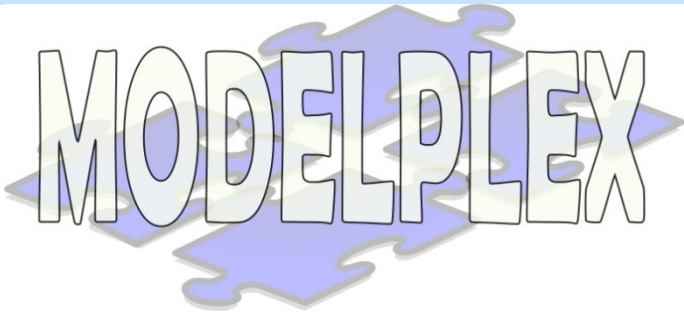
Composing Models: Principles & Techniques

Steven Völkel & Jendrik Johannes

TUBS & TUD

© Copyright TUBS & TUD 2008

Context of this work



- The present courseware has been developed within the context of the MODELPLEX European IST FP6 project (http://www.modelplex.org/).
- Co-funded by the European Commission, the MODELPLEX project involves 21 partners from 8 different countries.
- MODELPLEX aims to define and develop a coherent infrastructure specifically for the application of MDE to the development and subsequent management of complex systems within a variety of industrial domains.
- To achieve the goal of the large-scale adoption of MDE, MODELPLEX promotes the idea of the collaborative development of courseware dedicated to this domain.
- The MDE courseware provided here with the status of open-source software is produced under the EPL 1.0 licence.

© Copyright TUBS & TUD 2008

Outline

- Introduction
  - Terminology
- Principles
  - syntactical model composition
  - semantical model composition
  - relations between syntactical and semantical composition
  - properties of model composition
- Techniques
  - composing models with Reuseware
  - composing models through model weaving with AMW
  - Composing models through comparison and merging with Epsilon

© Copyright TUBS & TUD 2008

Introduction: Terminology

- Model composition process:
  - Integrating a number of models either by
    - Ensuring their interoperability as autonomous models or by
    - Comnining them into new composite models
  - Consists of several activities (etc. interoperability check or establishing weaving model)

- Participating models:
  - Input models
  - Output model(s)
  - Weaving model:
    - Contains links to elements of input models

© Copyright TUBS & TUD 2008

Introduction: Terminology II

- Participating models (cont'd):
  - Woven model:
    - Model that contains elements whcih were target of one or more weaving models
  - Composition model:
    - Special kind of Weaving model: links are useful for composition only (e.g., links to elements which have to be merged)

- Model Composition Activities:
  - Model weaving:
    - establishing a weaving model
  - Model merging:
    - integrating a number of input models to a one or more output models by exploiting information in one or more composition models

© Copyright TUBS & TUD 2008

Syntax and Semantics of Models

- Syntax:
  - Conrete Syntax: representation of the model
    - Diagrams
    - Text
    - XML
  - Abstract Syntax:
    - independent from representation
    - „essence" of the model
- Semantics:
  - „meaning of the model"
  - E.g., by mapping to a well known semantic domain

© Copyright TUBS & TUD 2008

Syntax and Semantics of Models

- Semantical composition:
  - User has a composition operator in mind
    - E.g. „merge 2 class diagrams using UML package merge and its rules"
  - i.e., the composition is first defined on the semantic level
- Syntactical composition:
  - Defines what the operator does on the syntactical level
    - E.g., copying attributes and methods of equal-named classes into a new class with the same name (without duplicates)
  - Composition is always executed on the syntactical level
- Central questions:
  - „What is the relationship between the syntactical and the semantical operator?"
  - „What is the relationship between the semantics of the input models and the semantics of the output model?" and especially „Are there any consequences?"

