CAL POLY
College of Engineering

Mining Developers' Coding Style from Software Repositories

Janani Sridhar & Kellie Banzon
Dr. Bruno C. da Silva, Computer Science & Software Engineering Dept.

2018 Summer Undergraduate Research Program Symposium

Motivation

Software development has become an increasingly collaborative process.

Employees from one team may collaborate with each other, another team, or independent developers on the other side of the globe.

As more and more files are written by more and more coders, the importance of program comprehension and code readability have also grown.

Many organizations have developed internal coding style guides for their employees to facilitate consistency and readability.

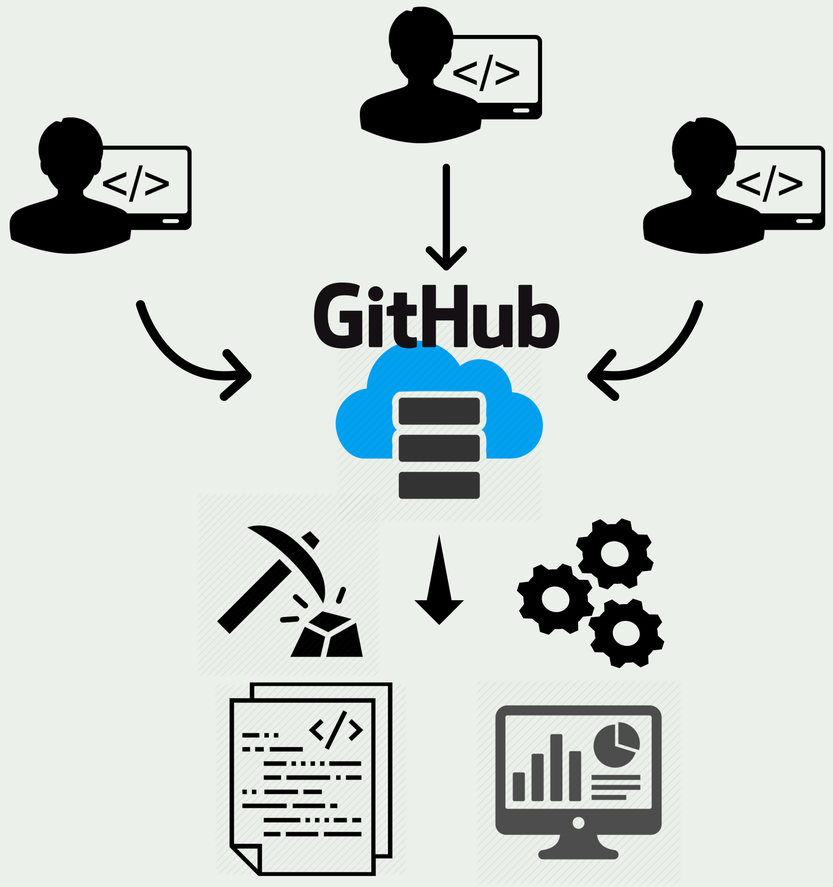
Moreover, little is known about how developers around the world actually use different coding styles.

Goals

Mine a large dataset of thousands of software repositories from GitHub to provide a broad view of developers' coding style choices in various projects and programming languages.

Furthermore, we aim to analyze whether developers' coding styles comply with well-known coding style sources such as Google's and Oracle's style guidelines .

By the end of the year we aim to be the first to mine developers' coding style over an expansive dataset of public projects and showcase that data to enable future analysis work.



Our repo: github.com/bcdasilv/code-style-mining

Coding Style Elements Analyzed

```
class CamelCase{…}   vs.  single_trailing_underscore()
```

```
processRequest(){
⟶indentTwoSpaces
} //Java
```

vs.

```
def process_request()
⟶⟶indentFourSpaces
//Python
```

[Line length: # of characters]

Tab vs. Space

```
import org.yourpackage.*;
      vs.
import your.package.org.ClassA;
import your.package.org.ClassB;
```

…and more: Blank lines and Curly braces

Completed Work

Over Summer/18, we developed custom parsing and analysis tools for projects written in **Java** and **Python**.

Repositories are explored through the GitHub API, and the Java and Python files are processed by our analyzers.

The results are exported in JSON format and collected for study.

Our Java and Python analyzers cover:

- 6 style elements: Naming, Indentation, Tabs vs. Spaces, Line length, Blank lines, Imports, Curly braces
- 6 source code elements: classes, constants, variables, methods/functions, control statements, import statements

Technology Used

Pycodestyle API for Python files.

JavaParser lib for Java files.

GitHub API and public repositories.

JSON format for storing the results.

Ongoing Work

Our team is currently endeavoring to write the JSON file analysis outputs to a cloud database.

Then, we will scale to analyze thousands of source files and save the results.

Create interactive infographics and searchable project results to showcase the data