Understanding the KDE Social Structure through Mining of Email Archive

Studer, Matthias; Mueller, Nicolas Séverin; Ritschard, Gilbert

Understanding the KDE Social Structure through Mining of Email Archive

Matthias Studer
Department of Econometrics, University of Geneva
Uni Mail, 40, Bd du Pont d'Arve
CH-1211 Geneva 4, Switzerland
+41 22 379 82 64
Matthias.Studer@metri.unige.ch

Nicolas S. Müller
Department of Econometrics, University of Geneva
Uni Mail, 40, Bd du Pont d'Arve
CH-1211 Geneva 4, Switzerland
+41 22 379 82 36
Nicolas.Muller@metri.unige.ch

Gilbert Ritschard
Department of Econometrics, University of Geneva
Uni Mail, 40, Bd du Pont d'Arve
CH-1211 Geneva 4, Switzerland
+41 22 379 82 33
Gilbert.Ritschard@metri.unige.ch

ABSTRACT[1]

In order to achieve a better understanding of FLOSS social structure, we need a definition of social position. From a theoretical perspective, we propose to think the participation as a trajectory. Empirically, we use optimal matching to build a typology of participation trajectories based on KDE email archives. We show how these trajectories structure the community as a whole by combining these results with a social network analysis.

Keywords

Analysis of trajectories, Community of Practice, Optimal Matching, Social Network Analysis, Social Position, Social Structure.

1. INTRODUCTION

It is often proposed that the distinctive social structure of FLOSS communities could be one of the key reasons of its success [6, 9, 11 and 12]. How can we achieve a better understanding of the social structure of a FLOSS project? In order to understand this structure, we need a definition of "social position". Indeed, unlike usual institutions with clearly established social statuses, there are no such statuses in FLOSS. In many projects, some information on this subject may be available, such as the name of the project maintainer or some important contributors. However, this information is usually too general to approach the social structure and it is often not up to date. The underlying fact is that these statuses do not provide specific rights (or allow specific practices) *just because of carrying them*. They are a *sign* of an *already acquired* social status.

Thus, we need a theoretical framework that will enable us to identify the key dimensions of the social position in order to be able to measure it. In this article, we base ourselves on the theory of the communities of practice (CoP) brought by Lave and Wenger [10 and 16]. In CoPs, the membership and the social position are defined by practice, i.e. what one does in the community. Thus, the first dimension of the social position is the activity carried out. This point is already highlighted in Studer [15].

According to the theory of the legitimate peripheral participation [8, 10 and 16], we need to think practice, and thus the participation, as a trajectory. The practice changes in the course of time, as it must be learned and the social structure must be interiorized. Practice starts as peripheral (e.g. particular to the newcomers and nonessential to the community) and it tends towards a form of "complete" participation (e.g. an ideal form of practice) for people who place themselves in a trajectory of insertion. Nobody represents the complete and ideal practice. Thus, each contributor modifies his own practice unceasingly in order to approach this ideal. It is a continuous adaptation.

Along the same way, legitimacy within the community is acquired in the course of time. Thus, we should not look only at the current situation, but also to the past actions that constitute the current legitimacy and prestige of a given contributor. This legitimacy is important in two ways. First, without legitimacy it is not possible to act within the community. Second, legitimacy and prestige should be thought of as a resource which can be used to act within the community (i.e. influence). As such, it is one of the key dimensions of the social position.

In order to take trajectories into account, we rely on optimal matching [1, 2, 3 and 4]. This method is based on "sequence alignments", that is comparison of all pairs of sequences to generate a matrix of distances between them. We then use a cluster algorithm to build a typology of trajectories. This method will allow us to add two dimensions to the type of action carried out: since how long one contributes and the regularity of this contribution. We build our trajectories from KDE[2] mail archives. We then investigate the relationships between the typology of trajectories and social positions identified through a social network analysis of KDE contributors.

The rest of the article is organised as follows. First, we briefly introduce our data set. After that, we describe the procedure followed to build our trajectories. Then, we describe the optimal matching approach and discuss our results. Finally, we look at the relationships between optimal matching and a social network analysis of the KDE mail archives.

2. THE DATA SOURCE

Our data source is constituted by e-mails sent to KDE mailing-lists and archived by MARC[3]. These e-mails come from the lists

[1] This study has been realized within the Swiss National Science Foundation project SNSF 100012-113998/1.

[2] K Desktop Environnent (http://www.kde.org)

[3] Mailing list ARChives (http://lists.kde.org)

of discussions within each project and sub-project. Two problems quickly arise: neither the e-mails addresses nor the names can be considered unique. Consequently, we used an in-depth search algorithm to put together "name-email" couples corresponding to a same contributor. Indeed, the algorithm suggests possible merges. Since all regroupings were human-supervised, we were forced to use a selection criterion. We considered only persons having sent at least ten messages during a period of at least one month and that were active between the first of January 2006 and the first of July of the same year.

We distinguish three types of measures: those concerning modifications in the revision control system, those about the bug tracking system and usual messages sent to different kind of mailing lists. Let us briefly review these measures before discussing how to measure participation.

2.1 Revision Control

There is a specific mailing list in our data set, « kde-commit », which gathers automatic notifications from the revision control system (RCS). The use of the mailing list instead of CVS or SVN information allowed us to use a unique source of information and to link this information with the one from other sources (namely Bugzilla and usual mailing lists) on the basis of names and e-mail addresses instead of using an account name.

The RCS should be thought of as a "shared repertoire" [16]. It contains the production of the community, i.e. the reason of its existence. As such, contributions made to the RCS are very important, at the production as well as at the symbolic level, since they can be considered as a "proof" of participation. We should remind that, in CoPs, participation and practice are some of the key dimensions of membership [16].

Data from RCS measure in an imperfect way only the participation of contributors since "write accesses" to RCS are usually given once the contributor is already considered as a "member". Thus, "commits[4]" made by a specific contributor can be the application of a patch submitted by someone else. On some part of KDE – translation for instance – this situation is very common, since only few people have "write accesses". As a consequence, measurement of participation cannot be based on commits only. Nevertheless, the latter is an important measure since it is a sign of the social position. The number of commits should be used alongside with other measures to get an overview of different kinds of participation.

We measure "commit" by the number of messages sent to the "kde-commit" mailing list. However, we did not count "silent" commits, nor usual messages sent to this mailing list.

