities and other types of online networks. It allows researchers to visualize relationships within complex networks and provide insights into these communities.

Recommended Reading

For a detailed analysis of the use of SNA measures in studying online communities, including the limitations of this approach and recommendations for researchers, see: "Validity Issues in the Use of Social Network Analysis for the Study of Online Communities" by Kevin Crowston, James Howison, and Andrea Wiggins (2010; http://tinyurl.com/4hw4ssv).

Q&A: Social Network Analysis in Open Source

Chulaka Ailapperuma, Senthilkumar Mukunda, Shruti Satsangi

Chulaka Ailapperuma is Senior Software Developer at Canada Border Services Agency and is a graduate student in the Technology Innovation Management program at Carleton University. Chulaka also holds a Computer Science degree from Carleton University. He has 14 years experience in the computer science industry, working as a consultant for various clients, mostly in government and the telecommunications industry.

Senthilkumar Mukunda is a graduate student in the Technology Innovation Management program at Carleton University. He has over 4 years experience in Telecommunication and Railway Signaling Domain as embedded software developer. He holds a Bachelor of Engineering degree in Electrical & Electronics from Anna University.

Shruti Satsangi is a Wireless Engineer for Ericsson. She is also a graduate student in the Technology Innovation Management program at Carleton University, where she is researching coalition and competition within business ecosystems. She is a member of CU-Women in Science and Engineering, IEEE WiE, and the IEEE Communications Society.

Open Source for America: Federal Open Technology Report Card

From the Executive Summary: "Using open technologies creates cost efficiencies, more responsive and innovative software, and can help governments, enterprises and individual users avoid being dependent on a single vendor for software solutions. A 2009 Meritalk study indicated the U.S. federal government could save $3.7 billion by switching to open source solutions. Further, open source code is publicly available for review, meaning that flaws are more easily discovered and fixed. Open technologies are also a key ingredient to achieving the administration's drive to align the Federal budget and acquisition process with the technology cycle, strengthen program management, increase engagement with the IT community, and adopt light technologies and shared solutions. In many respects, the success of this reform effort will be more likely with continued emphasis and utilization of open technologies.

In light of the benefits that open technologies can bring to governments, and ultimately its citizens, Open Source for America (OSFA) conducted a review of fifteen (15) Cabinet-level departments and agencies to determine their use of open source technologies, open formats, and technology tools for citizen engagement. The results are summarized in this Federal Open Technology Report Card."

http://opensourceforamerica.org/reportcard/

Upcoming Events

February 9 and 10

Privacy and Security Conference and Exposition

Victoria, BC

"The Annual Privacy and Security Conference and Exposition, hosted by the Province of British Columbia, has become a leading event in North America for those working in the information privacy and security fields. Held in beautiful Victoria, British Columbia, Canada, the two-day conference draws an international audience of over 1000 delegates with an interest in cutting edge policy, programs, research and technologies aimed at the protection of privacy and security."

http://rebootconference.com/privacy2011/

March 9 to 11

CanSecWest

Vancouver, BC

"CanSecWest, the world's most advanced conference focusing on applied digital security, is about bringing the industry luminaries together in a relaxed environment which promotes collaboration and social networking. The conference lasts for three days and features a single track of thought-provoking presentations, each prepared by an experienced professional and talented educator who is at the cutting edge of his or her field. We give preference to new and innovative material, highlighting important, emergent technologies, techniques, or best industry practices."

http://cansecwest.com/

March 9 to 11

ConFoo

Montreal, QC

"PHP Québec, Montréal-Python, Montreal.rb, W3Qc and OWASP Montréal are proud to announce the second edition of the ConFoo Conference. From March 9th to 11th 2011, international experts in Java, .Net, PHP, Python and Ruby will present solutions for developers and project managers."

http://confoo.ca/en

Contribute

The goal of the Open Source Business Resource is to provide quality and insightful content regarding the issues relevant to the development and commercialization of open source assets. We believe the best way to achieve this goal is through the contributions and feedback from experts within the business and open source communities.

OSBR readers are looking for practical ideas they can apply within their own organizations. They also appreciate a thorough exploration of the issues and emerging trends surrounding the business of open source. If you are considering contributing an article, start by asking yourself:

1. Does my research or experience provide any new insights or perspectives?

2. Do I often find myself having to explain this topic when I meet people as they are unaware of its relevance?

3. Do I believe that I could have saved myself time, money, and frustration if someone had explained to me the issues surrounding this topic?

4. Am I constantly correcting misconceptions regarding this topic?

5. Am I considered to be an expert in this field? For example, do I present my research or experience at conferences?

If your answer to any of these questions is "yes," then your topic is probably of interest to OSBR readers.

When writing your article, keep the following points in mind:

1. Thoroughly examine the topic; don't leave the reader wishing for more.

2. Know your central theme and stick to it.

3. Demonstrate your depth of understanding for the topic, and that you have considered its benefits, possible outcomes, and applicability.

4. Write in third-person formal style. Formal first-person style (we only) may also be acceptable.

These guidelines should assist in the process of translating your expertise into a focused article which adds to the knowledgable resources available through the OSBR.

Upcoming Editorial Themes

March 2011: **Co-creation**
Guest Editors: Stoyan Tanev, U. of Southern Denmark; Marko Seppä, U. of Jyväskylä

April 2011: **Communications Enabled Applications**

Contribute

Formatting Guidelines:

Indicate if your submission has been previously published elsewhere.

Do not send articles shorter than 1500 words or longer than 3000 words.

Begin with a thought-provoking quotation that matches the spirit of the article. Research the source of your quotation in order to provide proper attribution.

Include a 2-3 paragraph abstract that provides the key messages you will be presenting in the article.

Any quotations or references within the article text need attribution. The URL to an online reference is preferred; where no online reference exists, include the name of the person and the full title of the article or book containing the referenced text. If the reference is from a personal communication, ensure that you have permission to use the quote and include a comment to that effect.

Provide a 2-3 paragraph conclusion that summarizes the article's main points and leaves the reader with the most important messages.

If this is your first article, include a 75-150 word biography.

If there are any additional texts that would be of interest to readers, include their full title and location URL.

Include 5 keywords for the article's metadata to assist search engines in finding your article.

Copyright: