Johan Martinson

Herman Jansson

Chalmers University of Technology

Mukelabai Mukelabai

Thorsten Berger

Ruhr University Bochum

Chalmers | University of Gothenburg

Alexandre Bergel

University of Chile

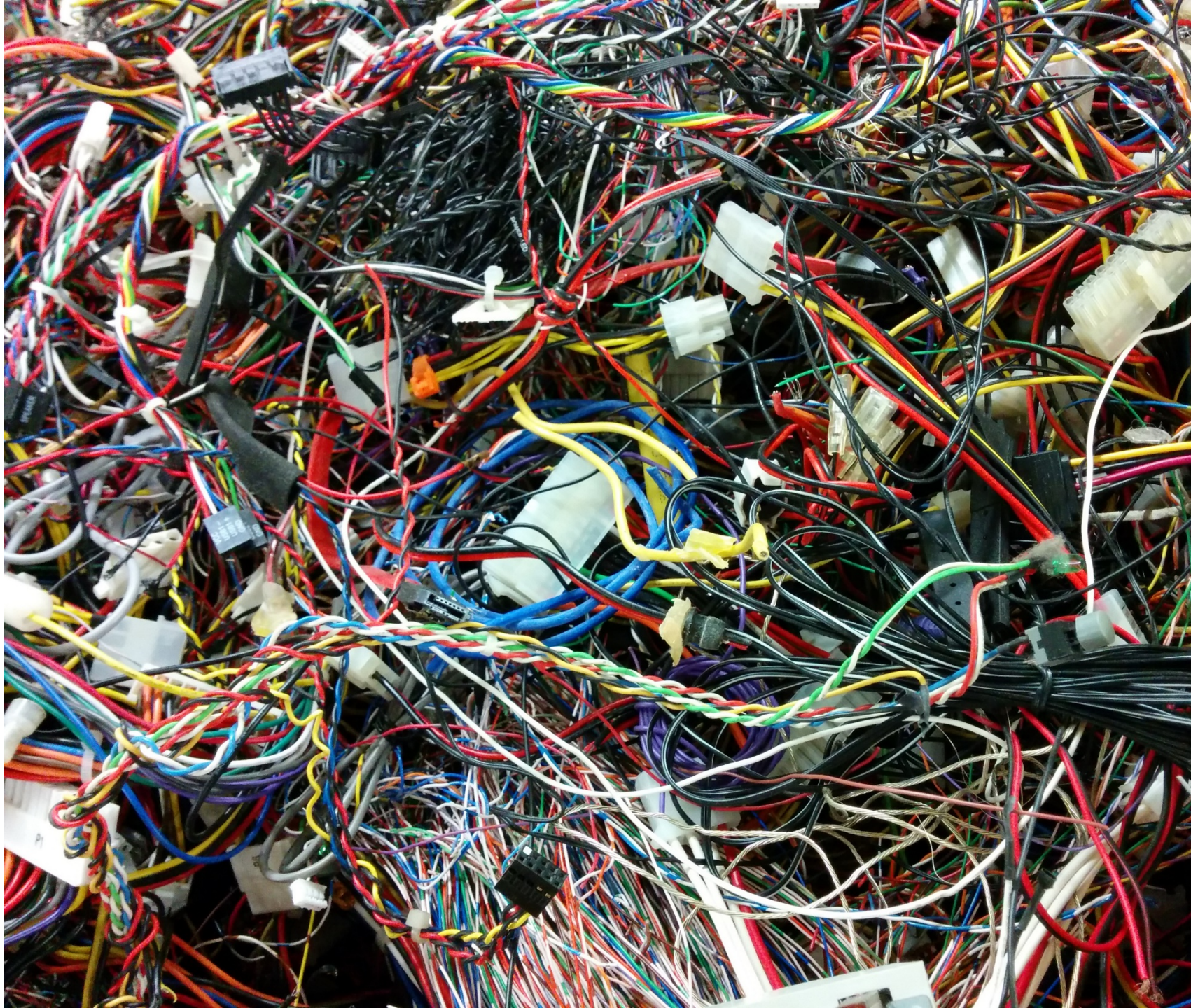
Truong Ho-Quang

Chalmers | University of Gothenburg

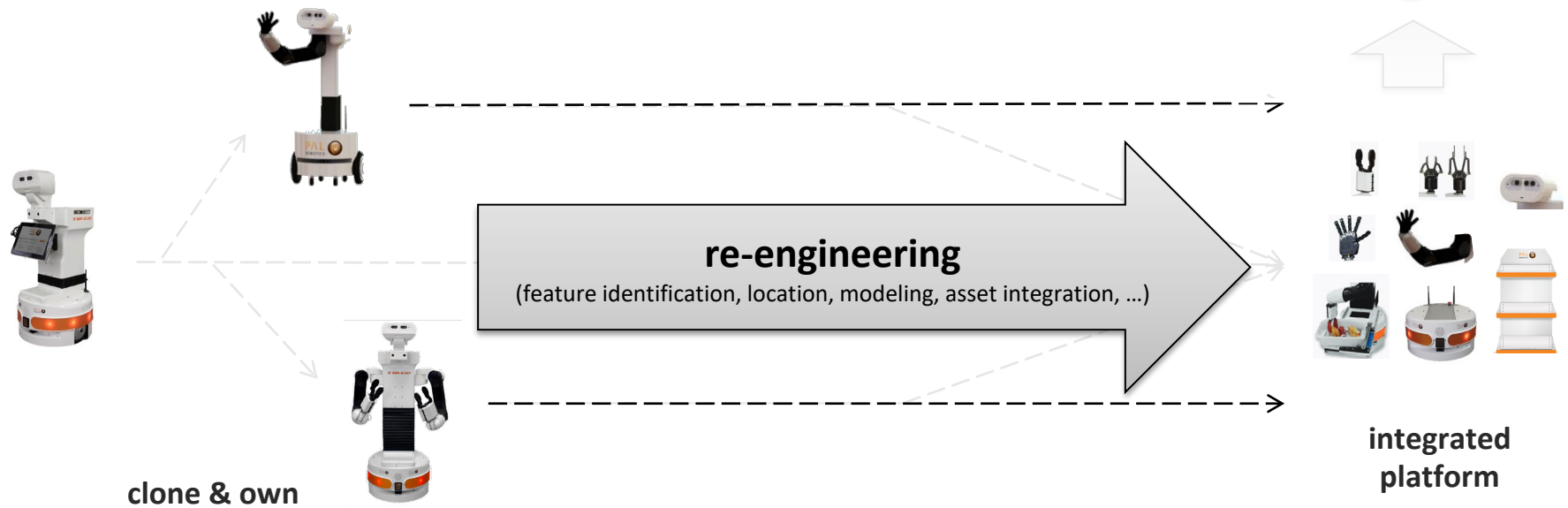
HAnS: IDE-Based Editing Support for Embedded Feature Annotations

