

Software Engineering Environment for Business Information Systems

Alar Raabe

Profit Software AS

Contents

- Introduction
- Usage of models in software engineering
- Using extended meta-models and reference models
- Software process steps
- Parts of software engineering environment
- Practical application
- Conclusions

Introduction

- Today's business
 - More dependent on software
 - Constantly changing
- Requirements for business information systems
 - Rapid delivery of initial results
 - Effortless change during the life-cycle
 - Main body of reusable assets should be independent of technologies
- Context
 - Insurance software product-line architecture, tools and methods

Model-Based Approaches

- Main artifact of software development are (implementation technology independent) models
- Model-based approaches:
 - Real-time and embedded systems
 - Model-based software synthesis (Abbott et al., 1993)
 - Model-based development (Mellor, 1995)
 - Integration and interoperability
 - Model-Driven Architecture (MDA) (OMG, 2001)

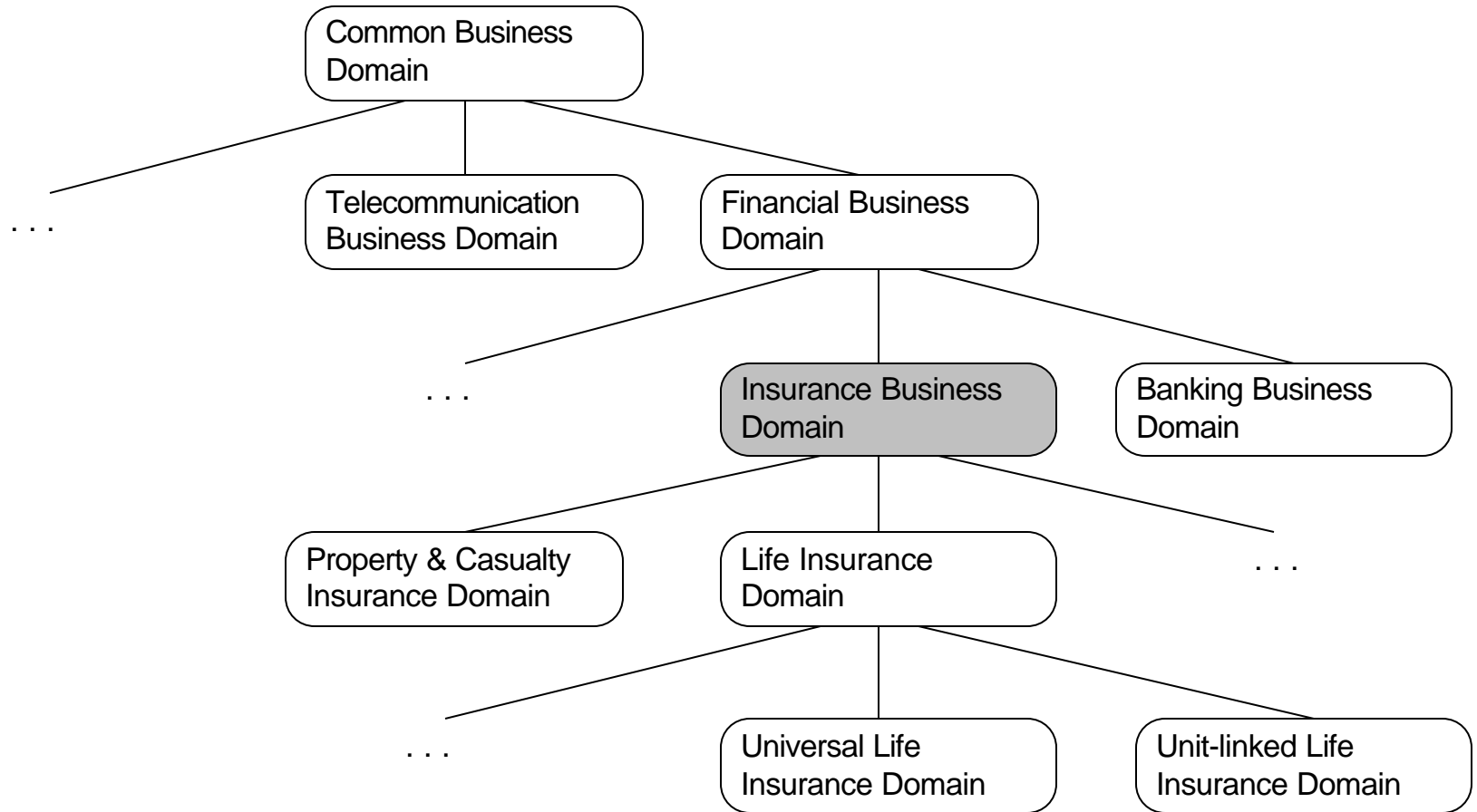
Contents

- Introduction
- Usage of models in software engineering
 - Definitions
 - Traditional approach with common meta-model
- Using extended meta-models and reference models
- Software process steps
- Parts of software engineering environment
- Practical application
- Conclusions

