Towards refactoring meta-models into multi-level models

Fernando Macías\textsuperscript{1}  Esther Guerra\textsuperscript{2}  Juan de Lara\textsuperscript{2}
fernandomacias.es

Campina Grande, November 21, 2017

\textsuperscript{1}Western Norway University of Applied Sciences
\textsuperscript{2}Universidad Autónoma de Madrid
This work: What, how and why

Two-level limitations

Multi-level refactoring process

Evaluation and experiments

What is MDSE?

Model-Driven Software Engineering.

Representing information in a structured way.

Uses:

- Code generation
- Simulation
- Documentation
- ...

...
Limitations of traditional approaches

- Complexity tends to grow pretty fast.
- Difficult to read
- Complex maintenance
- Low reusability
Multi-level modelling (MLM)

The good:
- Unbounded levels of abstraction
- Richer semantics
- Cleaner structures

The bad:
- Lack of consensus on foundations
- Technology lock-in
- Tool isolation
(a) “flat” meta-model

- **PolicyType**
  - name: String

- **RuleType**
  - name: String

- **ElementType**
  - name: String
  - hierarchy: bool

- **Rule**
  - name: String
  - rules: *

- **Parameter**
  - name: String
  - children: *
  - elements: *

- **Policy**
  - name: String

- **ruleTypes**
  - 1..*

- **parameters**
  - ruleTypes

- **elementTypes**
  - RuleType
  - ElementType

- **children**
  - Rule
  - Parameter

- **children**
  - Policy

- **elements**
  - Rule
  - Parameter
Security Policies with MLM

(a) “flat” meta-model

PolicyType
- name: String

RuleType
- name: String

ElementType
- name: String
  - hierarchy: bool

Rule
- name: String
  - rules: 1..*

Policy
- name: String

Element
- name: String
  - children: *
  - elements: *

RuleType
- ruleTypes: 1..*

ElementType
- elementTypes: 1..*

Policy
- rules: * elements*

(b) “flat” model

rbac:PolicyType
- name: “RBAC”
  - ruleTypes
    - userRole:RuleType
      - name: “UserRole”
    - user:ElementType
      - name: “User”
        - hierarchy: true
    - p1:Parameter
      - name: “Romain”
    - p2:Parameter
      - name: “Student”

role:ElementType
- name: “Role”
  - hierarchy: true

user:ElementType
- name: “User”
  - hierarchy: false

rule:Rule
- name: “R1”

policy:Policy
- name: “LibraryRBAC”
(a) “flat” meta-model

- **PolicyType**
  - name: String
- **RuleType**
  - name: String
  - rules: *
- **ElementType**
  - name: String
  - hierarchy: bool
- **Parameter**
  - name: String
  - rules: *
- **Rule**
  - name: String
- **Policy**
  - name: String

(b) multi-level specification

- **PolicyType**
  - name: String
- **RuleType**
  - name: String
- **ElementType**
  - name: String
  - hierarchy@1: bool
- **Parameter**
  - name: String
  - children: @2

(c) “flat” model

- **rbac:PolicyType**
  - name: “RBAC”
  - :type
  - :ruleTypes
    - **userRole:RuleType**
      - name: “UserRole”
    - **user:ElementType**
      - name: “User”
    - **role:ElementType**
      - name: “Role”
      - hierarchy: true
  - :type
  - :parameters
    - **p1:Parameter**
      - name: “Romain”
    - **p2:Parameter**
      - name: “Student”
  - :rules
  - **policy:Policy**
    - name: “LibraryRBAC”
Security Policies with MLM

(a) “flat” meta-model

(b) multi-level specification

(C) “flat” model

(d) multi-level model
Security Policies with MLM

(a) “flat” meta-model

(b) multi-level specification

(C) “flat” model

(d) multi-level model
Refactoring into MLM

Goals:
- Empirical data about MLM improvement
- Automatic refactoring tool
- Convergence of MLM frameworks
Step 1: Detecting smells

