Mining Evolutionary Data

• Version control data (e.g. commits history) enriches our vision of **software evolution**
  – Metadata about the software change: comments, user-ids, timestamps
  – Differences between the versions: addition, deletion or modification
  – Analysis of different software versions (snaphosts)

• Add the **time dimension** to codebase understanding
Mining Software Repositories

- Describe a broad class of investigations into the examination of software repositories data
- Empirical and systematic investigations
- Identify uncovered information, relationships or trends

- Data can be automatically **gathered** using collecting tools
  - VCS APIs (JGit, repodriller, pydriller…)
  - IDE APIs (plugins)
  - Supporting Tools (issue trackers, coverage reports, CI tools…)
  - Knowledge platforms and developers communities: (Stack Overflow, GitHub)

- Collected data can be **analysed** to reason about software evolution
  - Change history
  - Static code analysis
  - Software process insights

- Ease code comprehension by data **visualisation**
Example of analysis questions

- Who is the best developer to assign this task?
- Which parts of the system are having more issues?
- Which commit introduced a bug?
- What kind of changes do developer revert?
- Which part of the system are usually changed together?

<table>
<thead>
<tr>
<th>Evolutionary task category</th>
<th>Approaches</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evolutionary couplings/patterns</td>
<td>Bieman et al. [79], Canfora and Cerulo [56,57], Fischer et al. [23,27], Gall et al. [20,26,76], Hassan and Holt [69], Kagdi et al. [36], Shirabad et al. [37–39], Williams and Hollingsworth [48], Zimmermann et al. [15,33], Ying et al. [34]</td>
</tr>
<tr>
<td>Change classification/representation</td>
<td>Antoniol et al. [62], German [80], Hindle and German [29], Holt and Pak [75], Kim et al. [30], Mockus and Votta [77], Nikora and Munson [73]</td>
</tr>
<tr>
<td>Change comprehension</td>
<td>Beyer and Noack [81], Burch et al. [82], Chen et al. [22], Chen et al. [83], Cubranic et al. [60,61], Gall et al. [76], Görg and Weiβgerber [78], Hindle and German [29], Holt and Pak [75], Kim et al. [84], Purushothaman and Perry [67,68], Claudio [85], Robles et al. [44], Van Rysselberghe and Demeyer [53], Venolia [86]</td>
</tr>
<tr>
<td>Defect classification and analysis</td>
<td>Anvik et al. [63], German [21], Livshits and Zimmermann [35], Menzies et al. [52], Nagappan et al. [87], Ostrand and Weyuker [43], Sandusky et al. [42], Sliwerski et al. [28], Williams and Hollingsworth [45,46]</td>
</tr>
<tr>
<td>Source code differencing</td>
<td>Maletic and Collard [31], Neamtiu et al. [71], Raghavan et al. [70], Sager et al. [88]</td>
</tr>
<tr>
<td>Origin analysis and refactoring</td>
<td>Dig et al. [89,90], Godfrey et al. [72,91], Görg and Weiβgerber [49,78], Henkel and Diwan [92], Kimand Notkin [55], Ratzinger et al. [54], Tu and Godfrey [74], Weiβgerber and Diehl [93], Zou and Godfrey [24]</td>
</tr>
<tr>
<td>Software reuse</td>
<td>Selby [47], Van Rysselberghe and Demeyer [32], Xie and Pei [94]</td>
</tr>
<tr>
<td>Development process and communication</td>
<td>Dinh-Trong and Bieman [40], El-Ramly and Stroulia [95], Hayes et al. [59], Huang and Liu [64], Mockus et al. [96], Ohba and Gondow [58], Ohira et al. [65,66], Ying et al. [50]</td>
</tr>
<tr>
<td>Contribution analysis</td>
<td>Koch and Schneider [97], Mockus et al. [96], Robles et al. [41,98]</td>
</tr>
<tr>
<td>Evolution metrics</td>
<td>Capiluppi et al. [51], Godfrey et al. [72,91], Menzies et al. [52], Nagappan et al. [87], Nikora and Munson [73], Tu and Godfrey [74]</td>
</tr>
</tbody>
</table>
Use case: Reduce code complexity

- Software maintenance is both difficult and expensive: we need to **reduce complexity**

- **Metrics** can help in **identifying complex code**, that is hard to understand and tricky to modify (last snapshot)

- "If no one needs to read or modify a particular part of the code, does it really make a difference whether it’s complex?" [Adam Tornhill, 2015]
YES! IT’S A TIMEBOMB READY TO EXPLODE, BUT...