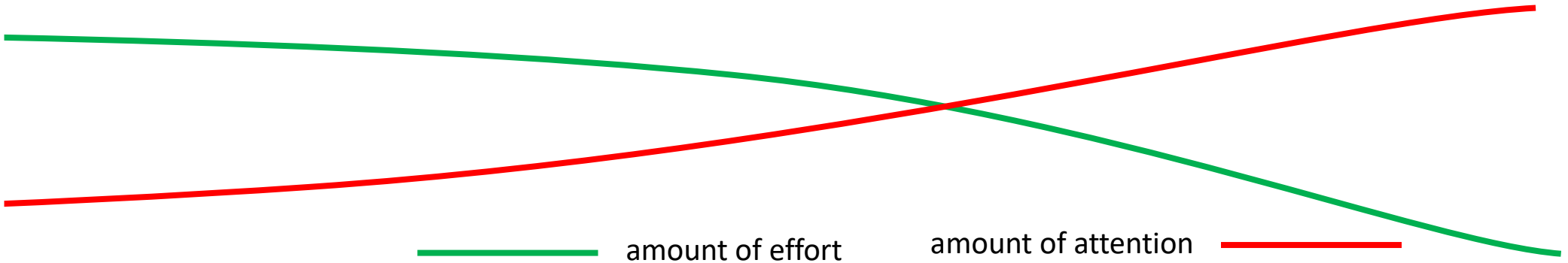
https://bitbucket.org/easelab/hans-text/

feature location
problem



variant-rich systems

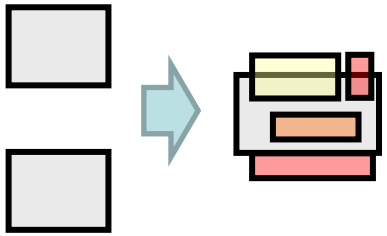




Berger et al. A survey of variability modeling in industrial practice. VaMoS, 2013.

traceability

traceability != variability



feature Init

mainline

```
void __init init_IRQ(void){
 ia64_register_ipi();
 platform_irq_init();
 reginster_percpu_irq(...);
 ...
}
```

traceability annotation

feature Init

fork

```
void __init init_IRQ(void){
 acpi_boot_init();
 platform_irq_init();
 reginster_percpu_irq(...);
 ...
 }
```

traceability annotation

variability annotation (#IFDEF)

```
void __init init_IRQ(void){
 #ifdef CONFIG_ACPI
  acpi_boot_init();
 #else
  ia64_register_ipi();
 #endif
 platform_irq_init();
 reginster_percpu_irq(...);
 ...
}
```

traceability annotation

traceability strategies

recover traceability when needed

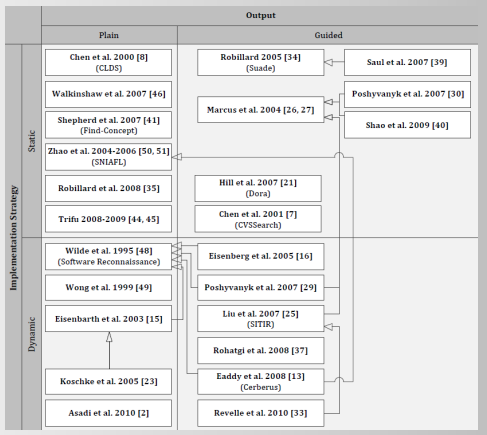
lazy strategy

record and edit traceability during development

eager strategy

automated traceability recovery?

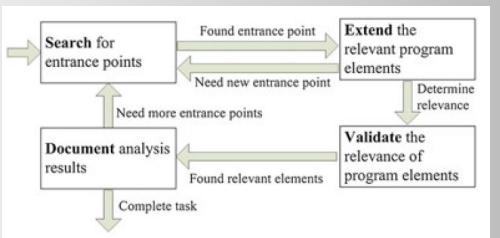
automated feature location



Rubin, Chechik. A survey of feature location techniques. Domain Engineering. 2013

