# Yet Another Three QVT Languages

Edward Willink, Willink Transformations Ltd

Horacio Hoyos, The University of York
Dimitris Kolovos, The University of York

ICMT 2013, Budapest

18th June 2013

# QVT History

- Models are good

- Model Transformation must be good

- OMG: Need a standard

  - 2002 RFP: Request for Proposal

- QVT: Query/View/Transformation standard

  - 2003 Competing Proposals        -      8 then 7
  - 2004,5 Merged Proposal          =>  3 languages
  - 2007 QVT 1.0 Standard
  - 2011 QVT 1.1 Standard             -      minor fixes

# QVTo - QVT Operational Mappings

Operational/Imperative paradigm

- SmartQVT from France Telecom

    - mature since 2008

- Eclipse QVT Operation Project (from Borland)

    - stagnated in 2009

    - moving again since 2012

        - 48 fixes in Kepler (June 26 2013)

# QVTr - QVT Relational

Powerful, multi-directional Declarative paradigm

- Medini QVT (from ikv++ technologies)
  - mature since 2011
  - disappointing performance

- ModelMorf (from Tata)
  - only Beta releases in public domain

- Eclipse QVT Declarative Project
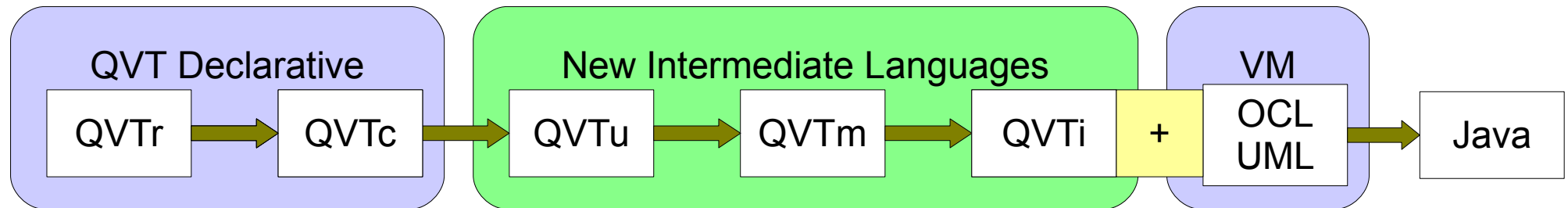  - models, parsers, editors
  - no execution

# QVTc - QVT Core

Simpler, multi-directional Declarative paradigm

- Eclipse QVT Declarative Project
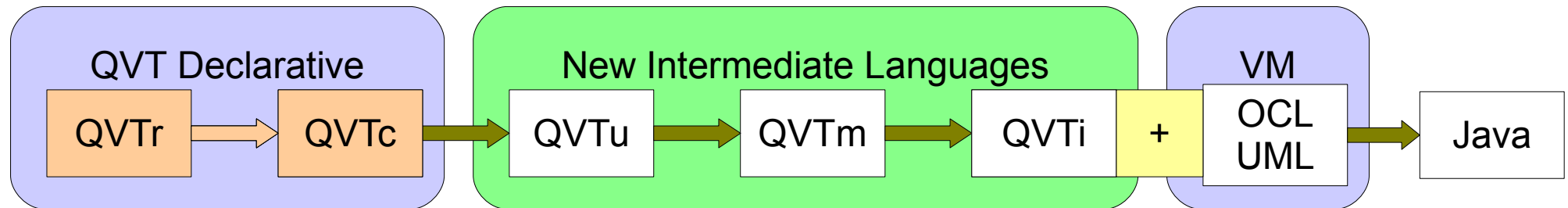  - models, parsers, editors
  - no execution

# Three more QVTs

| QVT Declarative | | New Intermediate Languages | | | | VM | |
|---|---|---|---|---|---|---|---|
| QVTr → | QVTc → | QVTu → | QVTm → | QVTi | + | OCL UML → | Java |