2.2 Bug Tracking

Despite its name, the bug tracking system (BTS) – Bugzilla in the case of KDE – should be thought of as an interface of formal communication between "users" and "contributors". Activity in BTS is very important, since BTS is a place of communication between the inside and the outside of the community. This kind of communication is often seen as one of the key dimension of the success of FLOSS communities [11 and 12].

We measured activities done in BTS in two ways: "bug opener" and "non bug opener". First, we counted the number of modifications done by the contributor who opened the concerned bug report. We built this measure by looking at the first entry with the given bug ID[5] and called it "bug opener". This measure should reflect a user practice of the BTS since the goal is to receive a response from the inside of the community.

Similarly, we computed the number of modifications done by other contributors. In this case, practice should be more oriented toward bug management. It is an "inside activity" and it should be distinguished from "user practice". We called this measure "non bug opener".

2.3 Mailing Lists

As we have seen, some kinds of contribution, such as translation, are not reflected by "commits". Therefore, we considered also usual mailing lists archives and measured the number of messages sent. Since these lists may address very different issues such as user assistance, translation, development issues, internal organisation and so on, our measure remains vague. However, it is worth taking this dimension into account. It reflects participation to decision processes and other community centred practices essential to define the social position. This dimension also measures a kind of personal commitment in the community.

2.4 Measuring Participation

We have thus the three above mentioned sources of information to measure participation. We cannot rely solely on one of these sources. Hence, we should look at characterizing usual configurations of these measures. By reducing all three measures into a single scale, we would loss too much information. Therefore, we choose to represent them through a categorical variable.

Let us recall that we are interested in the social structure of the community. This structure includes users as well as project maintainers. Of course, some positions may be more important than others for the community and especially for its reproduction. The social structure relies on the relationships between these positions. Hence, we have to depict all kinds of contribution in order to understand the social structure. These relevant patterns will be characterized in subsection 3.1.

3. OPTIMAL MATCHING

We argued that participation should be considered as a trajectory. Practices evolve over time as well as the social position. Using optimal matching, we build a typology of participation trajectories. We start by explaining how "participation trajectories" were built from our data set. We present the optimal matching approach and clustering technique used for building the typology.

3.1 Participation Trajectories

In optimal matching, trajectories must be provided as an ordered sequence of states (category). In sociology, the position in the sequence represents the position in time. Each period (or position) is of equal length.

[4] Modifications made to RCS.

[5] Each bug reports as a bug ID given by Bugzilla.

We built our trajectories using a community time of reference. In other words, we used an identical reference time for all contributors. Since we want to understand actual dynamics of contributors, we choose to end the trajectories at the time the data were collected. Our trajectories are built as sequences of 36 periods of four weeks each. Thus, trajectories represent the participation of each contributor included in the analysis between the 24th of September 2003 and the 28th of June 2006.

In our construction of trajectories, each period corresponds to a four week interval starting Thursday at midnight (GMT). We preferred this solution rather than periods measured in months, since there is a "day of the week effect" on participation. In Table 1, we present the average values of the variables commit, bug and message for each day of the week calculated over the period starting from 10.01.06 until 28.06.06. We can clearly distinguish this "day of the week effect". Week-ends show less average activity (in each measure) than week days.

Table 1. Average activity by day of the week

Day	Average Commit	Average Bug	Average Message
Monday	296.3	280.5	219.8
Tuesday	300.9	286.5	240.4
Wednesday	290.2	297.6	247.4
Thursday	268.8	290.1	221.5
Friday	288.4	274.7	203.9
Saturday	243.2	246.7	154.1
Sunday	263.2	251.2	169.8

After having defined the period – the position in the sequence – we should assign to each of them a category or a state. From a theoretical point of view, we have showed that we should include all three measures in our analysis. To define the categories of practices, we used a cluster analysis.

For this analysis, we organised the data in "person-period" form. This means that we have 36 records for each contributor, i.e. a total of 105'552 records (36 periods times 2932 contributors). Each record is of length four, two variables representing the bug dimension. We created an a priori cluster with all records without any activity, since there is a qualitative gap between a small activity and no activity at all. Since we already created a theoretical category for period without any activity, the cluster analysis was finally done on the remaining 37'513 records.

We did not use the measure directly. We used a logarithmic transformation of each of them with the function: $y = \ln(1+x)$ (we added one in order to avoid troubles with null values). This makes differences between high values less important than differences between low ones.

Then, we used the "k-means" method from SPSS [14][6] to build the typology. We tried several solutions and retained the one with five classes. Table 2 shows the mean values of the variables "message", "commit", "non bug opener" and "bug opener" for each cluster of activity.

Aside from periods without any activity, we have five clusters of activities. The "Small" type corresponds to periods with small contributions, such as one bug report opened and/or one "non bug opener". The next cluster, "Bug", regroups period with an activity centred on bug reports ("bug opener" is much higher than "non bug opener"). Some messages are sent on mailing lists, the average being one. The cluster "Message" shows periods with an activity mainly characterized by sending messages on mailing lists. Unfortunately, we do not know the type of message sent. The cluster "Commit" includes periods with messages sent on mailing lists, the average being 22 modifications in the RCS. We can interpret this cluster as that of usual developer's activity. The last cluster named "High" includes periods of very high involvement according to all measures except "bug opener".

Notice the clear separation between "non bug opener" and "bug opener", as expected. Cluster with commits are more linked with "non bug opener" than with "bug opener". On the other hand, "Bug" which includes periods with activity appearing almost only in BTS shows much higher "bug opener". This simple distinction is meaningful.

Table 2. Average activity by cluster of activity

	Message	Commit	Non Bug Opener	Bug Opener
No Activity (N) (64.5%)	0.00	0.00	0.00	0.00
Small (S) (15.1%)	0.27	0.16	0.70	0.74
Bug (B) (5.4%)	1.09	0.14	2.99	8.14
Message (M) (8.4%)	7.86	0.21	0.52	0.40
Commit (C) (3.8%)	5.57	22.38	1.98	0.56
High (H) (2.7%)	33.06	47.95	39.45	3.13
Total (100%)	1.86	2.19	1.45	0.69

Having computed these clusters of activity, we can now build the trajectory of each contributor. We simply represent this trajectory as the succession of clusters of activity assigned at each period. For instance, the sequence below represents the trajectory of a given KDE contributor.

