

Lucy Family Institute for **Data & Society**

Understanding Developer-AI Collaboration: A Behavioral and Cognitive Modeling Approach

Ningzhi Tang (ntang@nd.edu), Meng Chen, Zheng Ning, Collin McMillan, Toby Jia-Jun Li Department of Computer Science and Engineering, University of Notre Dame

BACKGROUND



GitHub Copilot [1] is a AI pair programmer that generate code snippets from code contexts.

Our project is aimed at understanding programmer behavior when they program in collaboration with GitHub Copilot, including IDE behavior (direct) and eye tracking behavior (indirect).

DATA COLLECTION

We have developed a plugin for IntelliJ IDEA. The plugin has 3 main functions (Fig. 1).

(1) Detecting the developer's IDE behaviors, including IDE actions (e.g., reformat code, save all, go to declaration) and code editing by programmers.

(2) Tracking file content changes, compiles, test case, and debugging behaviors.

(3) Collecting and analyzing eye tracking gazes. Our plugin can use an eye tracking device to detect where a person looks in terms of line, column, and token in a source file [2].

All data samples collected are aligned by timestamps.



<action id="ReformatCode"</pre> path="/src/Reptile.java" timestamp="1662928377436"/>

<typing column="4" length="13"</pre> line="8" path="/src/Reptile.java" string="private int E" timestamp="1662928450732"/>

<action id="copilot.applyInlays"</pre> path="/src/Reptile.java' timestamp="1662928415454"/>

<action column="23" line="42"</pre> path="/src/Reptile.java" id="GotoDeclaration" timestamp="1662928462655"/>

<action path="/src/Reptile.java"</pre> id="SaveAll" timestamp="1662928462698"/>

Figure 1. Example of programmer behavior data collected by our plugin.

</ast> <srcml token="getYear"> </srcml>

Figure 2. Example of extraction of normal AST and srcML hierarchy of token "getYear",

lucyinstitute.nd.edu





File Logging

<log id="fileLog" info="fileOpened"</pre> src_path="/src/ZooSystem.java" timestamp="1662928820593"/>

<log id="fileLog" info="fileChanged"</pre> src_path="/src/Lion.java" timestamp="1662928829436"/>

<log id="fileLog" info="fileClosed"</pre> src_path="/src/Lion.java" timestamp="1662928831515"/>

<log id="outputLog" timestamp="1662928831632"/>



Eye Tracking

<gaze column="4" validity="1.0"</pre> pupil diameter="4.2882" line="67" path="/src/Chameleon.java" timestamp="1662928567232" **x**="501" **y**="334" **token**="layEggs" ast_type="IDENTIFIER"/>

<gaze column="15" validity="1.0"</pre> pupil_diameter="4.3061" line="67" path="/src/Chameleon.java" timestamp="1662928567260" x="592" y="316" token="public" ast_type="PUBLIC_KEYWORD"/>

<gaze column="16" validity="1.0"</pre> pupil_diameter="4.3119" line="67" path="/src/Chameleon.java" timestamp="1662928567260" **x**="596" **y**="312" **token**=" ast_type="WHITE_SPACE"/>

```
<ast ast_type="IDENTIFIER" token="getYear">
 <level end="12:33" start="12:26" tag="PsiIdentifier:getYear"/>
 <level end="12:33" start="12:15"</pre>
        tag="PsiReferenceExpression:new Date().getYear"/>
 <level end="12:35" start="12:15"</pre>
        tag="PsiMethodCallExpression:new Date().getYear()"/>
<level end="12:61" start="12:15"</pre>
        tag="PsiBinaryExpression:new Date().getYear()
        - this.birthday.getYear()"/>
 <level end="12:62" start="12:8" tag="PsiReturnStatement"/>
 <level end="13:5" start="11:4" tag="PsiMethod:getAge"/>
<level end="14:1" start="2:0" tag="PsiClass:Student"/>
 <name pos:start="13:27" pos:end="13:33"/>
 <call pos:start="13:27" pos:end="13:35"/>
 <block pos:start="12:25" pos:end="14:5"/>
 <block content pos:start="13:9" pos:end="13:62"/>
 <return pos:start="13:9" pos:end="13:62"/>
 <expr pos:start="13:16" pos:end="13:61"/>
 <function pos:start="12:5" pos:end="14:5"/>
 <class pos:start="3:1" pos:end="15:1"/>
```

FUTURE WORK

1. Model programmer behaviors and cognitions.

2. Augment ML models for source code with programmer behaviors.

3. Improve performance in downstream tasks e.g., code generation, code summarization, bug detection, requirement analysis.

Fall Symposium Oct 11, 2022



Token: getYear()

BEHAVIOR ANALYSIS

import java.util.Date public class Student Time Spent: 1.023s private final String name TTFF: 4.136s private final Date birthday Respondence Ratio: 38/12 srcML: call, return public Student(String name, Date birthday Method: getAge this.name = name: Class: Student this.birthday = birthday;

oublic <mark>int getAge(</mark> return new Date()(getYear()) this.birthday.getYear()

Figure 3. Analyzing of gazes in token level. In the heatmap [3], the warmer colors represent higher percentage of fixation and the colder colors for lower percentage of fixation.

In an experiment, programmers will finish various kinds of programming tasks with GitHub Copilot: (1) algorithm design; (2) front-end and back-end development of a calculator; (3) an exercise of object-oriented programming. Eye tracking patterns based on AST/srcML hierarchy and IDE behavior patterns will be analyzed.

Reference

[1] GitHub Copilot: https://github.com/features/copilot. [2] D. T. Guarnera, C. A. Bryant, A. Mishra, J. I. Maletic, and B. Sharif, "itrace: Eye tracking infrastructure for development environments," in Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications, pp. 1–3, 2018. [3] B. Clark and B. Sharif, "itracevis: Visualizing eye

movement data within eclipse," in 2017 IEEE Working Conference on Software Visualization (VISSOFT), pp. 22–32, IEEE, 2017.