tools have low precision and require high manual effort

manual feature location



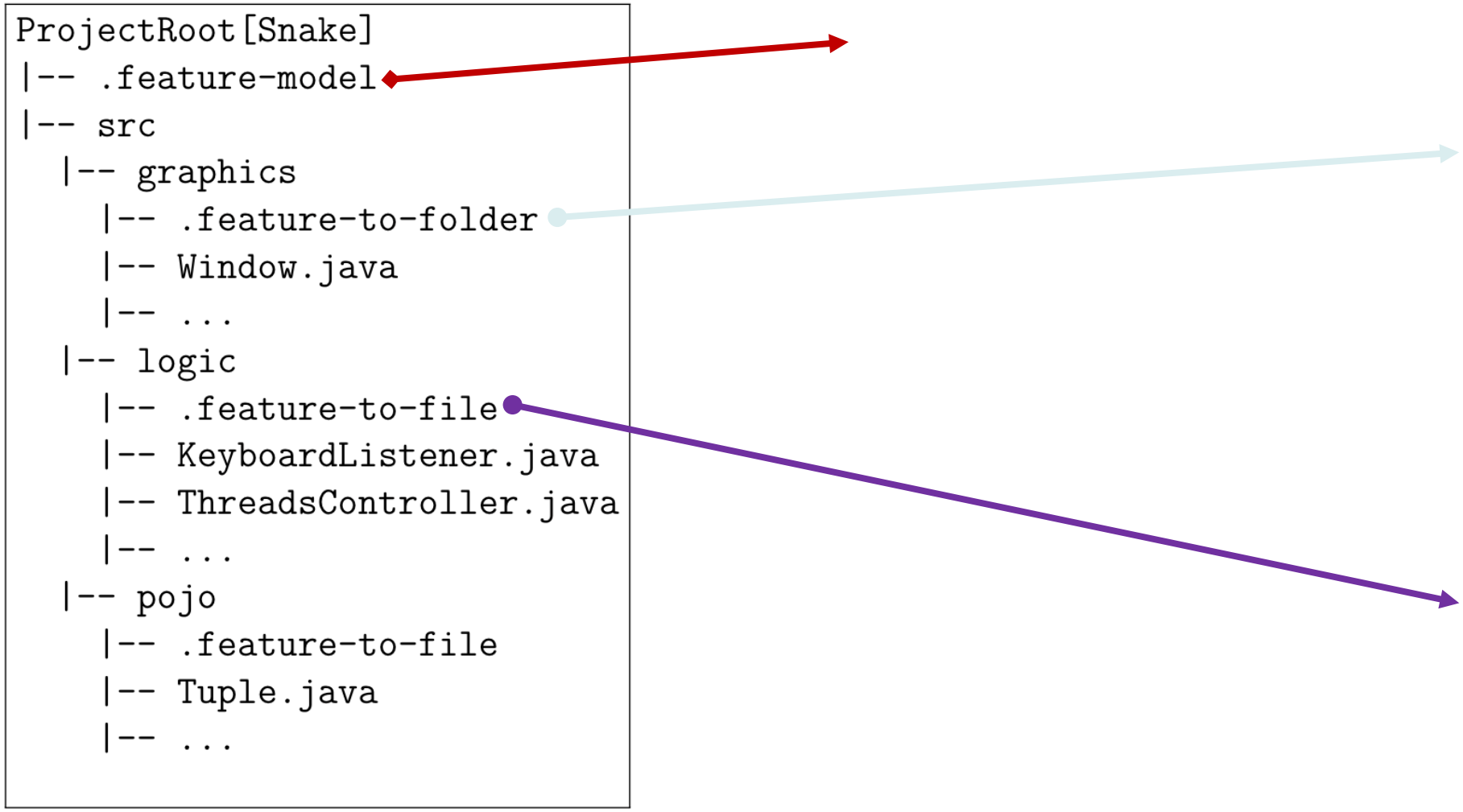
Wang, Peng, Xing, Zhao. How developers perform feature location tasks: a human-centric and process-oriented exploratory study. Journal of Software: Evolution and Process, 2013.

Krüger, Berger, Leich. Features and How to Find Them: A Survey of Manual Feature Location. Software Engineering for Variability Intensive Systems: Foundations and Applications. CRC Press. 2018

lazy strategy

systems with 73k, 2k, 43k, 19k LOC
average location time: 15min

embedded feature annotations



Schwarz, Mahmood, Berger. A Common Notation and Tool Support for Embedded Feature Annotations. SPLC. 2020
Ji, Berger, Antkiewicz, Czarnecki. Maintaining Feature Traceability with Embedded Annotations. SPLC. 2015.

feature references

```
// &begin[Food]
foodPosition = new Tuple( x: getWindowHeight() - 1,
         y: getWindowWidth() - 1);
spawnFood(foodPosition);//&line[Food::Spawn]
// &end[Food]
```

- Snake_Game
  - Playing_Area
    - Tile
      - Poison
        - Spawn
      - Food
        - Spawn

```
// &begin[Food::Spawn]
private void spawnFood(Tuple foodPositionIn) {
    Squares.get(foodPositionIn.x)
           .get(foodPositionIn.y)
           .lightMeUp(SquareToLightUp.FOOD);
}
// &end[Food::Spawn]
```

HAnS: **H**elping **An**notate **S**oftware

Jetbrains IntelliJ Plugin

https://bitbucket.org/easelab/hans-text/

mapping features

- annotation syntax
- code completion
- live templates
- surrounding live templates

