



Meta-modeling and meta-model extensions in the construction of software

Alar Raabe

Contents

- 1. Background
- 2. Meta-model extensions
 - Meta-model extensions in UML
- 3. Usage of meta-model extensions
- 4. Combining of meta-model extensions
- 5. Conclusions

1. Background

- 1.1. Insurance applications
 - Insurance business domain as an example of problem domain
 - big and complex domain models
 - need for quick application construction
- 1.2. Convergent engineering
 - Model-based business systems
- 1.3. Product-line architecture
 - Reuse of architecture
- 1.4. Application – Once&Done®

1.1. Insurance applications

- Insurance business domain as an example of problem domain
 - Big and complex domain models
⇒ guidance in analysis
 - Big and complex systems
⇒ guidance in design
 - Need for quick application construction
⇒ synthesis/generation of applications
 - Need for portability to new technologies
⇒ implementation independent descriptions

1.2. Convergent engineering

- Model-based business systems
 - Simplify the software engineering process and reduce the total amount of work
 - only single system that implements business model will be implemented
 - Eliminate the gaps between business processes and their supporting software
 - same system for execution, forecast and representation of business and business processes
 - Facilitate change by minimizing the problem of coordinating modifications

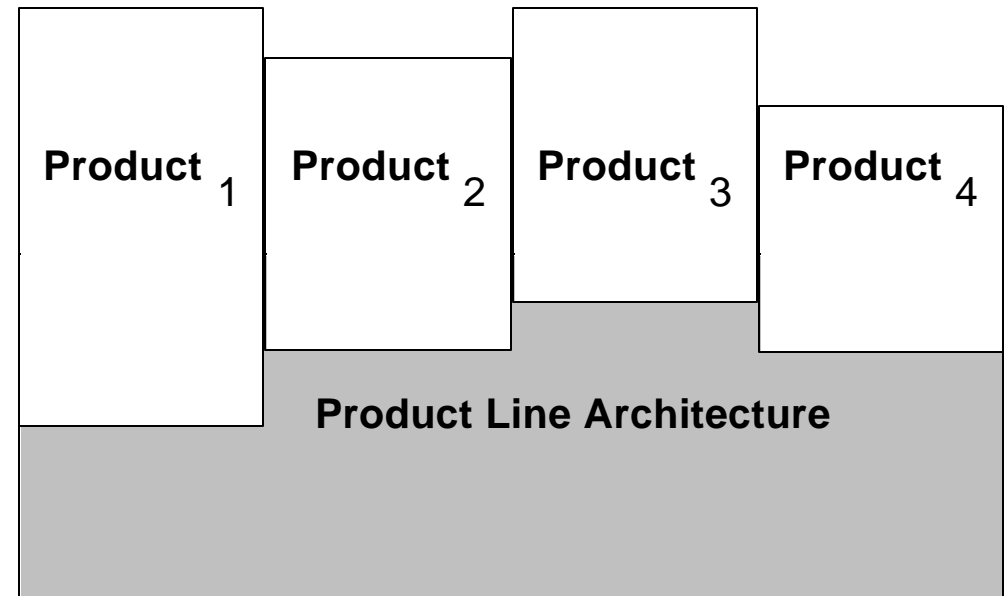
1.3. Product-line architecture

- Architecture
 - Set of components and interactions between these components
 - Organizational structure and associated behavior of a system
 - Sum of design decisions
- Product-line architecture
 - Sum of early design decisions
 - Defines what is fixed for all product-line members and what is variable

1.3. Product-line architecture

■ Reuse of architecture

- components
- personnel
- project planning
- defect elimination
- performance issues
- processes, methods, tools
- exemplar systems (demonstration prototypes)



1.4. Application – Once&Done®

- Once&Done® Architecture for Insurance Applications (a product-line architecture)
 - Once&Done® models
 - extended OOA and OOD meta-models
 - reference models for insurance domain
 - Once&Done® framework
 - reusable components
 - Once&Done® process
 - Once&Done® tools
 - repository (implementing extended meta-models)
 - generators (using extended models)

2. Meta-model extensions

- 2.1. Model and Meta-model
- 2.2. Four layer meta-modeling architecture
- 2.3. Meta-model extensions in UML
 - UML 1.3 extension mechanisms
 - MOF 1.3

2.1. Model and Meta-model

- Model is ...
 - Model is a representation in a certain medium of something in the same or another medium
 - Model is a language for expressing something
 - Model is meta-data (data about data)
- Meta-model is ...
 - Meta-model is a model that defines the language for expressing a model
 - Meta-model defines the structure and semantics of meta-data (data that describes information)

2.2. Four layer meta-modeling architecture (OMG)

Model layer	Description
M ₃ – meta-meta-model (MOF meta-model)	describes meta-models; defines language for meta-models
M ₂ – meta-model (UML meta-model)	describes models; defines language for models
M ₁ – model (UML model)	describes data; defines language for data
M ₀ – data	

2.3. Meta-model extensions in UML

- Meta-model extensions of UML meta-model
 - Implicit meta-model extensions (extending UML meta-model in UML)
 - stereotypes
 - constraints
 - tagged values
 - profile
 - Explicit meta-model extensions (extending UML meta-model in MOF)
 - new meta-model elements
 - sub-classing of meta-model elements