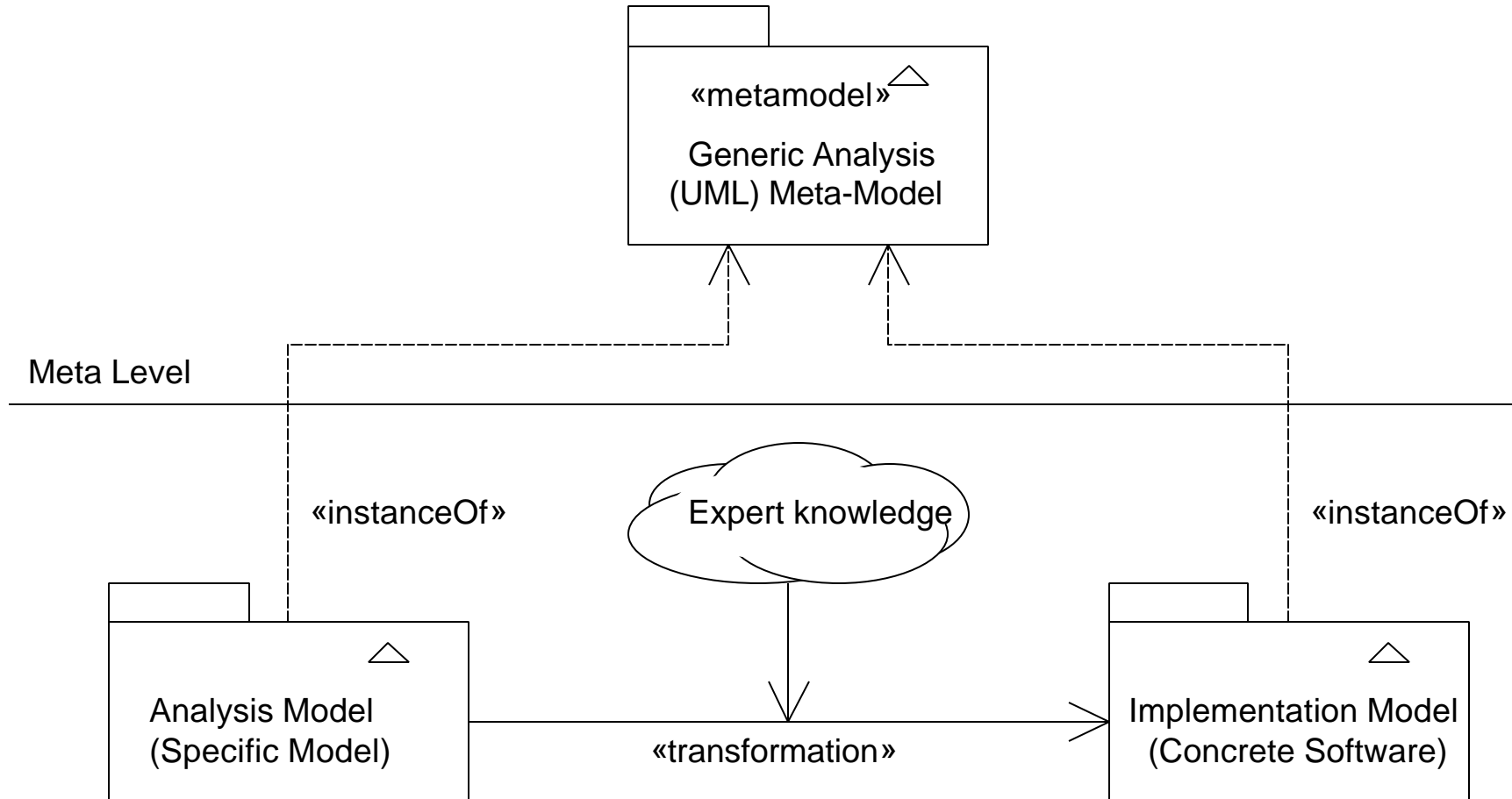
Definitions (UML)

- **Domain** is an area of knowledge or activity characterized by a set of concepts and terminology understood by practitioners in that area
- **Model** is an abstraction of a physical system with a certain purpose (viewed from certain viewpoint)
- **Meta-model** is a model that defines the language for expressing a model
- **Reference model** is a representation of knowledge about the problem domain combined with the standard solutions