H M S H H C N S N N B S N S N N S B N N B N N S M N S M N B M H H H C S

Each letter stands for an activity cluster membership (the first letter). For instance, this contributor had an activity characterized by cluster "S" ("small") at period 3 (i.e. between 18.11.2003 and 16.12.2003). At period 32, the activity of this contributor was best characterized by state "H" ("high").

Figure 1 shows an aggregated view of all trajectories. For each period on the x axis, it shows the distribution of our population among the different clusters of activity. We can remark that periods without activity are the most frequent (in black on the figure). There seems to be more activities during the last six

[6] Agglomerative clustering algorithms are not applicable because of the huge number of entries.

periods. This is due to our selection criteria. Our population is constituted by people having contributed during the last six periods.

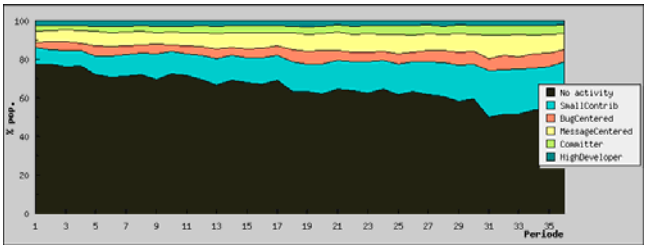

Figure 1. Aggregated trajectories of all contributors[7]

However, this presentation is too general and it does not permit to distinguish social positions. We will now present this part of the analysis: optimal matching and clustering sequences.

3.2 Clustering Sequences

Optimal matching gives an estimation of the "distance" between two sequences. The matrix of distances obtained by comparing all pairs of sequences can then be used to find clusters of trajectories (i.e. sequences) using classical methods. In other word, we can use this information to build a typology of different kinds of trajectories. We will now explain the general idea of optimal matching before presenting our results.

According to optimal matching, the distance between a given sequence A and another sequence B is equal to the minimum cost of transforming sequence A into sequence B. The total cost is the sum of the cost associated with each elementary operation required by this transformation. Since several solutions are possible, we select the minimum possible cost.

The elementary operations are of two types: insertion-deletion or substitution. Insertion-deletion, usually called "indel", corresponds to the insertion or the deletion of one element in one of the sequence. The cost associated with this operation is independent of the state concerned. For instance, the cost associated with inserting the state "Small" is the same as inserting the state "High". "Indel" operations deform the time structure of a sequence by allowing that a given subsequence appears later. The second kind of operation is substitution of an element (state) by another one. The cost associated with this operation depends on the state involved in the substitution. Table 3 shows the matrix of substitution costs used in our analysis. There are several ways for establishing these costs. Here, we derived the substitutions costs from the transition probabilities between two states. More precisely, we used the following formula to compute the cost (C) of substitution between states i and j based on transition probabilities (P):

$$C_{i,j} = C_{j,i} = 2 - P(i_t \mid j_{t-1}) - P(j_t \mid i_{t-1})$$

The indel costs were set to one. Thus, the cost of a deletion followed by an insertion is slightly higher than any substitution. We want to penalize this operation since it tends to introduce temporal distortion in the comparison of sequences and we want to take temporal structure into account.

[7] All figures are available in appendix in full size.

Table 3. Substitution costs between clusters of activity

	No activity	Small	Bug	Message	Commit	High
No Activity	0.00					
Small	1.38	0.00				
Bug	1.72	1.62	0.00			
Message	1.72	1.76	1.89	0.00		
Commit	1.93	1.97	1.97	1.85	0.00	
High	1.99	1.85	1.97	1.91	1.70	0.00

Substitution costs as well as distances between sequences were computed with the software TDA [13] ("Transition Data Analysis"). This software also implements several clustering algorithms. In the present study, we relied on SAHN (sequential, agglomerative, hierarchical, non-overlapping) algorithm with Ward criteria which minimizes intra-class inertia. This algorithm is widely used since it is close from a variance analysis and produces clusters with comparable sizes. In particular, it avoids small groups with only few individuals.

Finally, we retained the solution with nine clusters of trajectories. We briefly discuss these groups below. The figures showing aggregated trajectories for each cluster can be found in the appendix. At first look, we can distinguish three main dynamics: stable or regular trajectories, short term trajectories showing a strong involvement and finally a group of sporadic contributions.

3.2.1 Regular Contributors

The first four clusters of trajectories correspond to stable contributors in different area. The first cluster (Fig. 2 in appendix) shows contributors with a strong involvement on a long term period. By comparing with other trajectories, we see that cluster of activity "high" are mainly concentrated in this cluster of trajectories. The other types of trajectories show regular activity over a long term period mainly centred on message (Fig. 3), commits and development (Fig. 4) or bug reports (Fig. 5). These different clusters represent 17.2% of our population.

3.2.2 Short Term Contributors

Next trajectories show a strong involvement over a shorter period than the previous. It could be new contributors in a trajectory of integration or short term contributors that stop their activity once their contribution is finished. We found three clusters of this kind. Those centred on commits and development (Fig. 7), messages (Fig. 9) and bug reports (Fig. 8).

3.2.3 Punctual contributions

The last two clusters of trajectories show sporadic contributions. The first one (Fig. 6) shows very small contributions over a long time frame. The second one (Fig. 10) shows small contributions of several types over the last six month. These two clusters of trajectories are the most important since they represent more than two third of our population.

3.3 Conclusion from Optimal Matching

What have we learned from the results of optimal matching? We can mainly draw two conclusions, but results also raise some

questions. First, we have seen that some activities – in the way we measured them – are structured according to the trajectory. For instance, the cluster of "high" activity appears mainly in long term trajectories. This structuration leads us to think that some activities should be learned as well as the social structure that surrounds this activity. Thus, we can also think that the activity in a regular trajectory and the one in sporadic one may differ even if the measure (for instance, message) is the same. For instance, if a contributor sends messages on a "user oriented" mailing list regularly over a period of three years, it is not the same kind of contribution as a "user" sending several messages during a short period. Hence, we should also consider the regularity of the activity aside from intensity (as measured) and the kind of activity.

The percentage of each kind of trajectories show us the huge amount of sporadic and short term contributors in a FLOSS community. We should be aware that these percentages are under evaluated because of our selection criteria. This situation raises several questions. First of all, how are sporadic contributors "handled" by the community? For instance, are these contributions (or questions) handled by "strongly involved" contributors or by some contributors specialized in this kind of interaction? These interactions can be time-consuming and less prestigious.

