Freie Universität Berlin

qme software

Visual QVT/R

A concrete graphical syntax for QVT/R

qme Software
Gustav-Meyer-Allee 25
13355 Berlin

Telefon 030/46307-230
Teleffax 030/46307-649
Email info@qme-software.de
Web www.qme-software.de

Max Bureck– Eclipse DemoCamp Berlin – 09.06.2009

Content

- Who I am
- Motivation
- What is QVT?
- QVT/R Engines
- Graphical Syntax of QVT/R
- Problems
- Outlook

Who I am

Max Bureck

Student of Freie Universität Berlin

Employed at qme Software GmbH for work on Diploma Thesis:

"Development of a visual transformation-framework for QVT"

Motivation (I / II)

- **Model-to-Model (M2M) transformations**
  - Convert models of meta-model A into models of meta-model B (A may equal B)
  - Use Cases:
    - Model refinements
    - Keep models in sync (e.g. different views on same issue)
    - Transformation to a more concrete model
  - Advantages:
    - Separation of concerns
    - Can be checked statically (will target be valid?)

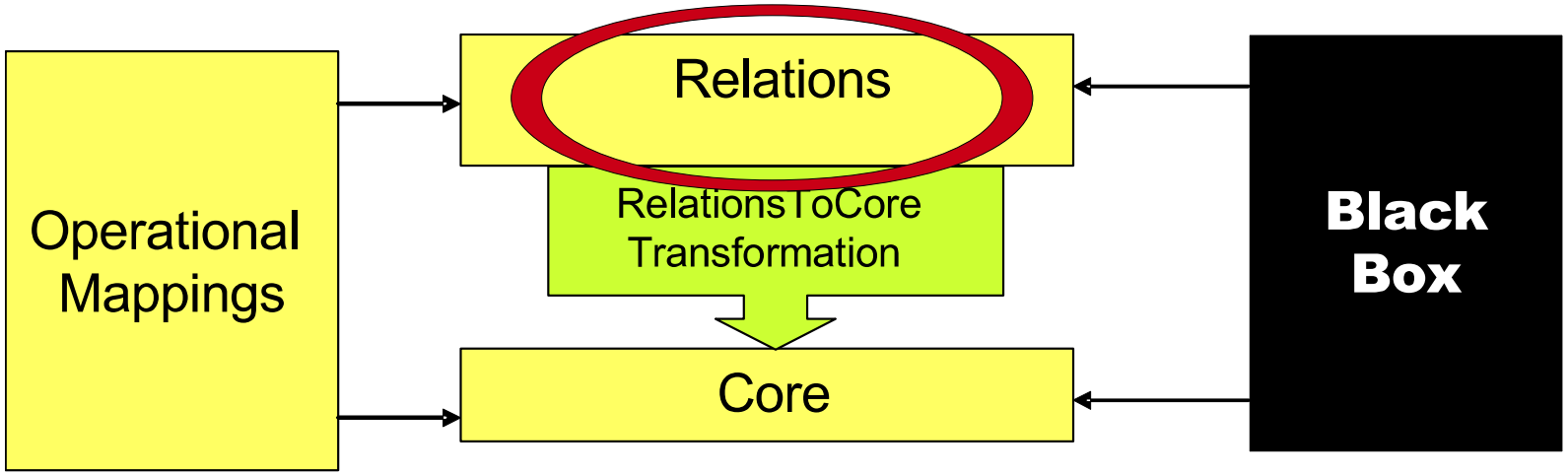
Motivation (II / II)

- **M2M-transformation DSLs**
  - Define transformations in an expressive way → less to write
  - Easy to understand due to concise syntax
- **Graphical syntax**
  - Often easier to conceive than textual representation

What is QVT? (I / III)

- **Standard of the Object Management Group**
- **Stands for Meta Object Facility (MOF) 2.0 Query/View/Transformation**
- **Defines languages for M2M-transformations**
- **Consists of three languages**
  - Core: declarative language, simple syntax → rules are complex to write
  - Relations: declarative language, has textual and graphical syntax
  - Operational Mappings: imperative language

What is QVT? (II / III)



Taken from the QVT standard

What is QVT? (III / III)

QVT Relations

- Advantages:
  - Standardized
  - On retransformation:
    - Modifies target-model, doesn't create it from scratch
    - Can keep changes in existing target-model
  - Multidirectional transformations are possible
  - Graphical syntax

- Disadvantages:
  - Source code a bit bloated
  - Declarative syntax needs to get used to

QVT/R Engines (I / II)

Eclipse QVT Declarative

- Scope: QVT/R and QVT/C interpreter, Eclipse integration
- Developed by: Obeo and Ed Willink
- State: Early in development
- License: EPL

QVT/R Engines (II / II)

