Facilitating Reuse in Model-Based Development with Context-Dependent Model Element Recommendations

Lars Heinemann

Technische Universität München, Germany

RSSE 2012, Zürich, Switzerland
Model-Based Development

• Models become primary development artifacts

Example: A/C control modeled in Matlab/Simulink
Challenge

• Modeling libraries can be large and complex
• How to find the right element for a given task?

Simulink Standard Library: 135 blocks in 16 categories
Recommendation Systems in SE

Machine Learning

Shared Knowledge Base

Task-specific Recommendations
Contribution

• Approach for context-dependent model element recommendation
• Instantiation for Simulink
• Case study with 165 model files
Approach

• Abstract model as the set of model elements used in it
• Learn knowledge base on a set of training models
• Recommend model elements for unfinished models
• Two recommender variants
  – Association Rules
  – Collaborative Filtering
Background: Association Rules

- Associations between items in shopping baskets

\[
\{\text{nachos, dipping sauce}\} \Rightarrow \text{cola}
\]

\[
support(I \rightarrow j) = \frac{\text{baskets containing all elements of } I \cup j}{\text{overall number of baskets}}
\]

\[
\text{confidence}(I \rightarrow j) = \frac{\text{baskets containing all elements of } I \cup j}{\text{baskets containing all elements of } I}
\]
AR Recommender

- Model elements are *items*
- Models are *shopping baskets*
- Recommend *right sides* of all association rules applicable to a given model
- Parameters
  - *support threshold*
  - *confidence threshold*
### Backgr.: Collaborative Filtering

#### Table: Users' Preferences

<table>
<thead>
<tr>
<th>Items</th>
<th>Joe</th>
<th>Ann</th>
<th>Mike</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moby Dick</td>
<td>☺</td>
<td></td>
<td>☺</td>
</tr>
<tr>
<td>The Time Machine</td>
<td>☻</td>
<td>☻</td>
<td></td>
</tr>
<tr>
<td>Star Wars</td>
<td>☻</td>
<td>☻</td>
<td>☻</td>
</tr>
<tr>
<td>Once Upon a Time in the West</td>
<td>☻</td>
<td>☻</td>
<td></td>
</tr>
</tbody>
</table>

We could recommend "Star Wars" to Joe.
CF Recommender

• Model elements are *items*
• Models are *users*
• Model usage is the *like-relation*
• *K-Nearest-Neighbor approach*
  – Recommend model elements from *k* most similar models (regarding block usage)
• Parameter: *k*
Case Study

• **Goals**
  – Assess adequacy of recommendations
  – Determine influence of parameters

• Comparison between
  – Baseline recommender (always recommends top-n blocks)
  – AR recommender
  – CF recommender
Case Study: Evaluation method

- 10-fold cross validation

1. Train subsystems
2. “Predict“

90% 10 “rounds“ 10%

Training subsystems Test subsystems
Predicting Model Elements

• Randomly remove all occurrences of 1/2 of the model elements
• User recommender to guess removed elements from remaining elements
Recommendation Adequacy

• How many of the recommendations are correct?
• How many of the expected recommendations are actually made?

\[
\text{precision} = \frac{\text{correct recommendations}}{\text{total recommendations}}
\]

\[
\text{recall} = \frac{\text{correct recommendations}}{\text{actually employed blocks}}
\]

\[
\text{F-measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}
\]
Study Objects

• 165 Simulink model files from Matlab Central
• 1103 subsystems
• 335 distinct library blocks used

Number of distinct blocks used per subsystem

<table>
<thead>
<tr>
<th>min</th>
<th>p25</th>
<th>median</th>
<th>p75</th>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td>16</td>
</tr>
</tbody>
</table>
The *Long Tail* of Block Usage
Results: Baseline

for $n=3$
17% precision
21% recall

Number of recommended blocks
Results: Association Rules

![Graph showing Precision, Recall, and F-Measure for various confidence thresholds. For a confidence threshold of 0.4, the graph indicates 32% precision and 30% recall.](graph.png)
Results: Collaborative Filtering

For $k=1$
66% precision
49% recall

Baseline F-measure
### Number of Recommended Blocks per Query

<table>
<thead>
<tr>
<th></th>
<th>AR (conf=0.4)</th>
<th>CF (k=1)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>min</strong></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>p25</strong></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>median</strong></td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td><strong>p75</strong></td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td><strong>max</strong></td>
<td>11</td>
<td>9</td>
</tr>
</tbody>
</table>
Discussion

- Both variants outperform baseline approach
- Trade-off between precision and recall depending on parameters
- AR achieves better precision than CF
- In terms of F-measure, CF better than AR
Conclusion

• Approach for context-dependent recommendation of model elements
• Instantiated and evaluated for Simulink
• Adoption of recommendation systems for model-based development promising
Future Work

• **Different notions of similarity for (partial) models**
  – e.g. take into account element interconnections

• **More fine-grained context** (currently complete subsystem)
  – e.g. take into account where element is to be inserted
    (consider predecessors in data flow)

• Extend studies to **other modeling languages**

• **User evaluation**: How much can a developer benefit from recommendations?
Thank you.
Questions?