Mining Concepts from Code
using Community Detection in co-occurrence Graphs

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

ACM SRC at Programming  11 Apr. 2018, Nice, France
Problem: Architectural Drift

With growing code bases...

» Concepts tend to **scatter** and **entangle**
» Programmers need **more time** to locate concepts

**modules**
separation of concerns

scattering / tangling
Goal: Concept Recovery and Location
Name-based Concept Model

**concept locations**
which concept a name belongs to

Canvas » `draw:` anObject
  ^ anObject `drawOn:` self

Morph » `drawOn:` aCanvas
  aCanvas `fillRectangle:` self `bounds`.

Morph » `bounds:` `newBounds`
  self `position:` `newBounds` `topLeft`;
  `extent:` `newBounds` `extent`.

**concepts**
prevalent names

**relations**
(e.g. usage)

- `draw`, `canvas`, `fill`, ...
- `bounds`, `position`, `extent`, ...
Graph-based Semantic Models

Nodes are names. Edges indicate they co-occur in close proximity.
Random Graph Model + Topic Model

Concept $c$
- distribution over names $P(n|c)$
- global frequency $P(c)$

$P(e = (n_1, n_2) \mid c) \propto P(n_1 \mid c)P(n_2 \mid c)$

$P(G = (V, E) \mid ... ) \propto \prod_{e \in E} \sum_c P(e \mid c) = \boxed{\text{null}}$
Random Graph Model + Topic Model

Concept $c$

distribution over names $P(n|c)$
global frequency $P(c)$

$$P(e = (n_1, n_2) | c) \propto P(n_1|c)P(n_2|c)$$

$$P(G = (V, E) | ... ) \propto \prod_{e \in E} \sum_{c} P(e|c) =$$
Determining $P(n|c)$ via Gibbs Sampling

Random assignment of latent variables $c$ to edges

Iterative Re-sampling

1. Decide on maximum number of concepts
2. Uniformly assign a concept to each edge
3. Re-assign each edge until near convergence

(clustering edges instead of nodes)
Random Graph Model + Topic Model
Random Graph Model + Topic Model
Multi-view Concepts

Co-located Names

Run-time Call Data

Git Commit (Diff)

(Multi-)Graph

Concept Distribution

Concept Labeling
Future Work: Concept-aware Tooling

» **Highlight** concepts

» **Improve relevance** of information displayed during
  › search
  › code completion
  › debugging

- graph, vertex, node
- city, road, speed
- draw, canvas, fill, ...

Debug: ZeroDivision in Edge » Cost

- Edge cost
- Graph aStar
- Vertex shortestPathTo
- City planRouteTo
- MapUI planRoute
- Button onClick

City >> planRouteTo: destination
  ^ Route new waypoints:
  (self |

  vertex shortestPathTo:
  roads [Road, ...]
  name "Potsdam"
  populationSize 167745
## Concept Coherence (Mimno et al.)

<table>
<thead>
<tr>
<th>Project</th>
<th>concepts</th>
<th>LDA</th>
<th></th>
<th>Co-occurrence Graph</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$C_2$</td>
<td>$C_4$</td>
<td>$C_8$</td>
<td>$C_{12}$</td>
</tr>
<tr>
<td>EPIC</td>
<td>10</td>
<td>-1.5</td>
<td>-11.0</td>
<td>-53</td>
<td>-135</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>-1.6</td>
<td>-9.9</td>
<td>-53</td>
<td>-143</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>-1.7</td>
<td>-10.6</td>
<td>-57</td>
<td>-144</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>-1.8</td>
<td>-11.7</td>
<td>-56</td>
<td>-144</td>
</tr>
<tr>
<td>Django</td>
<td>10</td>
<td>-1.9</td>
<td>-12.1</td>
<td>-65</td>
<td>-166</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>-2.4</td>
<td>-13.4</td>
<td>-68</td>
<td>-171</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>-2.3</td>
<td>-12.6</td>
<td>-67</td>
<td>-170</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>-2.0</td>
<td>-12.3</td>
<td>-69</td>
<td>-173</td>
</tr>
<tr>
<td>IPython</td>
<td>10</td>
<td>-2.0</td>
<td>-13.9</td>
<td>-67</td>
<td>-172</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>-1.9</td>
<td>-12.8</td>
<td>-68</td>
<td>-167</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>-2.1</td>
<td>-13.1</td>
<td>-70</td>
<td>-169</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>-1.7</td>
<td>-12.2</td>
<td>-63</td>
<td>-164</td>
</tr>
</tbody>
</table>
Table 4  Concepts inferred from the EPIC digital simulator

<table>
<thead>
<tr>
<th>Names</th>
<th>Comment (Interpretation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0  event if true mouse hand</td>
<td>A cross-cutting concept handling mouse interaction</td>
</tr>
<tr>
<td>1  morph layout panel create box</td>
<td>The editor window</td>
</tr>
<tr>
<td>2  canvas draw box center color</td>
<td>Drawing circuitry</td>
</tr>
<tr>
<td>3  is simulation if event not</td>
<td>Event-driven simulator</td>
</tr>
<tr>
<td>4  color rectangle string fill at</td>
<td>Drawing shapes and text</td>
</tr>
<tr>
<td>5  wire point anchor points bundle</td>
<td>Wires, bundles of wires, and their connections</td>
</tr>
<tr>
<td>6  input output values first with</td>
<td>Expectations encoded in unit tests</td>
</tr>
<tr>
<td>7  components panel component command all</td>
<td>The panel containing pre-defined components</td>
</tr>
<tr>
<td>8  xml circuit element named as</td>
<td>The (de)serializer</td>
</tr>
<tr>
<td>9  file name stream as named</td>
<td>File reading/writing</td>
</tr>
</tbody>
</table>

Figure 4  Abstract concepts (left) and how likely they relate to implementation-specific concepts (top). Values are logarithmically scaled.
Summary

» Graph-based concept modeling is a **framework** based on a co-occurrence relation over names

» **Future work:** extend tools to exploit conceptual information

» By giving programmers **feedback** how well their modules align with concepts, they can **counteract** architectural drift
Backup Slides
A Perspective on Modularity

**module entropy:**
- tangling

**concept entropy:**
- scattering

\[ H(m) = - \sum_c p(c|m) \log_2 p(c|m) \]

\[ H(c) = - \sum_m p(m|c) \log_2 p(m|c) \]

...high values indicate need for refactoring or cross-cutting concerns

Related Work

Topic Models

Random Graph Models with Community Structure
Topic Models

60% + 40%

topics
terms sharing a common distribution

60%
40%

route destination start
vertex edge graph
circle line color
route destination start
topics
vertex edge graph
circle line color
topic models
vertex edge path
circle line color
topic models
vertex edge
polygon color
topic models
vertex
canvas
polygon
color
topic models
vertex
edge
topic models
vertex
edge

route destination start
topic models
vertex
canvas
topic models
vertex
canvas
topic models
vertex
canvas
topic models
vertex
canvas
Disambiguating Names

« product »

```java
order.total += product.price;
product = matrix * vector;
```
Disambiguating Names

```java
order.total += product.price;
product = matrix * vector;
```
Disambiguating Names

```java
order.total += product.price;
product = matrix * vector;
```