• We can prioritize software-design improvements involving system parts that developers will most likely work with again in the future

• We need to understand how the developers work on the code

• We must look at how the system evolved (not just the last snapshot)
  – Treating version-control data as our evidence on developers behaviour
Code as a Crime Scene

- **Geographical profiling** has its scientific basis in statistics and environmental psychology.

- The locations where crimes occur are very rarely random:
  - Typically, the location of the next crime is in the opposite direction from the first scene.

- Geographic profiling can help track down Jack the Ripper.
Whitechapel Murders

Aaron Kosminski (1865-1919)

James Maybrick (1838-1889)
Geographical Profiling with Dragnet

Uses the locations of a series of crimes to *prioritise the surrounding* areas in relation to the likely location of the offender's base, known as *hotspot* (red area)

Investigators can *focus their efforts* on the smaller area instead of patrolling the entire city!

Explore the geography of code

• What if we could devise techniques that let us identify **hotspots** in **large software systems**?

• Geographical profiling would give us a **prioritized lists** of sections that need refactoring

• **Code Hot Spot**: complex code that is frequently changed
Hotspot analysis

Complexity

Efforts

[http://www.inf.usi.ch/phd/wettel/codecity.html]
Distributed components

Change coupling
Change coupling (co-change)
CCM Heat Map
MSR 2019 Mining Challenge

- Mining SOTorrent, a dataset providing the version history of Stack Overflow posts:
  - textual discussion
  - code blocks
  - linked references to GitHub files
- The overall goal is to study the origin, evolution, and usage of Stack Overflow code snippets

[https://arxiv.org/pdf/1809.02814.pdf]
Here is my go at it (no loops and handles both SI units and binary units):

```java
public static String humanReadableByteCount(long bytes, boolean si) {
    int unit = si ? 1000 : 1024;
    if (bytes < unit) return bytes + " B";
    int exp = (int) (Math.log(bytes) / Math.log(unit));
    return String.format("%.1f %s", bytes / Math.pow(unit, exp), pre);
}
```

Example output:

<table>
<thead>
<tr>
<th>SI</th>
<th>BINARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 B</td>
<td>0 B</td>
</tr>
<tr>
<td>27 B</td>
<td>27 B</td>
</tr>
<tr>
<td>999 B</td>
<td>999 B</td>
</tr>
<tr>
<td>1000 MB</td>
<td>1000 B</td>
</tr>
<tr>
<td>1.0 kB</td>
<td>1.0 kB</td>
</tr>
<tr>
<td>1.0 kB</td>
<td>1.0 kB</td>
</tr>
<tr>
<td>1.7 kB</td>
<td>1.7 kB</td>
</tr>
<tr>
<td>110.6 MB</td>
<td>110.6 B</td>
</tr>
<tr>
<td>7.1 MB</td>
<td>6.8 MB</td>
</tr>
<tr>
<td>453.0 MB</td>
<td>453.0 B</td>
</tr>
<tr>
<td>29.0 GB</td>
<td>27.0 GB</td>
</tr>
<tr>
<td>1.7 TB</td>
<td>1.7 TB</td>
</tr>
<tr>
<td>8.0 EIB</td>
<td>8.0 EIB</td>
</tr>
<tr>
<td>(Long.MAX_VALUE)</td>
<td></td>
</tr>
</tbody>
</table>

Related article: Java: Formatting byte size to human readable format

FileUtils.byteCountToDisplaySize(long size) would work if your project can depend on org.apache.commons.io.

JavaDoc for this method

share improve this answer  edited Oct 29 '16 at 3:58  answered Oct 21 '16 at 7:22

bacode  277k ● 62 ● 614 ● 804

share improve this answer  edited Jun 7 '16 at 13:27  answered Feb 3 '11 at 16:57

Zarathustra  1,439 ● 1 ● 14 ● 32

user601806  2,259 ● 1 ● 8 ● 2
Example of research questions

• How many **clones of code snippets** exist inside Stack Overflow?

• How frequently are code snippets **copied** from Stack Overflow to GitHub?

• How can we detect **buggy** versions of Stack Overflow code snippets and find them in GitHub projects?

