Families to Persons Case with UML-RSDS

K. Lano,

Dept. of Informatics, King's College London

S. Kolahdouz-Rahimi,

Dept. of Software Engineering, University of Isfahan, Iran

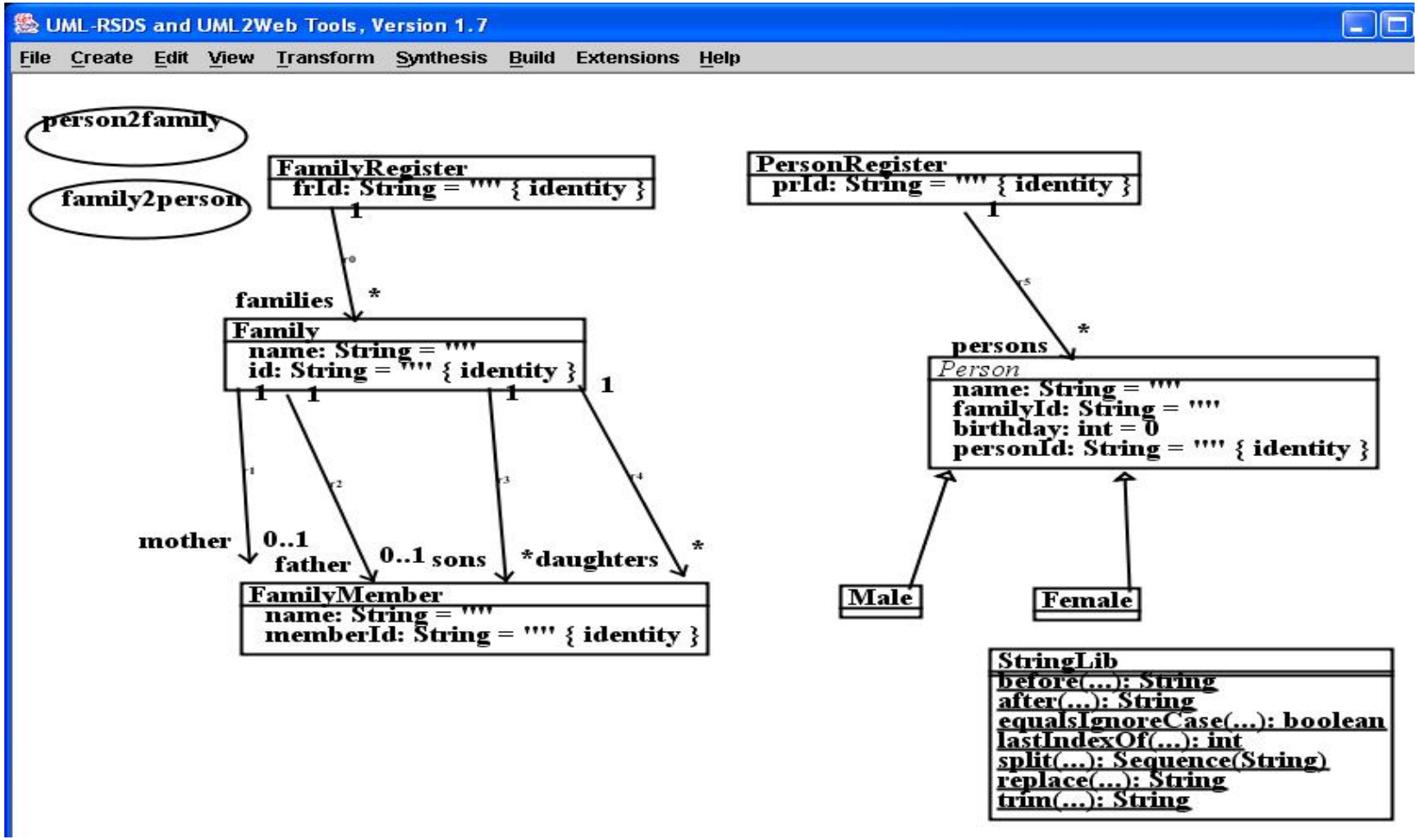
June 22, 2017

Approach

- Specification of bx using OCL
- Forward and reverse transformations derived from bx relation R
- Executable transformation code synthesised from transformation specifications.

