



# Overcoming the Difficulties of Implementing a QVT Execution Solution

Open Canarias, S.L.

ETSI Informática  
University of La Laguna



E. Victor Sánchez  
Víctor Roldán  
Adolfo Sánchez-Barbudo  
Orlando Avila-García

[research@opencanarias.com](mailto:research@opencanarias.com)

# Outline

- Implementing QVT: Problems
- Current QVT Solutions
- Our Solution: ATC-Based
- Our Support to QVT
  - Editors
  - Translation into ATC
  - Execution
- Overcoming Challenges
- EPL Contribution

# Outline

- **Implementing QVT: Problems**

- Current QVT Solutions
- Our Solution: ATC-Based
- Our Support to QVT
  - Editors
  - Translation into ATC
  - Execution
- Overcoming Challenges
- EPL Contribution

# Implementing QVT: Problems I

- Specification problems (semantic and syntactic):
  - Inconsistencies
  - Ambiguities
  - Insufficiencies
  - Lack of Reference Implementation
- Implementation Challenges:
  - Mixture Of Paradigms
  - Language Dependencies
  - Complexity
  - Maintenance
  - General vs. agile DSTLs

# Implementing QVT: Problems II

- Abstract Syntax and Semantics Superclasses mismatch in the QVT Operational Mappings Language
- QVToStdLib as a model instance of the QVTo language
- Element as instance of MOF::Element in the QVToStdLib (Sect. 8.3.1)
- Type Inconsistencies & Mind Breaker Considerations

