

QVT technologies and Eclipse

**Tony Clark¹, Tracy Gardner², Catherine
Griffin³, Laurence Tratt⁴**

¹ `anclark@dcs.kcl.ac.uk` King's College London

² `tgardner@uk.ibm.com` IBM UK

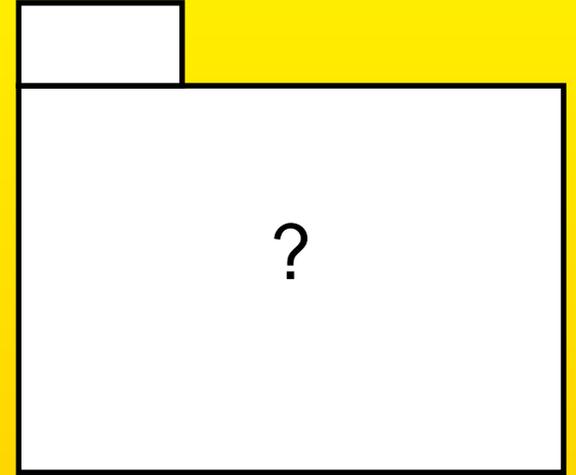
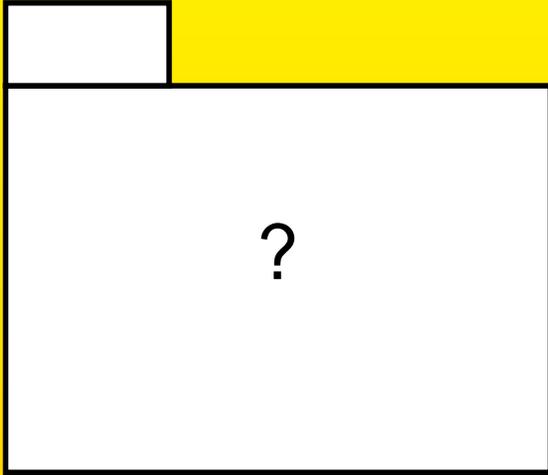
³ `catherine_griffin@uk.ibm.com` IBM UK

⁴ `laurie@tratt.net` King's College London

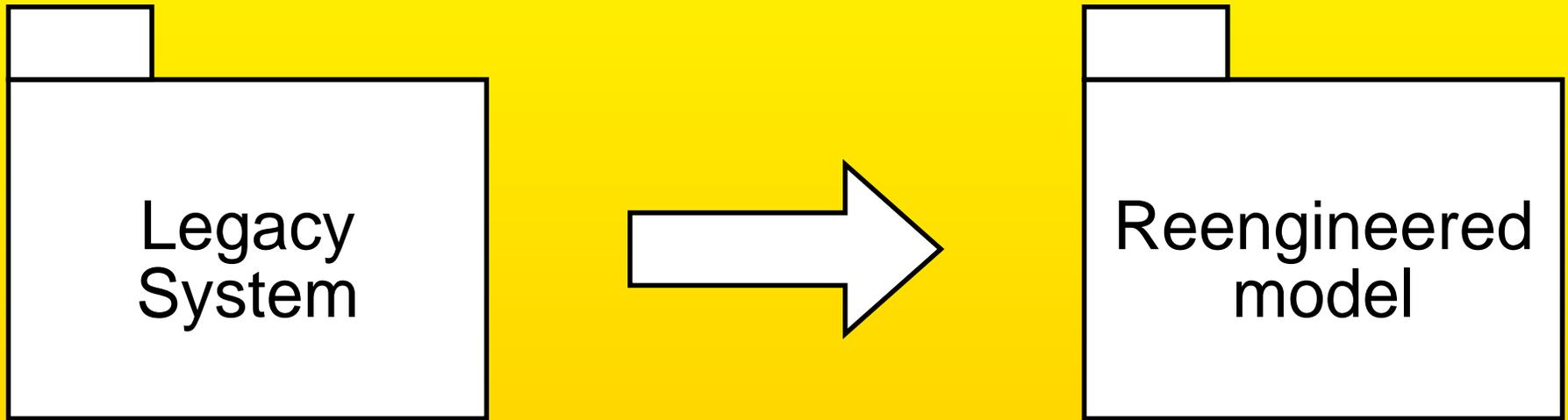
What are transformations

What problem is QVT trying to solve?