© Copyright TUBS & TUD 2008

Semantics of Models

- Compositional development implies open-world-assumption:
  - „What is not stated is considered unknown"
- More than 1 system may realize a model
- Thus: introduce semantic mapping which maps each model to the set of systems which are possible realizations
- Denote S / M as the set of all systems / models, sm as the semantic mapping, $\wp$ as powerset we have
  - sm : M -> $\wp$ (S)

© Copyright TUBS & TUD 2008

Semantics of Models

- Each model describes predicates over the set of systems

- Properties of sm:
  - $m \in M$ consistent $\Leftrightarrow sm(m) \neq \emptyset$
  - $m \in M$ does not contain information $\Leftrightarrow sm(m) = S$
  - m_2 refines $m_1 \Leftrightarrow sm(m_2) \subseteq sm(m_1)$
    - The more we know the fewer implementations are possible
    - Add information $\Rightarrow$ set of systems gets smaller

© Copyright TUBS & TUD 2008

Model composition

- Simplest form on the syntactic level:
  - $\otimes$: M x M $\rightarrow$ M
- Property Preserving (PP) composition operator:
  - On the left: $sm(m_1 \otimes m_2) \subseteq sm(m_1)$
  - On the right: $sm(m_1 \otimes m_2) \subseteq sm(m_2)$
  - PP: $sm(m_1 \otimes m_2) \subseteq sm(m_1) \cap sm(m_2)$
- PP $\Rightarrow$ no information/design decision is lost during composition
- No equality, but subset:
  - There can be added information (e.g., unnamed associations get a name)

© Copyright TUBS & TUD 2008

Model composition

- If we have equality ⇒ Fully Property Preserving composition operator:
  - $sm(m_1 \otimes m_2) = sm(m_1) \cap sm(m_2)$
- Advantages !!!
  - Analyze models seperateley
  - Trace properties/errors back to source models
  - PP ⇒ all requirements defined in source models are preserved in composition
  - FPP is a prerequisite to generate code out of each model separately

© Copyright TUBS & TUD 2008

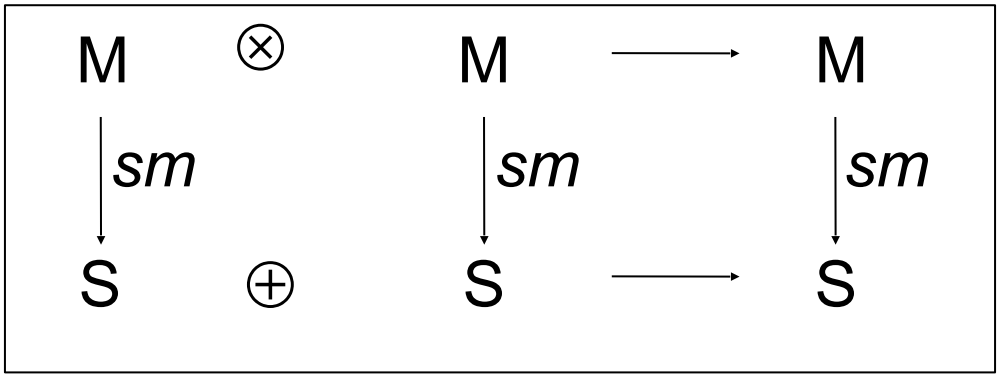
Model composition

- Repeated: PP composition operator:
  - $sm(m_1 \otimes m_2) \subseteq sm(m_1) \cap sm(m_2)$
- $\Rightarrow$ composition can add information
- This may lead to an inconsistent result ($sm(m) = \emptyset$) even though the originals are not incosistent with each other ($sm(m_1) \cap sm(m_2) \neq \emptyset$)
  - Difficult to handle such a composition operator
  - $\Rightarrow$ demand for Consistency Preservation (CP):
    - $sm(m_1) \cap sm(m_2) \neq \emptyset \Rightarrow sm(m_1 \otimes m_2) \neq \emptyset$
- FPP $\Rightarrow$ CP

© Copyright TUBS & TUD 2008

Semantic Composition Operator

- Similar to the syntactic level
  - $\oplus$: D x D $\rightarrow$ D
  - Example: intersection
- Resulting diagram:



- Diagram commutes iff
  - $sm(m_1 \otimes m_2) = sm(m_1) \cap sm(m_2)$ (FPP)

© Copyright TUBS & TUD 2008

Deeper investigation

- This presentation contains basic investigations only
- For more detailed discussions and examples see:

  Herrmann, Krahn, Rumpe, Schindler, Völkel:
  An Algebraic View on the Semantics of Model Composition, ECMDA 2007, Haifa, Israel