browsing features

refactoring features

HAnS: **H**elping **An**notate **S**oftware

mapping features

browsing features

- feature model view
- find usages
- syntax highlighting

refactoring features

HAnS: **H**elping **An**notate **S**oftware

mapping features

browsing features

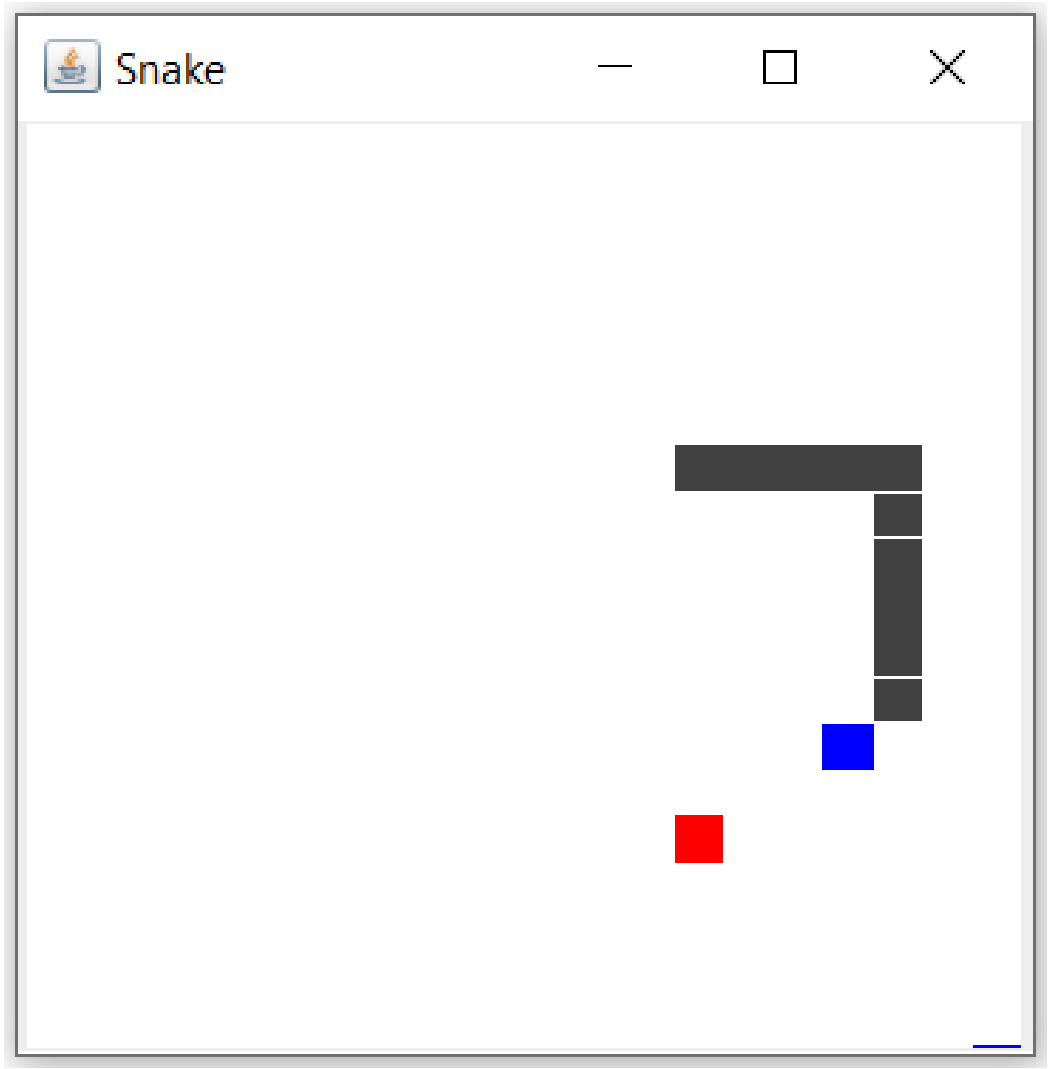
refactoring features

- rename a feature
- add / delete from the feature model

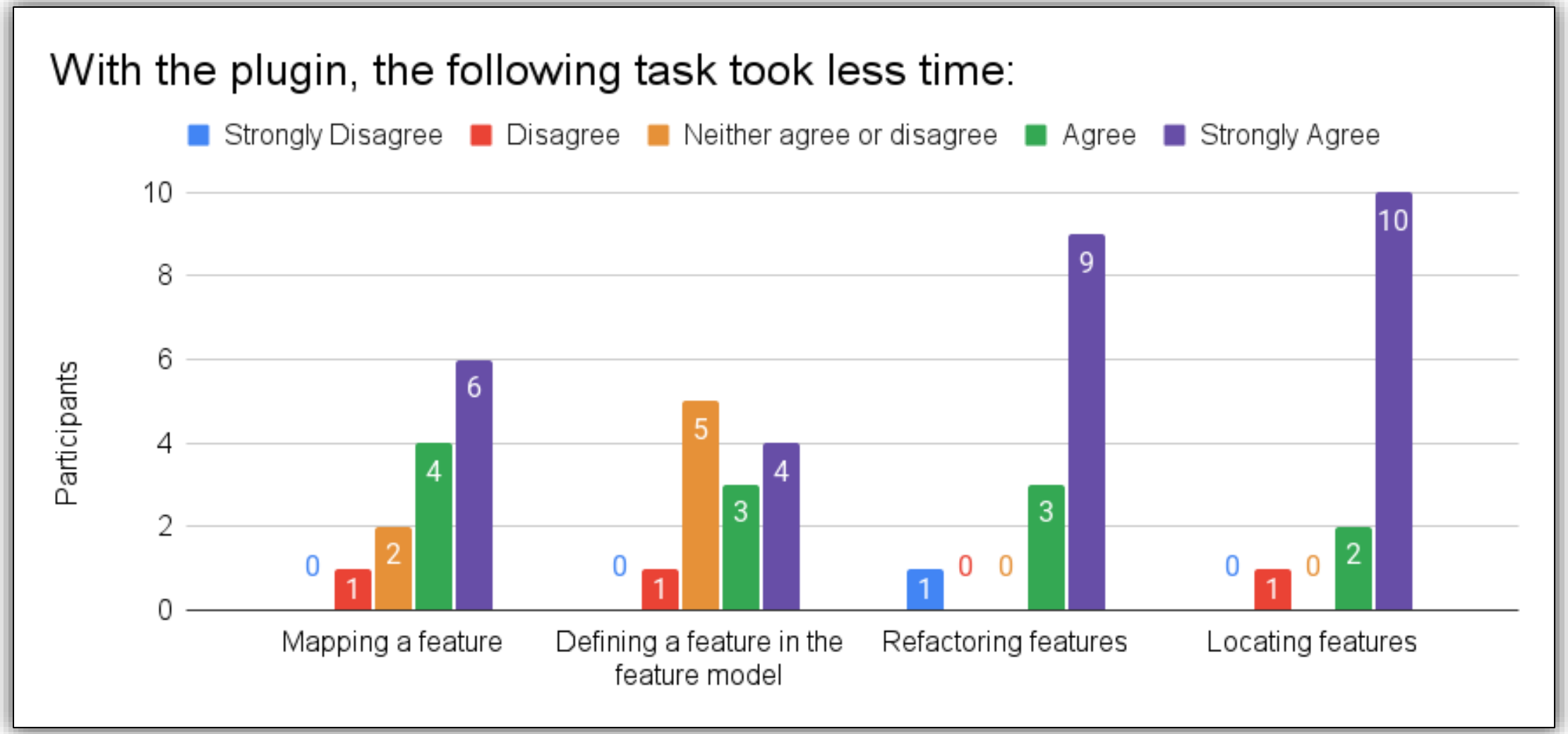
evaluation – user study

overall methodology
- design science (multiple iterations)
- evaluation: experiment

experiment with student developers
- subject system: a small *Snake* game
- cross-over design:
  - two groups, two sets of tasks
  - treatment: HAnS
- screen recording
- questionnaire



developer performance



mistakes made

Severity	Error	Plugin Enabled	Plugin Disabled
High	Feature reference	4	6
High	Syntax	2	6
Medium	Definition	4	6
High	Spelling	0	0
	Total:	10	18