Direct QVTr implementation too hard

- Break problem into smaller parts

- Progressive program-to-program transformation


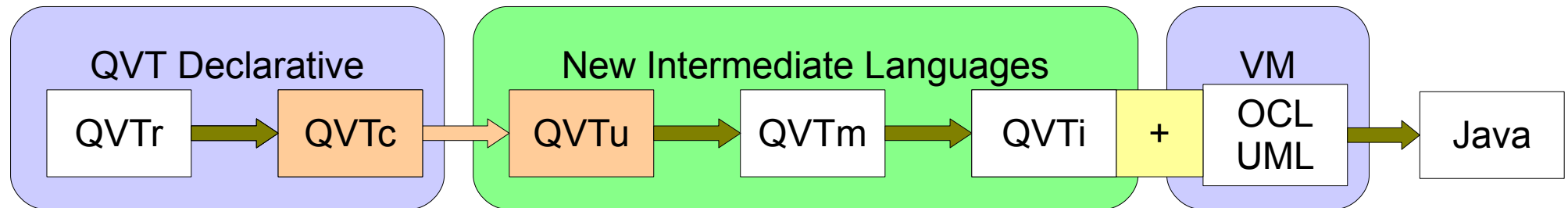- Exploit Eclipse OCL VM and Java code generator

# QVTr to QVTc

| QVT Declarative | New Intermediate Languages | VM | |
|---|---|---|---|
| QVTr → QVTc → | QVTu → QVTm → QVTi + | OCL UML → | Java |

'Provided' by the QVT specification

- single pass, hard to read

- many syntax errors, never exercised

- requires a working QVTr tool to convert QVTr

- re-implement in QVTc

# QVTc to QVTu

**QVT Declarative** — QVTr → QVTc

**New Intermediate Languages** — QVTu → QVTm → QVTi
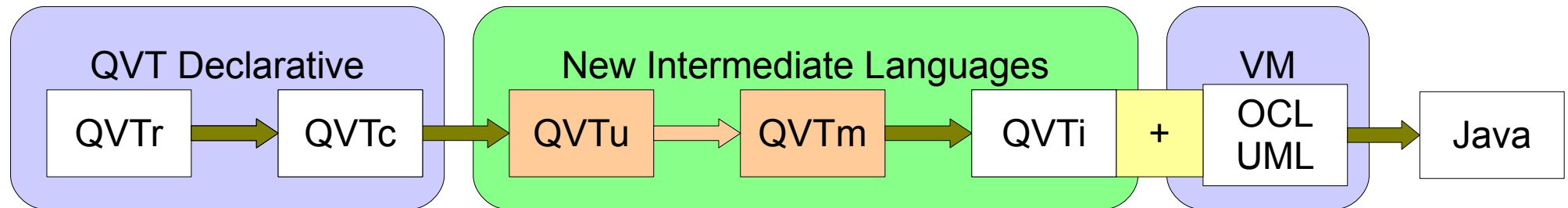
**+**

**VM** — OCL UML → Java

Multi-directional powerful for specification

- powerful specification capability

- unhelpful implementation of actual user requirement

- QVTu - Unidirectional subset of QVTc

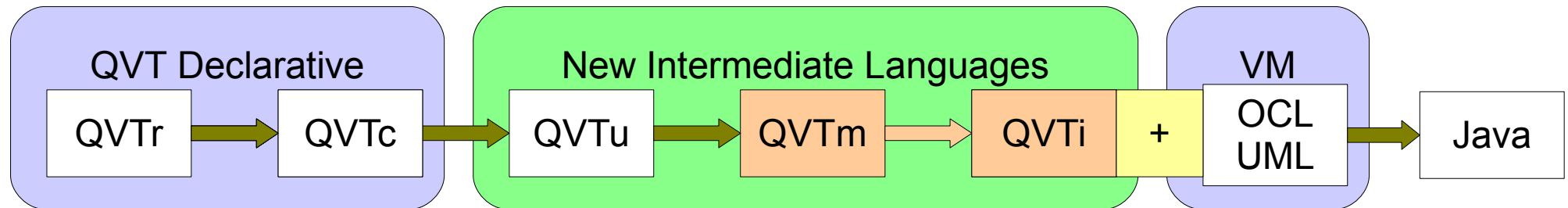- QVTc to QVTu - discard multi-directional bloat