© Copyright TUBS & TUD 2008

Outline

- Introduction
  - Terminology
- Principles
  - syntactical model composition
  - semantical model composition
  - relations between syntactical and semantical composition
  - properties of model composition
- Techniques
  - composing models with Reuseware
  - composing models through model weaving with AMW
  - Composing models through comparison and merging with Epsilon

© Copyright TUBS & TUD 2008

Composing Models With Reuseware

- Reuseware is...
  - A language independent modularisation approach [1]
  - A framework: Reuseware Composition Framework [2]
- Common concepts for different composition systems for arbitrary languages
  - Easy specification of new composition techniques and porting of techniques from one language to another
  - Reuse of composition tooling
  - Tailor tooling for composition system and language
- Support features of aspect-oriented systems
  - Support homogeneous crosscuts (and quantification)
  - Support heterogeneous crosscuts

[1] On Language-Independent Model Modularisation, Transactions on Aspect-Oriented Development, 2008
[2] http://reuseware.org

© Copyright TUBS & TUD 2008

Reuseware - Core Concepts

- Model Fragments
  - (Partial) models that may contain variation points
  - Offer a **Composition Interface**
  - **Composition Interface** consists of **Ports**
  - **Ports** point at elements of the model fragment that can be accessed for composition
  - One **Port** can point at several elements at arbitrary places in the model fragment (heterogeneous crosscut)
  - Similar Ports can be joined to one **HomogeneousPort** (homogeneous crosscut)
- Composition Programs
  - Define **composition links** between Ports
  - Are executed to produce a composed model where model fragments are woven at the elements pointed out by the linked Ports

