Collaboration Networks in Software Development: Perspectives from Applying different Granularity Levels using Social Network Analysis - Research in progress

Miguel Angel Fernandez, Gregorio Robles and Jesus Gonzalez Barahona

GSyC/LibreSoft, Rey Juan Carlos University

(ma.fernandezsa@alumnos, grex@)urjc.es; jgb@bitergia.com

July 7, 2015

Motivation

- Large software projects may involve a lot of developers (Sometimes thousands of them!).
- Our interest: understand better how developers collaborate + evolution over time
- New business phenomenon: *Coopetition*
  - Sociological concept of *homophily*

Collaboration network graphs

How do we build this graphs?

Mining repositories + SNA = Collaboration networks.

Nodes = Developers

Two developers (nodes) are connected if they have collaborated together.

Edges = Collaborations

Edges width represents the amount of collaboration.

Network graph example

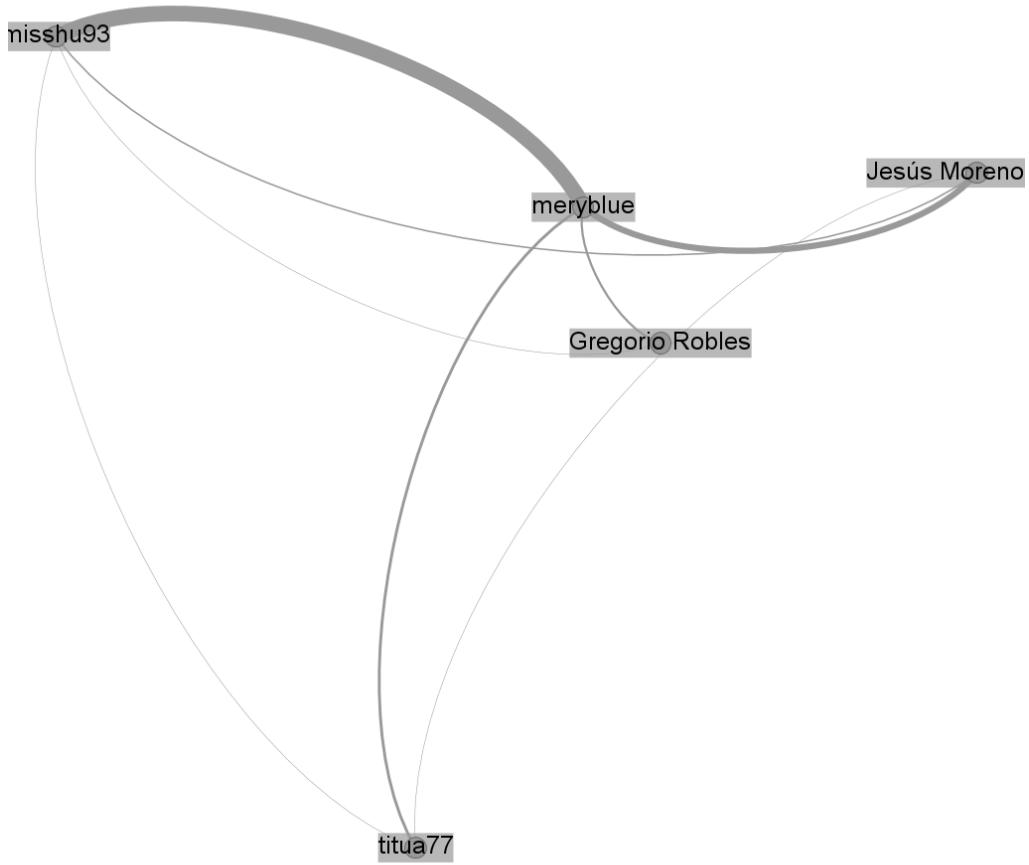


Figure : Collaboration network graph from DrScratch project (LibreSoft, Rey Juan Carlos University) - 1st semester, 2015

A different point of view

Up to now...

- In most SN studies, the resulting network is based on file/module data.
- When there are thousands of lines in a file, did collaboration really exist?

New-level analysis

- Resulting collaboration network depends heavily on the granularity level that is considered.
- We've been working to obtain collaboration graphs at function/method level.
- Better analysis (Excluding large functions/methods).