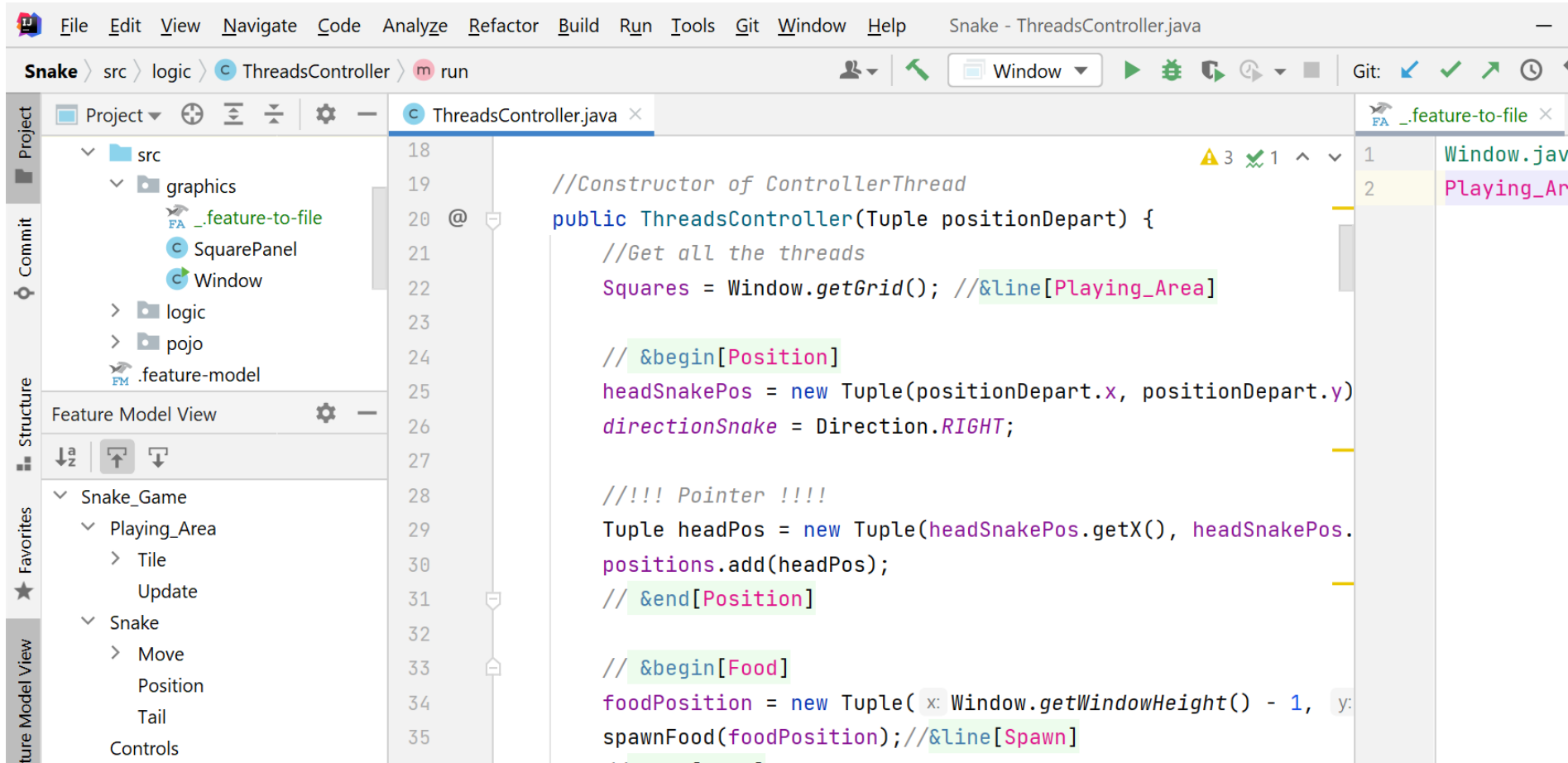
some responses

positively perceived:
feature browsing
feature referencing
feature refactoring

"Refactoring was very tedious without support for the built-in refactoring in IntelliJ."

"I would have liked to see some functionality were I can mark some code and then surround it with a begin and end tag."

"It would be nice to have more functionality like finding usages in the Feature Model View."