Medini QVT

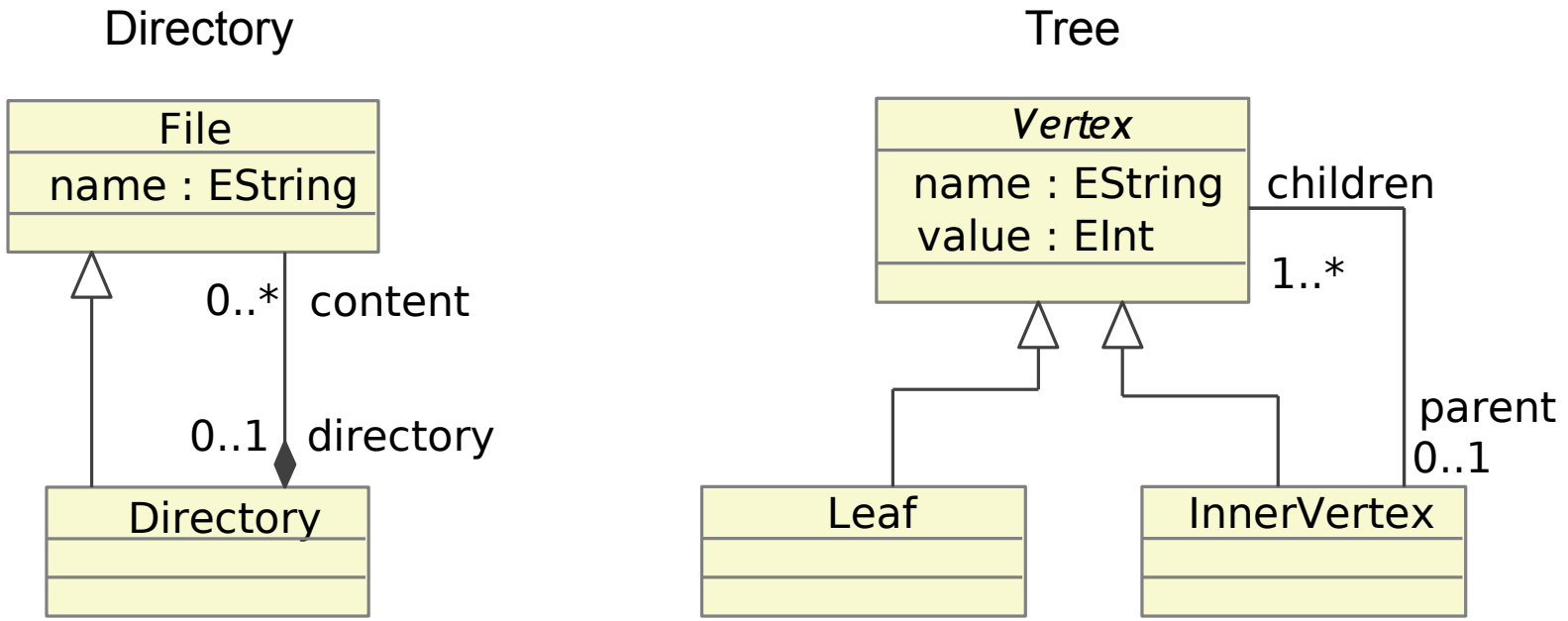
- Scope: QVT/R interpreter, Eclipse integration
- Developed by: ikv++ technologies ag
- State: Some language features are missing (e.g. Collection Templates)
- License:
  - EPL for the Interpreter
  - Eclipse plug-ins are closed source, but freely available for non-commercial use only

Graphical Syntax of QVT/R (I / XI)