# QVTu to QVTm



Normal forms more powerful for optimization

- powerful specification capability

- unhelpful implementation of actual user requirement

- QVTm - Minimal declarative subset of QVTc

- QVTu to QVTm - discard syntax sugar (eg refinement)

# QVTm to QVTi



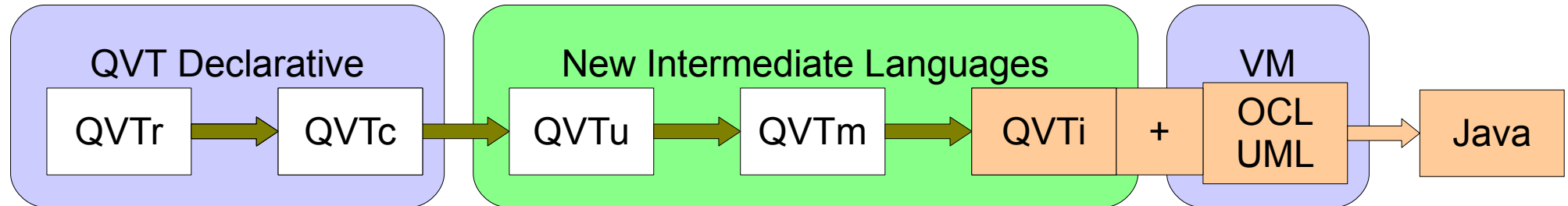| QVT Declarative | | New Intermediate Languages | | | VM | |
|---|---|---|---|---|---|---|
| QVTr → | QVTc → | QVTu → | QVTm → | QVTi + | OCL UML → | Java |

Declarative semantics inappropriate for implementation

- introduce imperative search schedule

- exploit metamodel


- QVTi - Imperative variant subset of QVTc
- QVTm to QVTi - synthesize an execution schedule

# QVTi + OCL to Java

| QVT Declarative | | New Intermediate Languages | | | VM | |
|---|---|---|---|---|---|---|
| QVTr → QVTc | | QVTu → QVTm → | QVTi | + | OCL UML | → Java |