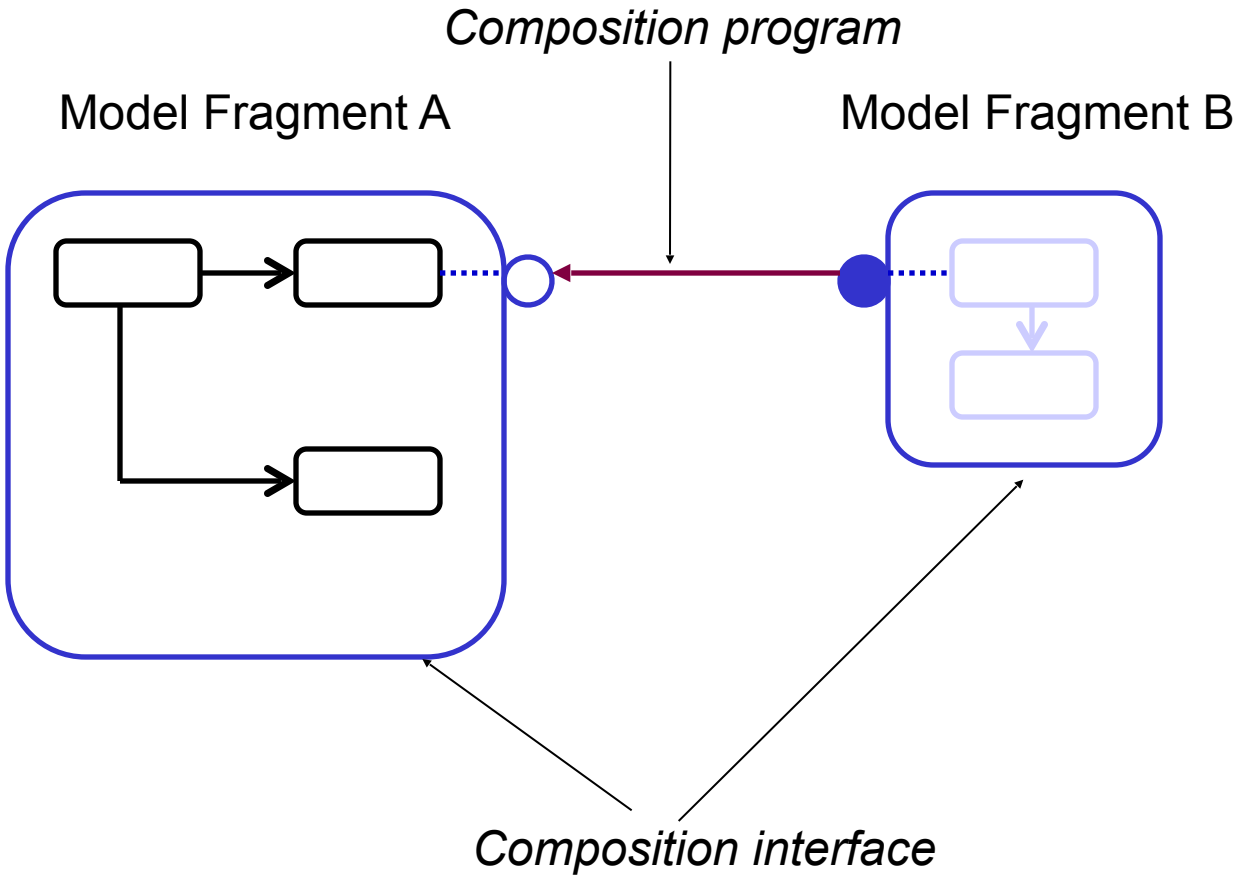
© Copyright TUBS & TUD 2008

Reuseware - Core Concepts

- Composition Systems
  - Define modularisation concepts (e.g., modules, packages, aspects)
  - Define relations between modularisation concepts (e.g, an aspect relates to a core)
- Reuse extensions (for a particular language)
  - Define how modularization concepts defined in a composition system are realized in a concrete language
  - Define which ports are related to which model elements of a model fragment