Relationship of domains



Traditional Approach



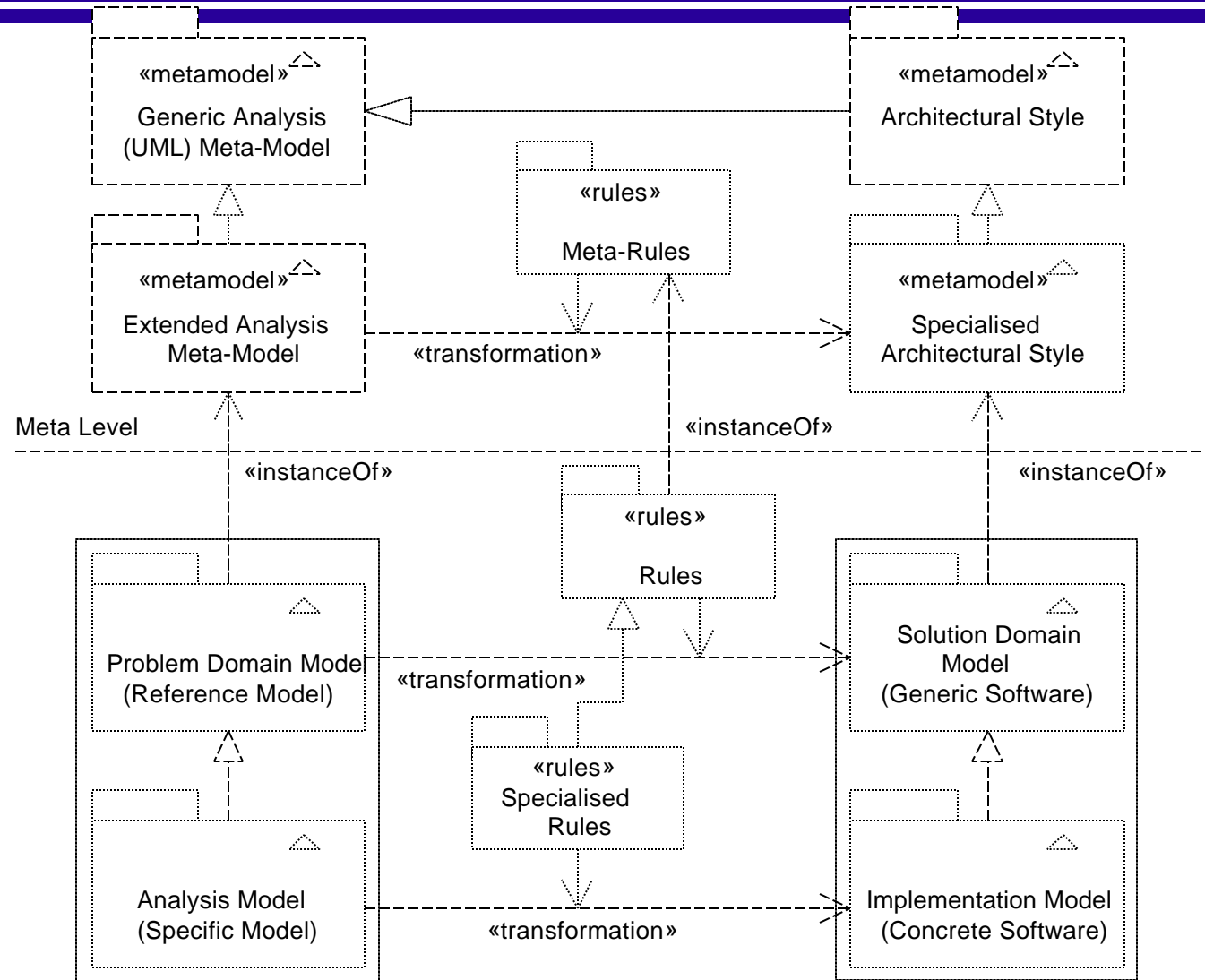
Problems of Traditional Approach

- Analysis model contains implicitly parts of the domain models of all the domains, which the given software system concerns
- Implementation model of a specific software system contains parts from architecture models inherent to the chosen implementation technology

Contents

- Introduction
- Usage of models in software engineering
- **Using extended meta-models and reference models**
 - Need for combination of meta-models
 - Need for combination of models
 - Creation of reference models
- Software process steps
- Parts of software engineering environment
- Practical application
- Conclusions