According to our typology there are a lot of short term contributors such as "new committers" (Fig. 7), 3.3% of our population, comparing to "regular commit", 4.5% of our population. In our sense, this high importance of "short term" contributors could be challenging for the community and its reproduction since it might reflect a high incoming-outgoing movement. Hence, a comprehension of the social structure of FLOSS communities should explain the movement between new and old contributors.

In order to understand how trajectories are linked with the social position and how it can structure the community as whole, we propose to link these trajectories with a social network analysis of contributors. This is what we present now.

4. SOCIAL NETWORK ANALYSIS

If the clusters of trajectories reflect some specific social positions or roles, we should see some differences in how they are inserted in the web of social interactions. Thus, we propose to link our results with a social network analysis. This will give us the opportunity to discuss the relevance of the proposed methodology to study FLOSS community structure.

4.1 Building the Network

We used the information from mailing list archives as well as the one from the bug tracking system to build a social network analysis of all participants to KDE using Pajek [6]. In this kind of analyses, two definitions are essential: inclusion and relationship.

We included the same contributors as for the analysis of trajectory. Regarding relationships, we followed the method used in Studer [15]. We used the definition of "thread" from MARC archive to constitute our network. We have then defined the relationship between two persons as:

> The relationship between a person A and B is equal to the sum of all messages sent by A in "threads" where B also sent at least one message.

The relationship has a direction (from A to B) and the value is different according to this direction. However, and this follows from the given definition, if A has a relationship with B, then B has a relationship with A. It will not be automatically the same value. The graph obtained is directed and valued. Our measurement also contains a scale about the "force" of the relationship. Thus, taking part in a discussion with a lot of different participants implies more "relations" than taking part in a small discussion. In other words, each message is not equal in our construction of the network. This corresponds to some logic. By taking part in a large discussion (which has more chance to be considered as important), one acquires a greater visibility than in a small discussion implying only two people.

After having exposed how we have built the network, let us look at where our clusters of trajectories are located in the social network and especially at the differences of positions.

4.2 Social Network and Trajectories

We computed several indicators from the network in order to compare the clusters of trajectories. The degree is simply the number of arcs connected to a given vertex (contributor) in our social network. According to our definition of the network, the sum of incoming arcs corresponds to the number of messages received. The sum of outgoing arcs can be interpreted as an indicator of influence in our network.

Table 4. Network indicators average by cluster of trajectories

	Degree	**Sum of incoming arcs**	**Sum of outgoing arcs**
Regular Small (22.4%)	3.81	4.25	4.46
Sporadic (45.3%)	7.73	12.74	14.75
New BugSmall (4.7%)	14.09	20.79	19.36
Regular Bug (4.2%)	20.85	32.04	32.80
New SmallMessage (7.0%)	28.74	57.07	63.50
New Committers (3.3%)	41.45	78.50	65.75
Regular Commit (4.5%)	43.34	90.16	82.62
Regular Message (5.2%)	57.45	125.52	110.80
Strongly involved (3.3%)	218.26	654.04	656.47
Total (100%)	21.54	48.28	48.27

Unsurprisingly, involved and regular contributors hold a much more central position than other. The "New Committers" who seem to be quite central are the only exceptions. However, by comparing the degree and the sum of incoming arcs of "New Committers" and "Regular Commit", we can conclude that newcomers tend to have weaker relationships than regular contributors. Bug related trajectories tend to have more decentralized positions, according to theses indicators. For instance, the average degree of "Regular Bug" is quite smaller than other "regular" trajectories. By looking at these differences, we can identify three main dynamics that seem to influence the position in the network:

- The activity performed.

- The dynamic of trajectory (long term, short term, sporadic).
- The intensity of the activity performed.

This discussion confirms that we should look at the participation in a FLOSS community with a trajectory perspective and not just at a given point in time.

4.3 Relation between Clusters of Trajectories

The clusters of trajectories hold different positions according to our network indicators. However, if we consider these clusters as "social positions" or "social roles", we may be interested in the pattern of relationships between different kinds of trajectories. Thus, we built a "supra-network" of relationships between the different clusters of trajectories. In this "supra network", the relationship $R_{A,B}$ between two clusters, A and B, is equal to the average relation between the members of A and B:

$$R_{A,B} = \frac{\sum_{i \in A} \sum_{j \in B} r_{i,j}}{|A| \cdot |B|}$$

where:

$r_{i,j}$ is the value of the relationship between contributors i and j. If there is no relationship, it is equal to zero.

|A| and |B| are respectively the size of clusters A and B[8].

We obtained the graph shown in appendix (Fig. 11) using the Pajek software [5 and 7]. In this graph, we discarded all relationships smaller than the overall average of relationships (0.0156) in order to highlight relevant links between clusters of trajectories. The size of the vertex reflects the number of members of the concerned cluster. The cluster "strongly involved" occupies the most central position. It is also the only cluster connected to all other clusters of trajectories. According to this graph, we may think that most relationships with the "outside" of the community are carried out by the most central contributors. We do not observe any "filter role" between sporadic and strongly involved contributors. Moreover, the graph leads us to think that coordination of the community could be closely related to the management of outside relationships. This graph shows a very strong centralization of the network and seems to support the hypothesis that large decentralized project are organized as a collection of small centralized projects [9].

5. CONCLUSION

Optimal matching showed us the importance of considering participation and practice in community in a trajectory perspective. Activity, practice, legitimacy and social position depend on the personal history of community members. This history should be thought of as a whole aside with the intensity of participation and the kind of activity performed.

Our analysis showed (Subsection 3.3) the huge amount of short term and sporadic contributors. It raises the question of the reproduction of the community that seems to rely primarily on a small group of strongly involved members. But we are not able to provide evidence on this point, a longitudinal analysis of the community could help us to gain a new insight on this point.

Our analysis also showed that most contacts with sporadic contributors seem to be with the most involved members. In particular, contacts with regular contributors seem to be less important. This leads us to think that the community is very centralized on a small group of coordinators.

[8] When A = B, we set the denominator to |A| (|A|-1), the number of possible relationships within cluster A.