© Copyright TUBS & TUD 2008

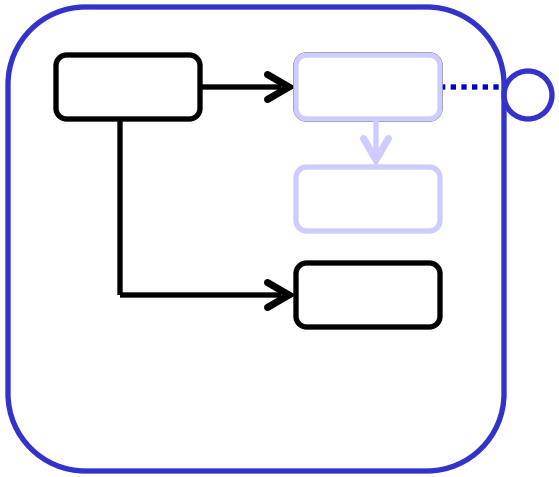
Reuseware - Core Concepts

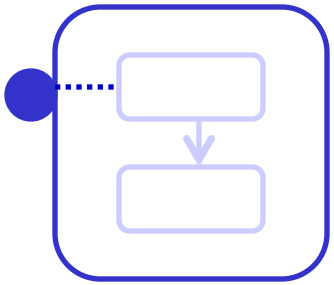


© Copyright TUBS & TUD 2008

Reuseware - Core Concepts

Model composed from A and B

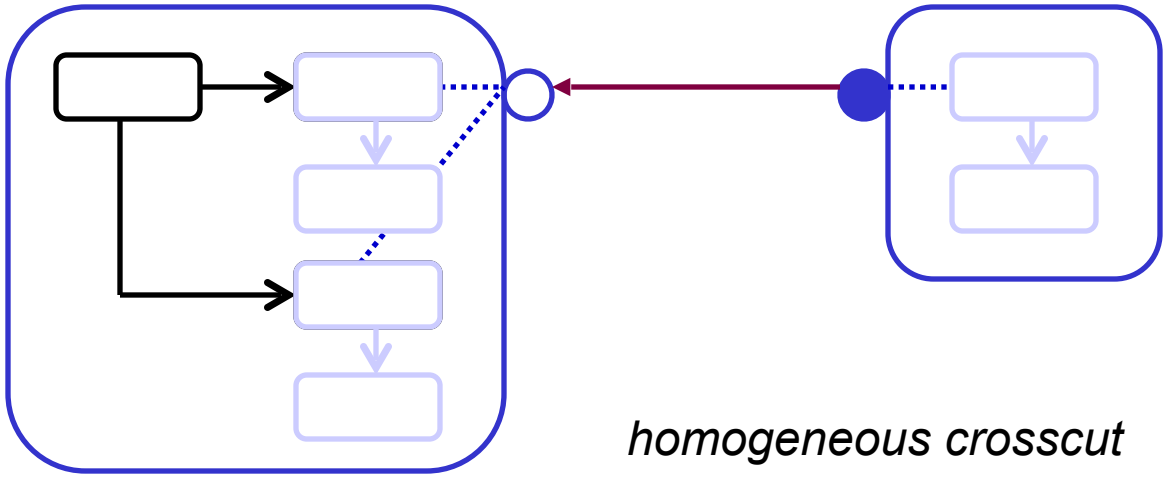




Composed model

© Copyright TUBS & TUD 2008

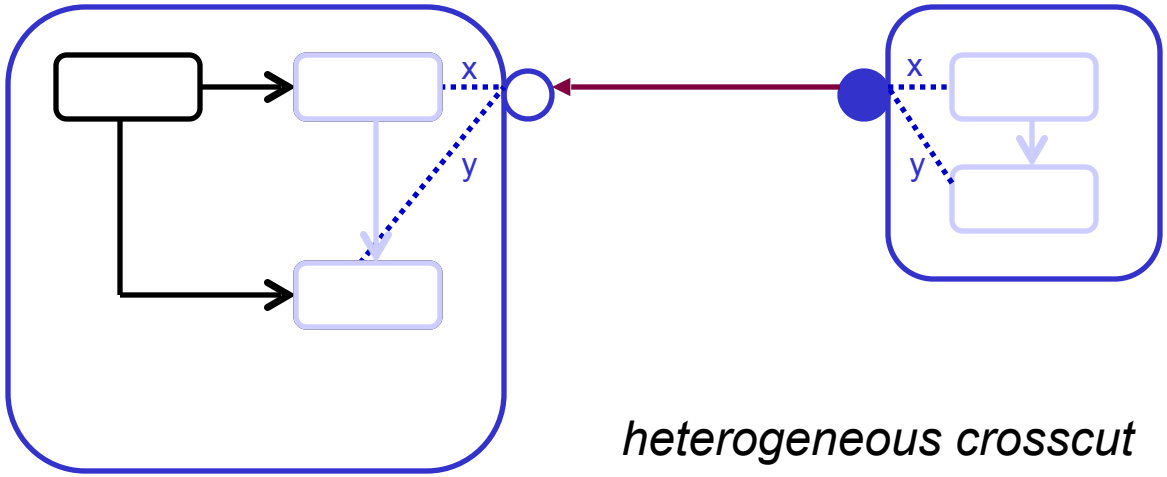
