

Using Hybrid Networks for the Analysis of Online Software Development Communities

Yevgeniy “Eugene” Medynskiy
Cornell University
ym66@cornell.edu

Nicolas Ducheneaut
Palo Alto Research Center
nicolas@parc.com

Ayman Farahat
PricewaterhouseCoopers
ayman.o.farahat@us.pwc.com

ABSTRACT

Social network-based systems usually suffer from two major limitations: they tend to rely on a single data source (e.g. email traffic), and the form of network patterns is often privileged over their content. To go beyond these limitations we describe a system we developed to visualize and navigate *hybrid networks* constructed from multiple data sources – with a direct link between formal representations and the raw content. We illustrate the benefits of our approach by analyzing patterns of collaboration in a large Open Source project, using hybrid networks to uncover important roles that would otherwise have been missed.

Author Keywords

Online communities, social networks, natural language processing, visualization.

ACM Classification Keywords

H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

The CHI community has had a long-standing interest in social networks, in great part because of their descriptive and analytical power. For instance social network graphs have been used as interface components to explore and navigate online social spaces [e.g. 3, 15], to examine patterns of scientific collaboration [8], or as the underlying basis for recommender systems in the context of group work [5, 9, 13], among others.

However, using social network analysis for the observation of online activities is not without limitations [12]. In particular, many structural analyses limit themselves to one kind of relationship (e.g. collaboration as expressed through email exchanges), whereas in practice social groups are

built on a much more heterogeneous web of relationships between diverse actors and resources. It is to reflect this heterogeneity that Science and Technology Studies scholars coined the term *hybrid networks* [e.g. 11] instead. Moreover, a second risk with structural analysis is to concentrate entirely on the form of network patterns [15], without giving any weight to the content generated by the members of the network. This pitfall could affect the reliability of conclusions drawn from analysis of the patterns, and therefore it is essential to preserve the links between formal representations and the original data [14].

Recent research has begun to address these issues. In sociology, [4] discusses “multi-modal” networks and presents one attempt at merging heterogeneous data sources. In the context of distributed software development [1, 2] describe systems linking various information repositories (in particular, email discussions and CVS records) from Open Source projects into hybrid networks. The Conversation Map [14] illustrates a different kind of hybridism by interconnecting the structural, semantic, and thematic networks constructed by participants in online discussions. Finally, [16] discuss how systems presenting hybrid, visual representations of online activities (such as theirs but also [14]) can be used as powerful images to trigger storytelling.

In this paper we describe our attempt at designing and using such evocative, hybrid representations of online activities. We have developed a system to observe and analyze collaborative activities in online groups merging three data sources: not only communication patterns (i.e. email traffic) but also topical and material relationships. Each component in this hybrid network allows easy access to the raw data, so that analysts can examine the qualitative information behind the structure of the network.

We illustrate our approach below by using our software to analyze development activities in a large Open Source project, Python. We show how the simultaneous visualization of heterogeneous data reveals collaboration patterns that would not have been visible using social networks exclusively. We conclude with a discussion of the current limitations of our software and propose avenues for future research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2006, April 22–27, 2006, Montréal, Québec, Canada.
Copyright 2006 ACM 1-59593-178-3/06/0004...\$5.00.

OBSERVING PYTHON USING HYBRID NETWORKS

We used PLSI to identify four main topics discussed by Python-dev's contributors in 2002. These four topics are represented as blues nodes in each of the four corners of Figure 1. They are: "code discussion" (represented by words such as 'class', 'object', 'method', 'integer'), "Unicode" (an important issue for Python in 2002, represented by words such as 'string', 'buffer', 'character', 'utf-8'), "release and maintenance" ('release', 'year', 'branch', 'stable') and finally "compiling and installation" ('module', 'build', 'package', 'configure').