- Heuristic-based annotation
- Based on structure and name similarity
- Adjustable confidence threshold
- Open to human intervention (i.e. semi-automatic)
Step 1: Detecting smells
Step 2: Refactoring

- Transformation applied to each annotated pattern
- Based on annotations from previous step
- Single-model output (for now)
- Automatic
Step 3: Recommendation

Based on framework support for MLM features

- Three degrees of support: supported, emulated, unsupported
- Scores and ranking output
- Automatic
### Step 3: Recommendation

<table>
<thead>
<tr>
<th>dimension</th>
<th>multi-level feature</th>
<th>start</th>
<th>end</th>
<th>depth</th>
<th>MetaDepth</th>
</tr>
</thead>
<tbody>
<tr>
<td>potency in clabjects and references</td>
<td>standard potency</td>
<td>1</td>
<td>1</td>
<td>n (n≥1)</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>leap potency</td>
<td>n (n&gt;1)</td>
<td>n</td>
<td>1</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>replicability</td>
<td>n (n≥1)</td>
<td>m (m&gt;n)</td>
<td>1</td>
<td>− (m=n)</td>
</tr>
<tr>
<td></td>
<td>deep leap potency</td>
<td>n (n&gt;1)</td>
<td>n</td>
<td>o (o&gt;1)</td>
<td>− (o=1)</td>
</tr>
<tr>
<td></td>
<td>deep replicability</td>
<td>n (n≥1)</td>
<td>m (m&gt;n)</td>
<td>o (o&gt;1)</td>
<td>− (m=n,o=1)</td>
</tr>
<tr>
<td>potency in attributes</td>
<td>attribute durability</td>
<td>1</td>
<td>n (n≥1)</td>
<td>1</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>attribute mutability</td>
<td>n (n≥1)</td>
<td>m (m&gt;n)</td>
<td>1</td>
<td>~ (m=n)</td>
</tr>
<tr>
<td>instantiation</td>
<td>shallow ref. cardinality</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>deep ref. cardinality</td>
<td>n (n&gt;1)</td>
<td>m (m&gt;n)</td>
<td>o (o&gt;1)</td>
<td>~ (OCL)</td>
</tr>
<tr>
<td></td>
<td>multiple typing</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+ (a-posteriori)</td>
</tr>
<tr>
<td></td>
<td>abstract types</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
</tbody>
</table>
### Step 3: Recommendation

<table>
<thead>
<tr>
<th>dimension</th>
<th>multi-level feature</th>
<th>start</th>
<th>end</th>
<th>depth</th>
<th>MultECore</th>
</tr>
</thead>
<tbody>
<tr>
<td>potency in clabjects and references</td>
<td>standard potency</td>
<td>1</td>
<td>1</td>
<td>n (n&gt;1)</td>
<td>~ (n=∞)</td>
</tr>
<tr>
<td></td>
<td>leap potency</td>
<td>n (n&gt;1)</td>
<td>n</td>
<td>1</td>
<td>~ (n=∞)</td>
</tr>
<tr>
<td></td>
<td>replicability</td>
<td>n (n≥1)</td>
<td>m (m&gt;n)</td>
<td>1</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>deep leap potency</td>
<td>n (n&gt;1)</td>
<td>n</td>
<td>o (o&gt;1)</td>
<td>~ (o=∞)</td>
</tr>
<tr>
<td></td>
<td>deep replicability</td>
<td>n (n≥1)</td>
<td>m (m&gt;n)</td>
<td>o (o&gt;1)</td>
<td>~ (o=∞)</td>
</tr>
<tr>
<td>potency in attributes</td>
<td>attribute durability</td>
<td>1</td>
<td>n (n&gt;1)</td>
<td>1</td>
<td>~ (n=∞)</td>
</tr>
<tr>
<td></td>
<td>attribute mutability</td>
<td>n (n≥1)</td>
<td>m (m&gt;n)</td>
<td>1</td>
<td>— (always mutable)</td>
</tr>
<tr>
<td>instantiation</td>
<td>shallow ref. cardinality</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>deep ref. cardinality</td>
<td>n (n&gt;1)</td>
<td>m (m&gt;n)</td>
<td>o (o&gt;1)</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>multiple typing</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+ (supplementary)</td>
</tr>
<tr>
<td></td>
<td>abstract types</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
</tbody>
</table>
### Step 3: Recommendation