6. REFERENCES

[1] Abbott, A. and Forrest, J. Optimal Matching Methods for Historical Sequences, *Journal of Interdisciplinary History*, 26, 1986, pp. 471-494.

[2] Abbott, A. A Primer on Sequence Methods, *Organization Science*, Vol. 1, No. 4, 1990, pp. 375-392.

[3] Abbott, A. and Hrycak, A. Measuring Resemblance in Sequence Data: An Optimal Matching Analysis of Musicians' Careers, *The American Journal of Sociology*, Vol. 96, No. 1 (Jul. 1990), pp. 144-185.

[4] Abbott, A. and Tsay, A. Sequence Analysis and Optimal Matching Methods in Sociology, *Sociological Methods & Research*, Vol. 29, No. 1 (Aug. 2000), pp. 3-33.

[5] Batagelj, V. and Mrvar, A. *Pajek 1.19 – Program for Large Network Analysis*, http://vlado.fmf.uni-lj.si/pub/networks/pajek/

[6] Crowston, K. and Howison, J. The social structure of free and Open Source software development, *First Monday* 10, 2 (2005).

[7] de Nooy, W, Batagelj V. and Mrvar, A. *Exploratory Social Network Analysis with Pajek* Cambridge University Press, Cambridge, 2005.

[8] Edwards, K., *Epistemic Communities, Situated Learning and Open Source Software Development*, http://opensource.mit.edu/papers/kasperedwards-ec.pdf, 2001.

[9] Howison, J., Inoue, K., and Crowston, K. Social dynamics of free and open source team communications, In *Proceedings of the IFIP 2nd International Conference on Open Source Software* (OSS2006) (Lake Como, Italy, 2006).

[10] Lave, J. and Wenger, E. *Situated learning: legitimate peripheral participation*, Cambridge University Press, Cambridge, 1991.

[11] Raymond, E. S., The cathedral and the bazaar, *First Monday* 3, 3 (1998).

[12] Raymond, E. S., Homesteading the noosphere, *First Monday* 3, 10 (1998).

[13] Rohwer, G. and Poetter, U. *Transition Data Analysis*, Bochum: Germany, ver. 6.4l.

[14] SPSS Inc. *SPSS 14.0 for Windows User's Guide*, SPSS Inc., Chicago IL, 2006

[15] Studer, M. Community Structure, Individual Participation and the Social Construction of Merit, In *Proceedings of the Third International Conference on Open Source System* (OSS 2007)(Limerick, Ireland, 11-14 June 2007).

[16] Wenger, E. *Communities of Practices: Learning, Meaning, and Identity*. Cambridge University Press, Cambridge, 1998.

Understanding the KDE Social Structure through Mining of Email Archive: Appendix

Matthias Studer, Nicolas S. Müller, Gilbert Ritschard
Department of Econometrics, University of Geneva
Uni Mail, 40, Bd du Pont d'Arve, CH-1211 Geneva 4, Switzerland
{matthias.studer, nicolas.muller, gilbert.ritschard}@metri.unige.ch

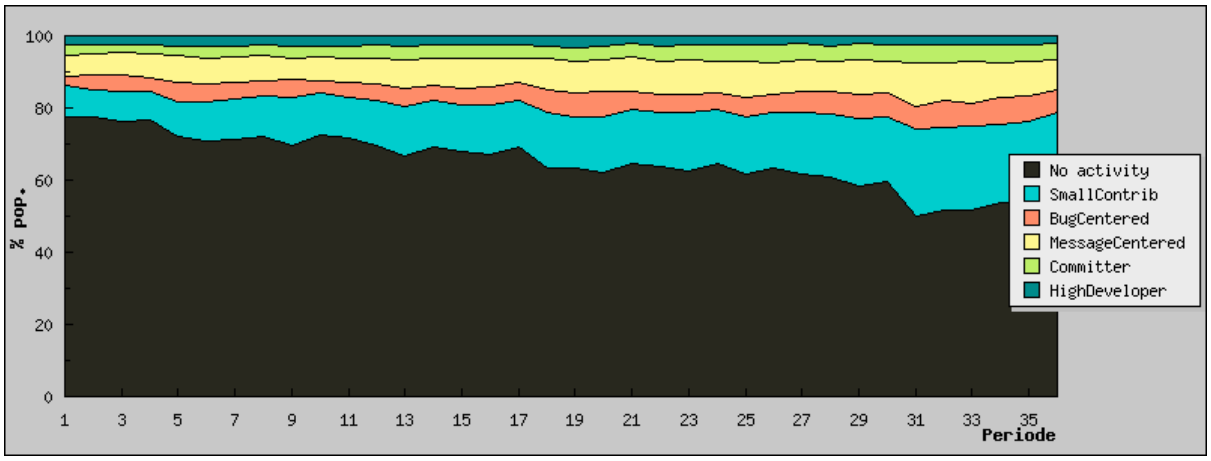

Figure 1. Aggregated trajectories of all contributors

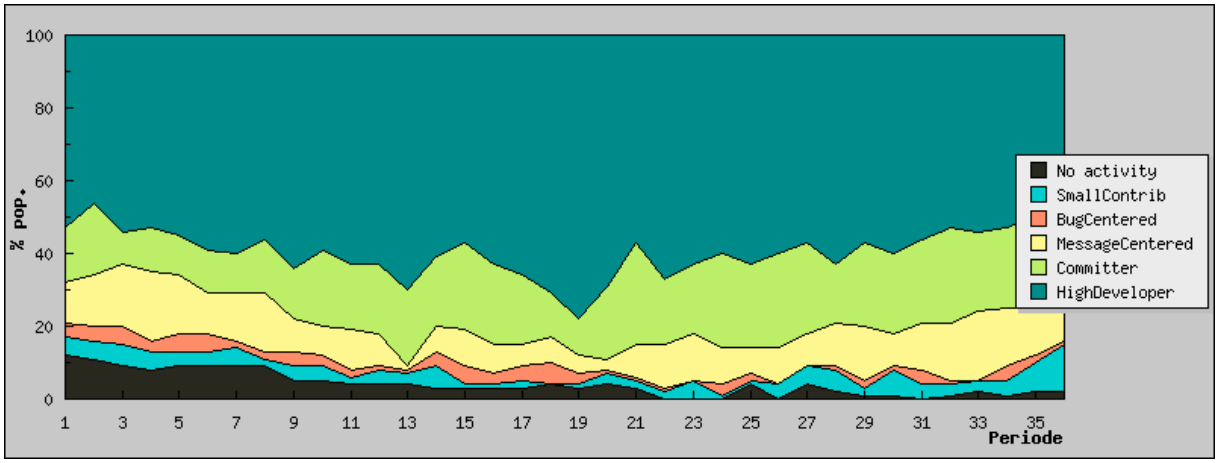

Figure 2. Strongly involved (3.3%)