Forward transformation

- forward transformation is defined by use case *person2family*
- inverse is defined by *family2person*.
- unique keys *personId*, *memberId* record 1-1 correspondence between persons and family members.



Metamodels of Families to Persons case

Bx relation

Formed of 4 invariants, $(I1)$:

$$
\begin{aligned}
&Family \rightarrow forAll(fam \mid \\
&\quad FamilyMember \rightarrow forAll(m \mid \\
&\qquad m \in fam.mother \rightarrow union(fam.daughters) \Rightarrow \\
&\qquad Female \rightarrow exists(f \mid f.personId = m.memberId \ \& \\
&\qquad\quad f.familyId = fam.id \ \& \\
&\qquad\quad f.name = fam.name + ", " + m.name)))
\end{aligned}
$$

$(I2)$:

$$
\begin{aligned}
&Family \rightarrow forAll(fam \mid \\
&\quad FamilyMember \rightarrow forAll(m \mid \\
&\qquad m \in fam.father \rightarrow union(fam.sons) \Rightarrow \\
&\qquad Male \rightarrow exists(f \mid f.personId = m.memberId \ \& \\
&\qquad\quad f.familyId = fam.id \ \& \\
&\qquad\quad f.name = fam.name + ", " + m.name)))
\end{aligned}
$$

Express that families model is consistent wrt persons model.

($I3$):

$$Female \rightarrow forAll(f \mid FamilyMember \rightarrow exists(m \mid m.memberId = f.personId \;\&\; Family \rightarrow exists(fam \mid fam.id = f.familyId \;\&\; m \in fam.mother \rightarrow union(fam.daughters) \;\&\; f.name = fam.name + ", " + m.name)))$$

($I4$):

$$Male \rightarrow forAll(f \mid FamilyMember \rightarrow exists(m \mid m.memberId = f.personId \;\&\; Family \rightarrow exists(fam \mid fam.id = f.familyId \;\&\; m \in fam.father \rightarrow union(fam.sons) \;\&\; f.name = fam.name + ", " + m.name)))$$

Express that persons model is consistent wrt families model.

I3, I4 are logical duals of I1, I2.

family2person transformation (from I1 and I2):

```
Family::
 m : mother->union(daughters) =>
    Female->exists( f | f.personId = m.memberId &
      f.familyId = id &
      f.name = name + ", " + m.name )

Family::
 m : father->union(sons) =>
    Male->exists( f | f.personId = m.memberId &
      f.familyId = id &
      f.name = name + ", " + m.name )
```

person2family:

Logically strengthen (I3) and (I4), enforce that persons are mapped to parents preferentially:

```
Female::
 FamilyMember->exists( m | m.memberId = personId &
   Family->exists( fam | fam.id = familyId &
     (fam.mother@pre.size = 0 =>
        m : fam.mother & m /: fam.daughters) &
     (fam.mother@pre.size > 0 & fam.mother@pre->excludes(m) =>
        m : fam.daughters) &
     fam.name = StringLib.before(name, ", ") &
     m.name = StringLib.after(name, ", ") ) )
```

```
Male::
 FamilyMember->exists( m | m.memberId = personId &
   Family->exists( fam | fam.id = familyId &
     (fam.father@pre.size = 0 =>
        m : fam.father & m /: fam.sons) &
     (fam.father@pre.size > 0 & fam.father@pre->excludes(m) =>
        m : fam.sons) &
     fam.name = StringLib.before(name, ", ") &
     m.name = StringLib.after(name, ", ") ) )
```