# Outline

- Implementing QVT: Problems

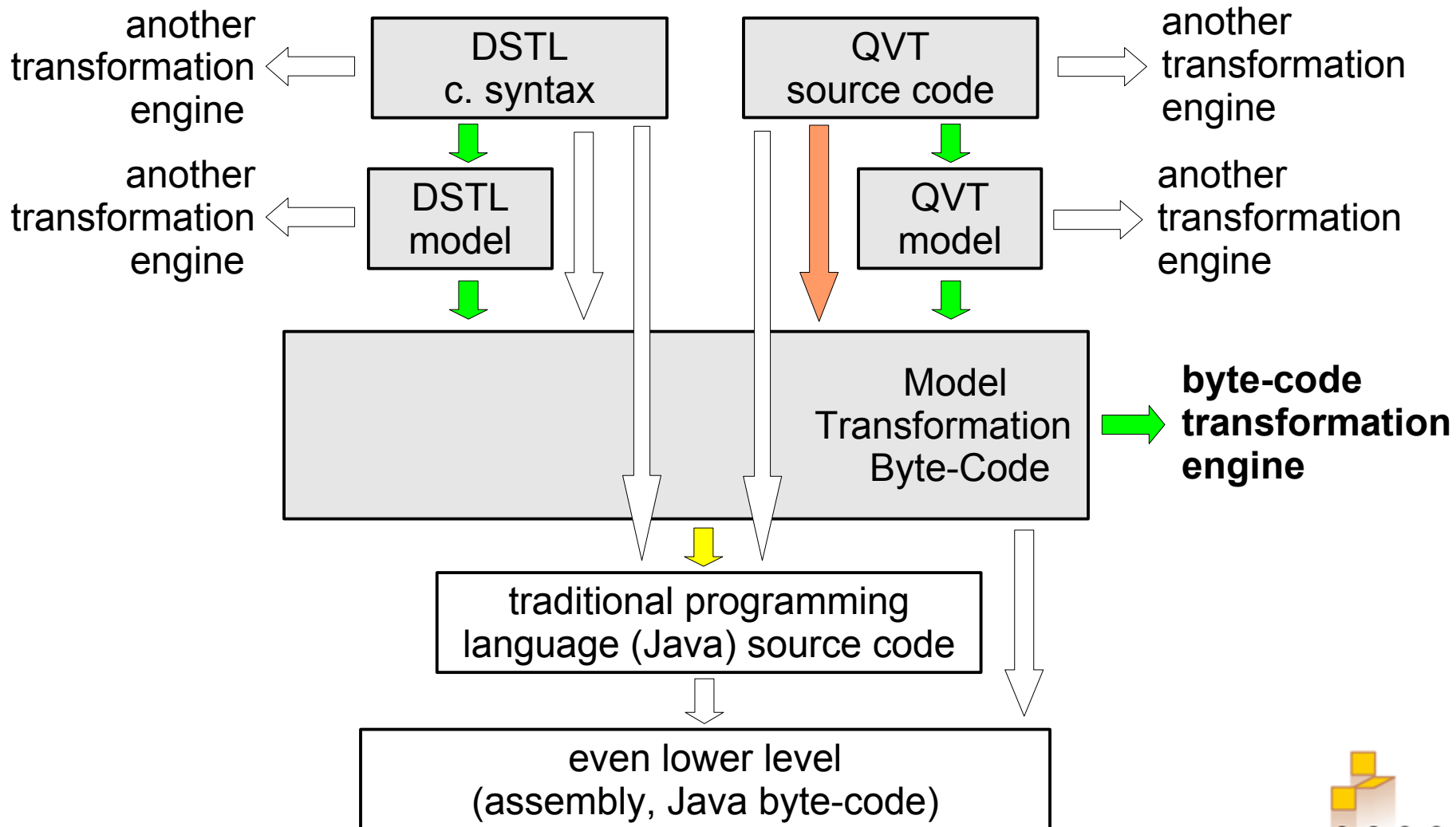
- **Current QVT Solutions**

- Our Solution: ATC-Based
- Our Support to QVT
  - Editors
  - Translation into ATC
  - Execution
- Overcoming Challenges
- EPL Contribution

# Current QVT Solutions (I)

- The Solutions:
  - growing and improving
- Based On:
  - Code Interpreters
  - Code Translators (compilers, e.g: into java)
  - Byte-Code Virtual Machine Approaches