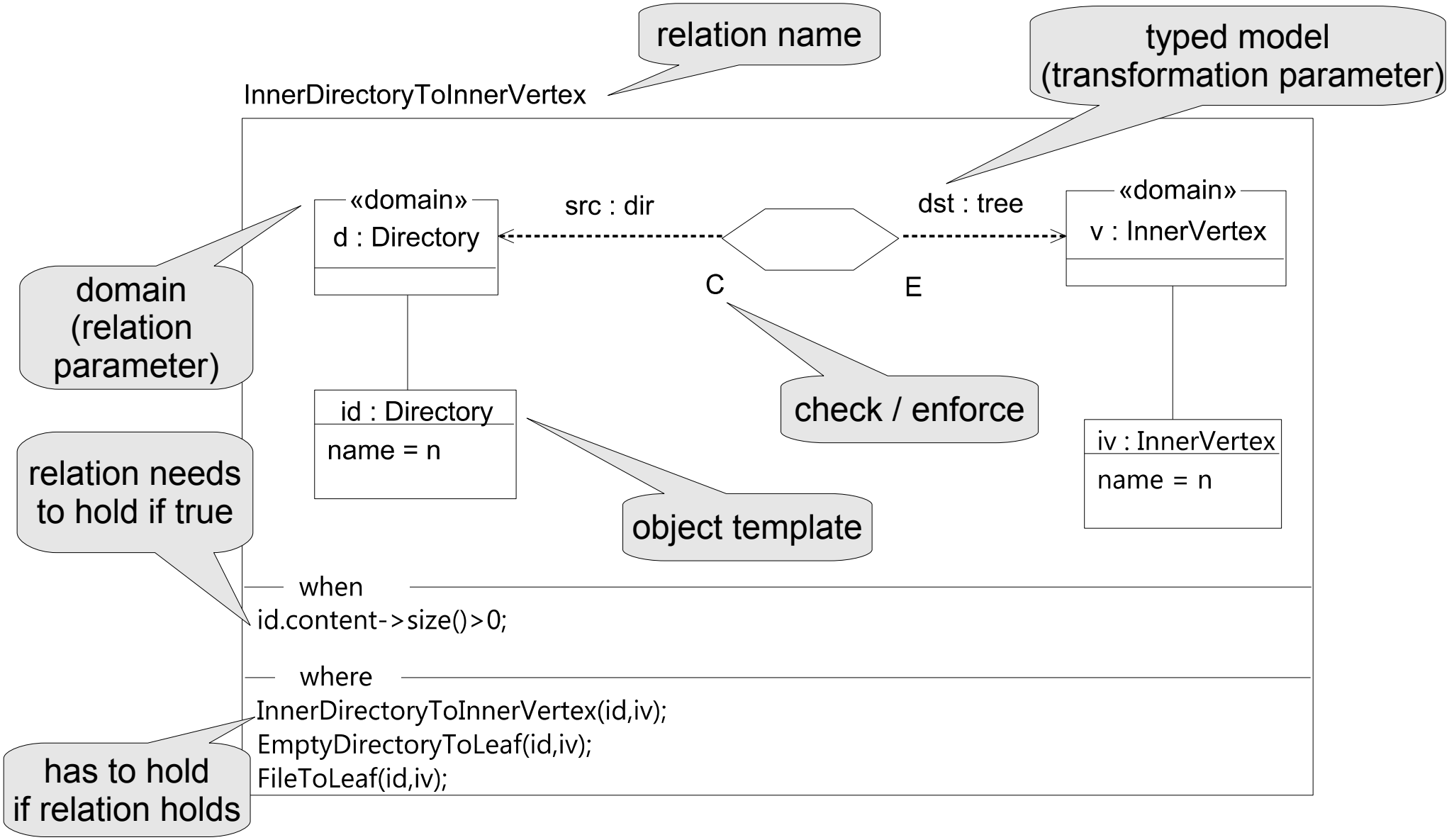
- **Similar to UML Object Diagrams**
- **Idea:**
  - Define object patterns (a set of templates)
  - Relation between patterns for different models
  - A transformation consists of one or more relations
  - Toplevel relations are entry-points for transformation

Graphical Syntax of QVT/R (II / XI)

Meta-models for example



Graphical Syntax of QVT/R (III / XI)



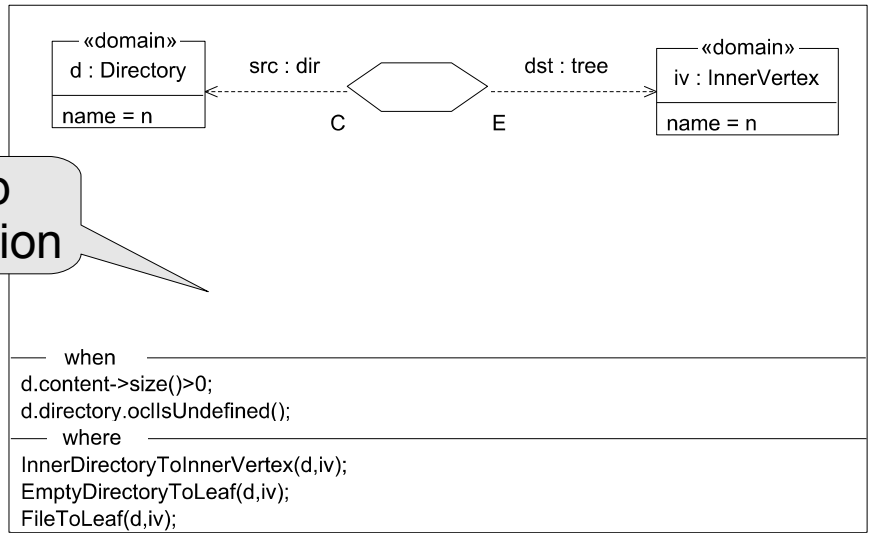
Graphical Syntax of QVT/R (IV / XI)