• Can we reliably **predict popularity** of Stack Overflow code snippets on GitHub?
usage: perceval [-c <file>] [-g] <backend> [<args>] | --help | --version

Repositories are reached using specific backends. The most common backends are:

- askbot: Fetch questions and answers from Askbot site
- bugzilla: Fetch bugs from a Bugzilla server
- bugzillarest: Fetch bugs from a Bugzilla server (>=5.0) using its REST API
- confluence: Fetch contents from a Confluence server
- discourse: Fetch posts from Discourse site
- dockerhub: Fetch repository data from Docker Hub site
- gerrit: Fetch reviews from a Gerrit server
- git: Fetch commits from Git
- github: Fetch issues, pull requests and repository information from GitHub
- gitlab: Fetch issues, merge requests from GitLab
- googlehits: Fetch hits from Google API
- groupsio: Fetch messages from Groups.io
- hyperkitty: Fetch messages from a HyperKitty archiver
- jenkins: Fetch builds from a Jenkins server
- jira: Fetch issues from JIRA issue tracker
- launchpad: Fetch issues from Launchpad issue tracker
- mattermost: Fetch posts from a Mattermost server
- mbox: Fetch messages from MBox files
- mediawiki: Fetch pages and revisions from a MediaWiki site
- meetup: Fetch events from a Meetup group
- nntp: Fetch articles from a NNTP news group
- phabricator: Fetch tasks from a Phabricator site
- pipermail: Fetch messages from a Pipermail archiver
- redmine: Fetch issues from a Redmine server
- rss: Fetch entries from a RSS feed server
- slack: Fetch messages from a Slack channel
- stackexchange: Fetch questions from StackExchange sites
- supybot: Fetch messages from Supybot log files
- telegram: Fetch messages from the Telegram server
- twitter: Fetch tweets from the Twitter Search API
MSR 2020

Mining the Software Heritage Graph Dataset

Software Heritage
preserves software source code for present and future generations

https://www.softwareheritage.org/
RepoDriller & PyDriller

• **RepoDriller**: a Java framework for mining software repositories
  • Extract information from Git repositories
    – commits, branches, tags
    – developers info
    – modifications and diffs
    – source codes
  • Quickly export CSV files
  • Integration with Static Analysis tools and Code Parsers (Eclipse JDT, Java Parser)
• **PyDriller**: Python version of RepoDriller

[https://github.com/mauricioaniche/repodriller]
[https://github.com/ishepard/pydriller]
import org.repodriller.*;

public class S1 implements Study {
    public static void main(String[] args) {
        new RepoDriller().start(new S1());
    }
    public void execute() {
        String gitUrl = "https://github.com/mauricioaniche/repodriller.git";
        String repoDir = "exp/repodriller";
        String cvsFile = "exp/devs.cvs";

        GitRemoteRepository
            .hostedOn(gitUrl)
            .inTempDir(repoDir)
            .buildAsSCMRepository();

        new RepositoryMining()
            .in(GitRepository.singleProject(repoDir))
            .through(Commits.all())
            .process(new DevelopersVisitor(), new CSVFile(cvsFile))
            .filters(
                new OnlyModificationsWithFileTypes(Arrays.asList(".java", ".xml")),
                new OnlyInBranches(Arrays.asList("master")),
                new OnlyNoMerge()
            ).mine();
    }
}
import org.repodriller.*

public class DevelopersVisitor implements CommitVisitor {

    public void process(SCMRepository repo, Commit commit, PersistenceMechanism writer) {
        writer.write(
            commit.getHash(),
            commit.getCommitter().getName());
    }
}
import os
from git import Repo
from pydriller import RepositoryMining

git_url = 'https://github.com/ishepard/pydriller.git'
repo_dir = 'clones/pydriller'

if not os.path.exists(repo_dir):
    Repo.clone_from(git_url, repo_dir)

for commit in RepositoryMining(repo_dir).traverse_commits():
    print('Hash {}, author {}'.format(commit.hash, commit.author.name))
PyDriller Example on Colab

• Jupyter Notebook are interactive computational environment that combines:
  – code execution
  – rich text
  – mathematics
  – plots
  – rich media

• Supports several languages (e.g. Python)

• Google Colab Notebook: cloud hosting for Jupyter Notebooks

• Save and share notebooks as simple google docs

• https://tinyurl.com/LISD-dmi-pydriller
Colab on Drive

Semplice esempio sull'uso di PyDriller

Installazione delle dipendenze con pip

```python
# Install a pip package in the current Jupyter kernel
import sys
!{sys.executable} -m pip install pydriller
```