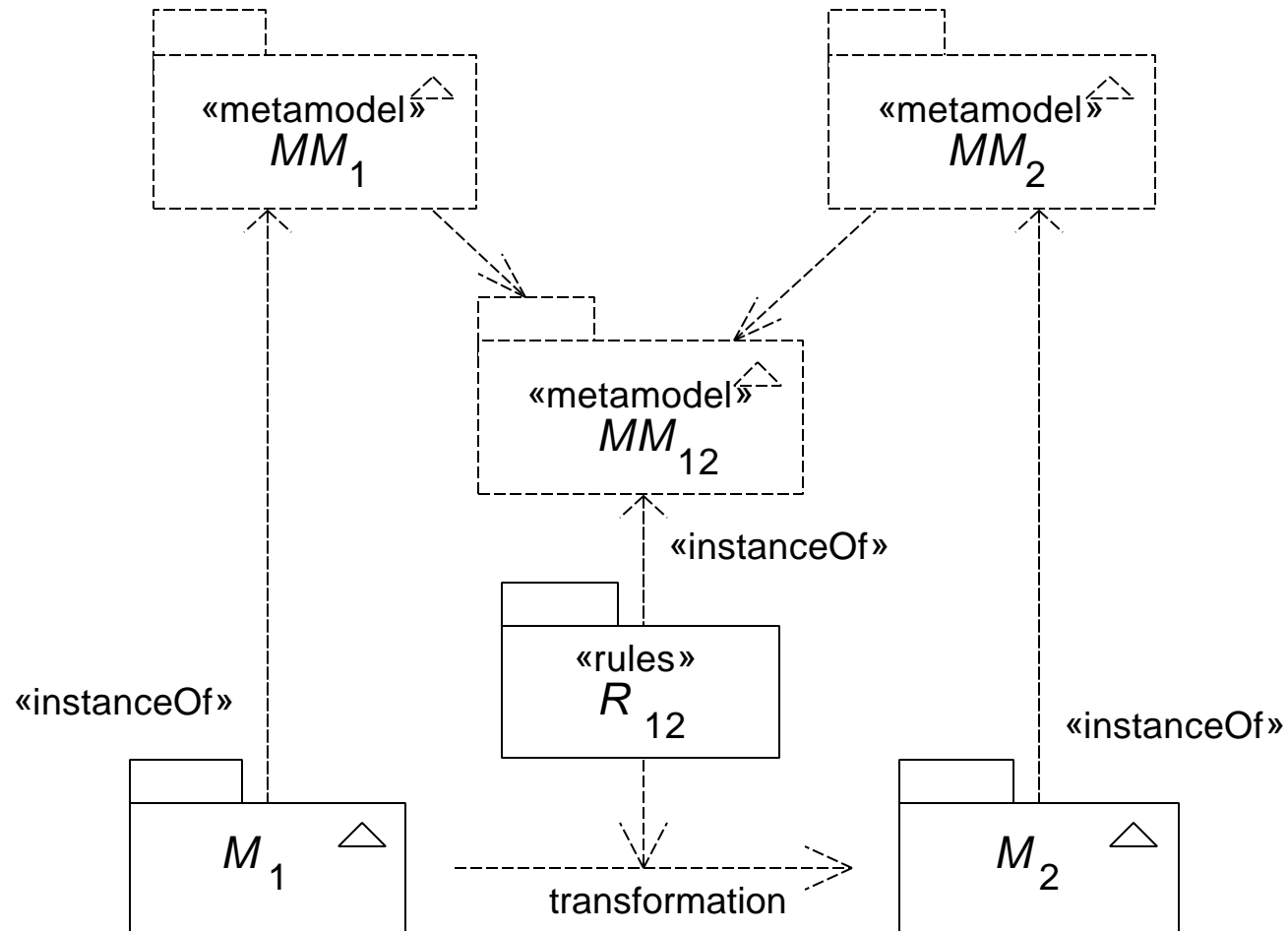
Approach with Extended Meta-Models and Reference Models