Equivalent in textual syntax

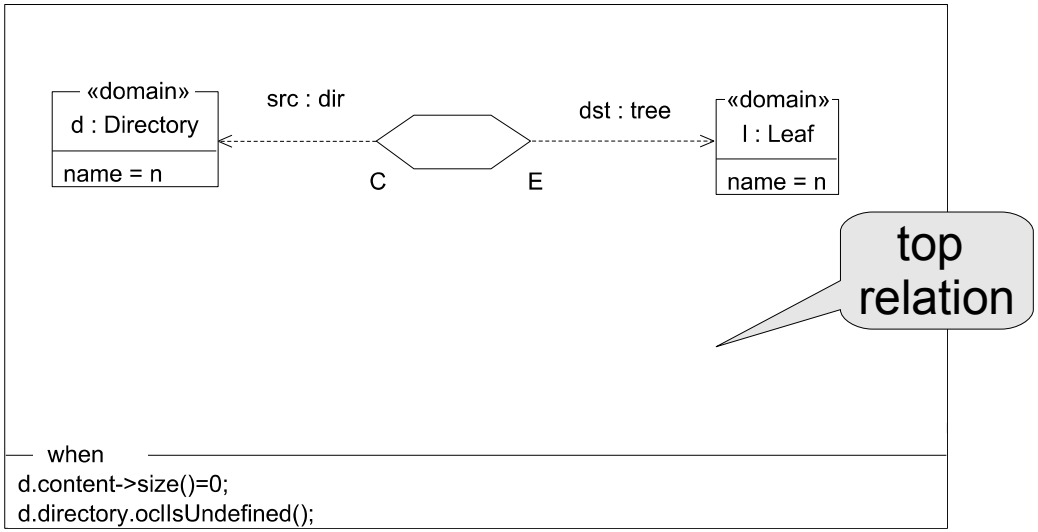
```
relation InnerDirectoryToInnerVertex {

  n : String;

  checkonly domain src d : dir::Directory {
    content = id : dir::Directory {
      name = n
    }
  };

  enforce domain dst v : tree::InnerVertex {
    children = iv : tree::InnerVertex {
      name = n
    }
  };

  when {
    id.content->size() > 0;
  }

  where {
    InnerDirectoryToInnerVertex(id, iv);
    EmptyDirectoryToLeaf(id, iv);
    FileToLeaf(id, iv);
  }

}
```

Graphical Syntax of QVT/R (V / XI)