Case of study

Studied project

Gedit, GNOME Text-editor

Date range

- Goes form April 15, 1998 until April 15, 2015. (17 years!)
- Divided into six-month periods

Graphic results: 1st semester, 2001

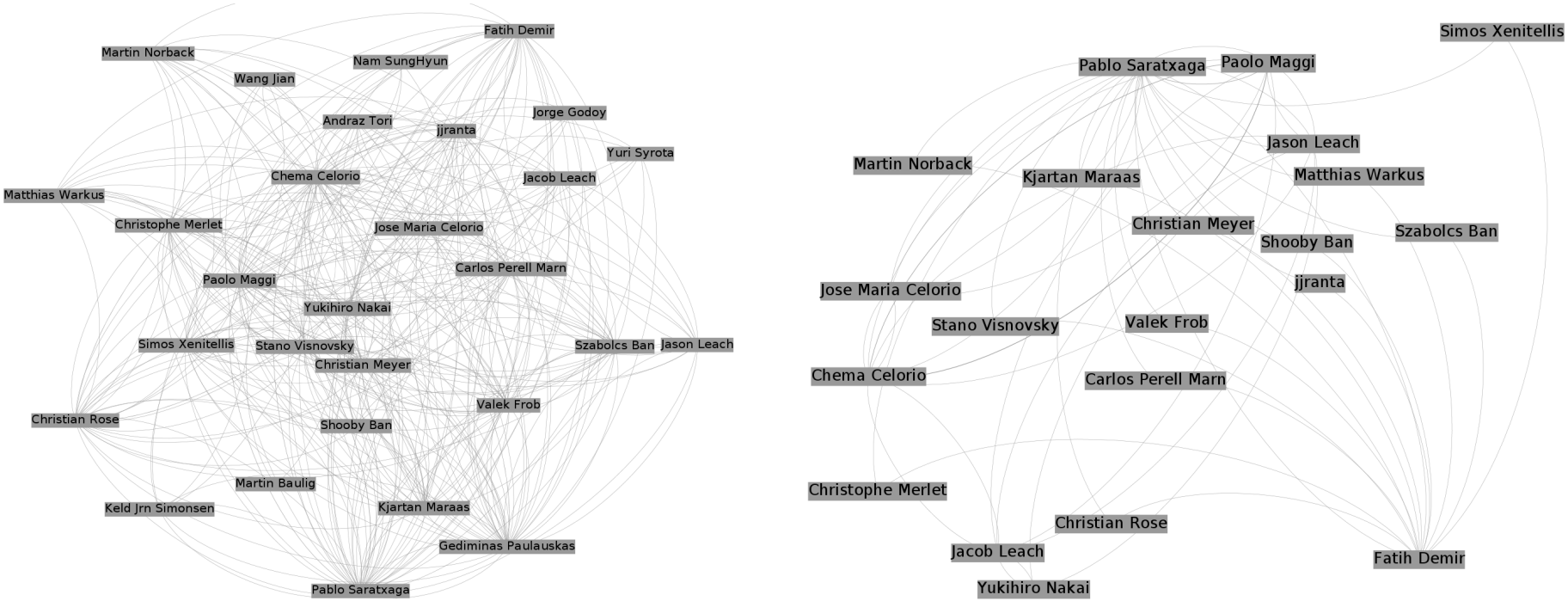


Figure : In-file (left) and In-method (right) collaboration network graphs