## OCL

- powerful side-effect free model evaluation

## QVTi

- disciplined model mutation, schedule

- reuse OCL interpreter (and debugger)
- reuse OCL to Java code generator

# QVTi enhancements

QVT Declarative (Core/Relational)
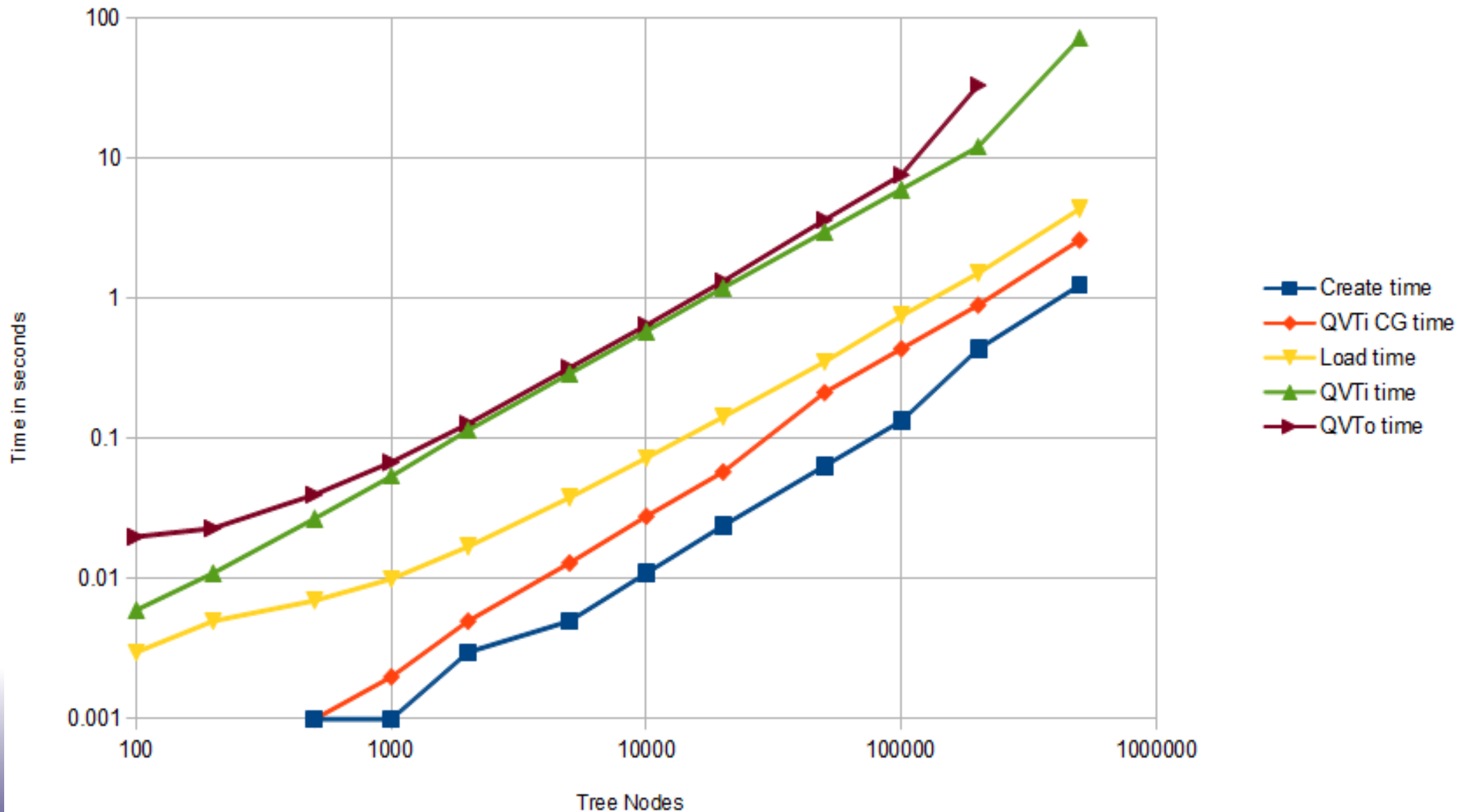
- Declarative rule matching

QVT Imperative

- Explicit Mapping Call with bindings

```
map NestedMapping {
    guardPatternVariable1 <= collectionOfBindings;
    guardPatternVariable2 := explicitBinding;
}
```