# Current QVT Solutions (II)





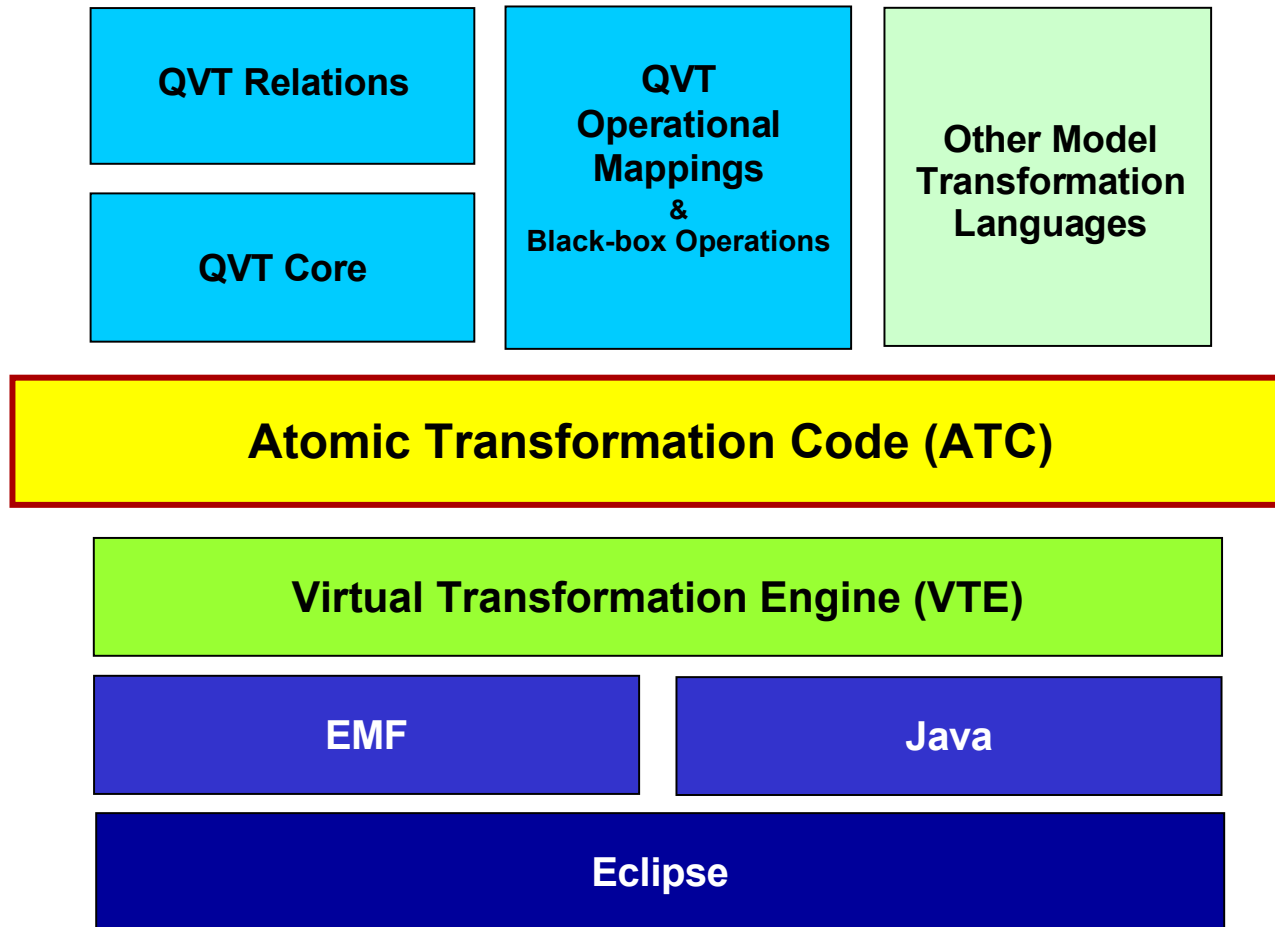
# Outline

- Implementing QVT: Problems
- Current QVT Solutions
- **Our Solution: ATC-Based**
- Our Support to QVT
  - Editors
  - Translation into ATC
  - Execution
- Overcoming Challenges
- EPL Contribution

# Our Solution: ATC (I)

- Atomic Transformation Code (ATC):
  - Low-level Model Transformation Language
  - Set of small Transformation Primitives (Byte-Code)
  - Imperative