<table>
<thead>
<tr>
<th>dimension</th>
<th>multi-level feature</th>
<th>start</th>
<th>end</th>
<th>depth</th>
<th>Melanee</th>
</tr>
</thead>
<tbody>
<tr>
<td>potency in clabjects and references</td>
<td>standard potency</td>
<td>1</td>
<td>1</td>
<td>n (n≥1)</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>leap potency</td>
<td>n (n&gt;1)</td>
<td>n</td>
<td>1</td>
<td>~ (n=1)</td>
</tr>
<tr>
<td></td>
<td>replicability</td>
<td>n (n≥1)</td>
<td>m (m&gt;n)</td>
<td>1</td>
<td>− (n=1,m=n)</td>
</tr>
<tr>
<td></td>
<td>deep leap potency</td>
<td>n (n&gt;1)</td>
<td>n</td>
<td>o (o&gt;1)</td>
<td>− (n=1,o=1)</td>
</tr>
<tr>
<td></td>
<td>deep replicability</td>
<td>n (n≥1)</td>
<td>m (m&gt;n)</td>
<td>o (o&gt;1)</td>
<td>− (n=1,m=n,o=1)</td>
</tr>
<tr>
<td>potency in attributes</td>
<td>attribute durability</td>
<td>1</td>
<td>n (n≥1)</td>
<td>1</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>attribute mutability</td>
<td>n (n≥1)</td>
<td>m (m&gt;n)</td>
<td>1</td>
<td>+</td>
</tr>
<tr>
<td>instantiation</td>
<td>shallow ref. cardinality</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>deep ref. cardinality</td>
<td>n (n&gt;1)</td>
<td>m (m&gt;n)</td>
<td>o (o&gt;1)</td>
<td>−</td>
</tr>
<tr>
<td></td>
<td>multiple typing</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>−</td>
</tr>
<tr>
<td></td>
<td>abstract types</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>~ (potency=0)</td>
</tr>
</tbody>
</table>
Step 4: Deployment

multi-level neutral meta-model

«conforms to»

recommend MLM approach

deploy

MLM tool

features supported by MLM tools

Melanee
MetaDepth
MultEcore
...

miso
Step 4: Deployment

multi-level neutral meta-model

«conforms to»

recommend MLM approach

deploy

MLM tool

features supported by MLM tools

Melanee
MetaDepth
MultiEcore
...

miso

MELANEE
## Evaluation: Gain

<table>
<thead>
<tr>
<th>meta-model</th>
<th>flat meta-model</th>
<th>multi-level version</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>classes</td>
<td>refs.</td>
</tr>
<tr>
<td>Security Policies</td>
<td>7</td>
<td>12</td>
</tr>
<tr>
<td>Agate</td>
<td>69</td>
<td>123</td>
</tr>
<tr>
<td>CloudML</td>
<td>21</td>
<td>28</td>
</tr>
<tr>
<td>CloudML-2.0</td>
<td>33</td>
<td>50</td>
</tr>
<tr>
<td>HAL</td>
<td>42</td>
<td>16</td>
</tr>
</tbody>
</table>
Evaluation: F-score optimization
Conclusions

- Empirical evaluation of MLM with third-party data
- Prototype tool support
- Tool-independent representation of MLM hierarchies
Future work

- Support more scenarios
- Large-scale evaluation (Github)
- Crystallize tool-independent metamodel as exchange format