# QVTi Preliminary Performance
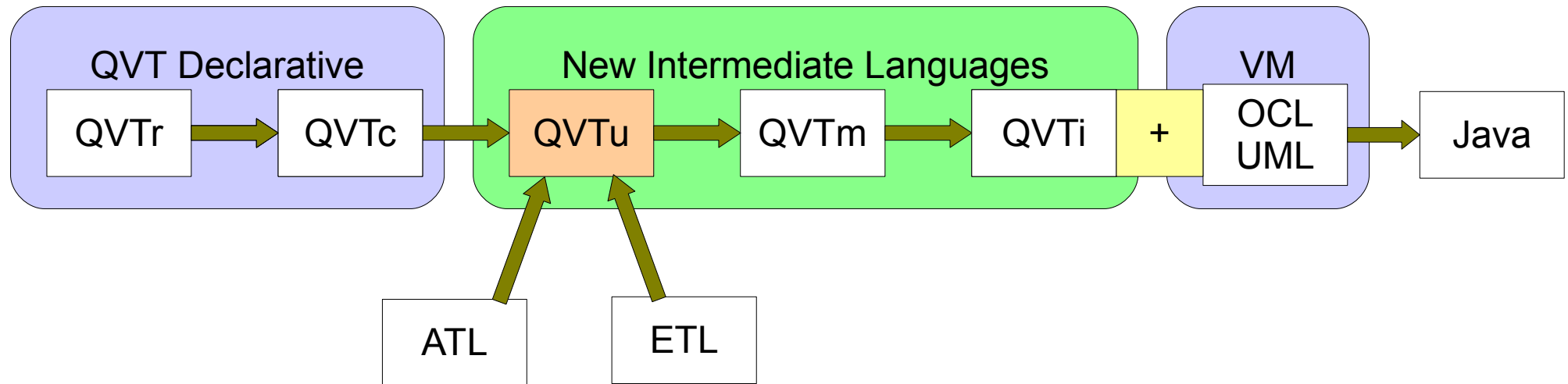


Tree Create/Load/Transform Speeds

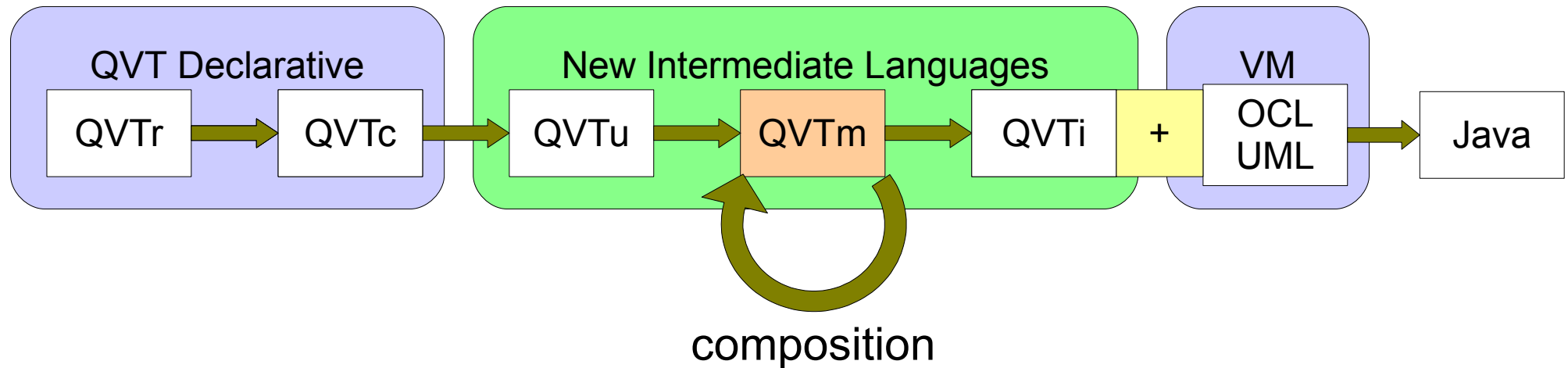# Demonstration / Poster

Eclipse QVT Declarative Project

- Kepler release (June 26 2013)
- QVTr Xtext editor/parser
- QVTc Xtext editor/parser
- QVTi Xtext editor/parser
- QVTr -> QVTc -> QVTm -> QVTi (Luna)
- QVTi interpreter
- QVTi to Java code generator
- QVTi debugger (Luna)

# QVTu Opportunities



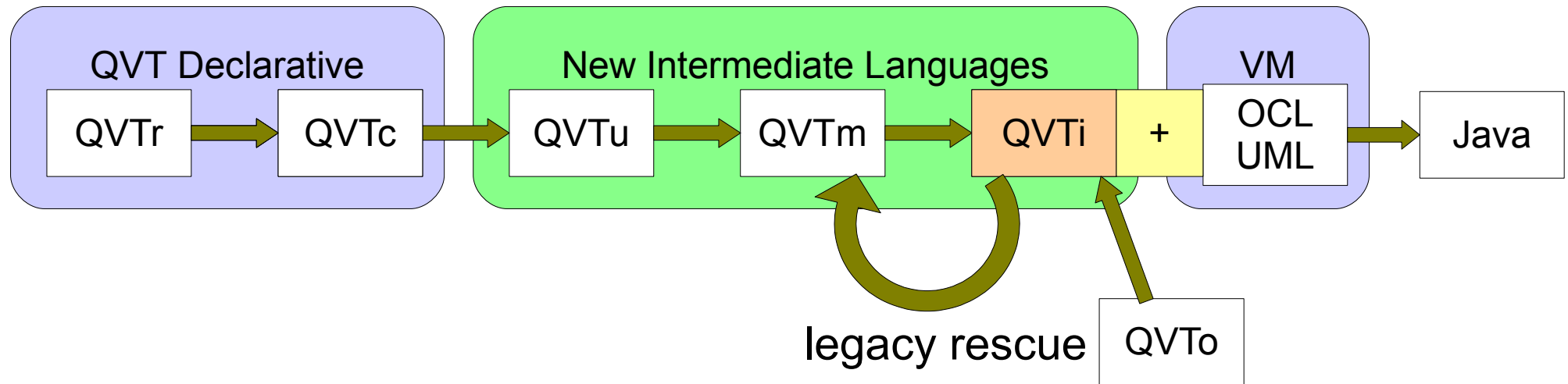Interchange point for alternative declarative syntaxes