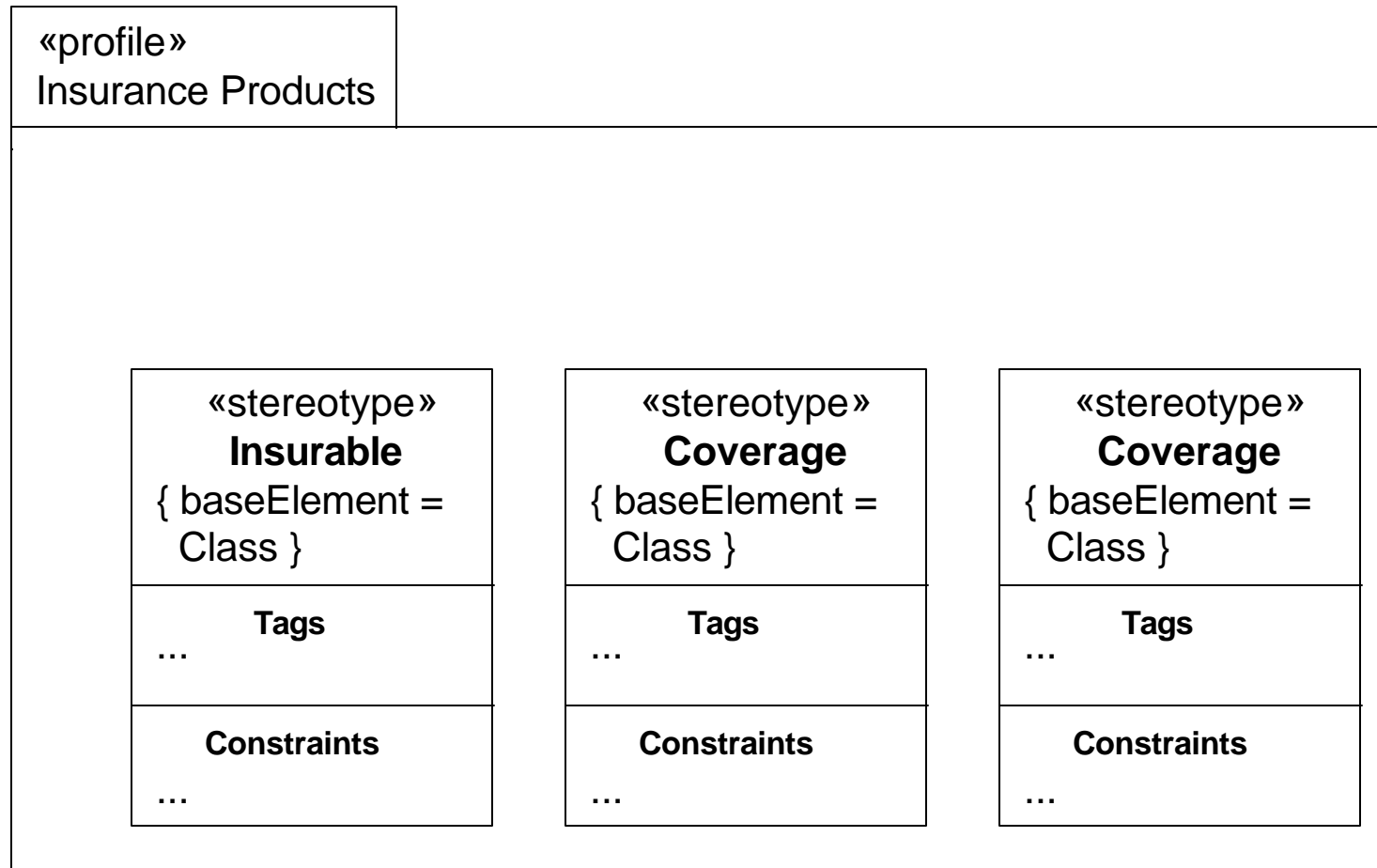
3. Usage of meta-model extensions

- 3.1. Example meta-model extension for describing insurance products
 - using UML profile
 - using MOF
- 3.2. Meta-model extensions in analysis
 - domain specific meta-model extensions
- 3.3. Meta-model extensions in design
 - architecture elements
 - design decisions
- 3.4. Meta-model extensions in implementation
- 3.5. Meta-model extensions in other areas

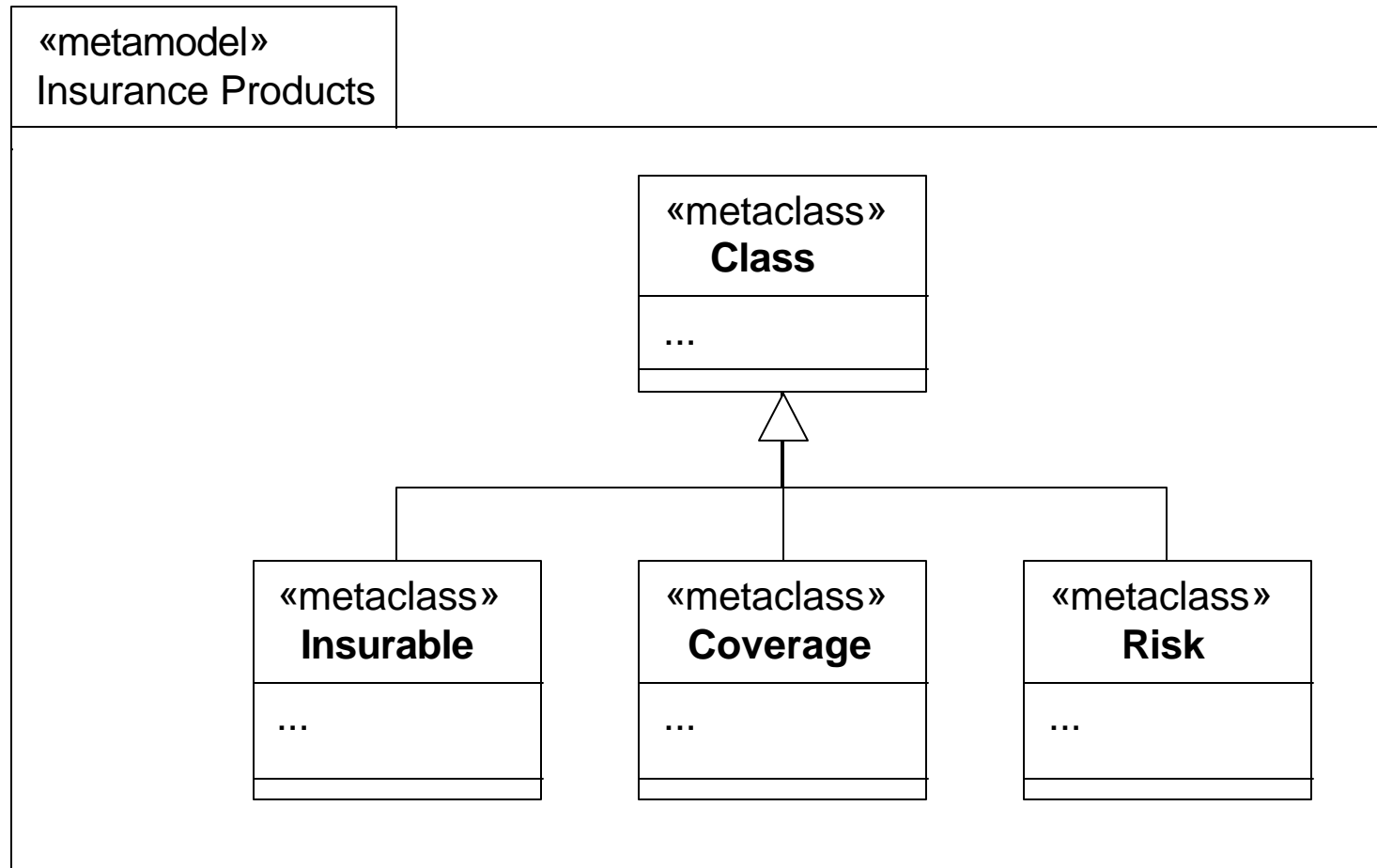
3.1. Example meta-model extension for describing insurance products