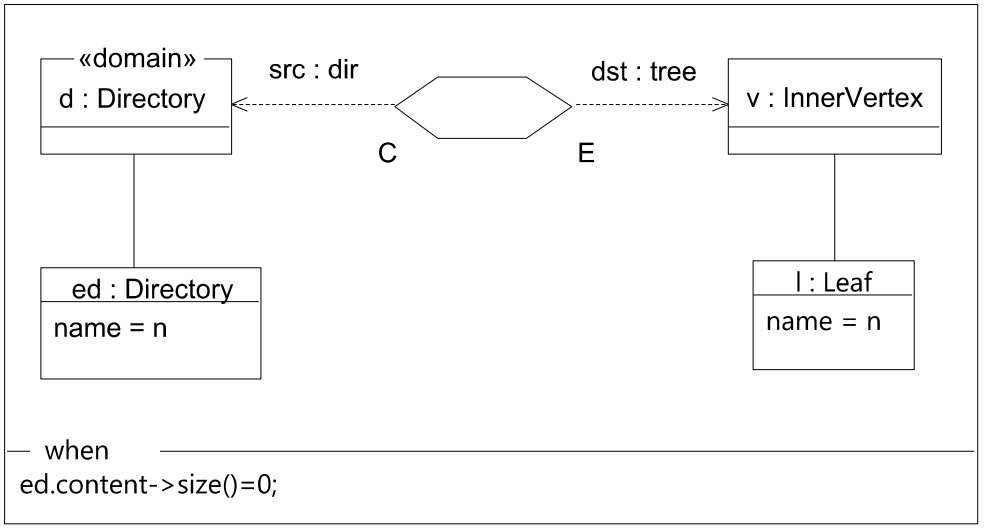
RootDirectoryToInnerVertex



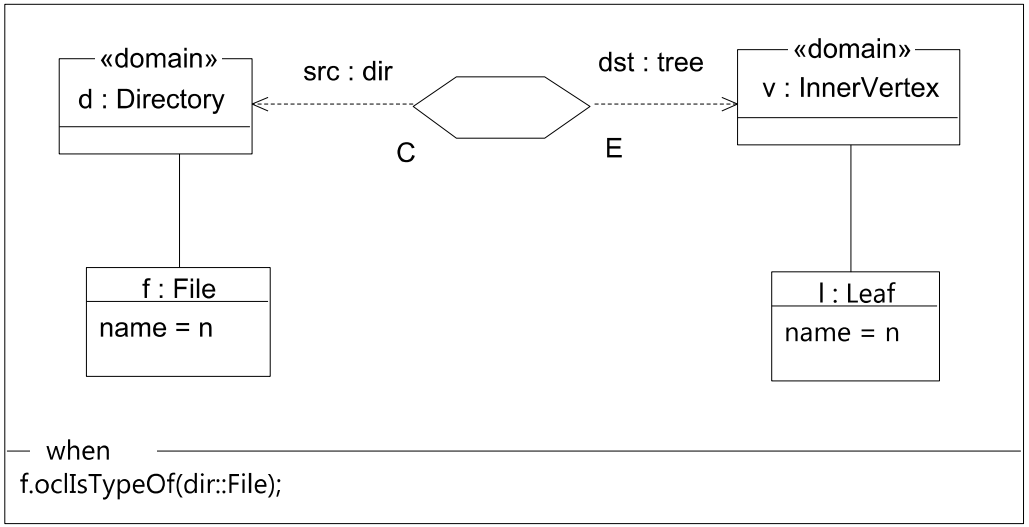
RootDirectoryToLeaf



EmptyDirectoryToLeaf



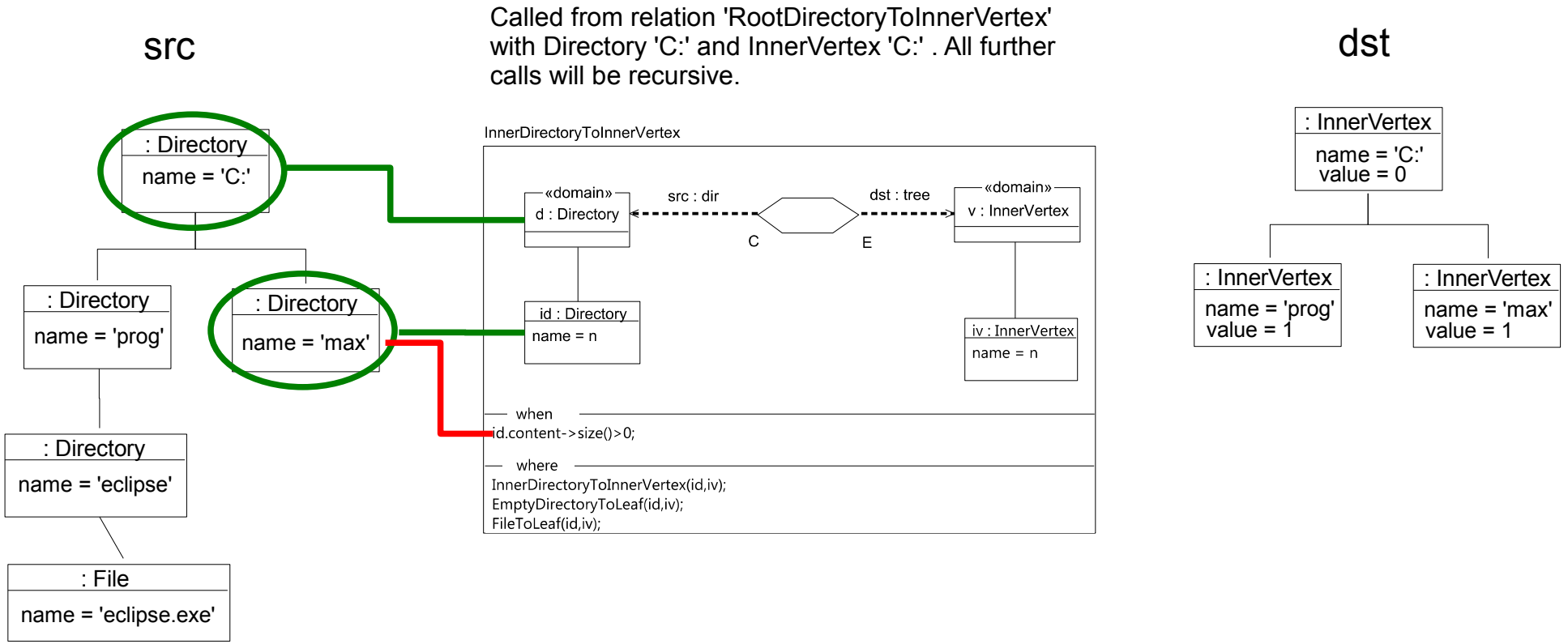
FileToLeaf



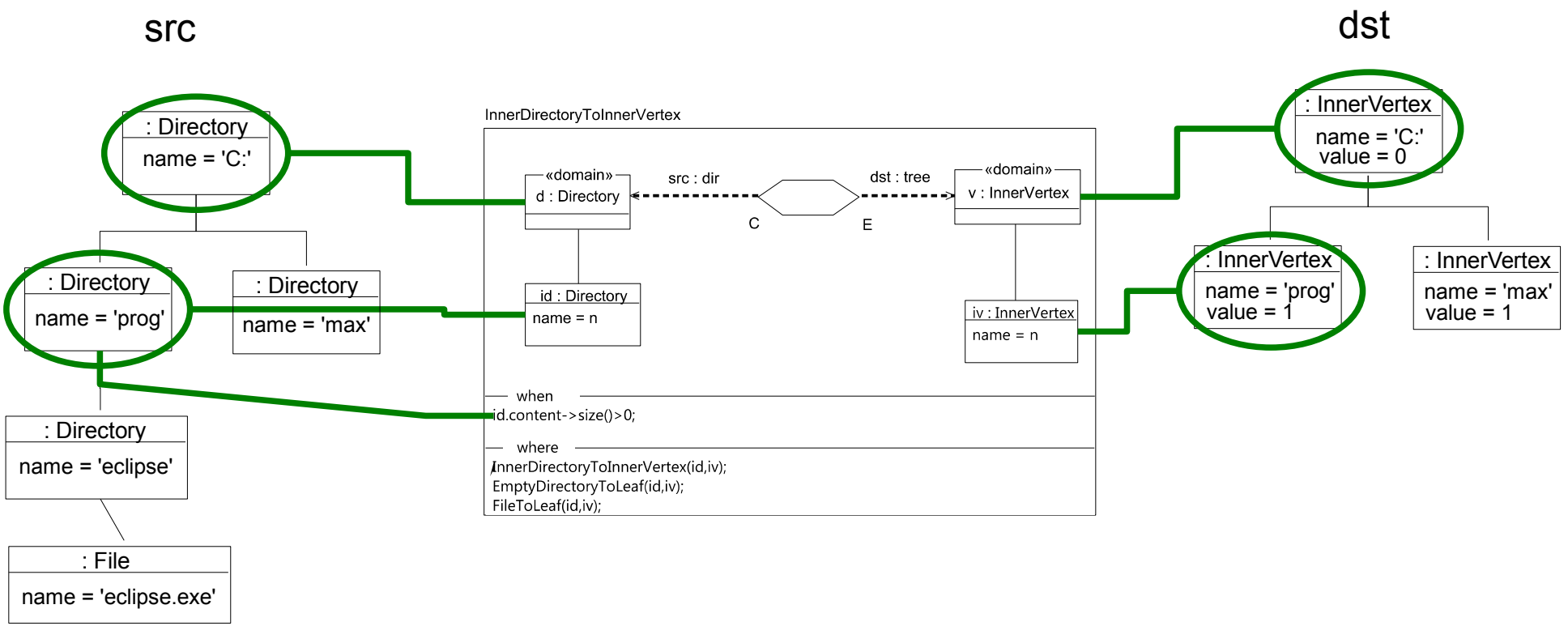
Graphical Syntax of QVT/R (VI / XI)