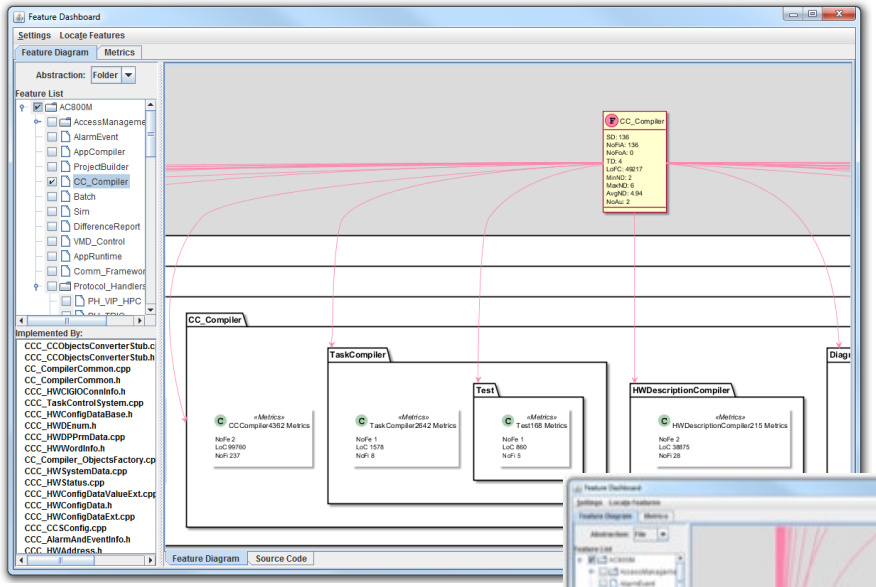


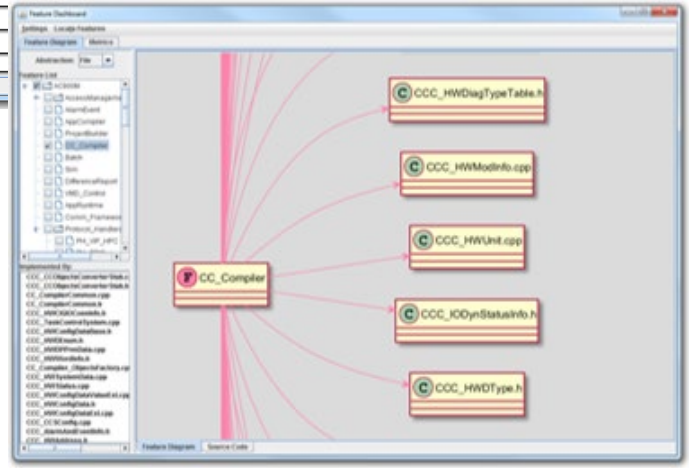
Need for more integrated tool support

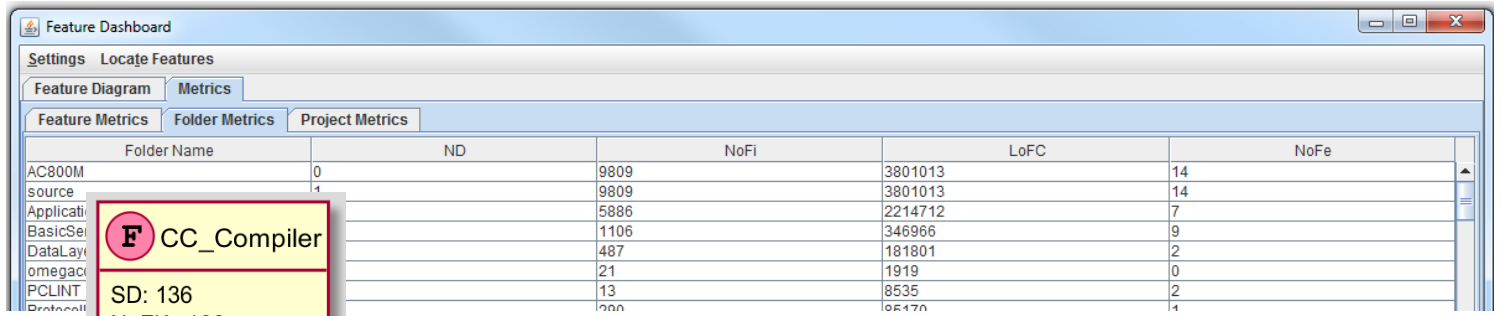
Visualization

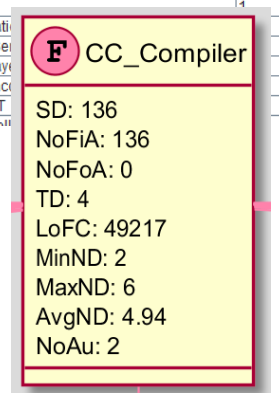
Reminders to continually annotate

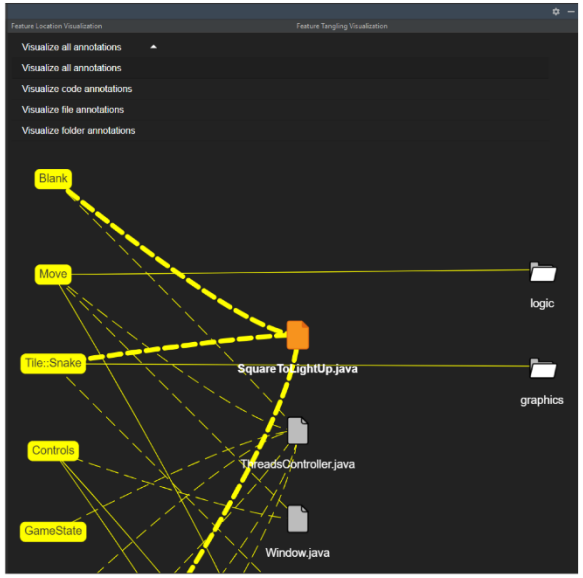
further tool support





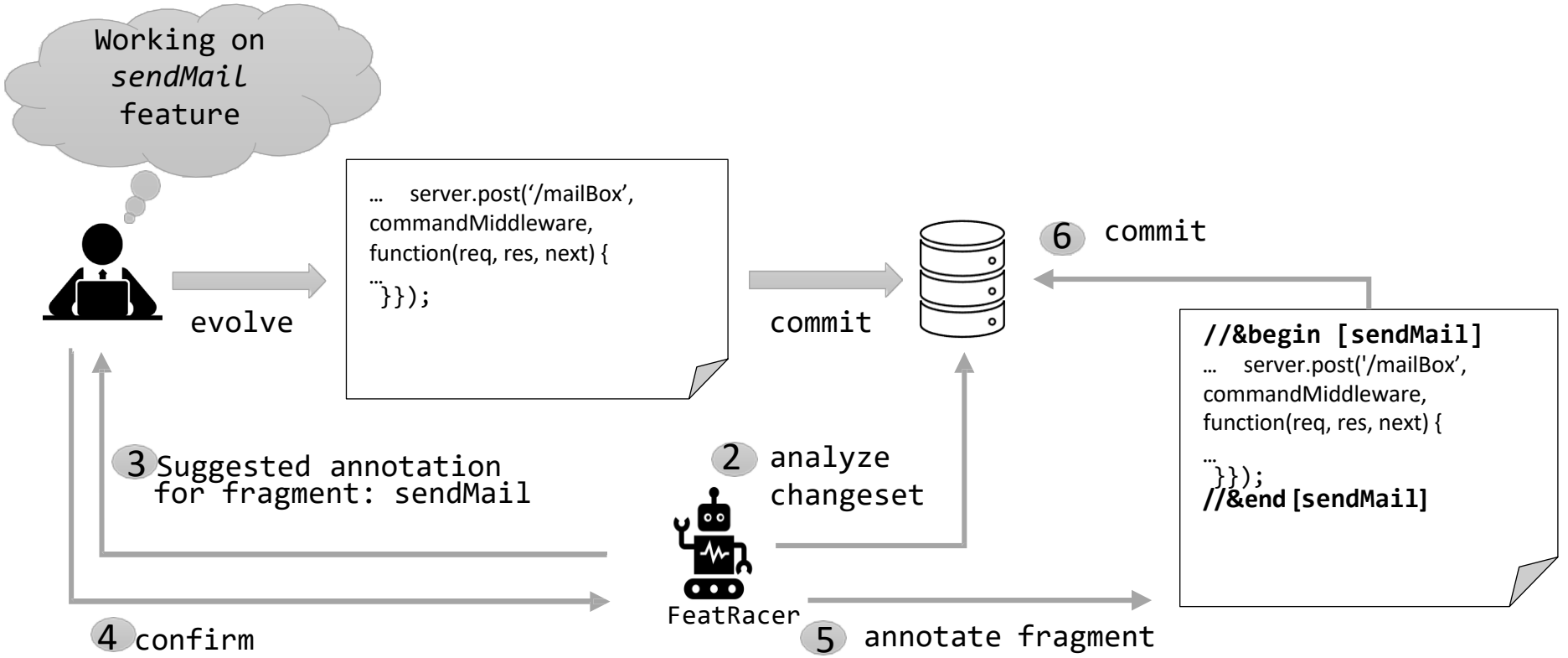




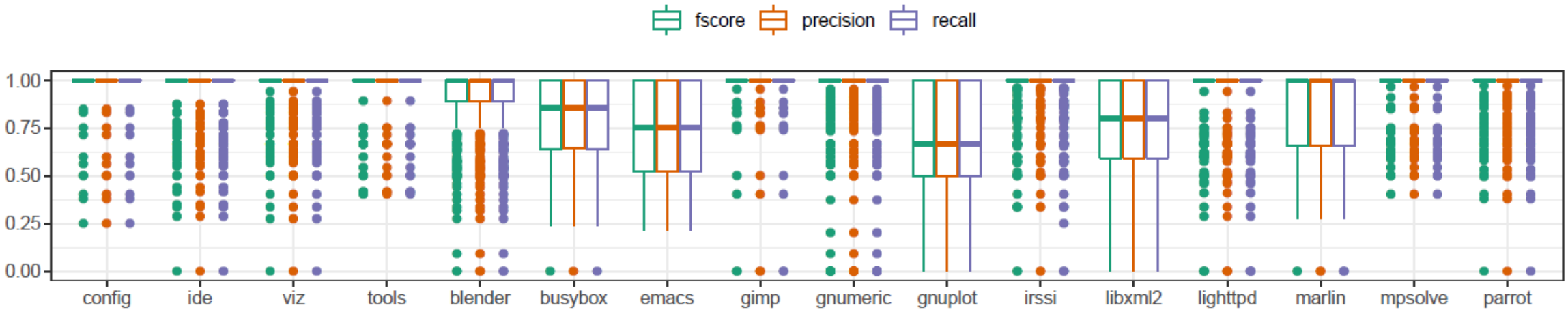


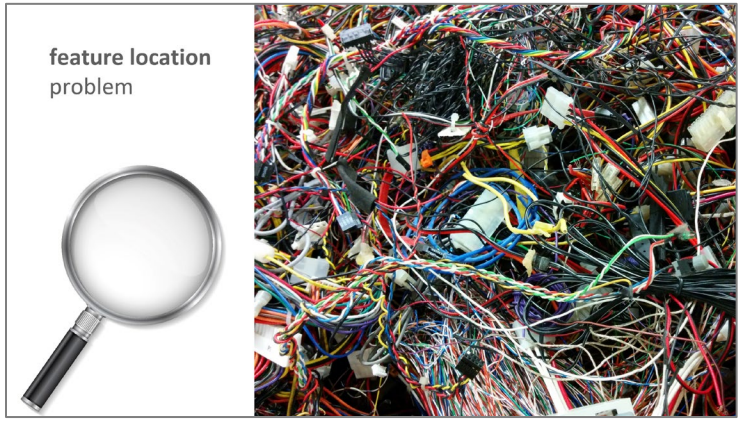