- Insurable
 - Insurable interests
- Coverage
 - Insurance products covering insurable interests
- Risk
 - Risks against which the insurable interests can be insured

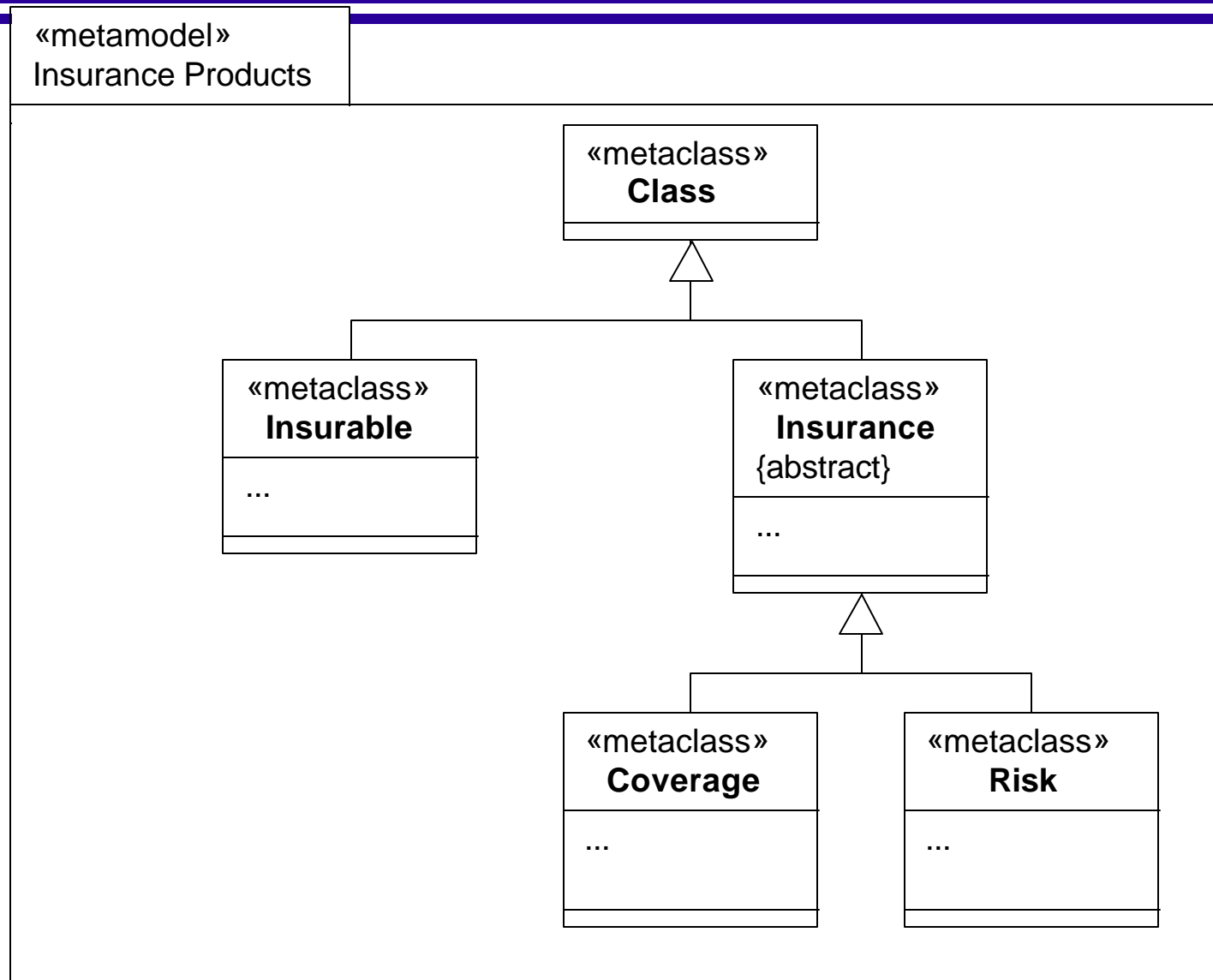
3.1.1. Example meta-model extension as UML profile