Reuseware - Core Concepts



homogeneous crosscut

© Copyright TUBS & TUD 2008

Reuseware - Core Concepts

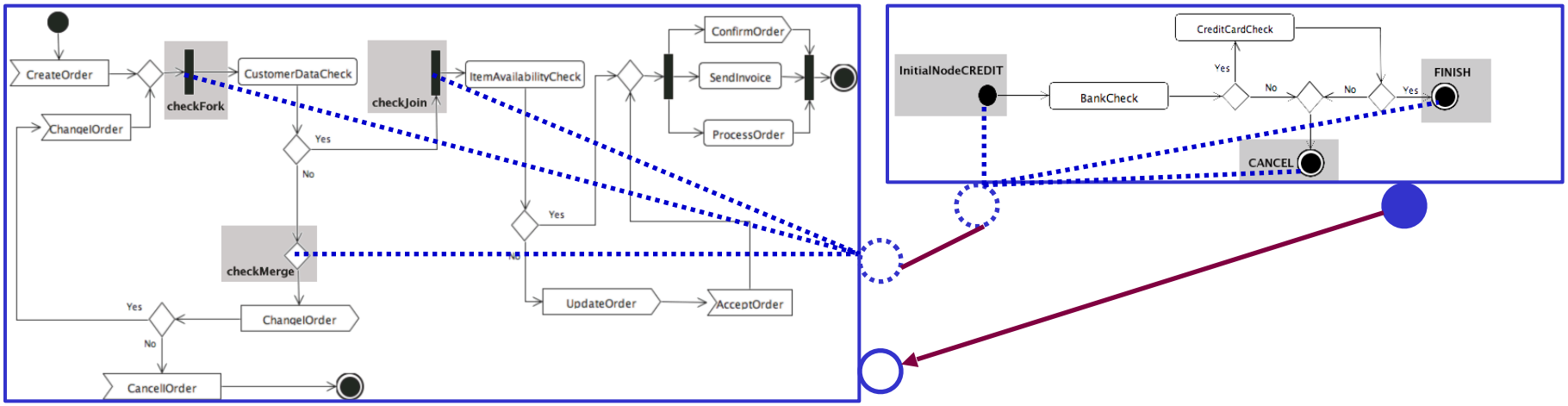


heterogeneous crosscut

© Copyright TUBS & TUD 2008

Reuseware Example: Activity Diagram Composition

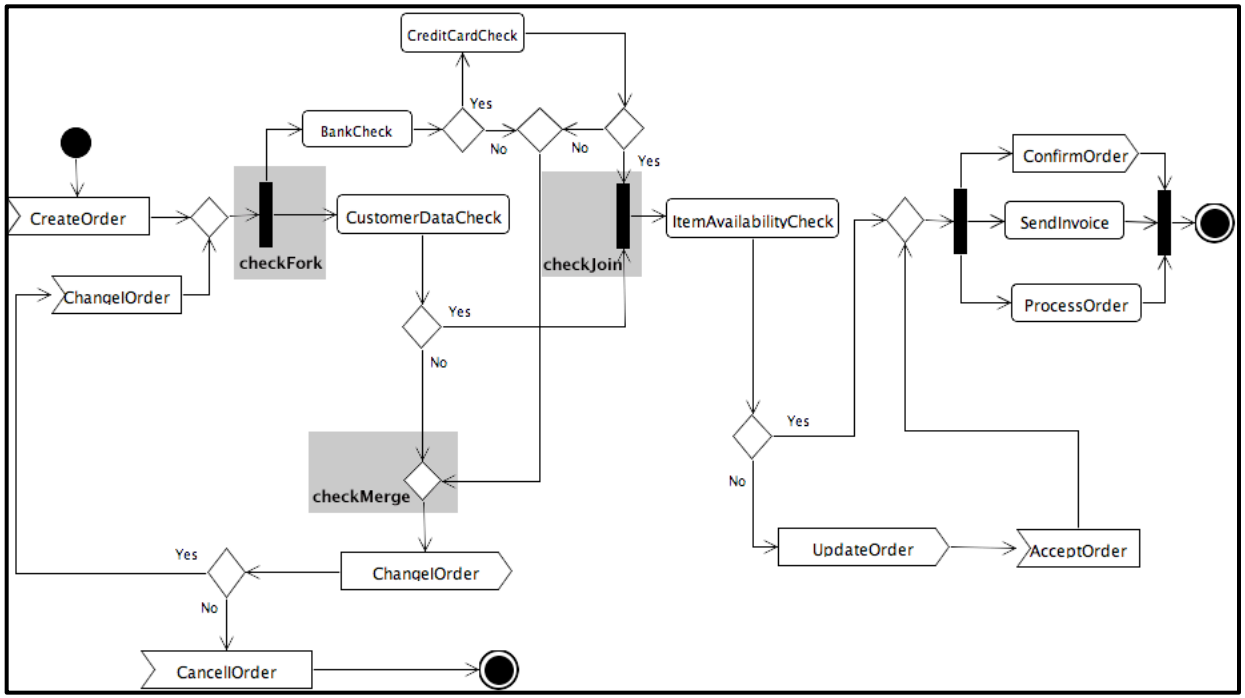
Composition of two activity diagram model fragments