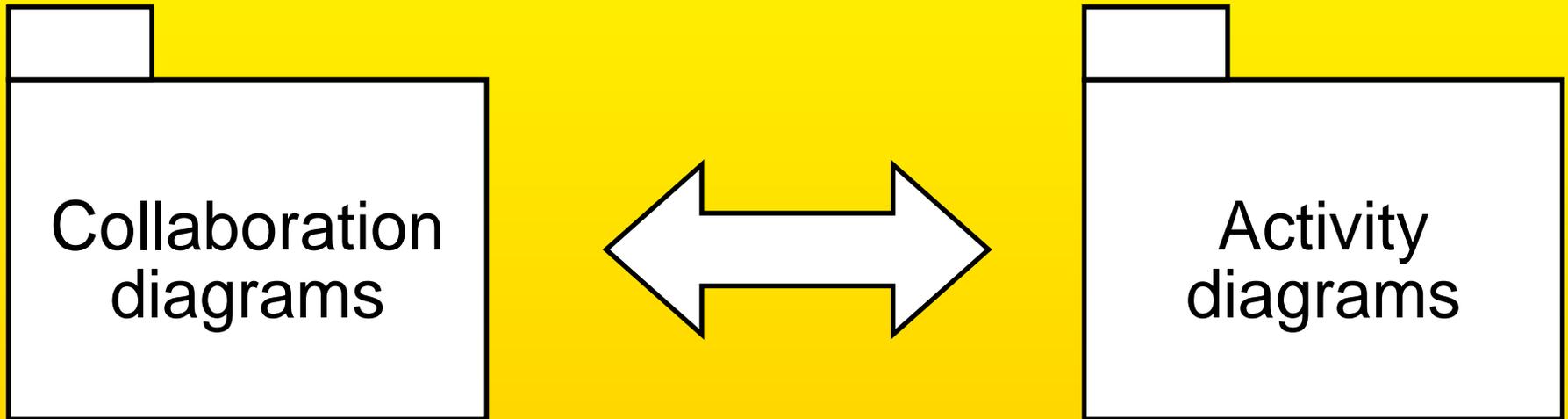
What are transformations



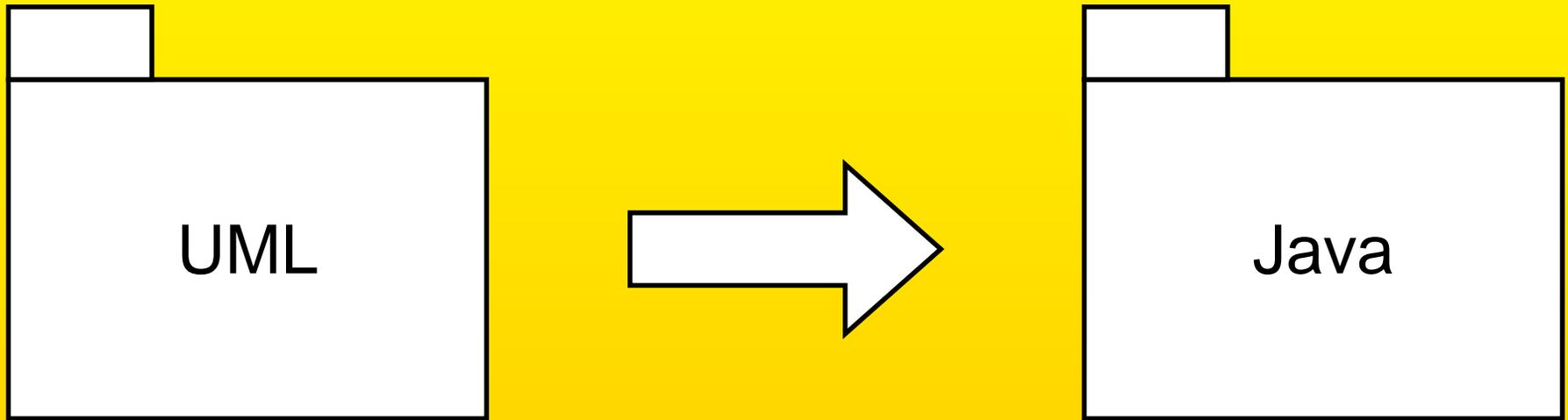
What are transformations



What are transformations



What are transformations



Introduction to QVT

Transformations are:

- key to MDA.
- currently difficult to implement.
- done in an ad hoc fashion

hence the Queries/Views/Transformations (QVT) RFP issued by the OMG.

As of writing, there are 7 QVT submissions on the table.

QVT issues

- Several QVT submissions - what are the common elements in them?
- Are any of them practical?
- Can they be implemented by multiple vendors?
- Would anybody use them?
- Can/how do the submissions integrate with the wider world?