- **Example:**
  - Run a transformation from directory to tree with an existing target-model
  - We observe what relation "InnerDirectoryToInnerVertex" does

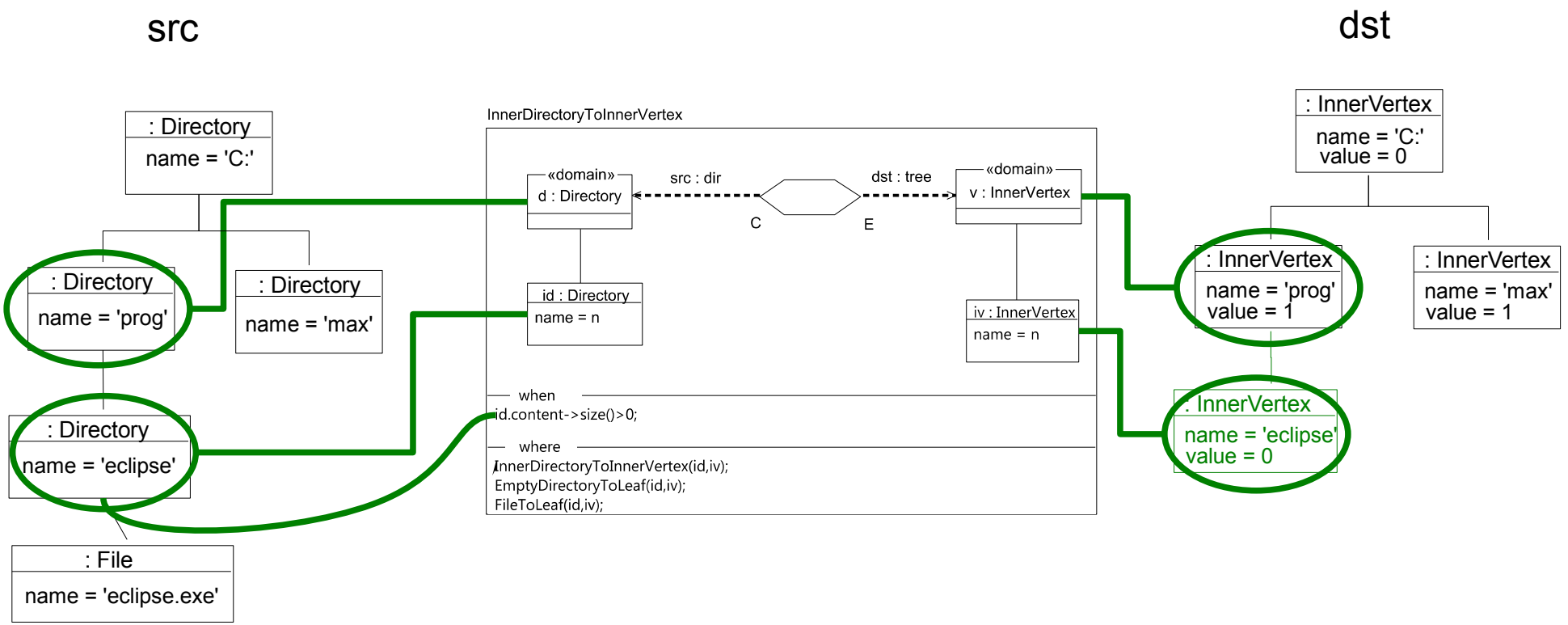
Graphical Syntax of QVT/R (VII / XI)



