Process Mining Software Repositories with FRASR

Bram Schoenmakers

April 27, 2011

Capita Selecta (2IS99)
Supervised by: A. Serebrenik

Introduction

- FRASR: Framework For Analysing Software Repositories
- Graduation project of Wouter Poncin (2010)
- Contribution to CSMR 2011
- Capita Selecta: extend and improve FRASR
Purpose

- Try to answer software engineering related questions such as:
  - Who are the architects in a project and who is doing the real work (implementation)?
  - Is there a relation between the day of the week and the number of bugs introduced?
  - Has the V-model been applied during a software project?
- Apply process mining techniques to data in software repositories.

FRASR input

- Version control systems
- Bug trackers
- Email messages
- Forum posts
- Trac Wiki pages
Schematic output of FRASR

Steve B.
- Check-in revision: 888
- Email: [RFC] Bugfix in iPushButton.cpp

Steve J.
- Check-in revision: 890
- Email: Re: [RFC] Bugfix in iPushButton.cpp

Linus T.
- Check-in revision: 891

ProM output
State of FRASR

- Support for:
  - Version control systems: Subversion, CVS
  - Bug trackers: Bugzilla, JIRA, Trac, SourceForge
  - Mailing lists: SourceForge, MARC, Pipermail, Tigris.org
  - Misc: Trac Wiki pages

- A few bugs
- SourceForge data sources broken
- Suboptimal user interface

Extensions

- Git integration (implemented)
- Regexp filters (implemented)
- Content awareness
- Different developer matching technique(s)
Git integration

- Used by major open source software projects:
  - Qt
  - GNOME
  - Android
  - Rails
  - Eclipse
- Git tracks more information than SVN
- Implemented with JGit library
- Challenge: multiple branches

Regexp filters

- Apply regular expressions to data source fields
- Example commit message from the KDE project:
  
  Load and save settings when printing. This will set sane initial values, fixing bug 116893 for example.

  BUG:116893
  BUG:111301
  BUG:126592

- Filter: \b(BUG|FEATURE) :\d{5,6}
Content awareness

- Goal: make FRASR aware of programming languages
- Enables FRASR to:
  - extract metrics from code
  - ignore insignificant commits
  - more accurate linking between emails and software artifacts
  - ...

Developer matching (1)
Developer matching (2)

- Match sets of aliases to real persons
- Using identifiers, names and email addresses
- Algorithms:
  - Refer to GPG servers, contributor lists, etc.
  - Permute aliases to find a match
  - Simple matching

Questions

Questions or suggestions?