3.1.2. Example meta-model extension using MOF (1)



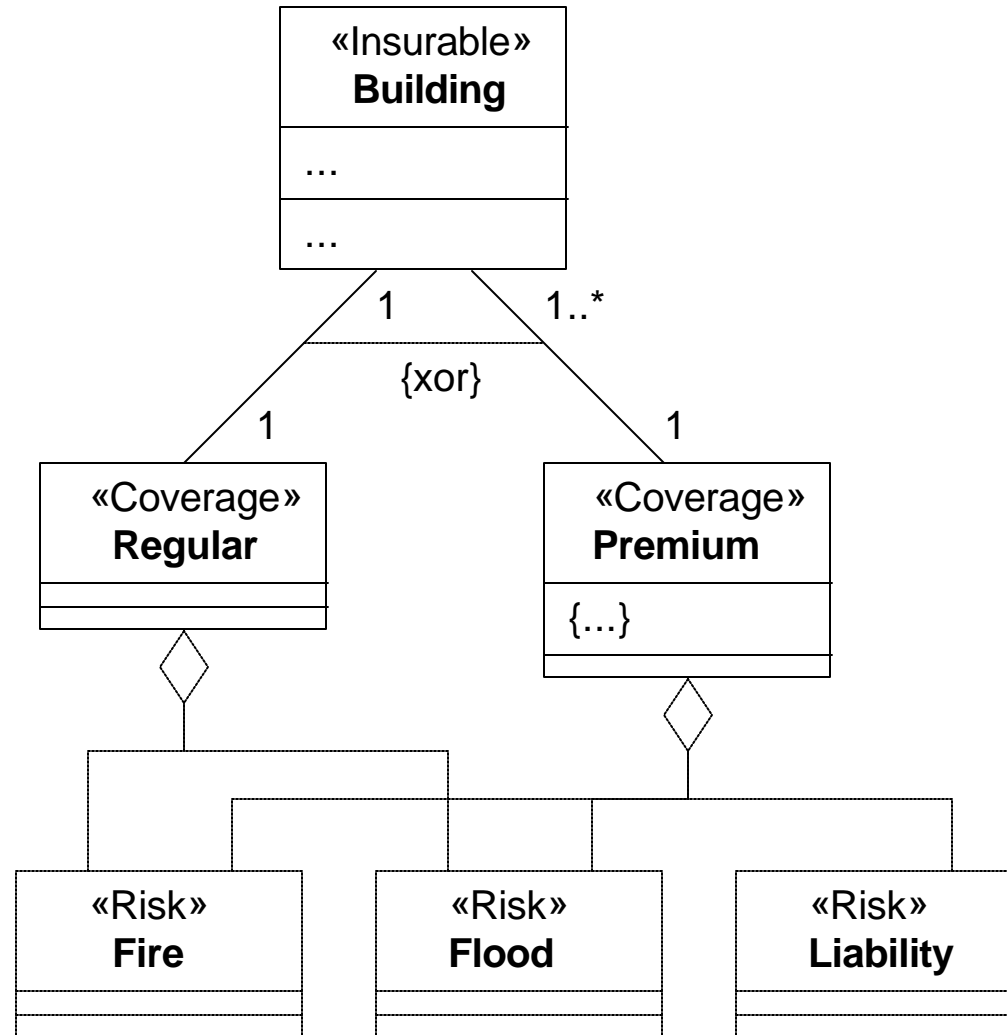
3.1.2. Example meta-model extension using MOF (2)



3.2. Meta-model extensions in analysis

- Domain specific meta-model extensions
 - Guide analysis process for a given domain
 - Formalize domain knowledge
 - Example:
Model of a simple insurance products
 - Regular building coverage can cover one building against fire and flood
 - Premium building coverage can cover several buildings which are not covered by regular coverage against fire, flood and owners liability
 - Premium building coverage is more constrained than ordinary coverage (e.g. has additional eligibility rules)

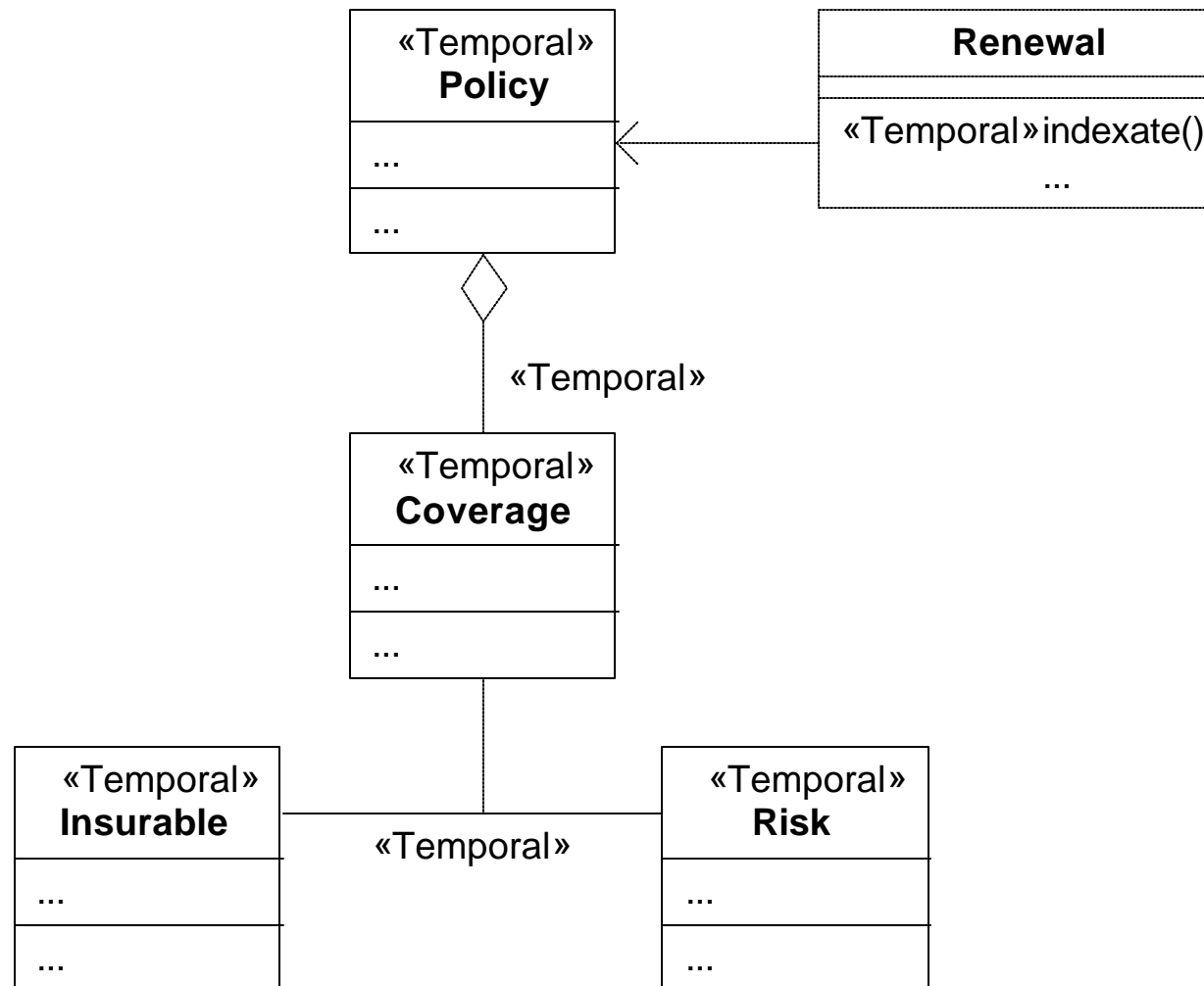
3.2.1. Model of a simple insurance product