© Copyright TUBS & TUD 2008

Reuseware Example: Activity Diagram Composition

Composed activity diagram



© Copyright TUBS & TUD 2008

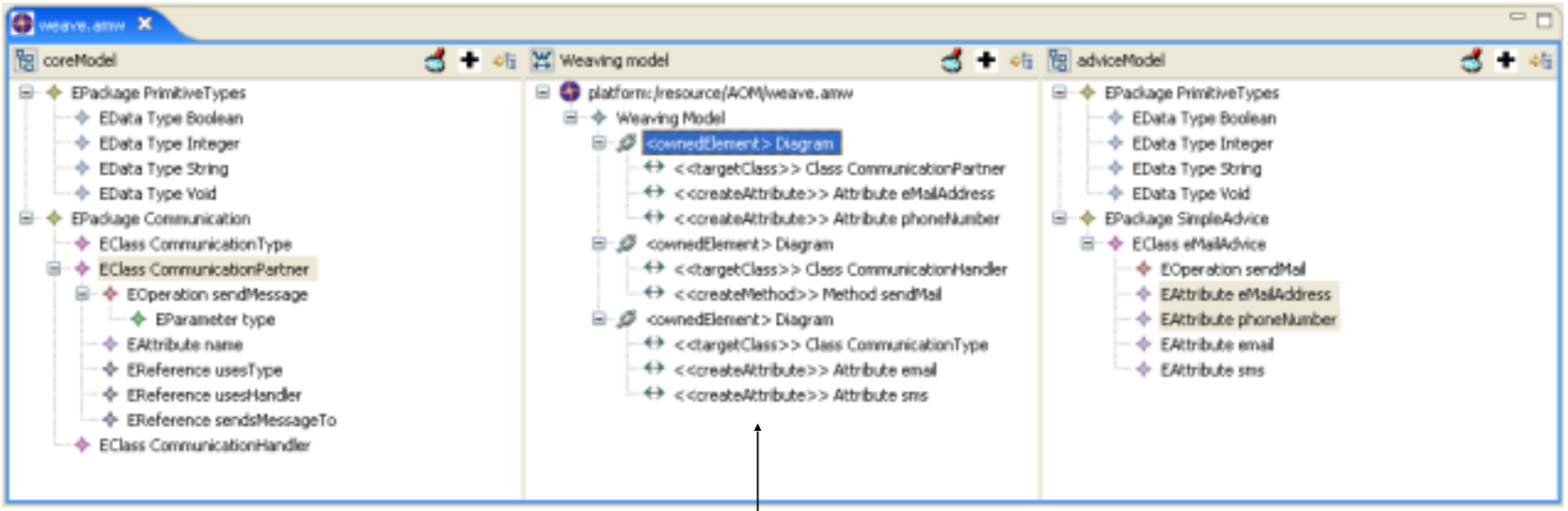
Compositing Models through Model Weaving with AMW

- AMW
  - Eclipse modelling tool for model weaving
  - http://www.eclipse.org/gmt/amw/
- Allows to link models through weaving models
  - Weaving models can act as **composition models**
  - Model transformation interprets such weaving models and merges them

© Copyright TUBS & TUD 2008

Model Composition with AMW - Example

- http://www.eclipse.org/gmt/amw/usecases/AOM/



Weaving links are defined between elements of two models to express composition

© Copyright TUBS & TUD 2008

Model Composition with AMW - Example

```
--
rule newAttribute {
	from
		attr : AMW!Attribute
	to
		out : MOF!EAttribute (
			name <- attr.name,
			lowerBound <- 1,
			upperBound <- 1,
			eType <- MOF!EModelElement.allInstancesFrom('core')->
				select (e | e.name =
MOF!EModelElement.getInstanceById('advice',attr.element.ref).eType.name) ->first()
		)
}
----------------------------------------------------------------------------------------
----------------------

rule newMethod {
	from
		method : AMW!Method
	to
		out : MOF!EOperation (
			name <- method.name,
			eType <- MOF!EDataType.allInstances()->select ( e | e.name ='Void')->first()
		)
}
```