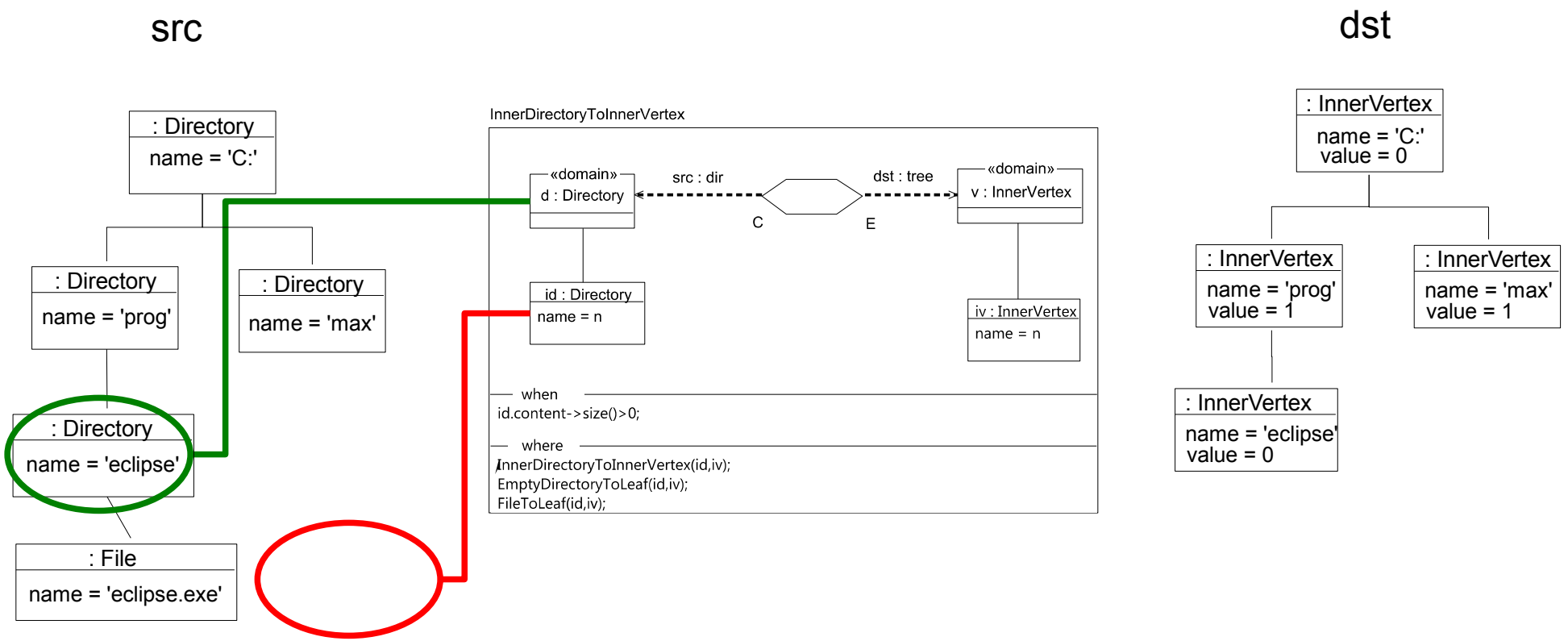
Graphical Syntax of QVT/R (VIII / XI)



Graphical Syntax of QVT/R (IX / XI)