3.3. Meta-model extensions in design

- Architecture elements
 - Guide design toward certain architecture
- Design decisions
 - Representation and application of certain design decisions
 - Example:
Model of an insurance policy – showing temporal semantics of model elements
 - Insurance companies need to manage data in two orthogonal time dimensions: validity time and transaction time

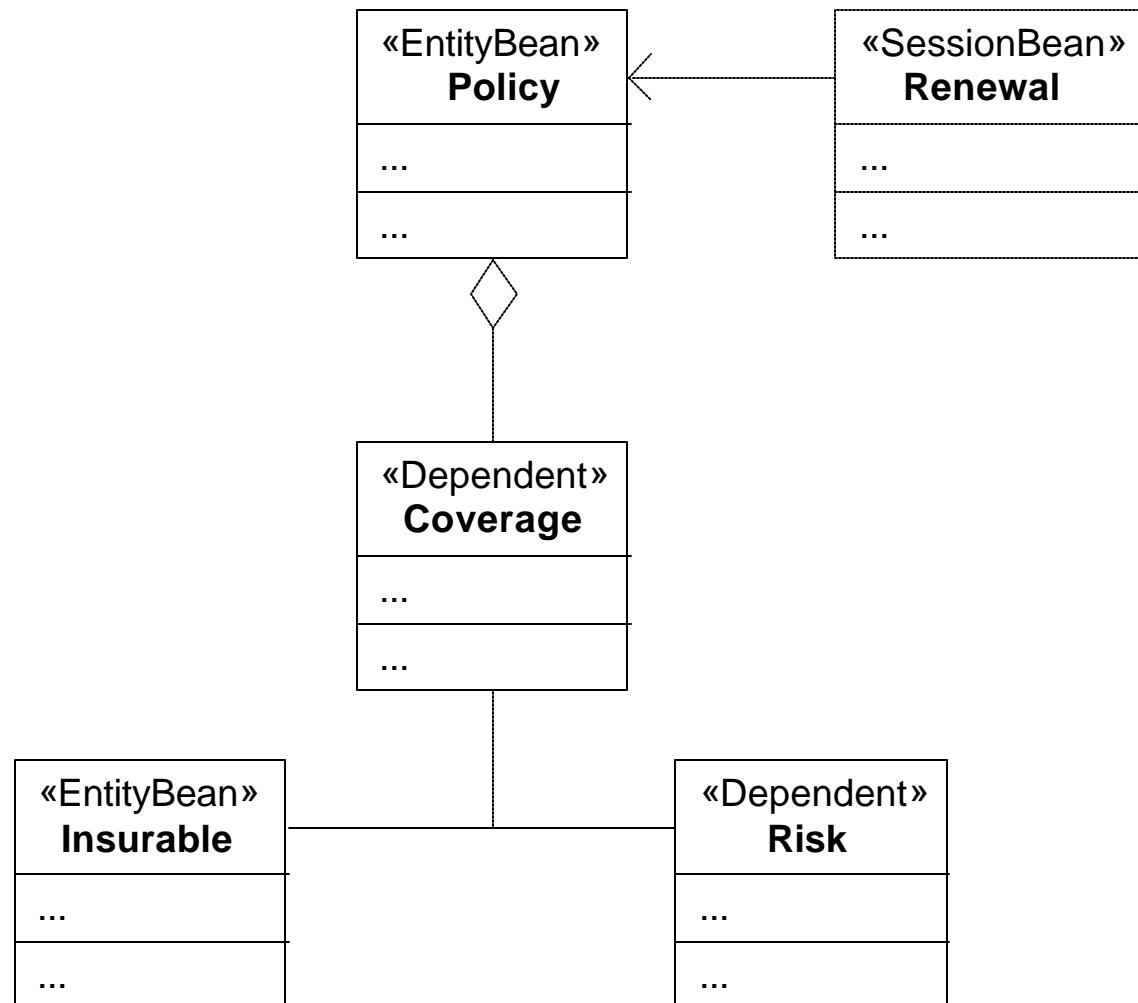
3.3.1. Model of an insurance policy (showing temporal elements)



3.4. Meta-model extensions in implementation

- Implementation decisions
- Implementation details
- Driving implementation generators

