Correctness and Completeness for Incremental Model Synchronisation Based on TGGs

BANFF Bidirectional Transformations
04-DECEMBER-2013

Frank Hermann, frank.hermann@uni.lu

SECAN-Lab
Interdisciplinary Centre for Security, Reliability and Trust
Université du Luxembourg
OVERVIEW

• **Formal results** for model synchronization via TGGs

• **Challenges for theory and tool support**

• Conclusion
Interrelated Models in Model Driven Engineering

Model Transformations

1. **PIM↔PIM (horizontal):**
   - DSL1↔DSL2,
   - Model Translation/ Integration/ Synchronisation, e.g.:
   - UML Class Diagram↔RDBM
   - BPMN↔BPEL
   - Sequence Diagram↔State Machines

2. **PIM↔PSM (vertical):**
   - Model/Code generation, reverse engineering, e.g.:
   - Class Diagram↔Class Diagram
   - Class Diagram↔Java

[HEO+13]
Key Idea of Triple Graph Grammars (TGGs)

- **Specify** pattern by pattern how **consistent integrated models** can be constructed **simultaneously**
Key Idea of Triple Graph Grammars (TGGs)

- **Specify** pattern by pattern how **consistent integrated models** can be constructed **simultaneously**
- **Generate** operations for interoperability: Model **Translation/Integration/Synchronisation**

![Triple Graph Diagram]
Key Idea of Triple Graph Grammars (TGGs)

• **Specify** pattern by pattern how **consistent integrated models** can be constructed **simultaneously**

• **Generate** operations for interoperability: Model **Translation/Integration/Synchronisation**

---

**Triple Graph**

- **Source Model**
- **Correspondence Model**
- **Target Model**
Concurrent Synchronization Problem

Source Domain

<table>
<thead>
<tr>
<th>Name</th>
<th>Base</th>
<th>Bonus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alex Archer</td>
<td>3000</td>
<td>1000</td>
</tr>
<tr>
<td>Willy Wilson</td>
<td>500</td>
<td>0</td>
</tr>
</tbody>
</table>

Target Domain

<table>
<thead>
<tr>
<th>Name</th>
<th>Base</th>
<th>Bonus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alex Archer*1975-02-14</td>
<td>4000</td>
<td></td>
</tr>
<tr>
<td>Willy Wilson*1990-06-17</td>
<td>500</td>
<td></td>
</tr>
<tr>
<td>Molly Murphy*1975-02-14</td>
<td>4050</td>
<td></td>
</tr>
</tbody>
</table>

\(r_0\)

\(d^S_1\)

\(d^T_1\)

\(G^S_0 \leftrightarrow G^T_0\)

\(G^S_1 \leftrightarrow G^T_1\)

\(:CSynch\)

\(d^S_2\)

\(d^T_2\)

\(G^S_2 \leftrightarrow G^T_2\)

\(r_2\)
Position Statement: Model Synchronisation

Forward Propagation

\[ G^S \xrightarrow{r} G^T \]

\[ \vdash a \downarrow \]

\[ G'^S \]

\[ G^S \xleftarrow{r} G^T \]

\[ \vdash \text{:fPpg} \]

\[ G'^S \xrightarrow{r'} G'^T \]

Backward Propagation

\[ G^S \xrightarrow{r} G^T \]

\[ \vdash b \downarrow \]

\[ G'^T \]

\[ G^S \xleftarrow{r} G^T \]

\[ \vdash \text{:bPpg} \]

\[ G'^S \xrightarrow{r'} G'^T \]

References:


Conflict Resolution

Resolution Strategy

Preservation over deletion

- Elements are **preserved**, if
  - not deleted by any update or
  - required by one update
- Elements are **deleted**, if:
  - deleted by one update and **not required** by the other
  - deleted by both updates

Correctness

1. **Consistency Law:**
   result is always consistent

2. **Identity Law:**
   no change, if input is already consistent
Formal Result

Definition (Completeness)
Model synchronisation can be performed for any input.

Theorem (Correctness and Completeness)
The derived (non-) deterministic concurrent synchronisation framework CSync(TGG, CSync) is correct and complete.

References:
Achievement

Incremental synchronization – making it applicable in practice (non-invasive and efficient)

Restriction

Concurrency: no concurrent updates that could cause conflicts

Restriction

Types of TGGs: e.g., deterministic, node creating rules, rule dependencies
### Key Challenges

<table>
<thead>
<tr>
<th>Problem</th>
<th>Challenge</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Diversity</strong>: different concepts and implementations for incremental propagation</td>
<td>Provide a <em>generalized notion</em> of incremental propagation (least change)</td>
</tr>
<tr>
<td><strong>Gap</strong>: between formal results and implementations of incremental synchronisation</td>
<td><strong>Challenge</strong>&lt;br&gt;Extend <em>formal approaches</em> to achieve a close relation to implementations&lt;br&gt;<strong>Challenge</strong>&lt;br&gt;Extend <em>implementations</em> to the <strong>concurrent</strong> case</td>
</tr>
</tbody>
</table>
Conclusion and next steps

SUMMARY

• Concurrent model synchronisation via TGGs
• Guarantees: syntactical correctness, completeness, termination

NEXT STEPS

• Closing the gap between formal theory and implementations by extending both
<table>
<thead>
<tr>
<th>Reference</th>
<th>Authors</th>
<th>Title</th>
<th>Source</th>
</tr>
</thead>
</table>
## Further Reading

<table>
<thead>
<tr>
<th>Reference</th>
<th>Title</th>
<th>Authors/Details</th>
</tr>
</thead>
</table>