  - Lightweight Thin Layer Built Upon EMF
  - Different Abstraction Layers Available (Generative ATC)
- Current Uses:
  - Model Transformation Development (ATC, QVT)
  - Modernization (ADM, KDM...)
  - DSTL support (MTTL, GATC...)

# Our Solution: ATC (II)



# Outline

- Implementing QVT: Problems
- Current QVT Solutions
- Our Solution: ATC-Based

- **Our Support to QVT**

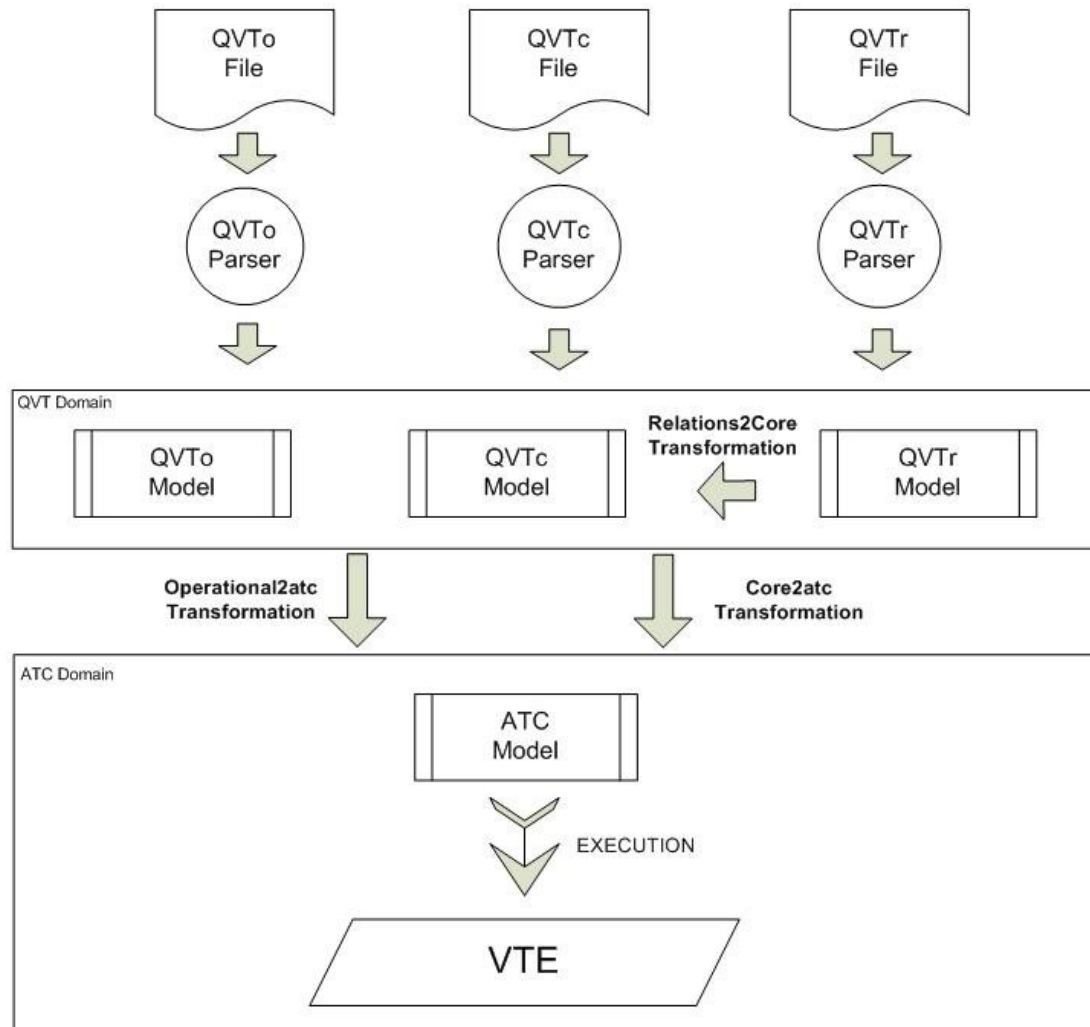
- **Editors**
- **Translation into ATC**
- **Execution**