3.4.1. Model of an insurance policy (showing implementation decisions)



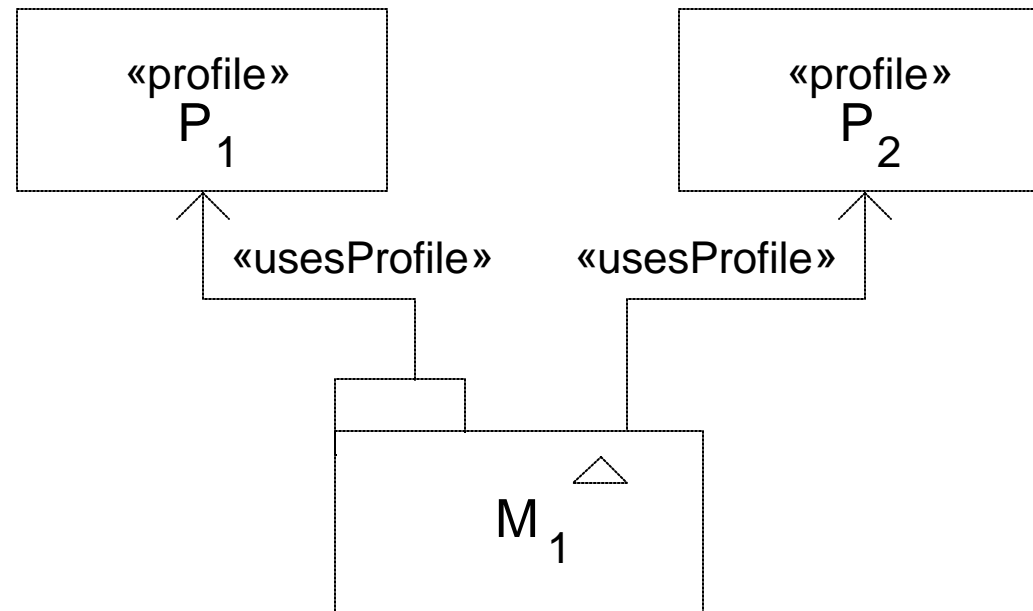
3.5. Meta-model extensions in other areas

- Software process organization
- Software process modeling
- Distributed systems modeling
- Test planning
- Data warehousing

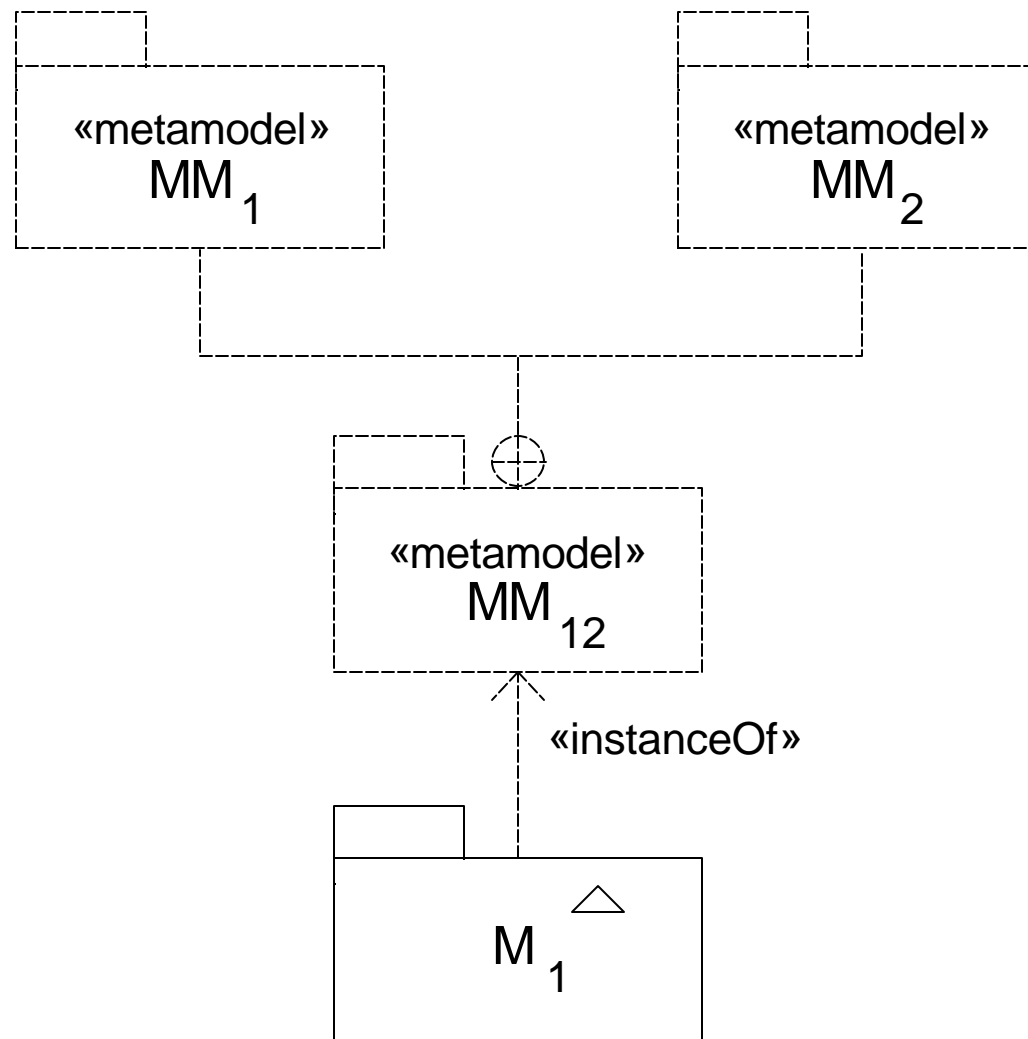
4. Combining of meta-model extensions

- 4.1. Combining of meta-model extensions
 - combining of multiple profiles
 - combining meta-models
- 4.2. Example of combining meta-model extensions
- 4.3. Problems
- 4.4. Solutions