Andam, Burger, Berger, Chaudron. *FLOrIDA: Feature LOcatIon DAshboard for Extracting and Visualizing Feature Traces.* VaMoS. 2017
Entekhabi, Solback, Steghöfer, Berger. Visualization of Feature Locations with the Tool FeatureDashboard. SPLC. 2019
Abukwaik, Burger, Andam, Berger. Semi-Automated Feature Traceability with Embedded Annotations. ICSME. 2018.
Schwarz, Mahmood, Berger. A Common Notation and Tool Support for Embedded Feature Annotations. SPLC. 2020

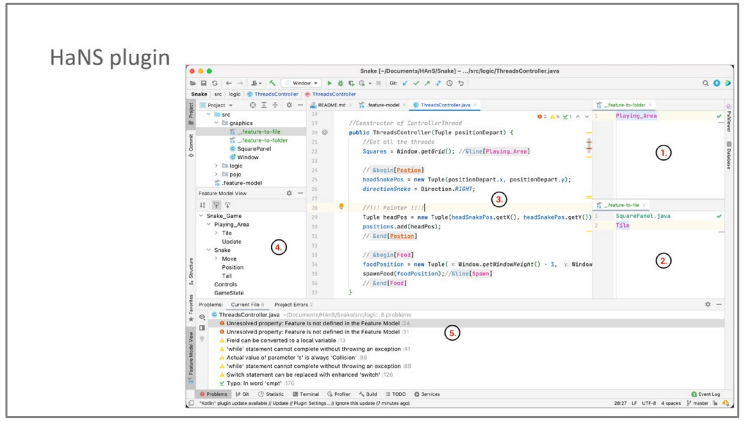
FeatRacer: Feature traceability recommender