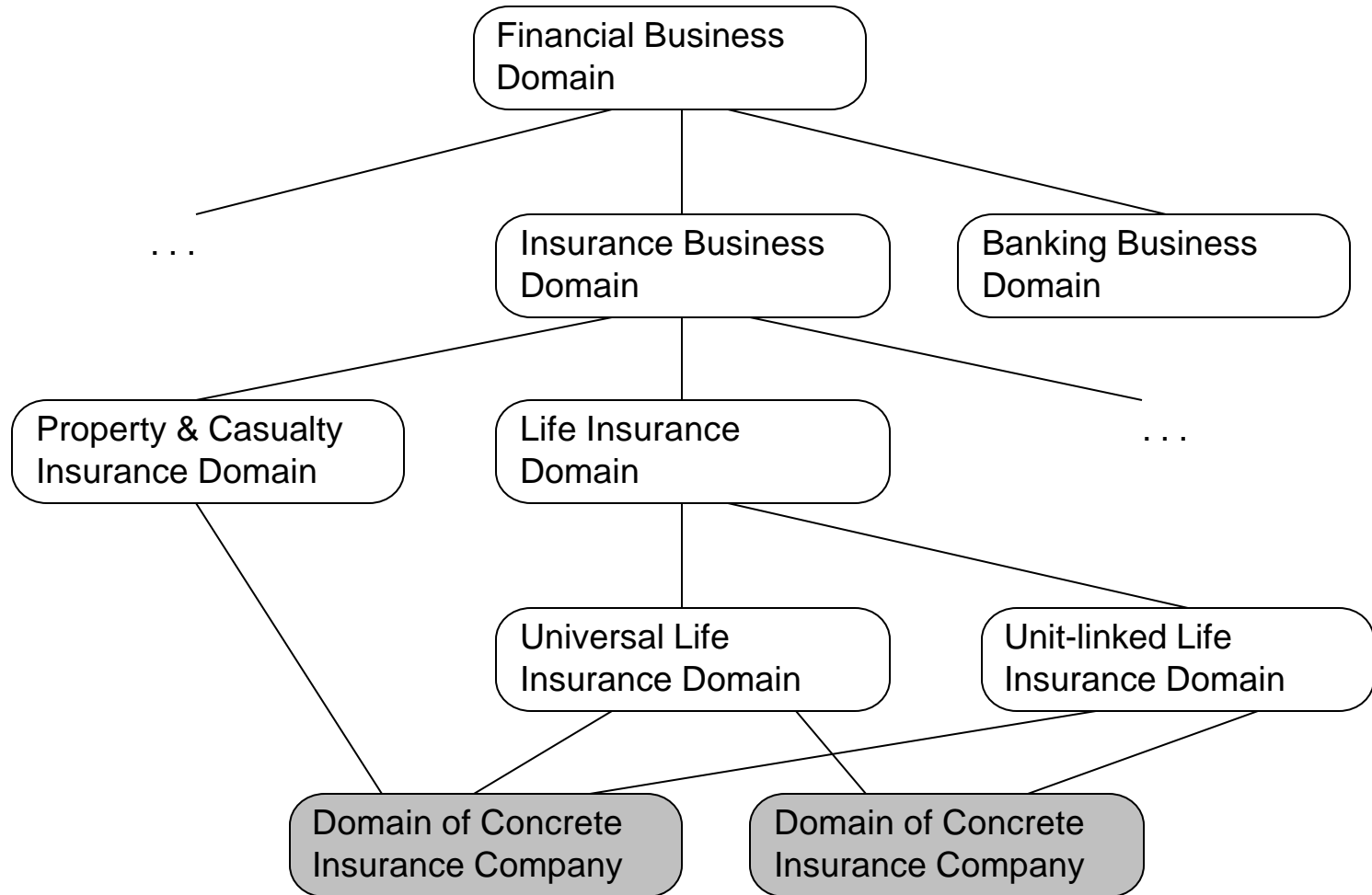
Approach with Extended Meta-Models and Reference Models

- Separation of domain models from analysis model of specific system
- Separation of architecture model from implementation model of specific system
- Context for precise transformation rules between several levels of models

Need for Combination of Meta-Models



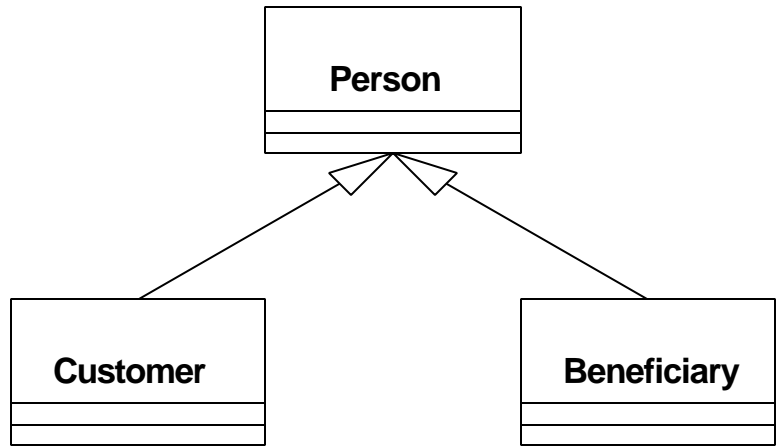
Need for Combination of Models



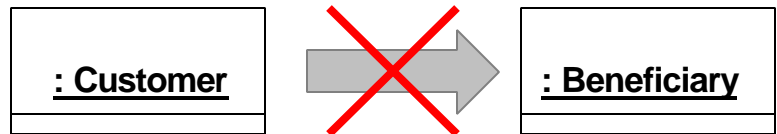
Creation of Reference Models

- Reference models should support combination of models
- Modeling techniques for reference models:
 - Role-Based Modeling
 - Clear identification of extension
 - Separation of variable parts
 - Clustering of model elements