We are particularly interested in how a QVT implementation would operate in practise.

Motivation

An implementation of QVT will:

- make sure the standard is sound and implementable.
- help identify commonality and differences between submissions.
- allow people other than the submitters to put the standard through its paces.
- be separate from, but complementary to, the standard.

High-level requirements

High-level requirements are to:

- have an open architecture.
- be performant and scaleable.
- allow integration with external tools.
- have mechanisms for checking and enforcing model consistency.

More requirements

Some more specific, but important, requirements:

- Good error handling and debugging support.
- Composition of black box transformations.
- Must be able to trace elements.

And finally: there should be a reference implementation which is open source.

What is Eclipse?

From `eclipse.org`:

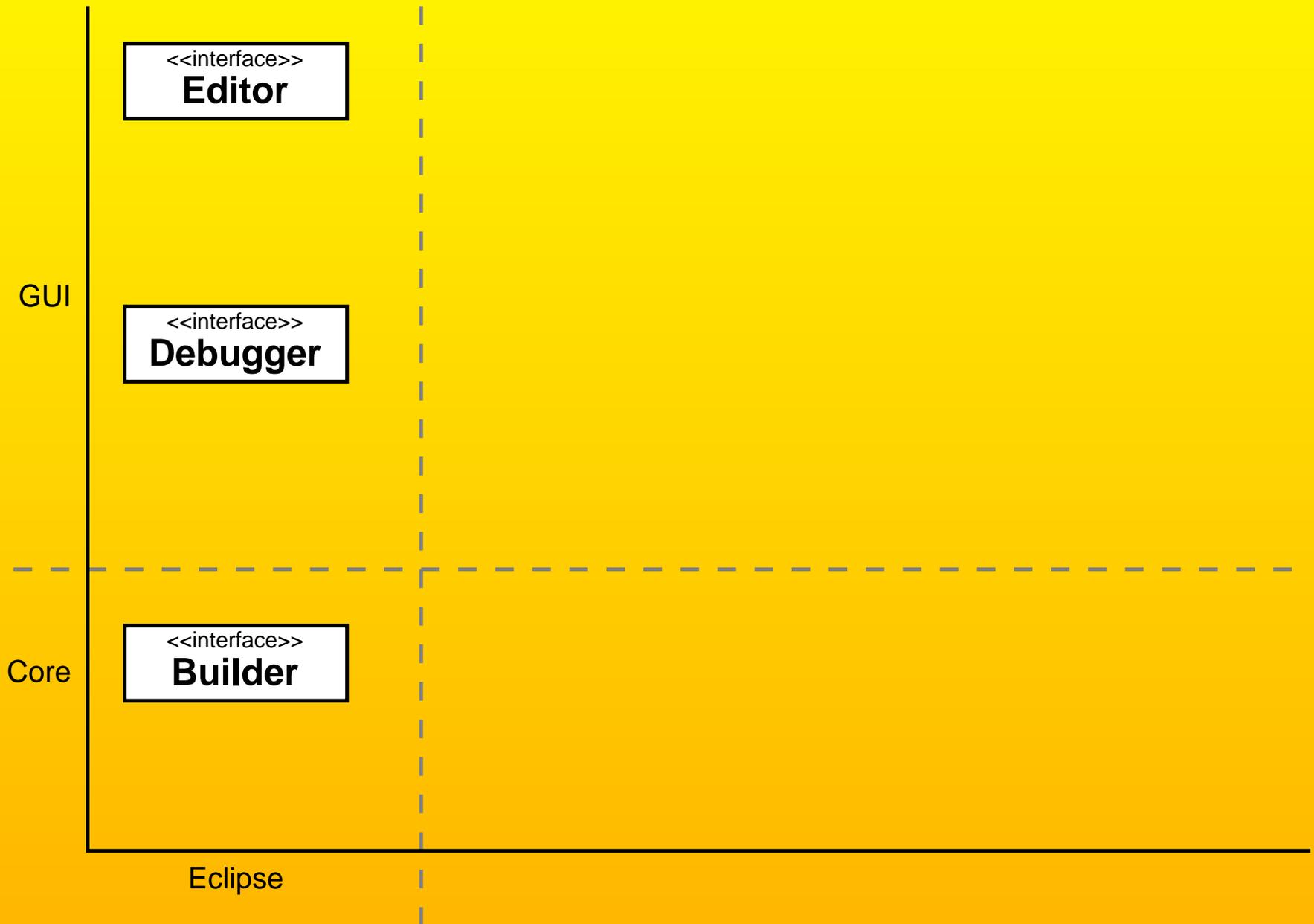
Eclipse is a kind of universal tool platform - an open extensible IDE for anything and nothing in particular.