A transformation interprets the weaving links as model compositions

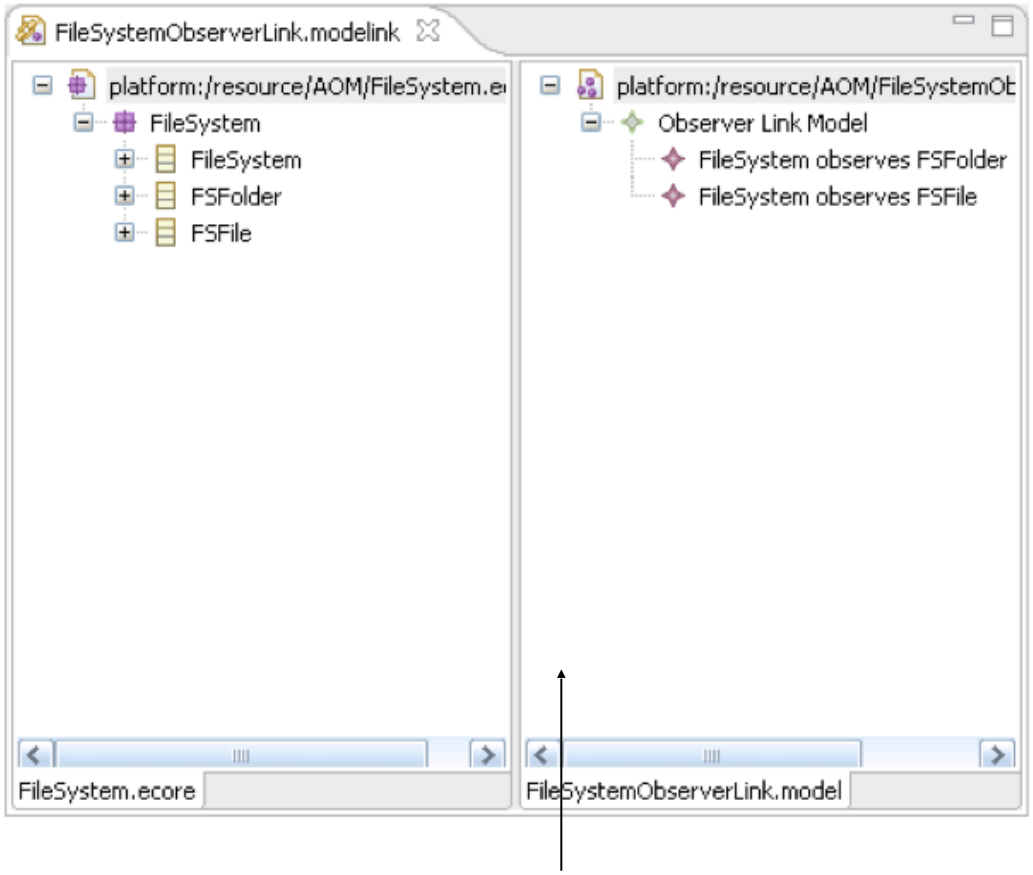
© Copyright TUBS & TUD 2008

Composing models through comparison and merging (Epsilon)

- Epsilon
  - Eclipse model management platform
  - http://www.eclipse.org/gmt/epsilon/
- Allows to link models through link models
  - Link models can act as **composition models**
  - Model comparison and merging are used to interpret link models as model compositions
  - Epsilon language ECL and EML are used for this

© Copyright TUBS & TUD 2008

Model Composition with Epsilon - Example



Links define relations between model elements required for a composition

© Copyright TUBS & TUD 2008

Model Composition with Epsilon - Example

```
rule ClassWithObserver
        match c : Core!EClass
        with o : ObserverPattern!EClass {

        guard : o.name = 'Observer'

        compare : ObserverLink!ObserverLink.allInstances.exists(
                        l|l.observer = c)
}

...
```

Comparison rules (written ECL) compute correspondences between models involved in the compositon

© Copyright TUBS & TUD 2008

Model Composition with Epsilon - Example

```
autorule ClassWithSubject
        merge c : Core!EClass
        with s : ObserverPattern!EClass
        into t : Target!EClass
        extends ClassWithOperationsContributingClass {

        guard : o.name = 'Subject'
}
```

... Merge rules (written in EML) compose models

© Copyright TUBS & TUD 2008

Model Composition Techniques - Summary

- More detail about the presented model composition techniques can be found in:

- MODELPLEX Public Deliverable Document
  D3.1.b: Model Weaving M14

  Available at:
  https://www.modelplex.org//index.php?option=com_remository&Itemid=0&func=fileinfo&id=164

© Copyright TUBS & TUD 2008