Transformation Tool Contest 2016
Live case study

Louis Rose
Filip Krikava
Antonio Garcia-Dominguez (@antoniogado)

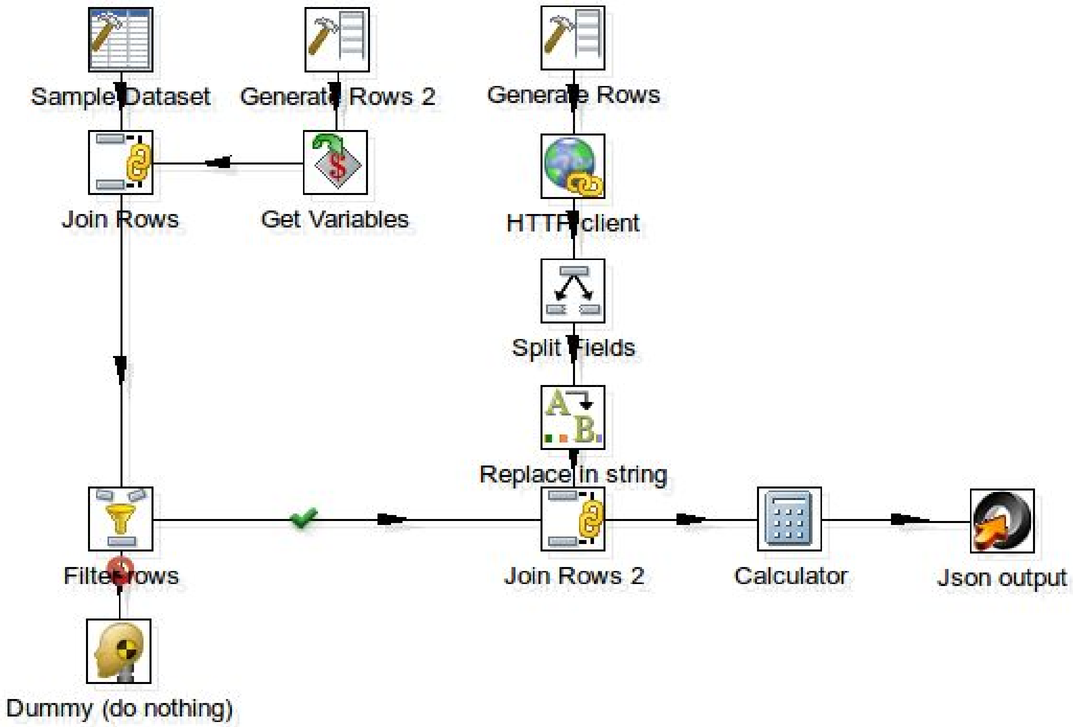
@ttc_workshop
http://www.transformation-tool-contest.eu/

Motivation

- With traditional (“batch”) transformations, if the input model changes we have to transform it again
- Incremental transformations only process the change - much faster for large models
- Studied in literature, but not simple to implement:
  - ReactiveATL - turn ATL rules into event-based system
  - VIATRA - event-driven framework (requires coding)
- Could there be a simpler approach, e.g. for porting?