- Overcoming Challenges
- EPL Contribution

# Our Support to QVT (I)

- 3 Repeatable Independent Steps:
  1. Edition + Parsing of QVT concrete syntax notation  
Currently hosted at the UMLX project (QVTo pending?)
  2. Transform edition + parsing output QVT models into ATC
  3. Ready to Execute the  
Translated QVT Model Transformation!
- Transparency to the User is Required

# Our Support to QVT (II)

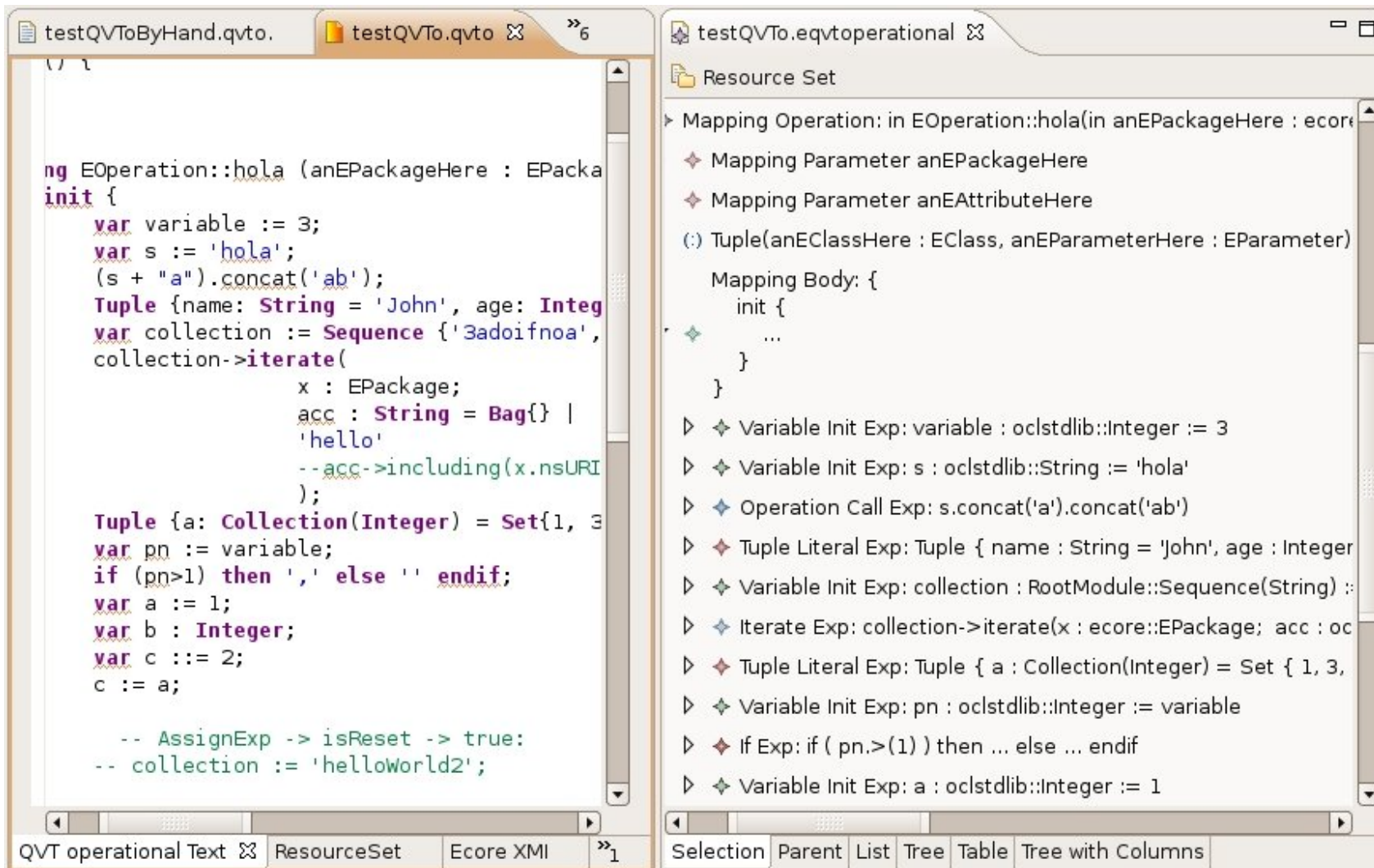


© Copyright 2008 Open Canarias S.L.

# Support to QVT: Editors (I)

- Hosted at UMLX:
  - Dr. Edward D. Willink
    - Relations (QVTr)
    - Core (QVTc)
  - Adolfo Sánchez-Barbudo
    - Operational Mappings (QVTo)
- Multipage Editors
  - Output Models conforming to Ecore and EMOF
  - XMI-executable
- Reuse of the Eclipse MDT OCL parser

# Support to QVT: Editors (II)



The screenshot shows the Eclipse IDE with two QVT editors open. The left editor, 'testQVToByHand.qvto', displays a QVT transformation script. The right editor, 'testQVTo.eqvtooperational', displays the operational mapping for the same transformation.

**testQVToByHand.qvto:**

```

ng EOperation::hola (anEPackageHere : EPacka
init {
    var variable := 3;
    var s := 'hola';
    (s + "a").concat('ab');
    Tuple {name: String = 'John', age: Integer
    var collection := Sequence {'3adoifnoa',
    collection->iterate(
        x : EPackage;
        acc : String = Bag{} |
        'hello'
        --acc->including(x.nsURI
    );
    Tuple {a: Collection(Integer) = Set{1, 3
    var pn := variable;
    if (pn>1) then ',' else '' endif;
    var a := 1;
    var b : Integer;
    var c ::= 2;
    c := a;

    -- AssignExp -> isReset -> true:
    -- collection := 'helloWorld2';
  
```

**testQVTo.eqvtooperational:**

Resource Set

- Mapping Operation: in EOperation::hola(in anEPackageHere : ecore
  - Mapping Parameter anEPackageHere
  - Mapping Parameter anEAttributeHere
  - (:) Tuple(anEClassHere : EClass, anEParameterHere : EParameter)
    - Mapping Body: {
      - init {
        - ...