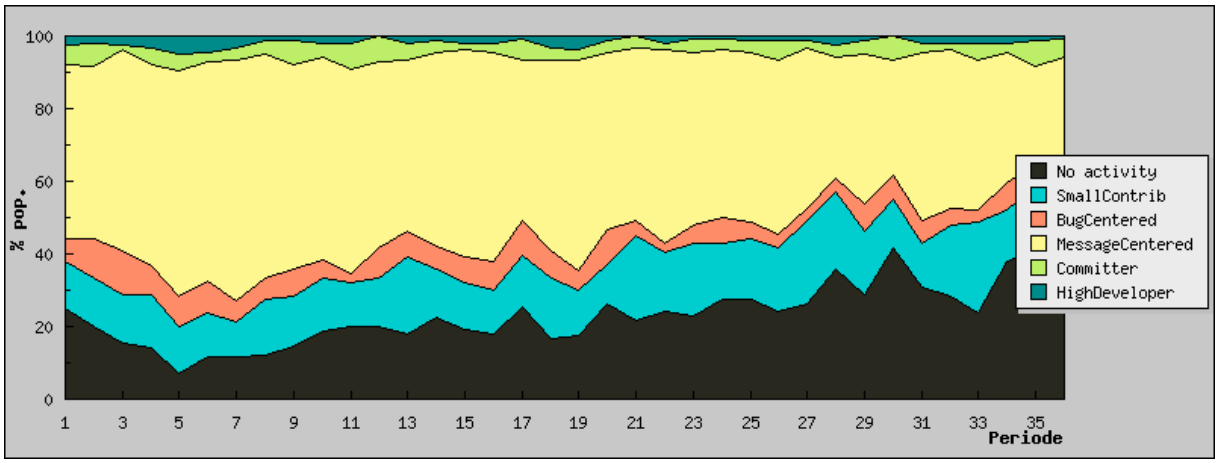

Figure 3. Regular Message (5.2%)

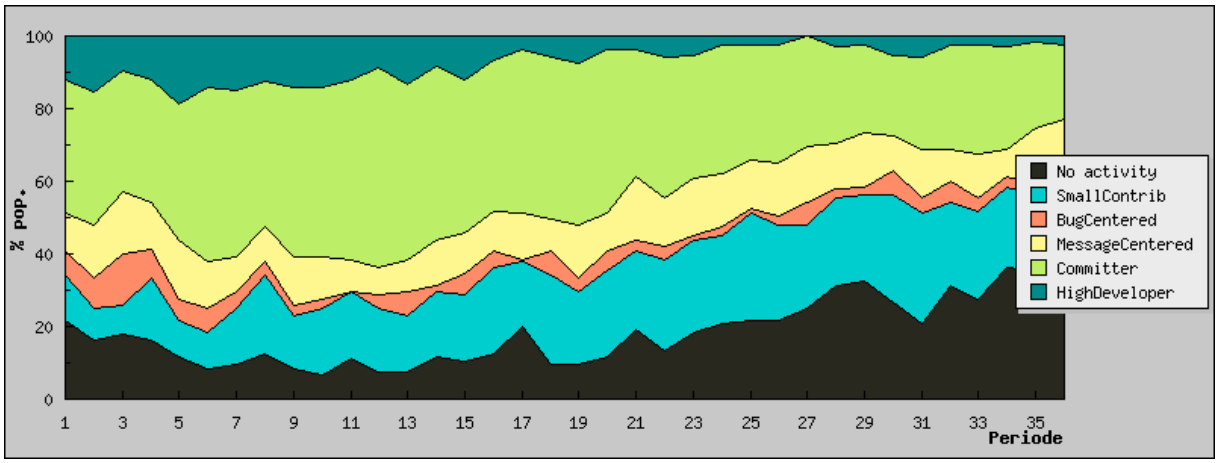

Figure 4. Regular Commit (4.5%)

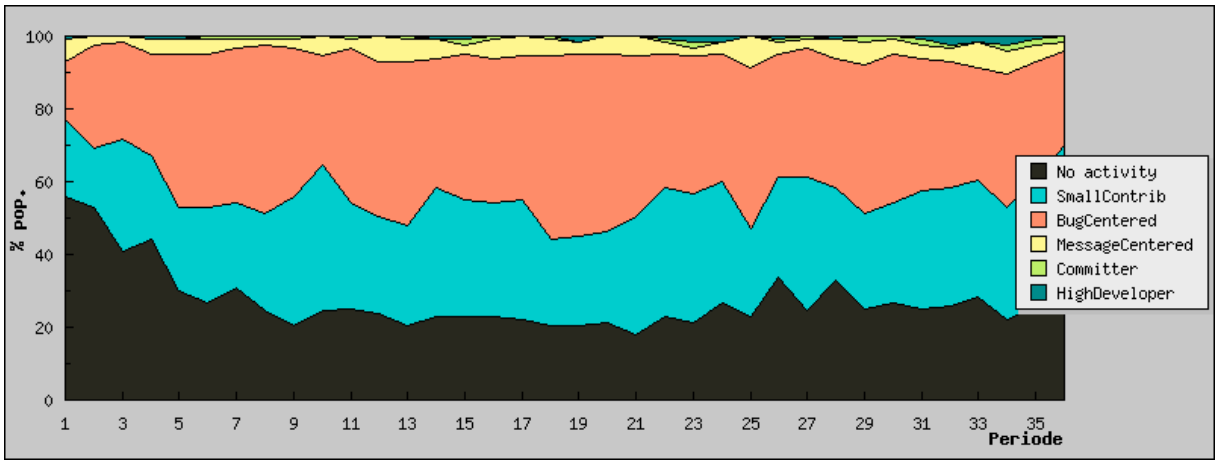

Figure 5. Regular Bug (4.2%)