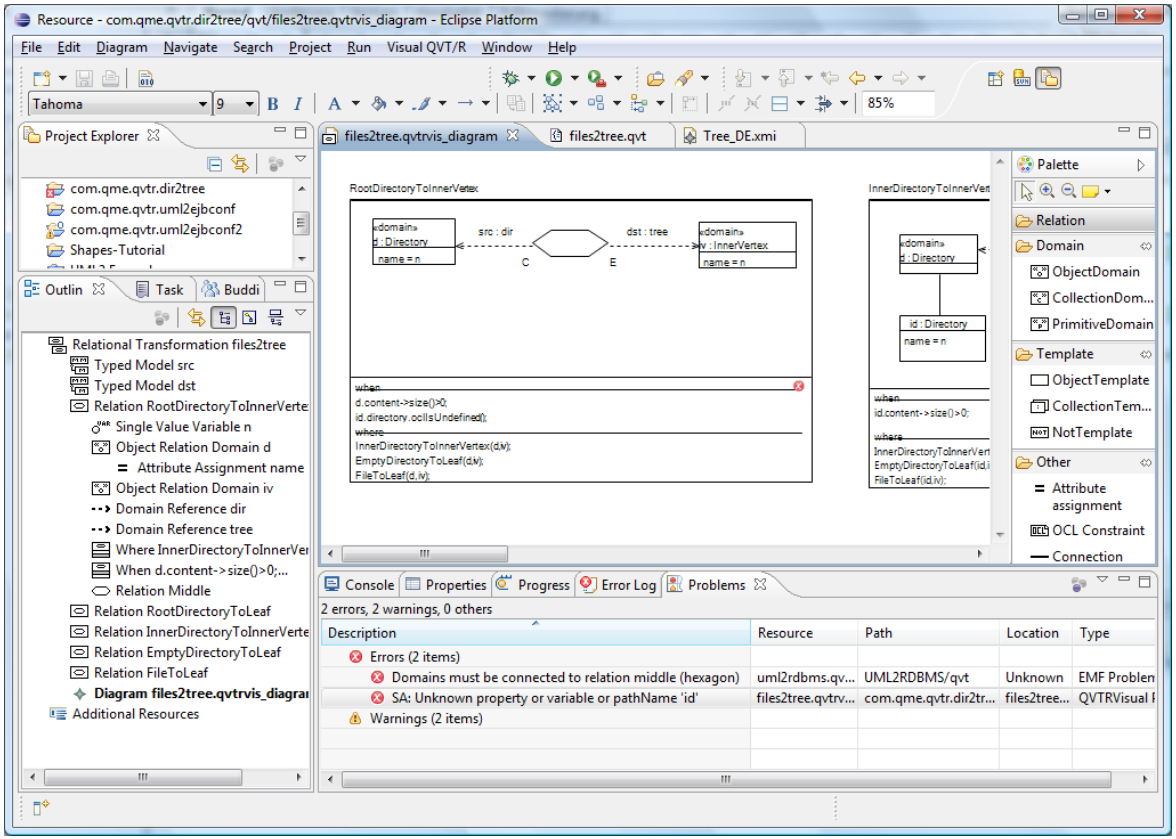


Graphical Syntax of QVT/R (X / XI)



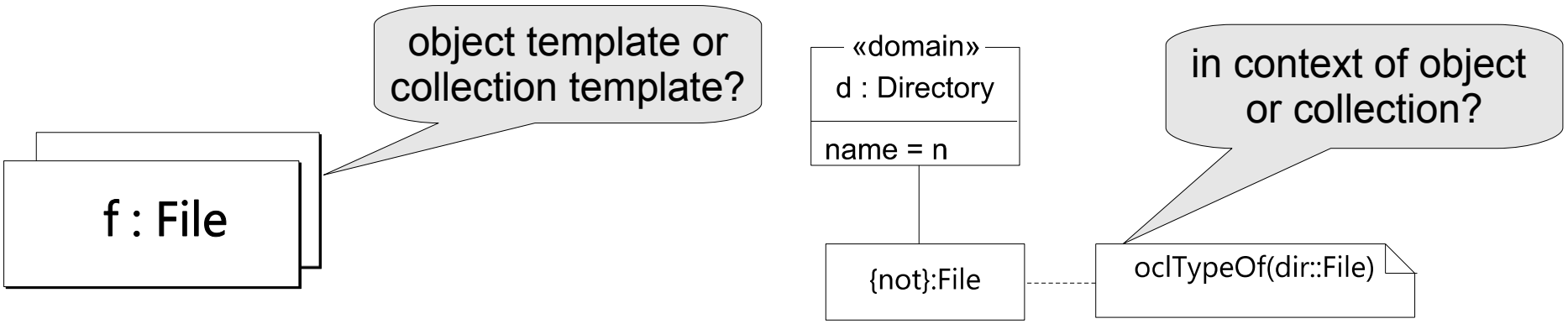
Graphical Syntax of QVT/R (XI / XI)

- Visual QVT/R provides an editor for the syntax (and more)
- More information soon to come



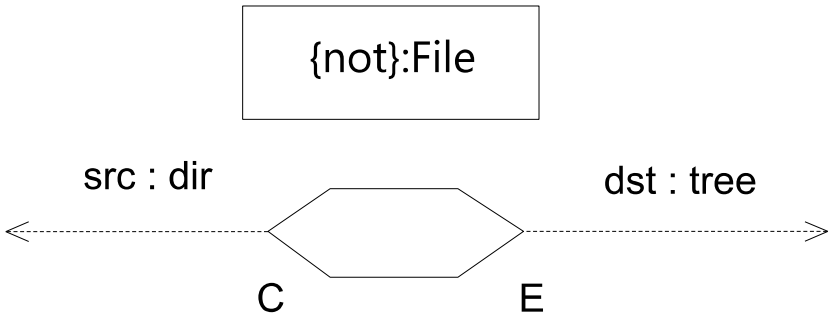
Problems (I / II)

- **For several elements in abstract syntax there is no graphical Syntax**
  - Key definitions
  - Queries
  - No indication for toplevel relations
  - ...
- **Some Elements of the graphical syntax have unclear semantics**



Problems (II / II)

- **Problem for Implementation: Some syntax elements have no correspondence in abstract or concrete textual syntax**
  - Not template
  - Relation middle



Outlook

- **Visualization of trace-data is interesting (→ last Eclipse DemoCamp)**
  - Show transformation on 3D-trace visualization to show cause of errors
  - Or simply dye transformation-model according to selection in source or target-model
- **Graphical debuggers possible**
  - Add breakpoint-conditions to model-elements as OCL-constraints
  - Mark currently processed template and show match for template

Thank you for your attention

Questions?

max.bureck@fu-berlin.de

qme Software
Gustav-Meyer-Allee 25
13355 Berlin

Telefon 030/46307-230
Teleffax 030/46307-649
Email info@qme-software.de
Web www.qme-software.de