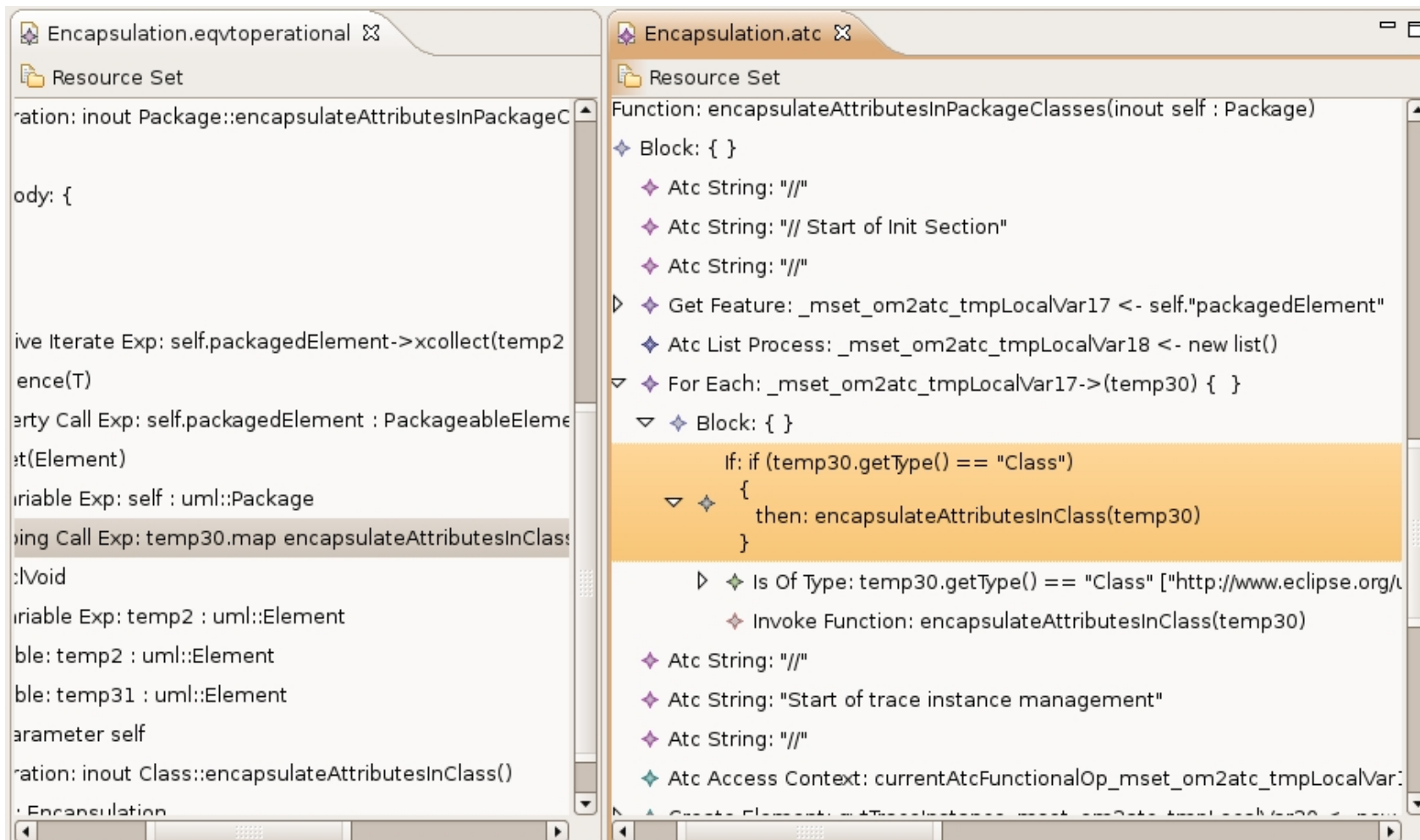
Selection Parent List Tree Table Tree with Columns



# Support to QVT: Translation (I)

- Models are Translated into ATC
  - Via Model (to Model) Transformations
  - Semantically equivalent (of course!)
  - Translation comprises:
    - Victor Sánchez:
      - OCL + MOF (Ecore) + OCLStdLib
      - Operational Mappings (QVTo) + StdLib
        - Includes Trace Infrastructure
    - Víctor Roldán: Core (QVTc) + OCLStdLib
- Written in GATC
- Support for QVT Relations: Indirect
  - Via the Relations to Core transformation

# Support to QVT: Translation (II)



The screenshot displays two Eclipse IDE windows side-by-side, illustrating the translation of a QVT operation from EQV to ATC.