Collecting pydriller
  Downloading https://files.pythonhosted.org/packages/b7/6d/44aaf5bf8f200de492238605b96cafdcc93b595f00ae0bcb22f6f9697d99/PyDriller-1.9.2.tar.gz
Collecting gitpython
  Downloading https://files.pythonhosted.org/packages/20/8c/4513981439d23c4ff65b2a62dd7f7ebe9a8a654a9be67a840d1e2730d3/GitPython-3.0.5.tar.gz
Collecting lizard
  Downloading https://files.pythonhosted.org/packages/24/7a/7ca04e3d84998d7344caaf3d534a71ba8a80d293bf32afbd2ef3956282d1c2/lizard-1.16.6.tar.gz
Collecting gitdb2>=2.0.0
  Downloading https://files.pythonhosted.org/packages/03/6c/9926f89bad2ef85626e1df9f877acbeee8885bb0434d82ad1ed4746d2325/gitdb2-2.0.6-py3.7.egg
Collecting mmap2>=2.0.0
  Downloading https://files.pythonhosted.org/packages/55/d2/866d45e3a12lee15a1dc013824d50072fd5c7799c9c34d01378eb262ca8f/mmap2-2.0.5-py3.7.egg
Installing collected packages: pydriller, gitdb2, gitpython, lizard, pydriller
Successfully installed gitdb2-2.0.6 gitpython-3.0.5 lizard-1.16.6 pydriller-1.9.2 mmap2-2.0.5

SHIFT+ENTER to run a cell
Colab on Drive

- **import dei moduli necessari**

```python
import os
from git import Repo
from pydriller import RepositoryMining
```

- **Scelta repository remoto e directory locale per il clone**

```python
git_url = 'https://github.com/ishepard/pydriller.git'
repo_dir = 'clones/pydriller'
```

- **Se non è già presente, effettua il clone del repository remoto**

```python
if not os.path.exists(repo_dir):
    Repo.clone_from(git_url, repo_dir)
```

- **Esempio di esecuzione di un comando di shell (inizia con '!')**

```bash
!ls clones/pydriller
```

```bash
C:\\> exemples pydriller README.md setup.py test-repos.zip tests
```

Attraversa tutti i commit e ne stampa hash e autore

```python
for commit in RepositoryMining(repo_dir).traverse_commits():
    print('Hash {}, author {}, format(commit.hash, commit.author.name))
```

<table>
<thead>
<tr>
<th>Hash</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>c363b3b63ec34554a367041d077fe9b93c6ac</td>
<td>ishepard</td>
</tr>
<tr>
<td>20b0f7386f862841421b33e870711758011c08bc7b</td>
<td>ishepard</td>
</tr>
<tr>
<td>3e0e545c693424cd5e8fe28de0778ec53ae9853a</td>
<td>ishepard</td>
</tr>
<tr>
<td>3c39658418921e2fd20f9e560dc082b0569afccf</td>
<td>Spadini Davide</td>
</tr>
<tr>
<td>2a76d0e41440ce72db8be76c2af2c1afa3f3d57c0a27b3</td>
<td>Spadini Davide</td>
</tr>
<tr>
<td>a38a88c596b2b0b9d66cdca20ed34f675a4d55</td>
<td>Spadini Davide</td>
</tr>
<tr>
<td>9321e0c2e80b3696f09ef79fccc2699b97384e1e0</td>
<td>ishepard</td>
</tr>
<tr>
<td>3e757de698fd59c693fa5d24e0fbae5e7f97824d</td>
<td>ishepard</td>
</tr>
<tr>
<td>af5b67038bd63d0f90d17d070440fd01997ac94</td>
<td>ishepard</td>
</tr>
<tr>
<td>3bbea6194eb9010d71d0110b193458a8c35493</td>
<td>ishepard</td>
</tr>
<tr>
<td>acc904ce560745f86991e960e06ef4c595c2fed3</td>
<td>ishepard</td>
</tr>
<tr>
<td>a09cd1749c905c0975823261d9dcd8b5f9db9e50</td>
<td>ishepard</td>
</tr>
<tr>
<td>6c2a423dea56276218723eb73f9b684be967140</td>
<td>Spadini Davide</td>
</tr>
<tr>
<td>4d379b1392829f2849e68316fe18437e1f2428aa</td>
<td>ishepard</td>
</tr>
</tbody>
</table>
References

- A. Tornhill: Your Code As a Crime Scene