# QVTm Opportunities

| QVT Declarative | | New Intermediate Languages | | | VM | |
|---|---|---|---|---|---|---|
| QVTr → | QVTc → | QVTu → | QVTm → | QVTi | + | OCL UML → | Java |

composition

## Normalized form

- composition of multiple transformations
  - variety of declarative languages
- optimization of the composition
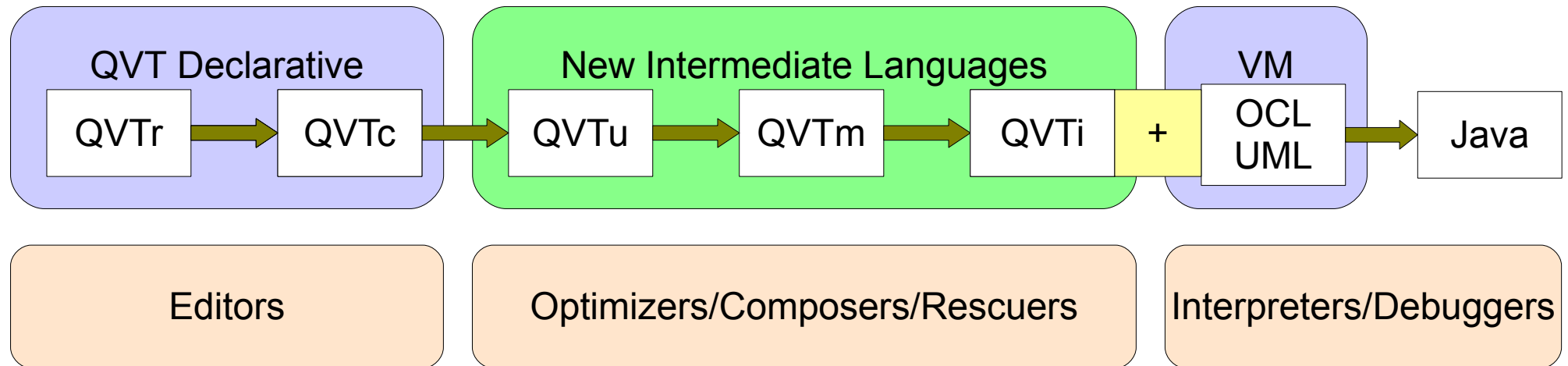
# QVTi Opportunities



Interchange point for alterative imperative syntaxes

Legacy rescue of imperative transformations

- 🟧 exploit composition of declarative transformations

# Shared tool chain



New transformation language

- leverage interpreter/debugger/optimizers/composers

Better debugger

- reuseable by many languages

# Summary

- Eclipse QVT Declarative moving at last
  - QVTc/QVTr editors in Helios (2010)
  - QVTi execution in Kepler (2013)
  - QVTi code generation in Kepler (2013)
    - 20 times speed up
  - QVTr execution targeted for Luna (2014)
  - QVTi/c/r debugger targeted for Luna (2014)

- Eclipse QVT Operational moving again
  - Xtext editor targeted for Luna (2014)
  - alignment with QVTd and OCL projects