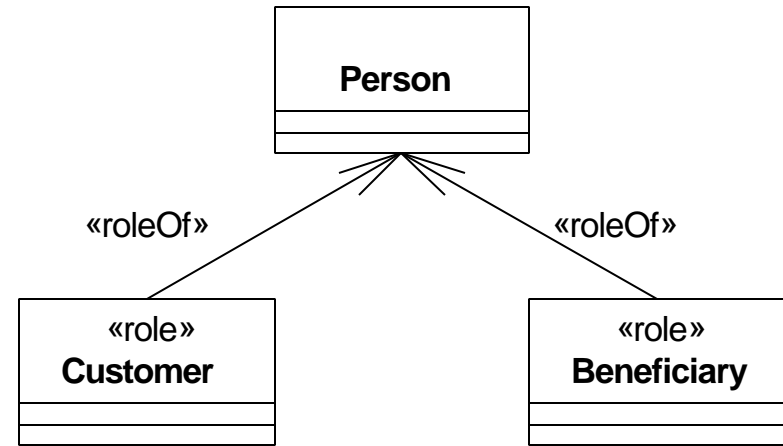
Difference of Roles from Classes



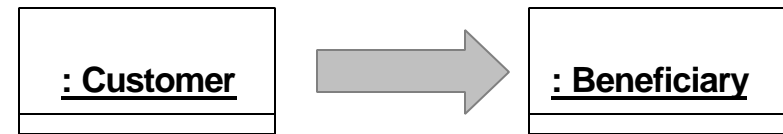
Migration not allowed



Different identities



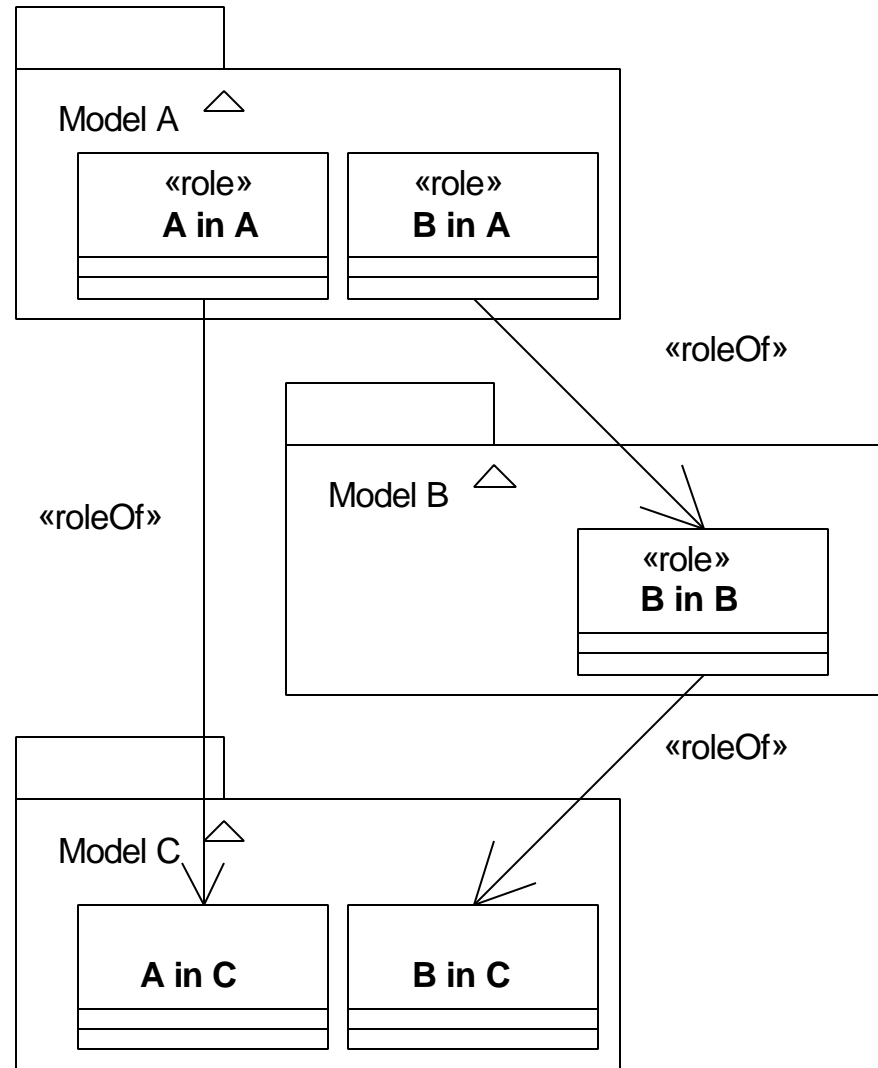
Migration allowed



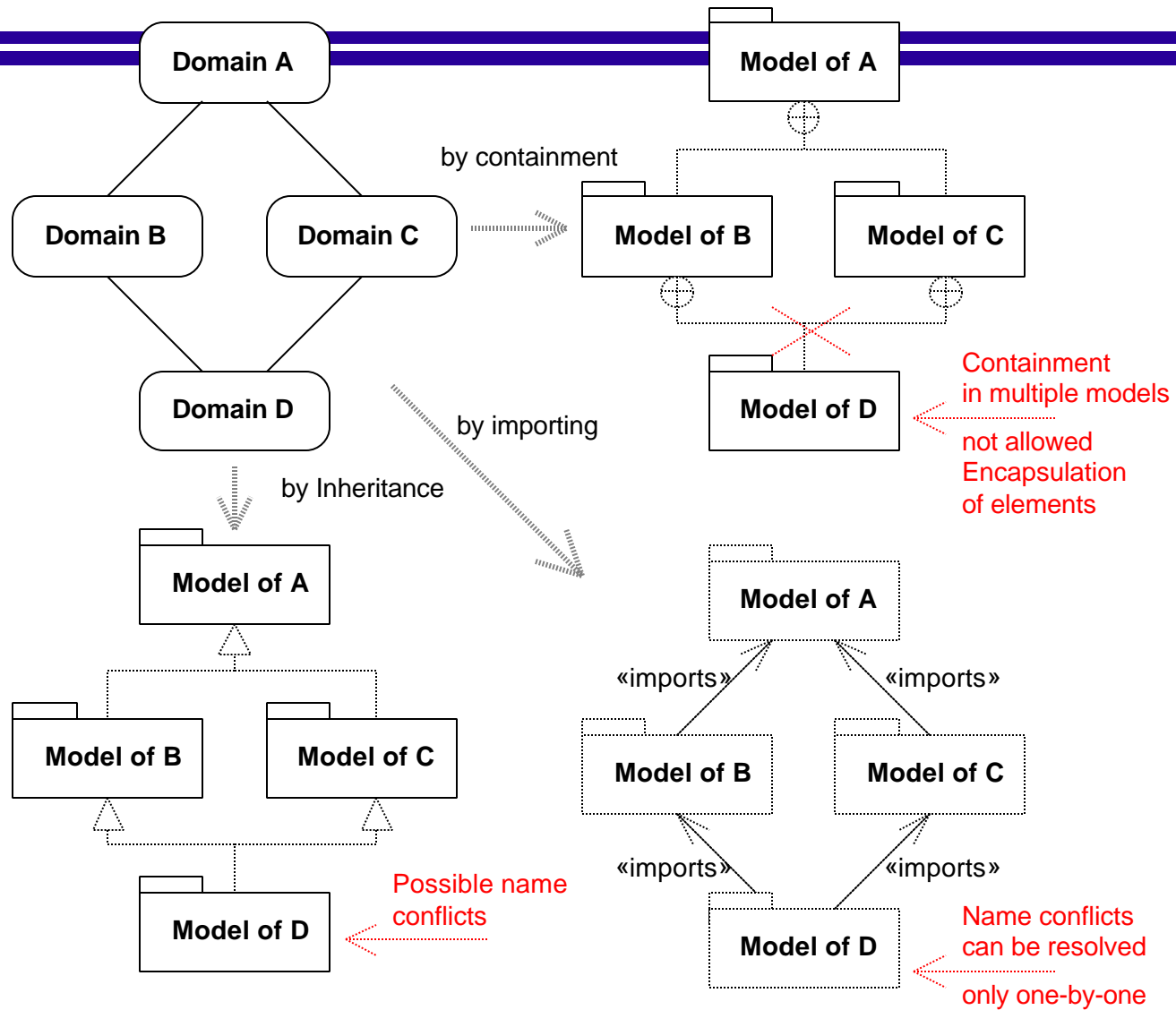
Same identity



Usage of Roles to Support Reuse



Model Combination in UML



Problems of Model Combination in UML

- Name conflicts between elements from different models
- Conflicting model elements (conflicting features, relationships and constraints)
- Cluttered resultant model (because all of the combination methods in UML are only additive)
- Difficulty in changing the used meta-model extensions of the model

Functionality of Model Combination

- Massive renaming
- Selective (filtered) combination
- Deferring of model elements
- Overriding and replacing of model elements

