Using Obfuscating Transformations for Supporting the Sharing and Analysis of Conceptual Models

Dr. Hans-Georg Fill
Research Group Knowledge Engineering
University of Vienna
Agenda

• Motivation
• Foundations
• Obfuscating Transformations for Conceptual Models
• Use Case
• Conclusion and Outlook
Motivation

• Several initiatives for sharing of conceptual models, reference models, modeling languages and modeling tools
• Based on principles similar to Open Source
• Goals:
  – Provide large repositories of models for learning, research, and development purposes
  – Establish a community of practitioners and academics
  – Exchange of knowledge and expertise
  – Access to IT concepts and tools
Example: Open Model Initiative

"...intends to establish a community of people who focus on the creation, maintenance, modification, distribution, and analysis of models."

(Karagiannis et al., 2008, p.3)

Current stats:

• approx. 17 categories of reference models available on www.openmodels.org that include >40 model instances
• approx. 22 projects that develop/provide modeling tools and services on www.openmodels.at

www.openmodels.org www.openmodels.at
Aspects of Sharing Models

- Intellectual property aspects of models
- Confidential information contained in models:
  - Work practices
  - Business relationships
  - Other information that is potentially harmful when disclosed to competitors
- Benefits of sharing models:
  - Enable comparisons, benchmarks
  - Receive feedback for improvements
  - Provide documentation for interaction with customers and suppliers

Balance?
Foundations
Privacy Preserving Data Mining (PPDM)

- Sanitizing data sets for shielding sensitive information and disallowing derivation of certain associations
  - Before the mining process
  - After the application of data mining techniques

- Preservation of properties similar to original data sets to receive reasonably accurate results

- Secure multi-party computations:
  - Several parties provide input for computations, e.g. benchmarking
  - Input of parties is not disclosed
Source Code Obfuscation

• Convert source code of programs into one with same observable behavior:
  – Same input produces the same output
  – Input is harder to understand and reverse-engineer

• Three types of obfuscations:
  – Layout transformations: change the formatting of source code and identifiers of variables
  – Control transformations: modify the control flow of a program to hide its functioning
  – Data transformations: target the data structures of an application
Obfuscating Transformations for Conceptual Models
Goals of Obfuscating Transformations for Conceptual Models

• Hide confidential and potentially harmful information in models
• Adapt techniques from PPDM and source code obfuscation
• Preserve main structure and contained semantic information as much as possible
• Preserve applicability of certain algorithms
• Focus on one-way and highly resilient transformations
• Avoid adding complexity to models to maintain comprehensibility
Concept for Applying Obfuscating Transformations

![Diagram]

- Meta Model
- Model Instance
- Obfuscating Transformation
- Algorithm
- Model Instance

- Conforms to
- Based on
- Works on
- Conforms to subset of

α → α'
β
α₀
Types of Transformations

- Representation transformations
- Structural transformations
- Data transformations
- Semantic obfuscation transformations
Representation Transformations I

- Derived from layout transformations in source code obfuscation:
  - Hiding information
  - Abstraction of models
Representation Transformations II

A → B
C → D → E → F → I
G → H

Task-68605
→ Task-68608
→ Task-68611
→ Task-68617
→ Task-68620
→ Task-68623

M
Structural Transformations I

- Change the structure and complexity of the models to hide model details
- Use nesting mechanisms to embed models in each other and hide parts of models
Data Transformations I

- Obfuscation of labels and attributes of elements:
  - Scrambling
  - Numerical Value Transformations
## Data Transformations II

### Original process

- **Task 1**: Call to Sub-Process A
- **Task 2**: Call to Sub-Process A

### Attributes

<table>
<thead>
<tr>
<th></th>
<th>Original Attributes</th>
<th>Obfuscated Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Execution Time</strong></td>
<td>00:02:30:20</td>
<td>00:02:00:00</td>
</tr>
<tr>
<td><strong>Waiting Time</strong></td>
<td>00:00:10:05</td>
<td>**XX:**XX:**XX:<strong>XX</strong></td>
</tr>
<tr>
<td><strong>Transport Time</strong></td>
<td>00:00:00:40</td>
<td><strong>0 &lt; X &lt; 00:00:00:80</strong></td>
</tr>
<tr>
<td><strong>Cost</strong></td>
<td>200,43</td>
<td><strong>20 &lt; X &lt; 300</strong></td>
</tr>
</tbody>
</table>

### Execution Time

<table>
<thead>
<tr>
<th></th>
<th>Execution Time</th>
<th>Waiting Time</th>
<th>Transport Time</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Original</strong></td>
<td>00:02:30:20</td>
<td>00:00:10:05</td>
<td>00:00:00:40</td>
<td>200,43</td>
</tr>
<tr>
<td><strong>Obfuscated</strong></td>
<td>00:07:10:03</td>
<td>01:01:12:40</td>
<td>00:00:00:00</td>
<td>23,56</td>
</tr>
</tbody>
</table>

### Rounding Obfuscation

- **Fixed Interval Obfuscation**
- **Variable Interval Obfuscation**

### Removal of Information

- **Fixed Interval Obfuscation**
- **Variable Interval Obfuscation**
Semantic Obfuscation Transformation

- Using the subsumption hierarchy of an ontology and semantic annotations
Application to a Use Case

• Based on a publicly available description of an account opening process of a Swiss bank
• Concrete issues that may necessitate obfuscations:
  – Competitive issues, i.e. information that should not be disclosed to competitors
  – Security issues, i.e. information that potentially permits to access non-public systems or could facilitate such access
  – Privacy issues, i.e. information that concerns personal data
• Application of the following techniques:
  – Hiding of information
  – Condensing of elements
  – Label scrambling
  – Variable interval obfuscation for process activity costs
  – Semantic obfuscation transformation for names of IT systems
Example Models from the Use Case

\[ M^1 \]
- Host name: Finstar01
- IP-Address: 169.34.122.12

\[ M^2 \]
- Customer Service Center
- Mrs White
- Service Center Manager
- Mr Green
- Mrs Black
- Mr Red
- Mr Blue
- Availability: 38 hours / week

\[ M_1^{OB} \]
- Host name: Finstar01
- IP-Address: 169.34.122.12

\[ M_2^{OB} \]
- Customer Service Center
- Availability: 38 hours / week

Acquire customer data (personal data, address, ID)
Create new customer profile and enter data
Customer fills out application form W4801
Exec. time: 00:00:00:05:30
Avg. Cost: 2.39 EUR

Exec. time: 00:00:00:05:30
Avg. Cost: \(1 \leq X \leq 3\)
Conclusion and Outlook

• Obfuscating transformations allow to preserve the core structures of the models
• Algorithmic analyses of certain models parts are still feasible – e.g. in regard to the execution time and capacity analyses in the use case
• Also obfuscated models can provide a lot of value for research and knowledge exchange

Next steps:
• Detail transformations using mathematical notation
• Implement them in a modeling tool
• Apply them to further use cases for evaluation
Thank you for your attention!
hans-georg.fill@dke.univie.ac.at

www.openmodels.org
www.openmodels.at
Tutorial
„Metamodellierungsplattformen im Einsatz am Beispiel ADOxx® & SOM“
Modellierung 2012, 14.3.2012, Bamberg