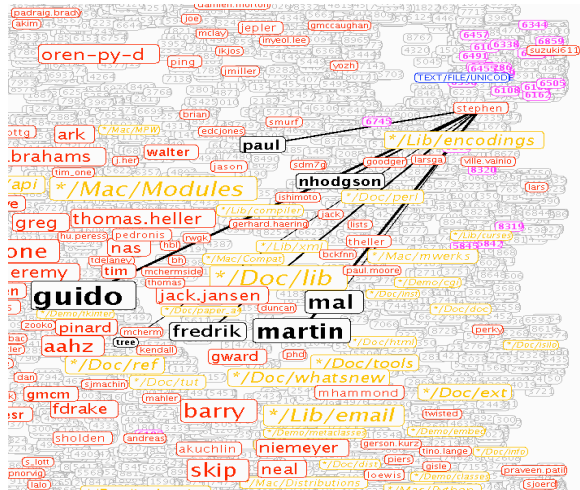


Figure 2 - Stephen's messages and connections

The messages sent to the python-dev mailing list are represented as gray ovals. As described above, they are laid out depending on their association with one of the four topics. The pile of messages directly under "code discussion," for instance, tends to be concerned almost exclusively with this topic, while messages at the center of the display are not as narrowly focused.

Authors are represented as red rectangles proportional in size to the volume of messages that have been written. Branches of Python's source tree are displayed as yellow rectangles, laid out in proximity to the topic they are associated with. Here size is proportional to the volume of CVS commits (e.g. /Lib/test in the lower part of Figure 1 was a major center of activity in 2002). When an author is selected, his or her links to other authors are shown in black (see Figure 2); connections with source code tree branches are shown in yellow. When a source tree branch is selected, all the authors who have worked on the branch are similarly highlighted in black and have yellow links to them (see Figure 3). In both cases, the width of the connecting lines is log-proportional to the amount of activity between the two nodes.

At this stage, without having selected any nodes, our system can already reveal interesting information about Python's organization. For instance, we can identify a core group of contributors writing a lot of messages about

most of the four topics (e.g. Guido, Martin). It is widely known that Python is the brainchild of Guido and that he plays a central role in the project: an analysis of his communication patterns places him at the very center of the project's communication network [2]. However our system nuances this picture. Indeed, it looks as if Python, at least in 2002, has two or three heads instead. Guido's messages tend to focus on code discussion and maintenance/release (he is positioned slightly to the left of the display) while Martin and Mal focus more on Unicode and tech support. Highlighting each author's messages by clicking on their nodes confirms this hypothesis.

Our visualization also points at possible work foci for the group in 2002. The upper right section of the display highlights the importance of Unicode-related developments for Python during the year, with subsequent modifications to the /Lib/encodings branch of the architecture. Selecting this library reveals an interesting division of the work: while several project members contributed to Unicode-related discussions (e.g. Stephen, see Figure 2), the purpose of these discussions was to convince the core members to implement certain changes (note how Stephen's messages are essentially addressed to Guido and his lieutenants). The core members could then decide to implement (or not) the member's ideas – all CVS commits to the Encodings library were made by Guido and his associates (Figure 3).

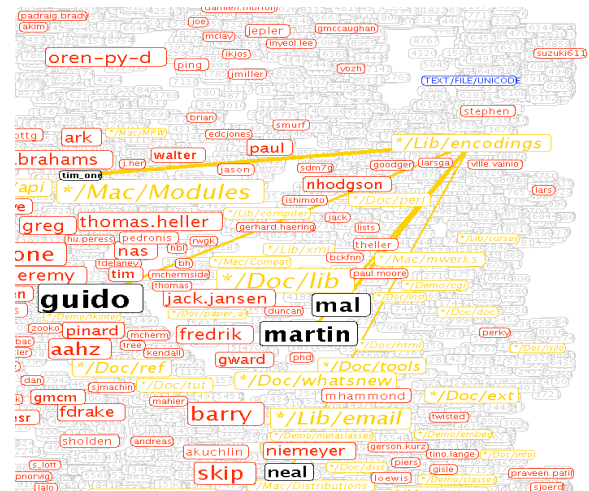


Figure 3 - Work on the /Lib/encodings branch of the project

The examples above illustrate our attempt at providing a more nuanced picture of communication flows in this project. Here our system suggests the possibility of a boundary between the material and discursive sides of the project. In Python it looks as if the transformation of ideas into code requires "translation" by the core members [2, 11]: holding the keys to the project's material infrastructure is an important source of power. Our system also shows how participants who would look peripheral from a purely structural point of view still play

