Concept-aware Programming Environments
for Program Comprehension and Modularity

Toni Mattis
Robert Hirschfeld
Software Architecture Group
Hasso Plattner Institute, University of Potsdam, Germany

NII Shonan Seminar No. 147  25 – 28  Feb. 2019
“Working Definitions”

Modularity

The quality of a system that allows parts to...

› change or run independently (module as unit of variation/distribution)

› be understood independently (module as concept)

Concept

(Named) unit of comprehension and communication
Problem: Architectural Drift

Many software projects start with good **modularity**

» Low effort to locate and understand concepts

modules

separation of concerns
Problem: Architectural Drift

With growing code bases...
- Concepts tend to scatter and entangle
- Programmers need more attention to recognize concepts

modules
separation of concerns

scattering / tangling
Goal

Help programmers…

» Find, navigate, and relate existing concepts to code
» Improve architecture to better express underlying concepts
Working Hypothesis

Modularity may not be perceived, but **concepts** leave statistically quantifiable **footprints**:

- shared identifier **names**
- simultaneous **changes**
- shared **control flow** and test coverage
- shared **data structures** and setup code
- shared **authors**
- …
Working Hypothesis

Modularity may not be perceived, but concepts leave statistically quantifiable footprints:

... can we make modularity perceivable by providing a qualitatively different view that emphasizes which concepts exist and where they are implemented/used?
Approach

1 “Concept Mining”  2 Tooling

- Concept 1
  - draw
  - color
  - pen

- Concept 2
  - bounds
  - point
  - origin
Basic Concept Model

**concept labels**

which concept a name belongs to

Canvas » \texttt{draw: anObject}

\texttt{^ anObject drawOn: self}

Morph » \texttt{drawOn: aCanvas}

\texttt{aCanvas fillRectangle: self bounds.}

Morph » \texttt{bounds: newBounds}

\texttt{self position: newBounds topLeft;}

\texttt{extent: newBounds extent.}

**concepts**

prevalent names & features

- \texttt{draw, canvas, fill, ...}

**relations**

(e.g. usage)

- \texttt{bounds, position, extent, ...}
Concept Mining as ML Problem

Multi-view learning: Link concepts to features (names) such that

1. Sharing a concept reflects proximity in AST, call graph, edit history, ...
   “clusters”

2. Relations between concepts are consistent with individual feature’s relations
   “inter-cluster relations”
Concept Mining as ML Problem

\[ P(a, b | c) \propto P(a | c)P(b | c) \]

probabilistic clustering (bi-term topic model)

\[ P(a \rightarrow c_1, c_2) \propto P(a | c_1)P(b | c_2)P(c_2 | c_1) \]

Gibbs sampler

inter-concept relations (hidden Markov model)

HMM transition
Workflows

» **Reverse Engineering:** Help programmers understand the conceptual structure of a large system

```
grep -r -i --include /*.java "search"
```
Workflows

» Reverse Engineering: Help programmers understand the conceptual structure of a large system

Goal: Concept Navigation

Search Concept
- name, type, package

Matching Concept
- pattern, rule, case

uses

 org.eclipse.jdt/.../Editor.java

 org.eclipse.jdt/.../SearchProvider.java
Reverse Engineering: Help programmers understand the conceptual structure of a large system.
Workflows

» **Reverse Engineering**: Help programmers **understand** the conceptual structure of a large system

» **Metrics**: Quantify how architecture deviates from conceptual structure
Workflows

» **Reverse Engineering**: Help programmers **understand** the conceptual structure of a large system

» **Metrics**: **Quantify** how architecture deviates from conceptual structure

» **Forward Engineering**: Maintain and **improve** modularity by real-time feedback and recommendations
Scenarios

» Reverse Engineering
  › Semantic Information Retrieval
  › Semantic Navigation

» Metrics
  › Modularity Metrics (Coupling, Cohesion, ...)
  › Concept Linting / Code Critique Tools

» Forward Engineering
  › Automated Refactorings
  › Recommendation based on concepts
Example: Live Assistance

User » **query:** sql
    | cursor |

... 

**“database”** concept is not exposed by class **User**.

- move method to **DBConnection**
- rename method
- add “database” concept to **User**
Example: Conceptual Test Prioritization

» Keep track of which *concepts* are touched during program modification

» **RQ:** How much faster can we detect errors by prioritizing tests by conceptual relatedness?

---

**Recall**

- 77.2% after 0.1s
- 93.4% after 1.0s

---

**Simple topic model to detect concepts**

**Python web framework with seeded faults**
Challenge: Programmer Override

- Programmer override
  - Concept stability under added constraints

- draw
- fill
- origin
- point
- bounds
Challenge: Programmer Override

» Programmer override
  › Concept stability under added constraints

- draw
- fill
- origin
- point
- bounds
- extent
- rectangle
Challenge: Programmer Override

» Programmer override
  › Concept stability under added constraints
Open Questions

» How do our user interfaces need to look like to
  › help programmers understand the *conceptual context* they are currently exploring, editing, debugging, ...
  › keep programmers *aware of modularity issues* without distracting them?

» How can we balance the trade-off between *automated* (potentially surprising) and *manual* concept maintenance?

» How can the proposed concept model be maintained *collectively*?