On the Realization of TractsTool

Loli Burgueño, Manuel Wimmer, Javier Troya, and Antonio Vallecillo

University of Málaga & Vienna University of Technology
Testing Model Transformations
A Tract defines
- a set of **constraints on the source and target metamodels**, 
- a set of **source-target constraints**, and 
- a **tract test suite** (a collection of source models satisfying the source constraints)
Principles

- Reuse existing languages and techniques of MDE
  - Contract specification
  - Test model generation
- Light-weight, black-box testing approach
  - Partial specification
  - Transformation language & implementation independent
- Support for various transformation kinds
  - Model-to-model
  - Model-to-text
  - Text-to-model
- Open design
  - Several testing scenarios
  - Integration with different transformation tools
TractsTool's Ecosystem

- **EMF (Eclipse Modeling Framework)**
  - Metamodelling (Ecore-based) & modeling support

- **USE Tool (UML-based Specification Environment)**
  - Analyzes model structure (with respect to its metamodel & OCL constraints)

- **OCL (Object Constraint Language)**
  - Allows to define model constraints

- **ASSL (A Snapshot Sequence Language)**
  - Imperative programming language with features for randomly choosing attribute values or association ends
  - Allows to automatically generate models
  - ASSL supports backtracking for finding models with particular properties

- **ATL (ATLAS Transformation Language)**
  - Hybrid language to define model transformations
Tracts for the Families2Persons MT

Source Metamodel: Family
Target Metamodel: Person

FamilyMM

<table>
<thead>
<tr>
<th>Family</th>
<th>Fatherhood</th>
<th>Member</th>
</tr>
</thead>
<tbody>
<tr>
<td>+lastName</td>
<td>String</td>
<td>+firstName</td>
</tr>
<tr>
<td>0..1</td>
<td>1</td>
<td>+age</td>
</tr>
<tr>
<td>+famMother</td>
<td>Motherhood</td>
<td>+gender</td>
</tr>
<tr>
<td>0..1</td>
<td>1</td>
<td>Gender</td>
</tr>
<tr>
<td>+famSon</td>
<td>Sonhood</td>
<td>female</td>
</tr>
<tr>
<td>0..1</td>
<td>0..*</td>
<td>male</td>
</tr>
<tr>
<td>+famDaughter</td>
<td>Daughterhood</td>
<td></td>
</tr>
<tr>
<td>0..1</td>
<td>0..*</td>
<td></td>
</tr>
</tbody>
</table>

PersonMM

<table>
<thead>
<tr>
<th>Person</th>
<th>Parenthood</th>
</tr>
</thead>
<tbody>
<tr>
<td>+parent</td>
<td>0..2</td>
</tr>
<tr>
<td>+name : String</td>
<td></td>
</tr>
<tr>
<td>+age : int</td>
<td></td>
</tr>
<tr>
<td>+child</td>
<td>*</td>
</tr>
</tbody>
</table>

Male

<table>
<thead>
<tr>
<th>+husband</th>
<th>+wife</th>
</tr>
</thead>
<tbody>
<tr>
<td>0..1</td>
<td>0..1</td>
</tr>
</tbody>
</table>

Female

+enumeration

Gender

female
male
Example of Tract: “Members only”

Tract: Members only

Interested in families consisting only of members

Precondition

```
inv SCR_MembersOnly: src_Member.allInstances -> forAll (m | m.famFather->size() + m.famMother->size() + m.famSon->size() + m.famDaughter->size() = 0)
```

Test Source Model

```
<table>
<thead>
<tr>
<th>MembersOnly</th>
</tr>
</thead>
<tbody>
<tr>
<td>m1 : Member</td>
</tr>
<tr>
<td>firstName = &quot;Brigita&quot;</td>
</tr>
<tr>
<td>m2 : Member</td>
</tr>
<tr>
<td>firstName = &quot;Martin&quot;</td>
</tr>
<tr>
<td>m3 : Member</td>
</tr>
<tr>
<td>firstName = &quot;Carmen&quot;</td>
</tr>
<tr>
<td>m4 : Member</td>
</tr>
<tr>
<td>firstName = &quot;Antonio&quot;</td>
</tr>
</tbody>
</table>
```
Example of *Tract*: “Mother2Female”

**Tract: Mother2Female** – a female has to be created from each mother

**Constraint** on the source-target relationship

```plaintext
inv SRC_TRG_Mother2Female:
src_Family.allInstances -> forall (fam|trg_Female.allInstances-> exists(m|
fam.mother.firstName.concat(' ').concat(fam.lastName) = m.name))
```

![Diagram of FamilyMM and PersonMM](image)
Internal Model Transformation Chain (M2M transformations)

**Input Models**
- **Intensional**
  - ASSL program
- **Extensional**
  - EMF models
  - Text artefacts

**Transformation**
- **Intensional**
  - ATL transformation
- **Extensional**
  - EMF models
  - Text artefacts
On the Realization of TractsTool

Input and Output Metamodels

OCL Constraints

Input (or ASSL code) and Output (or ATL trans) Models

Temporal and auxiliary folder

http://atenea.lcc.uma.es/index.php/Main_Page/Resources/Tracts
On the Realization of TractsTool

http://atenea.lcc.uma.es/index.php/Main_Page/Resources/Tracts
On the Realization of TractsTool
context trg_Male inv SRC_TRG_Male2FatherSon:
trg_Male.allInstances->forAll(m|src_Family.allInstances->exists(fam |
fam.father.firstName.concat('-').concat(fam.lastName)=m.name
or fam.sons->any(true).firstName.concat('-').
cconcat(fam.lastName)=m.name))

For all trg_Male, it has to exists (at least) a father or a son whose first name concatenated with ‘-’ and with his family name is like the name of the male
For all trg_Male, it has to exists (at least) a father or a son whose first name concatenated with ‘-’ and with his family name is like the name of the male.

@Male_13.fullName := 'Jim March'
@Male_14.fullName := 'Peter Sailor'
@Female_15.fullName := 'Cindy March'
@Female_16.fullName := 'Jackie Sailor'
@Male_17.fullName := 'Brandon March'
@Male_18.fullName := 'David Sailor'
@Male_19.fullName := 'Dylan Sailor'
Dealing with M2T and T2M Transformations

Testing problem is transformed to M2M testing problem
- Text artefacts are represented as models
- Then TractsTool can be simply reused

public class CalculateArrayAverageExample {
    public static void main(String[] args) {
        //define an array
        int[] numbers = new int[]{10,20,15,25,16,60,100};
        //calculate sum of all array elements
        int sum = 0;
        for(int i=0; i < numbers.length; i++)
            sum = sum + numbers[i];
        //calculate average value
        double average = sum / numbers.length;
    }
}
Next steps

- Generate ASSL code automatically
  - Metamodel Coverage/Transformation Coverage

- Enhance output of TractsTool
  - Text report → diagnostic model

- Allow transformation written in other languages to be tested
The Tracts Team:
Loli Burgueño, Lars Hamann, Martin Gogolla, Antonio Vallecillo, Manuel Wimmer, and Javier Troya
Universidad de Málaga/Universität Bremen/Technische Universität Wien

The Tracts Tool:
http://atenea.lcc.uma.es/index.php/Main_Page/Resources/Tracts

More information on Tracts:

Acknowledgements:
This work has been supported by Universidad de Málaga, Campus de Excelencia Internacional Andalucía Tech