Why Eclipse?

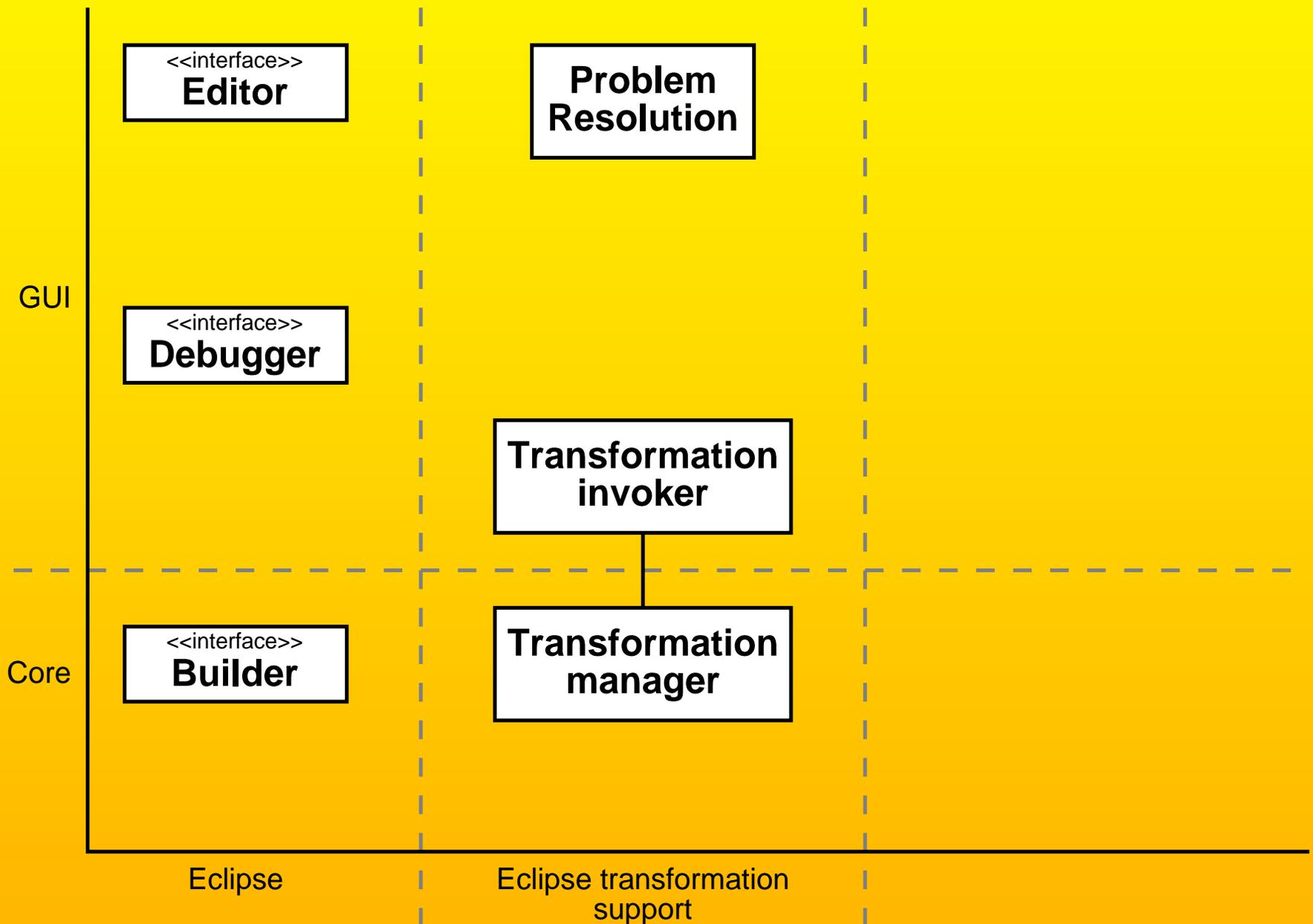
General reasons:

- Open source.
- EMF (Eclipse Modelling Framework).
- Plugin architecture.
- Good operating system coverage.

Architecture



Architecture



Demo

BPEL transformation & Transformation manager

Future work

- Create an open source reference implementation.
- Feedback experience into the QVT standard.
- Involve the wider community.
- Make sure QVT implementation is compatible with existing and emerging MDA tools.
- Implement the QVT language when it is agreed upon, and is close to finalization.