Monitoring the Execution of Declarative Model Transformations

R. Groner, S. Gylstorff, M. Tichy
Model of an Electronic Control Unit (170 000 elements)

Transformation Script

Transformation Engine

Transformed Model

12 hours!
We need monitoring to understand why a transformation takes so long
Related Work


We need monitoring to understand why a transformation takes so long

- We implemented monitoring for the declarative transformation language Henshin with Kieker
Henshin

transferMoney(in amount, in fromId, in toId)

**Left-hand side (LHS)**

- `clientFrom:Client`
  - `from:Account`
    - `id=fromId`
    - `credit=from.credit-amount`

- `clientTo:Client`
  - `to:Account`
    - `id=told`
    - `credit=to.credit+amount`

**Right-hand side (RHS)**

- `clientFrom:Client`
  - `from:Account`
    - `id=fromId`
    - `credit=from.credit-amount`
    - `c2:Client`
      - `name="Bob"`
    - `credit=from.credit+amount`

- `clientTo:Client`
  - `to:Account`
    - `id=told`
    - `credit=to.credit+amount`

- `a1:Account`
  - `id=1`
  - `credit=517.93`

- `a2:Account`
  - `id=2`
  - `credit=200.00`

- `a3:Account`
  - `id=3`
  - `credit=1012.63`

- `a4:Account`
  - `id=4`
  - `credit=17.45`
Henshin

transferMoney(in amount=5, in fromId=2, in toId=4)

1. Search plan
   - \{clientFrom, clientTo, from, to\}

2. Domains
   - clientFrom={c1, c2, c3}
   - clientTo={c1, c2, c3}
   - from={a1, a2, a3, a4}
   - to={a1, a2, a3, a4}

3. clientFrom
   - from={a3, a4}

4. clientTo
   - to={a3, a4}

5. from

6. Backtracking

7. clientTo
**Henshin**

- **Function**: `transferMoney(amount=5, fromId=2, toId=4)`

**Diagram Description**:

```
search plan | clientFrom | clientTo | from     | to        
candidates  | {c1,c2,c3} | {c1,c2,c3} | {a1,a2,a3,a4} | {a1,a2,a3,a4} 
```

- `c3` is selected as the clientFrom candidate.
- `c3` and `{a3,a4}` are chosen as the clientTo and from candidates.
- `c2` is selected as the next step in the process.
- `a2` is the final destination.

The diagram illustrates the flow of transactions from the client to the destination, highlighting the selection process and candidate preferences.
transformation ENGINE

```plaintext
transferMoney(in amount=5, in fromId=2, in toId=4)
```

Transformation Engine
Related Work


- Prolog is a declarative programming language

- Prolog uses a similar concept of backtracking during program execution
Relevant Execution Information

- Search plan
- Number of investigated model elements
- Changes in the domains
- Backtracking
- Execution duration
Search Plan

- Q1 How do we receive the order in which the elements of the LHS are chosen to find an isomorphic node in the input model?

```
transferMoney(amount=5, fromId=2, toId=4)
```

<table>
<thead>
<tr>
<th>search plan</th>
<th>clientFrom</th>
<th>clientTo</th>
<th>from</th>
<th>to</th>
</tr>
</thead>
<tbody>
<tr>
<td>candidates</td>
<td>{c1,c2,c3}</td>
<td>{c1,c2,c3}</td>
<td>{a1,a2,a3,a4}</td>
<td>{a1,a2,a3,a4}</td>
</tr>
</tbody>
</table>

NC: Number of Candidates
Number of investigated Model Elements

- **Q2** How do we get the number of model elements examined for each element in the LHS?

![Diagram of search plan and model elements]
Changes in the Domains

- **Q3** How can we monitor how binding decisions of a model element to an element of the LHS affect candidate sets for other LHS elements?

```
transferMoney(amount=5, fromId=2, told=4)
```

<table>
<thead>
<tr>
<th>search plan</th>
<th>clientFrom</th>
<th>clientTo</th>
<th>from</th>
<th>to</th>
</tr>
</thead>
<tbody>
<tr>
<td>candidates</td>
<td>{c1,c2,c3}</td>
<td>{c1,c2,c3}</td>
<td>{a1,a2,a3,a4}</td>
<td>{a1,a2,a3,a4}</td>
</tr>
</tbody>
</table>

S1 \rightarrow c3 \rightarrow E1

S2 \rightarrow c3 \rightarrow E2

S3 \rightarrow a4 \rightarrow a3 \rightarrow E3

S4 \rightarrow c2 \rightarrow E4

S \rightarrow \text{Start}

E \rightarrow \text{End}
Backtracking

- **Q4** How can we monitor where and when backtracking occurs?
Execution Duration

- **Q5** How can we measure how long the transformation execution takes?
Measuring Points

transferringMoney(amount=5, fromId=2, toId=4)

<table>
<thead>
<tr>
<th>search plan</th>
<th>clientFrom</th>
<th>clientTo</th>
<th>from</th>
<th>to</th>
</tr>
</thead>
<tbody>
<tr>
<td>candidates</td>
<td>{c1, c2, c3}</td>
<td>{c1, c2, c3}</td>
<td>{a1, a2, a3, a4}</td>
<td>{a1, a2, a3, a4}</td>
</tr>
</tbody>
</table>
Overhead

- Duration without Monitoring
- Duration with Monitoring

Execution duration [ms] vs Number of Accounts
Overhead

- SE & EE
- NC
- S & E
- BT
Related Work

Usage

- Search Plan

```
+-----------------+-----------------+-----------------+-----------------+
<table>
<thead>
<tr>
<th>clientFrom:Client</th>
<th>clientTo:Client</th>
<th>from:Account</th>
<th>to:Account</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

- Search Plan Progress

Average possible Candidates

- 2: from:Account
  - 1: clientTo:Client
    - 0: clientFrom:Client
      - -1: initial LHS Domain

R. Groner, S. Gylstorff, M. Tichy
Future Work

• Extend our monitoring to support also control structures

• Investigate the monitoring overhead with bigger examples
Summary