Idea

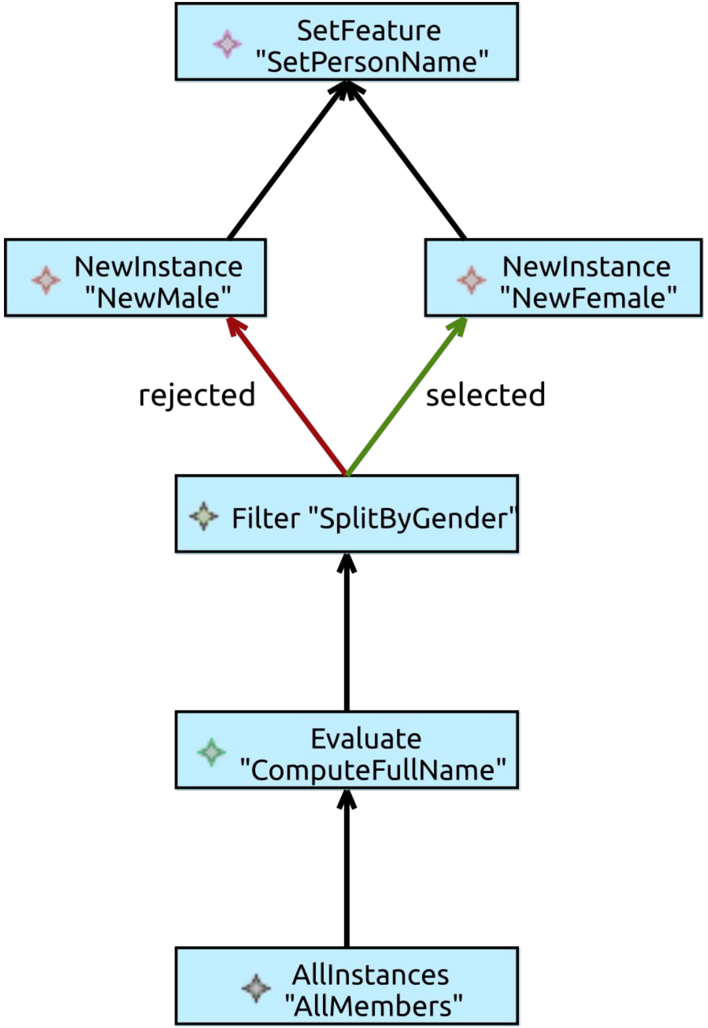
- Borrow from data integration tools (e.g. Pentaho)
- They reshape large volumes of complex data
- Graph of row streams



Mapping to MDE

- Replace data integration primitives with model-specific primitives
- Embed a small expression language

```
AllInstances AllMembers {
  field member
  type Families!Member
  target ComputeFullName
}
Evaluate ComputeFullName {
  field fullName
  expression member.firstName + ’ ’ +
    member.eContainer.lastName
  target SplitByGender
}
....
```



Goal of the contest

- Four transformations are given in this notation:
  - Tree2Graph
  - Families2Persons
  - Class2RDB
  - Flowchart2HTML
- Available as Xtext DSL + Sirius diagrams originally
  - XMI representation added later
- Contestants need to implement an execution engine
- Reference solution written in EOL (interpreter)

Tasks

- Task 1 is running Tree2Graph and Families2Persons, requires:
  - All Instances, New Instance, Filter, Evaluate, Set Feature
  - Embedded expressions: +, or, string literals, field refs, property refs, .eContainer
- Task 2 is running Class2RDB and Flowchart2HTML, requires:
  - Task 1 + ForEach, AddToContainer
  - Embedded expressions: =, .eClass, .eClass.name
- Evaluation:
  - Batch/incremental correctness (did it run successfully?)
  - Batch/incremental completeness (did it produce the right results?)
  - Batch/incremental performance (is it faster than the others?)
  - Maintainability and extensibility
  - Incremental version: make a change, update model (preferred over batch)
- Manual small model + synthetic models generated with EMG

Submitted solutions

- Sent as PRs to https://github.com/bluezio/ttc2016-live
- Interesting mix of technologies! In order of arrival:
  - NMF (Hinkel) - C#
  - Mofongo (Hoyos) - Python
  - ATL (Jouault, Wimmer) - Java
  - SIGMA (Krikava) - Scala
- Solutions limited to batch scenario :-(

Performance results (seconds, synthetic model)

- Running Thinkpad T450 laptop, i7-5600U @ 2.6GHz
- Ran Mofongo, ATL and SIGMA using GNU/Linux 4.4.0 + Ubuntu 16.04
- Ran NMF using Windows 10 (latest update, 8 July 2016)

	T2G (1.4MB)	F2P (503KB)	C2D (1.4MB)	F2H (503KB)
Epsilon (ref.)	18.65	10.24	37.08	6.72
NMF	(“children”)	2.53 (1.57 g/c)	53.30 (1.88 g/c)	N/A
Mofongo	(opposite refs)	5.99	N/A	N/A
ATL	163.99	19.79	N/A	N/A
SIGMA	2.36	1.22	N/A	N/A