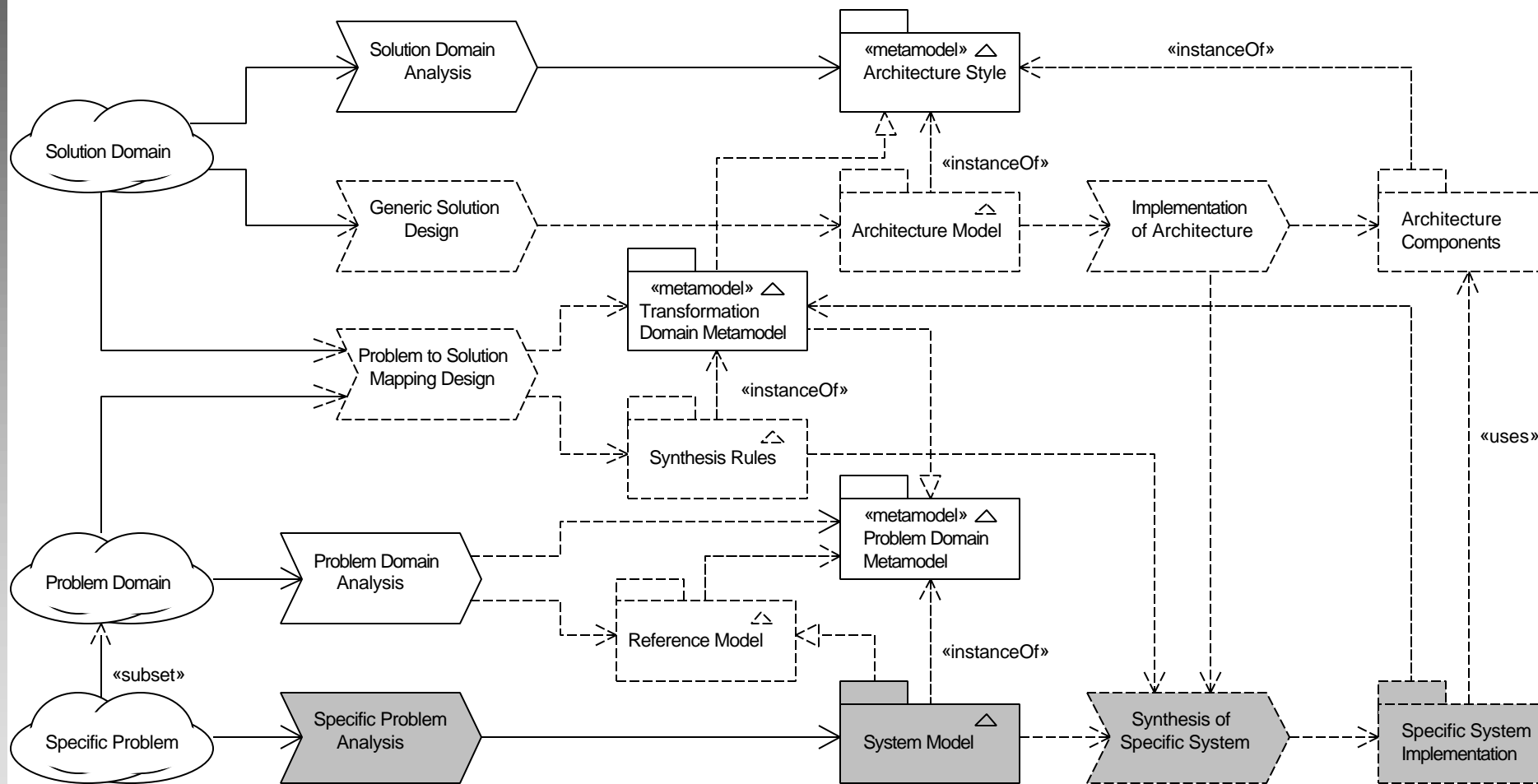
Contents

- Introduction
- Usage of models in software engineering
- Using extended meta-models and reference models
- **Software process steps**
- Parts of software engineering environment
- Practical application
- Conclusions

Software Process Steps

- *Problem domain analysis*
- *Solution domain analysis*
- *Generic solution design*
- *Implementation of architecture*
- *Problem to solution mapping design*
- Specific problem analysis
- Synthesis of a specific system

Steps of Model-Oriented Software Development



Contents

- Introduction
- Usage of models in software engineering
- Using extended meta-models and reference models
- Software process steps
- **Parts of software engineering environment**
- Practical application
- Conclusions

Parts of Software Engineering Environment

- Repository of models
- Tools for manipulating the models and extended meta-models
- Reference models of the needed problem domains
- Changeable implementations of base architectures
- Rule-driven generators which implement the model transformations

Contents

- Introduction
- Usage of models in software engineering
- Using extended meta-models and reference models
- Software process steps
- Parts of software engineering environment
- **Practical application**
- Conclusions

Practical Application

- Once&Done® software environment
 - OD Models
 - Extended meta-models
 - Reference models for insurance domains:
Non-life (Property and Casualty), Life and Claims
 - OD Tools
 - Repository of models (extended meta-models)
 - Model combination tool
 - Rule driven generators
 - OD Framework
 - OD Process

Practical Application

- Once&Done® product-line (1995-2001)
 - 4 Systems for Property and Casualty Insurance
 - 3 Systems for Life Insurance
 - Claim Handling System
- Once&Done® models
 - Property and Casualty Reference Models
 - Private (380 entities + 394 relations)
 - Commercial (569 entities + 894 relations)
 - Life Reference Model
 - Claims Reference Model (96 entities + 43 relations)
- Change of technology
 - Client-Server → Three-tier → Web-based

Contents

- Introduction
- Usage of models in software engineering
- Using extended meta-models and reference models
- Software process steps
- Parts of software engineering environment
- Practical application
- **Conclusions**

Conclusions

- Differences from other methods:
 - Separate step of solution domain analysis
 - Systematic use of meta-model extensions and reference models
 - Combination operations on models
- In line with OMG MDA approach
 - Clarifies the role of meta-model extensions
 - Offers a clear development process to support the MDA approach
 - Defines model operations for combination of models
 - Offers a framework for transformation rules on different model levels

Thank You!

Questions?