Graphic results: 1st semester, 2014

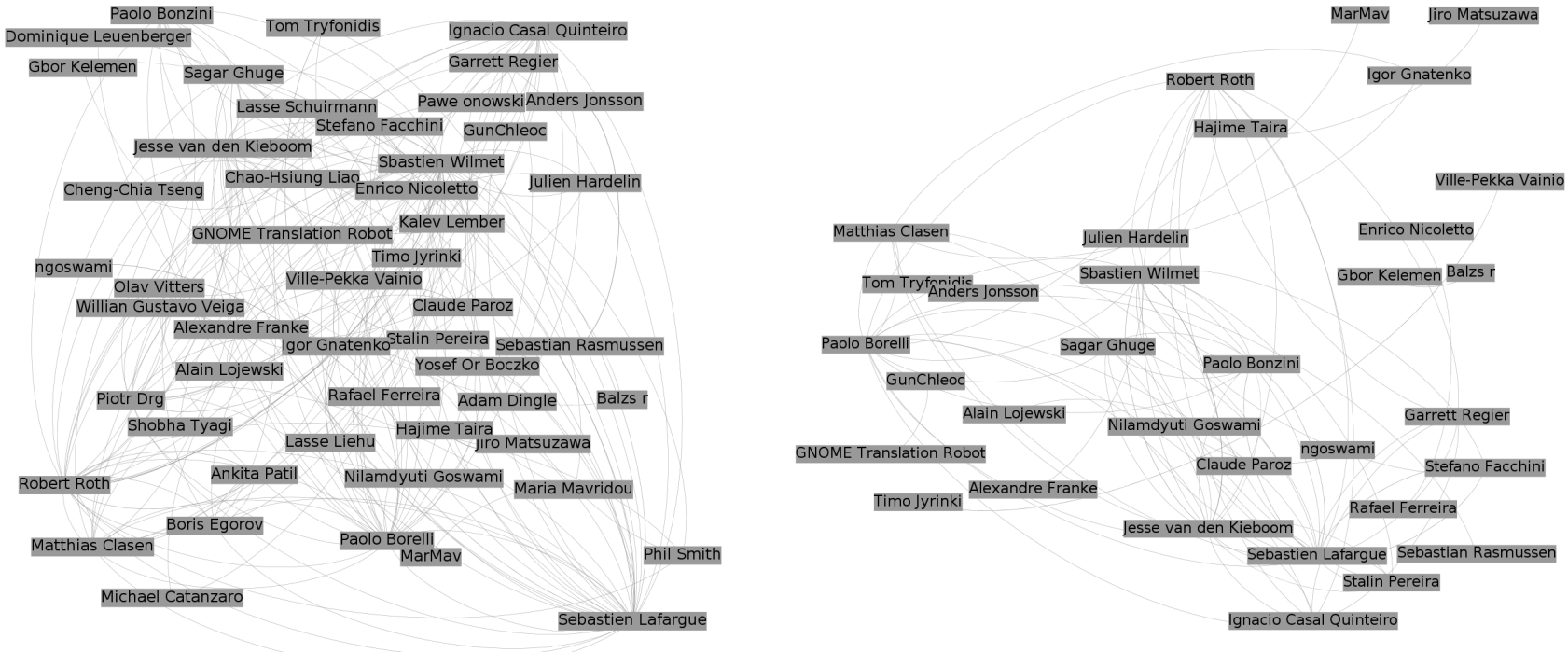
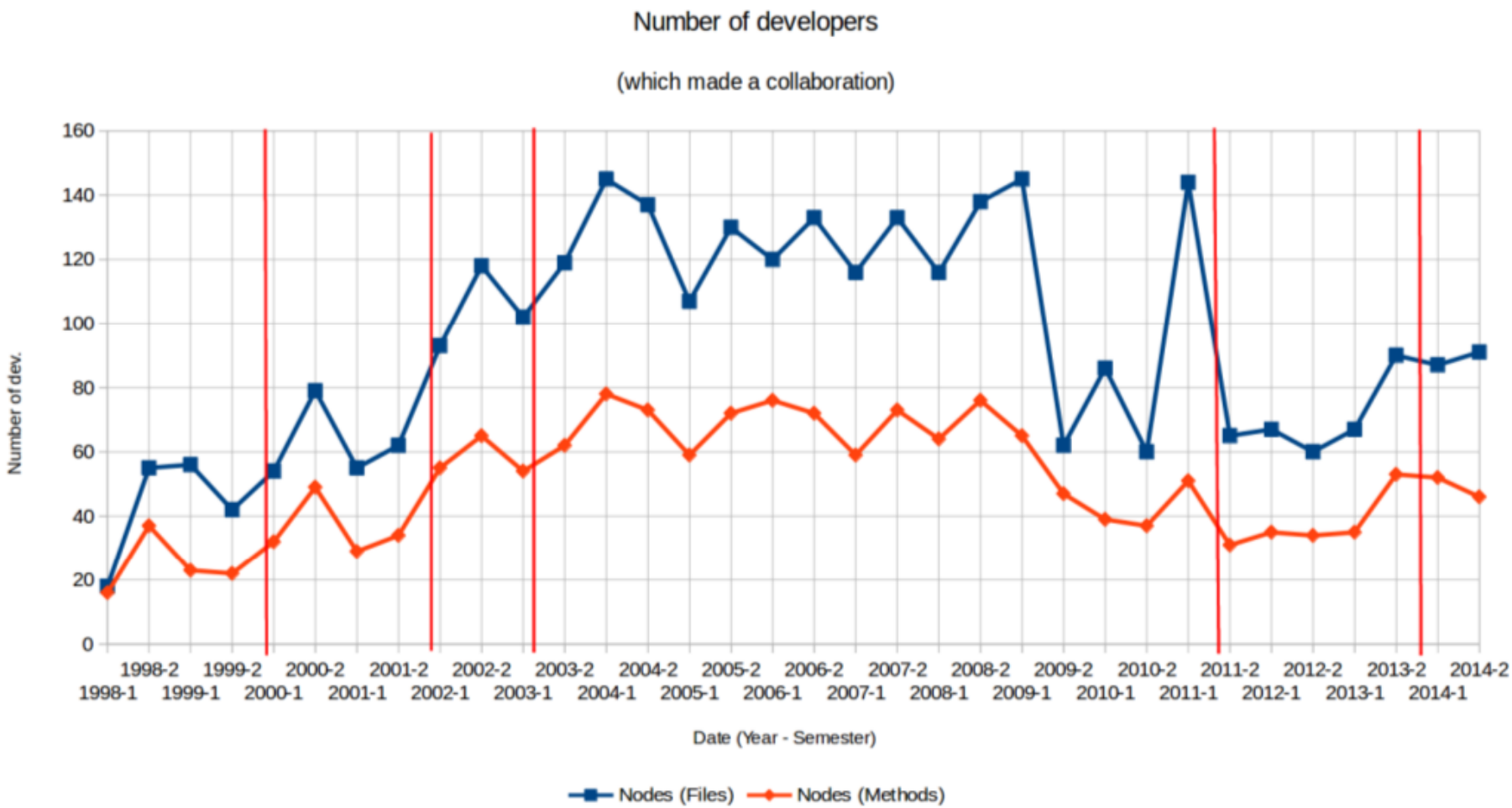
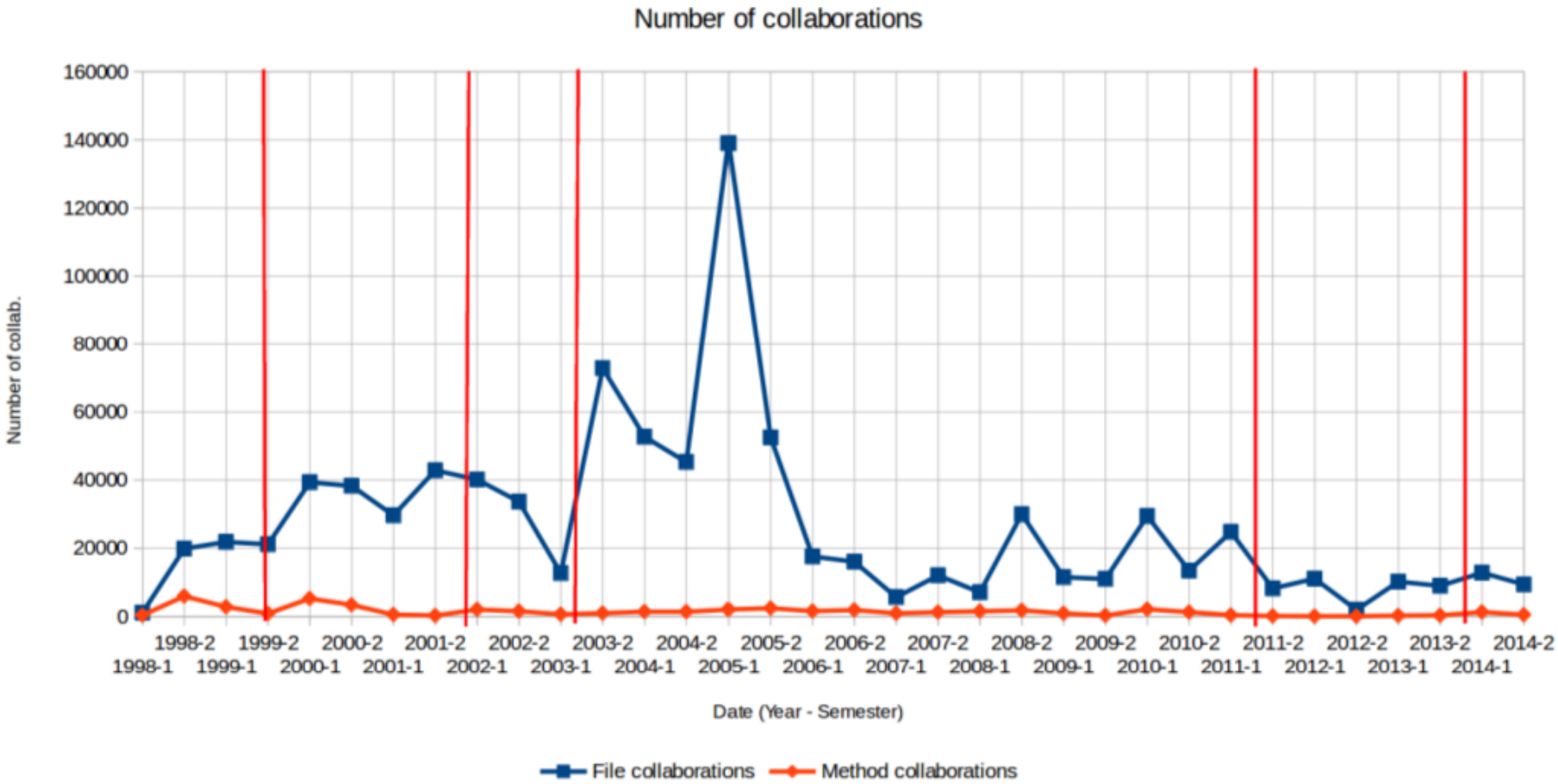


Figure : In-file (left) and In-method (right) collaboration network graphs

Numeric results: Number of developers



Numeric results: Number of collaborations



Future work

- Reproduce some of the studies done in the past now at method/function level
- Track function name changes and merge developer aliases
- Add developer affiliation information
- Improve graph visualization

Thank you for your attention!