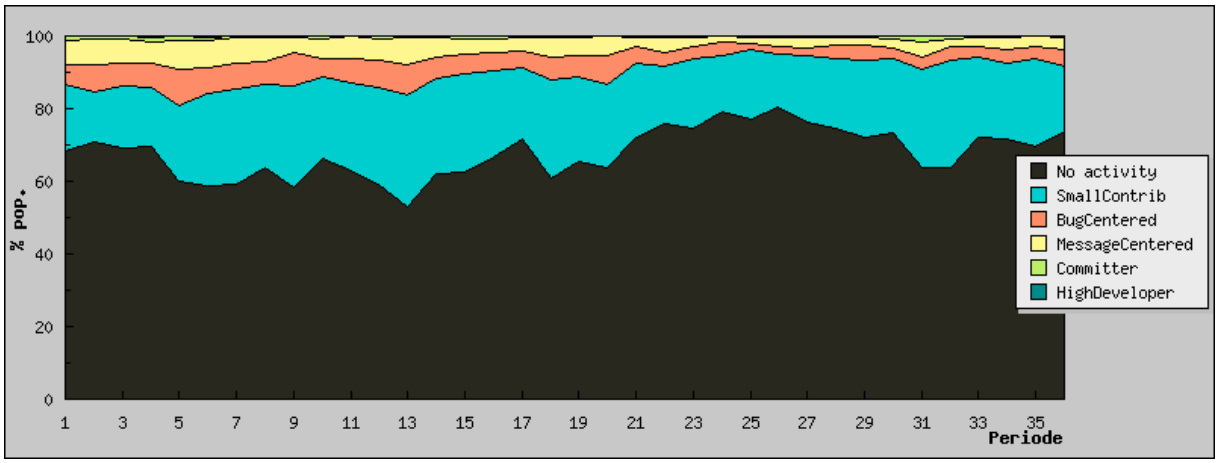

Figure 6. Regular Small (22.4%)

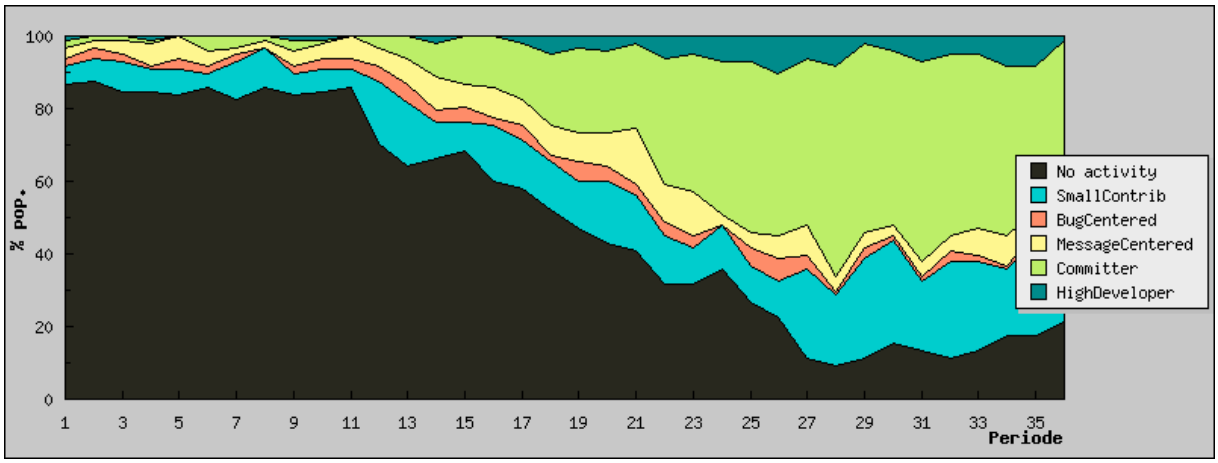

Figure 7. New Committers (3.3%)

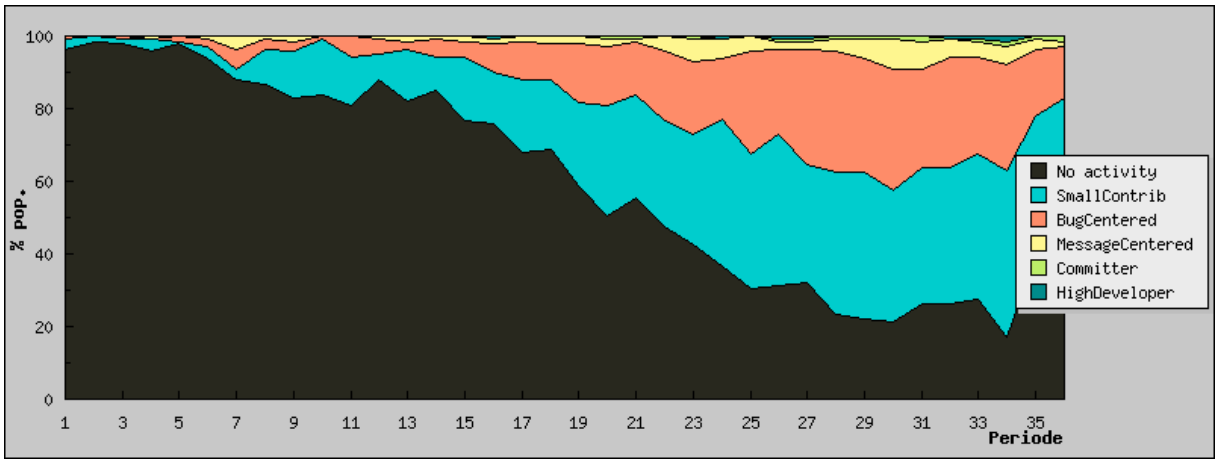

Figure 8. New BugSmall (4.7%)