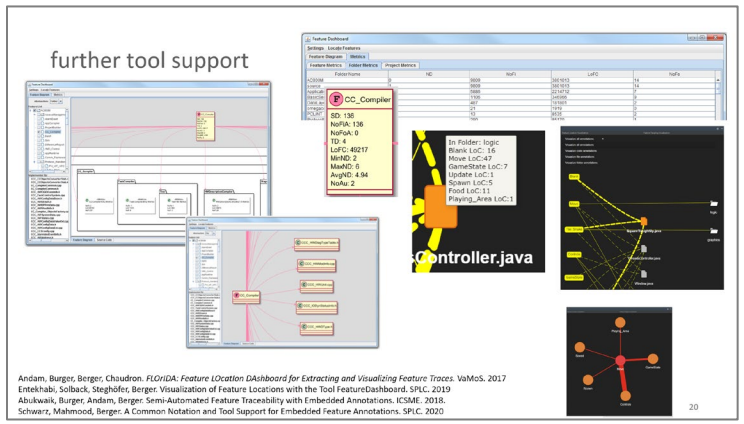


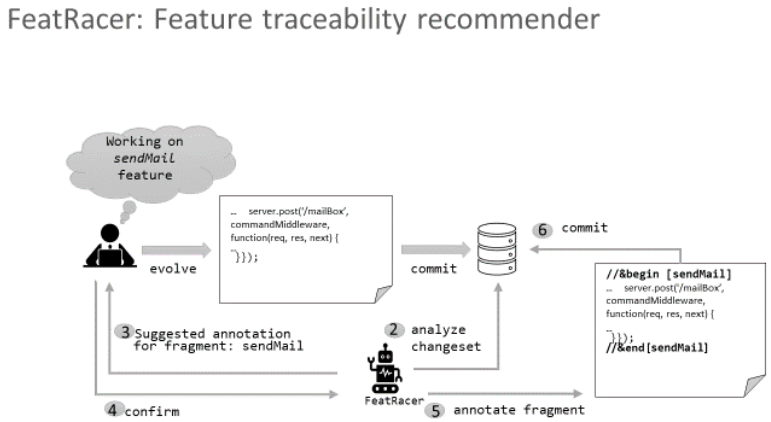
Needs IDE integration











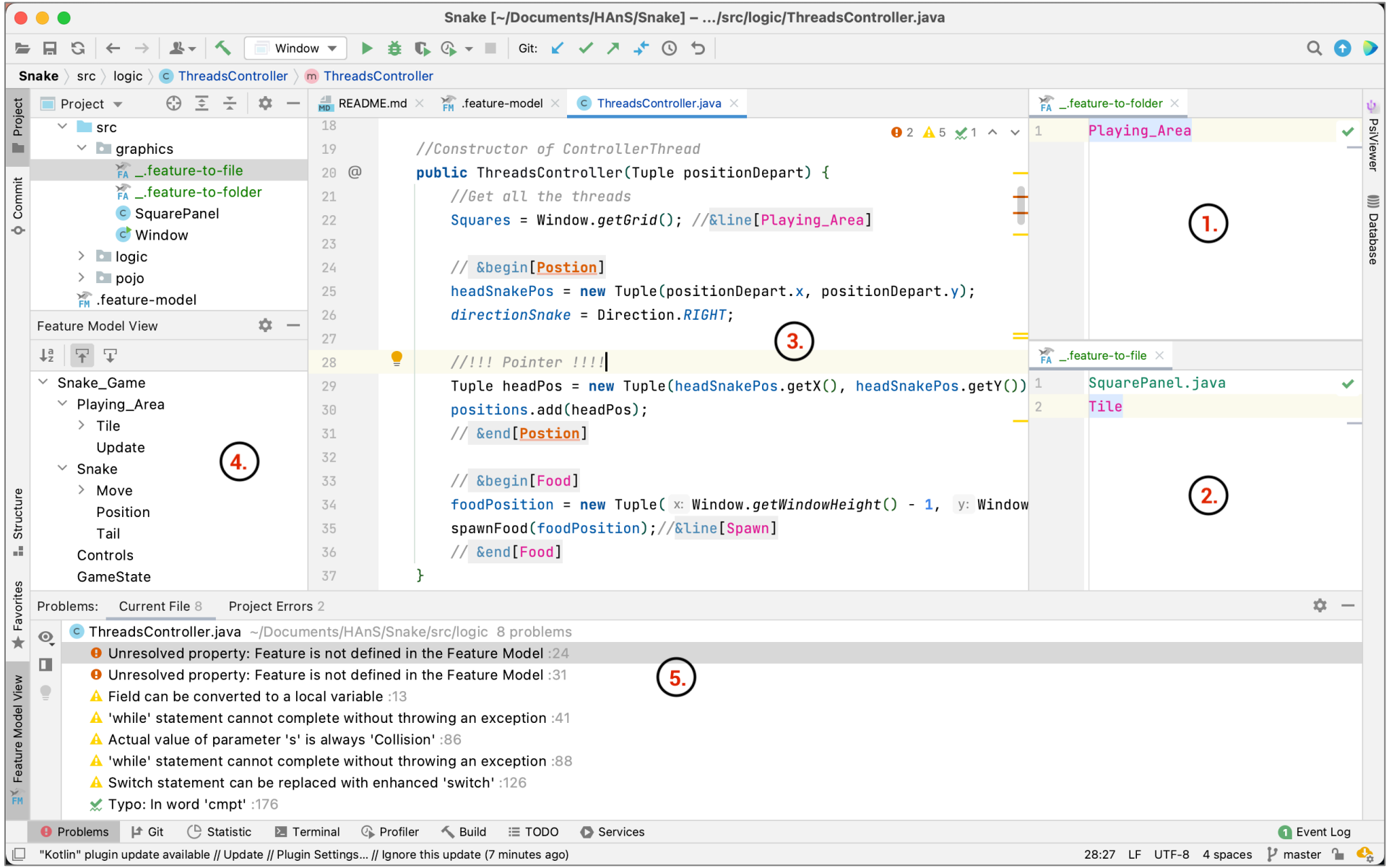
22

https://bitbucket.org/easelab/hans-text/

HAnS: IDE-Based Editing Support for Embedded Feature Annotations

Johan Martinson, Herman Jansson, Mukelabai Mukelabai, Thorsten Berger, Alexandre Bergel and Truong Ho-Quang

HaNS plugin


Snake [~/Documents/HAnS/Snake] – .../src/logic/ThreadsController.java
Snake › src › logic › ThreadsController › ThreadsController
Project
src
graphics
_.feature-to-file
_.feature-to-folder
SquarePanel
Window
logic
pojo
.feature-model
Feature Model View
Snake_Game
Playing_Area
Tile
Update
Snake
Move
Position
Tail
Controls
GameState
4.
README.md
.feature-model
ThreadsController.java
//Constructor of ControllerThread
public ThreadsController(Tuple positionDepart) {
//Get all the threads
Squares = Window.getGrid(); //&line[Playing_Area]
// &begin[Postion]
headSnakePos = new Tuple(positionDepart.x, positionDepart.y);
directionSnake = Direction.RIGHT;
3.
//!!! Pointer !!!!
Tuple headPos = new Tuple(headSnakePos.getX(), headSnakePos.getY())
positions.add(headPos);
// &end[Postion]
// &begin[Food]
foodPosition = new Tuple(x: Window.getWindowHeight() - 1, y: Window
spawnFood(foodPosition);//&line[Spawn]
// &end[Food]
}
_.feature-to-folder
Playing_Area
1.
_.feature-to-file
SquarePanel.java
Tile
2.
Problems: Current File 8 Project Errors 2
ThreadsController.java ~/Documents/HAnS/Snake/src/logic 8 problems
Unresolved property: Feature is not defined in the Feature Model :24
Unresolved property: Feature is not defined in the Feature Model :31
5.
Field can be converted to a local variable :13
'while' statement cannot complete without throwing an exception :41
Actual value of parameter 's' is always 'Collision' :86
'while' statement cannot complete without throwing an exception :88
Switch statement can be replaced with enhanced 'switch' :126
Typo: In word 'cmpt' :176
Problems Git Statistic Terminal Profiler Build TODO Services Event Log
"Kotlin" plugin update available // Update // Plugin Settings... // Ignore this update (7 minutes ago)
28:27 LF UTF-8 4 spaces master

Paper Reference

Martinson, J., Jansson, H., Mukelabai, M., Berger, T., Bergel, A., & Ho-Quang, T. (2021, September). HAnS: IDE-based editing support for embedded feature annotations. In *Proceedings of the 25th ACM International Systems and Software Product Line Conference-Volume B* (pp. 28-31).