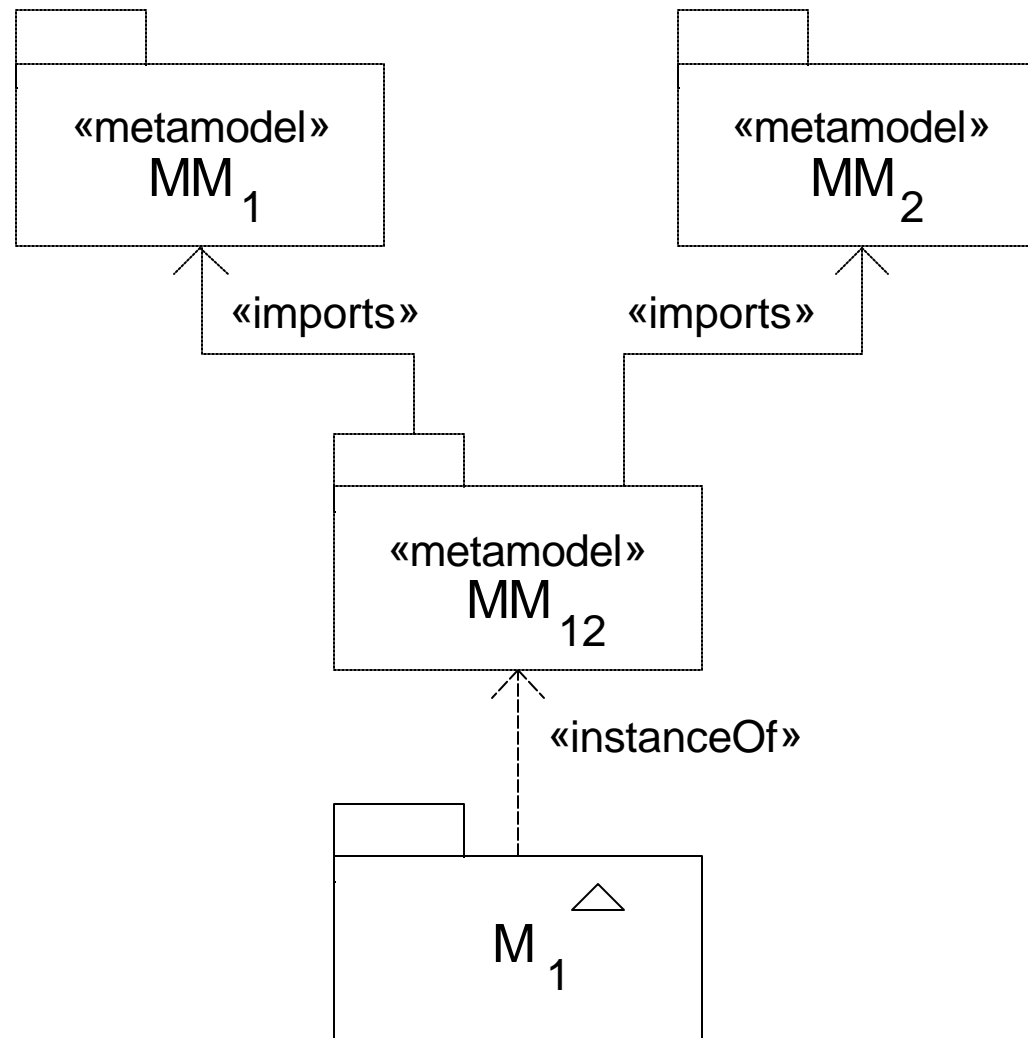
4.1.1. Combining of multiple profiles



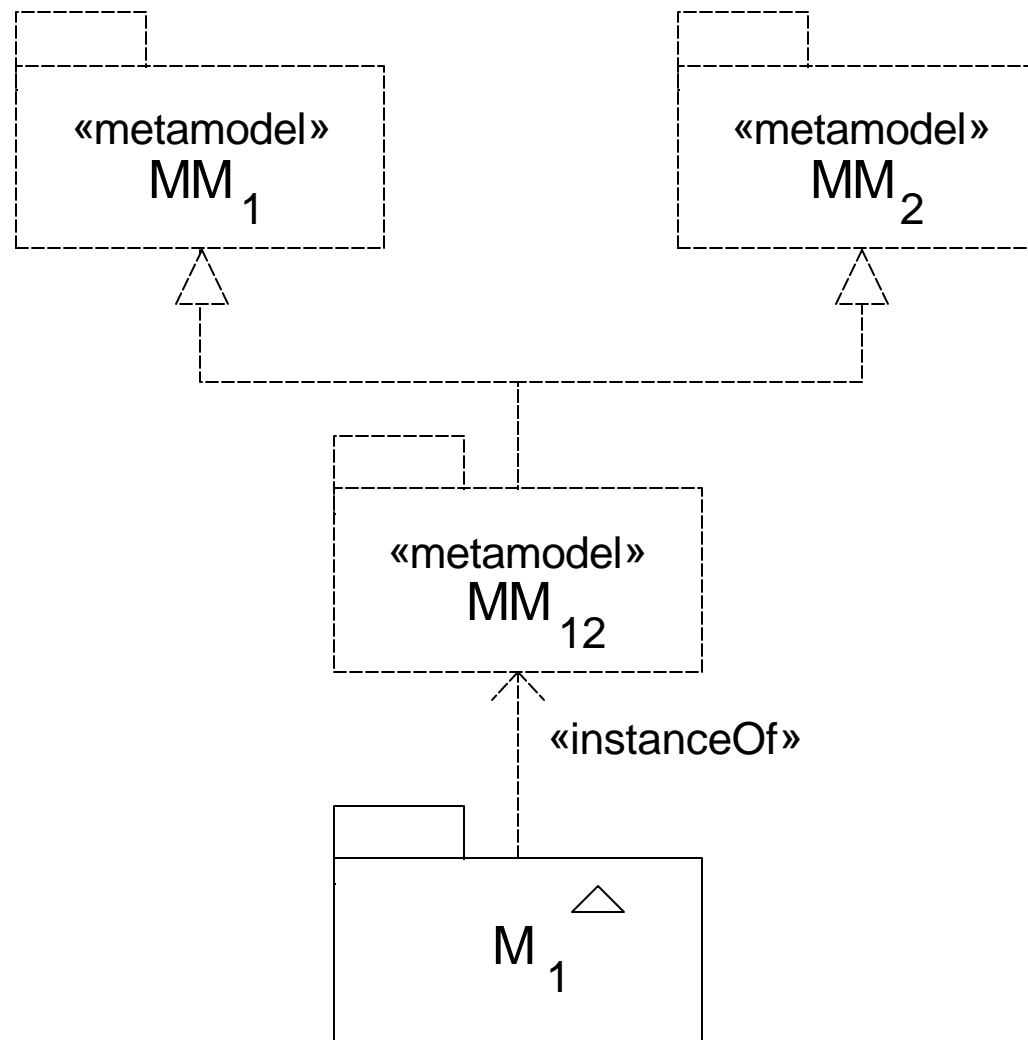
4.1.2. Combining of meta-models (containment)