Change propagation

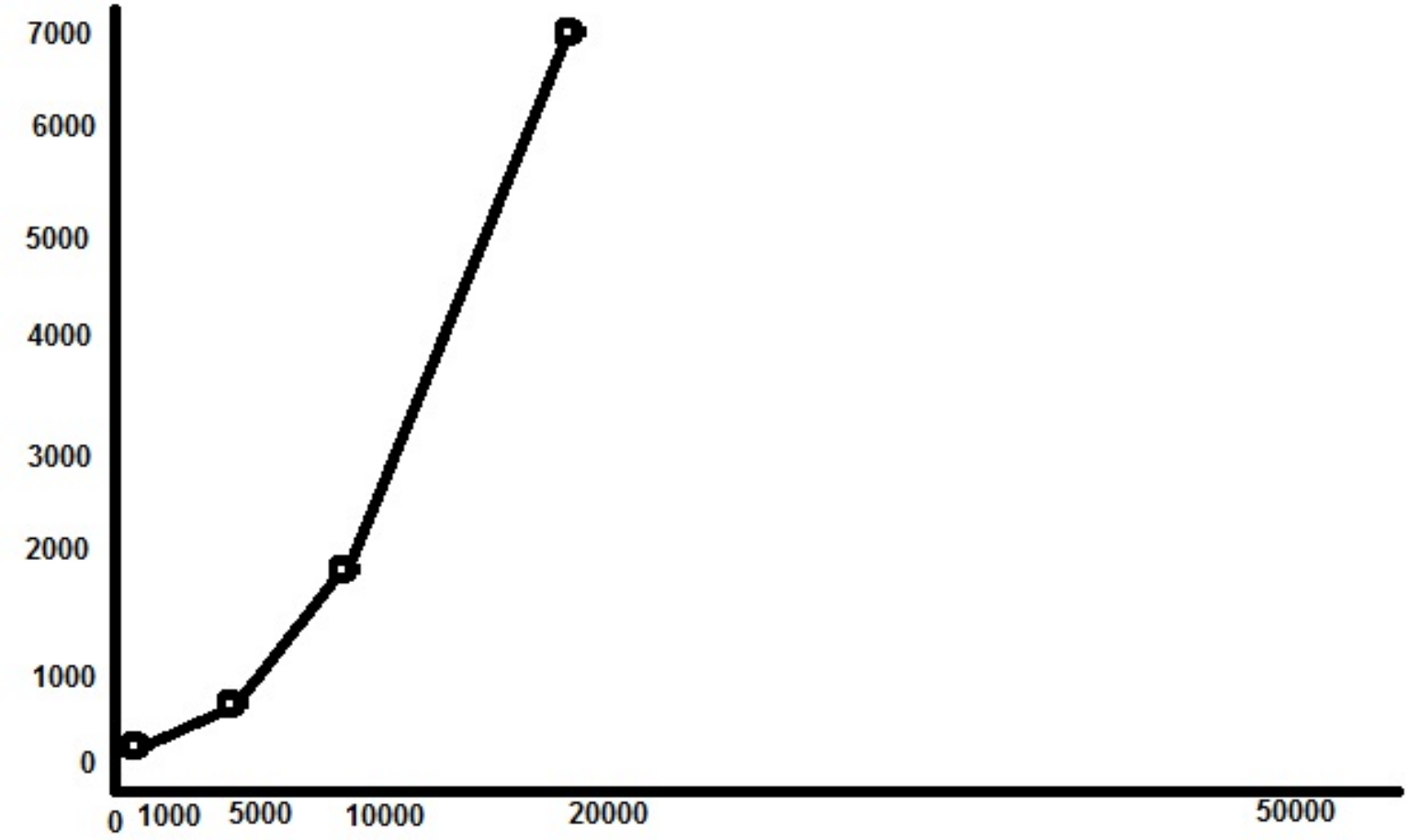
family model change	*person model change*
New *FamilyMember*	new *Male* or *Female*
New empty *Family*	no change
Changed *Family* :: *name*	changed *name* for each person from the family
Changed *FamilyMember* :: *name*	changed *name* for corresponding person
Move a member from *father* to *sons* in family	no change

person model change	*family model change*
New *Person*	new *FamilyMember*, possibly new *Family*
Changed *Person* :: *familyId*	moves corresponding member to new or modified *Family*
Changed *Person* :: *name*	changes *name* of corresponding member and possibly of its family
Changed *Person* :: *birthday*	no change

Evaluation

Test	*Description*	*Execution time*
1	Changed Person name, familyId	10ms
2	Changed Person birthday	10ms
3	New Persons (10000)	35s
4	Changed Family name	10ms
5	Changed FamilyMember name	10ms
6	New Family	10ms
7	New FamilyMember (50000)	27s

Invariants (I1) to (I4) are established in each case.



Execution time of test7

Conclusions

- Solution is concise (30 LOC)
- Declarative, close to logical statement of problem
- Incremental change propagation for attribute, creation, move updates
- Efficient up to 50,000 model elements.