**Left Window: Encapsulation.eqvtoperational**

Resource Set

```

ration: inout Package::encapsulateAttributesInPackageC
ody: {

ive Iterate Exp: self.packagedElement->xcollect(temp2
ence(T)
erty Call Exp: self.packagedElement : PackageableElemen
st(Element)
riable Exp: self : uml::Package
ing Call Exp: temp30.map encapsulateAttributesInClass
:Void
riable Exp: temp2 : uml::Element
ble: temp2 : uml::Element
ble: temp31 : uml::Element
arameter self
ration: inout Class::encapsulateAttributesInClass()
: Encapsulation
  
```

**Right Window: Encapsulation.atc**

Resource Set

```

Function: encapsulateAttributesInPackageClasses(inout self : Package)
◆ Block: { }
◆ Atc String: "/"
◆ Atc String: "// Start of Init Section"
◆ Atc String: "/"
▷ ◆ Get Feature: _mset_om2atc_tmpLocalVar17 <- self."packagedElement"
◆ Atc List Process: _mset_om2atc_tmpLocalVar18 <- new list()
▽ ◆ For Each: _mset_om2atc_tmpLocalVar17->(temp30) { }
  ▽ ◆ Block: { }
    If: if (temp30.getType() == "Class")
      {
        ▽ ◆ {
          then: encapsulateAttributesInClass(temp30)
        }
      }
    ▷ ◆ Is Of Type: temp30.getType() == "Class" ["http://www.eclipse.org/uml2/1.0.0/uml2.xmi"]
      ◆ Invoke Function: encapsulateAttributesInClass(temp30)
  ◆ Atc String: "/"
  ◆ Atc String: "Start of trace instance management"
  ◆ Atc String: "/"
  ◆ Atc Access Context: currentAtcFunctionalOp_mset_om2atc_tmpLocalVar18
  ◆ Create Element: atcTempLocalVar18_mset_om2atc_tmpLocalVar18 <- new
  
```

# Support to QVT: Execution

Name: Encapsulation

**Model Extents** **Common**

Transformation Source

Transformation URI:

Transformation Engine:

Model Extents

Model Extent: model

Direction:

Conformance:

URI Location:

☐ Transient

Model Extent: QVTTraceModelParameter\_mset\_om2atc\_tmpLocalVar1

Direction:

Conformance:

URI Location:

# Outline

- Implementing QVT: Problems
- Current QVT Solutions
- Our Solution: ATC-Based
- Our Support to QVT
  - Editors
  - Translation into ATC
  - Execution
- **Overcoming Challenges**
- EPL Contribution

# Overcoming Challenges

- Syntactic and Semantic Ambiguities
  - Issues to MOF 2.0 QVT Task Force
- OCL language extensibility
  - Several Open Bugs in the Eclipse Bugzilla
  - It also Affects the QVT 2 ATC model transformations!
- ATC Alignment with Source Languages
  - Declarative vs. Imperative
- Trace Support
  - Relying on Unique Trace Metamodel
  - Trace Infrastructure Injection

# Outline

- Implementing QVT: Problems
- Current QVT Solutions
- Our Solution: ATC-Based
- Our Support to QVT
  - Editors
  - Translation into ATC
  - Execution
- Overcoming Challenges
- **EPL Contribution**

# EPL Contribution

- ATC and GATC
- Operational Mappings Editor and Parser
- QVT2ATC model Transformations
  - Includes OCL2ATC!
- Everything under the EPL!!! (Coming soon)

[www.opencanarias.com](http://www.opencanarias.com)

# Thank you very much!!!



FIT-340000-2007-162  
PTQ2004-1495  
PTQ05-02-03062  
IDT-TF-07/013

- Víctor Roldán: **Core\_2\_ATC** model transformation
- Adolfo Sánchez-Barbudo: **Operational Mappings** (OM) parser & editor
- Victor Sánchez: **ATC** language & **OM\_2\_ATC** model transformation

## Special Thanks:

- Nuria Verde: **MOF2Text** parser & editor
- Orlando Avila-García: **MTTL** implementor, user feedback & insight
- Christian Damus: **OCL** project & personal patience
- Ed Merks: **EMF** project & personal patience
- Ed Willink: **UMLX QVT** editors & overall kindness

and

- **Eclipse & EclipseCON**