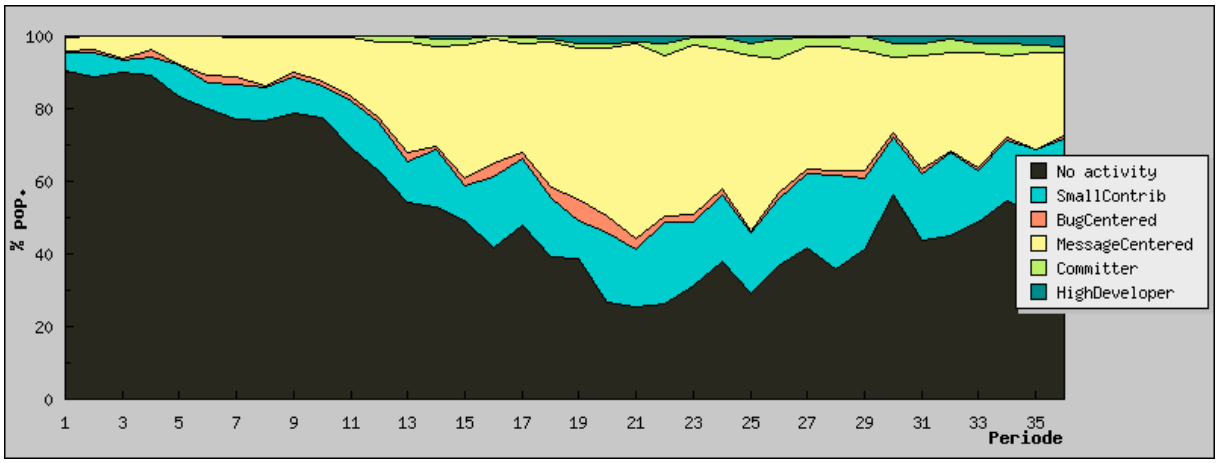

Figure 9. New SmallMessage (7.0%)

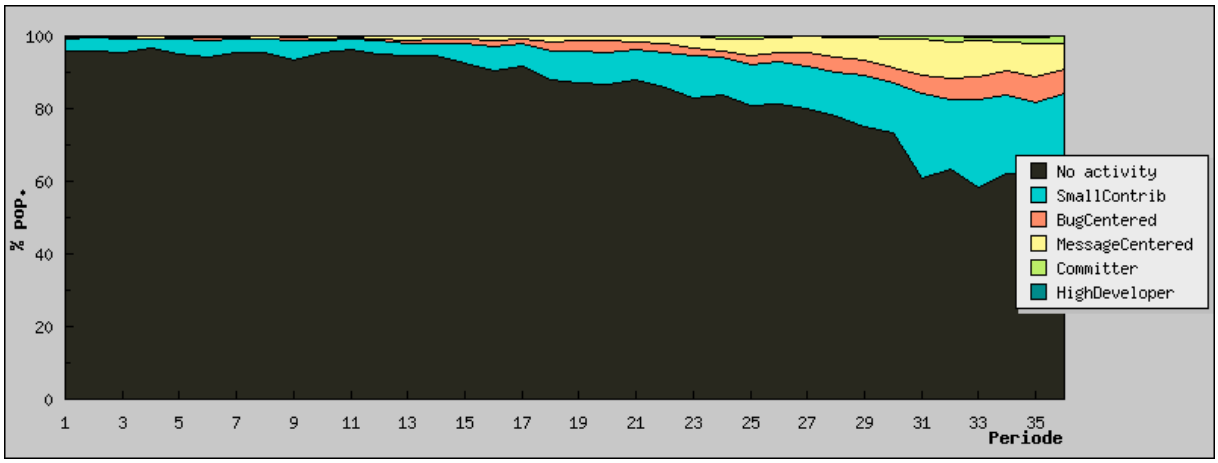

Figure 10. Sporadic (45.3%)

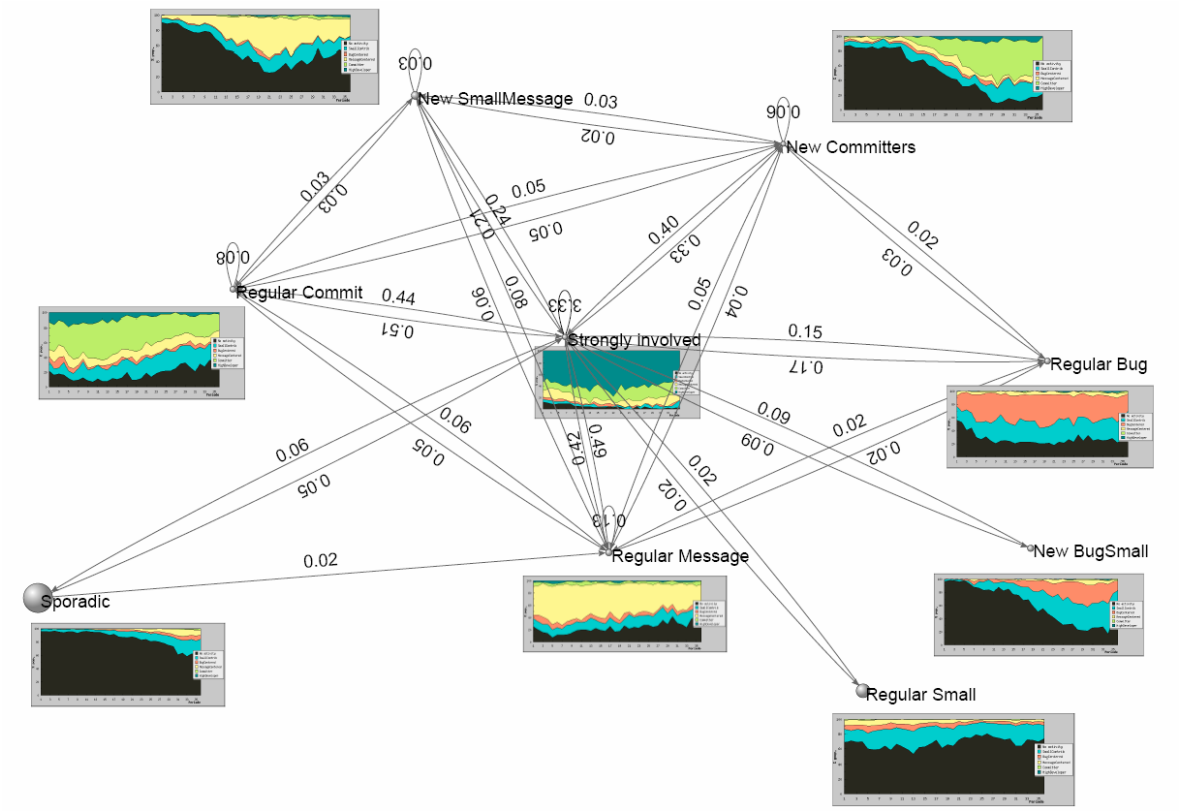

Figure 11. Collaboration between clusters of trajectories (average relationship)