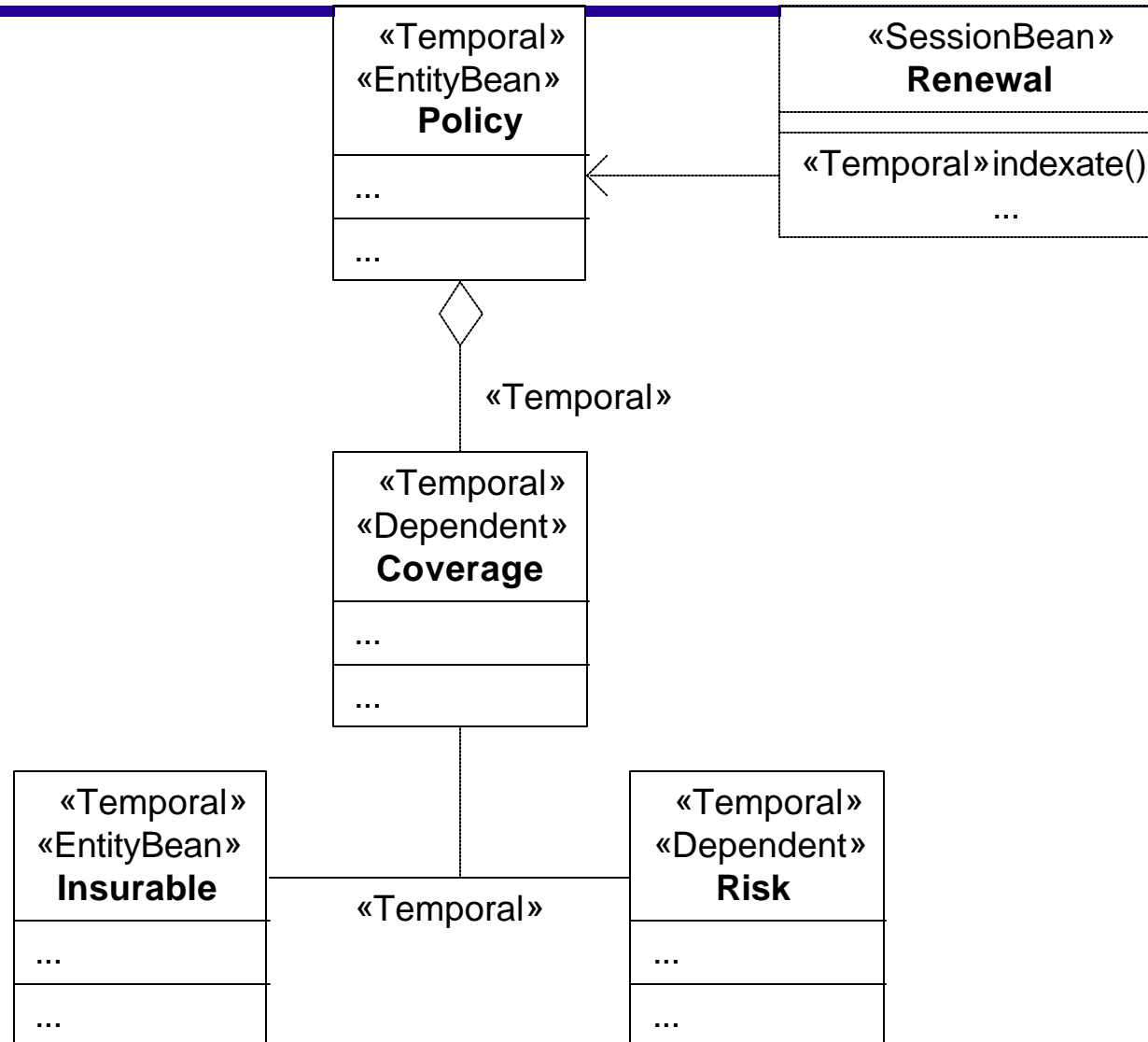
4.1.3. Combining of meta-models (importing)



4.1.4. Combining of meta-models (multiple inheritance)



4.2. Example of combining meta-model extensions (multiple profiles)



4.3. Problems

- Problems of meta-model extension methods
 - Problems of implicit meta-model extensions (UML profiles)
 - Problems of explicit meta-model extensions (using MOF)
- Problems of combining meta-model extensions

4.3.1. Problems of implicit meta-model extensions (UML profiles)

- Can only be used to extend UML meta-model strictly additively
- Meta-model elements are mixed with model elements
- Only one stereotype per model element is allowed
- One global namespace for meta-model elements and stereotypes
- One global namespace for tags
- Tagged values have no type

4.3.2. Problems of explicit meta-model extensions (using MOF)

- No support from “off-the-shelf” tools
- Model interchange between tools needs, that extended meta-model is available for both tools
- Harder to combine and apply to same model
 - Several options for combining meta-models:
 - containment
 - importing
 - multiple inheritance

4.3.3. Problems of combining meta-model extensions

- Name conflicts
- Conflicting meta-model elements
 - Conflicting features
 - Conflicting relationships
 - Conflicting constraints
- Cluttered meta-model (because all combination techniques are additive)
- Difficult to change

4.4. Solutions

- Solutions for extending meta-models
 - implicit meta-model extensions
 - explicit meta-model extensions
- Solutions for combining meta-model extensions

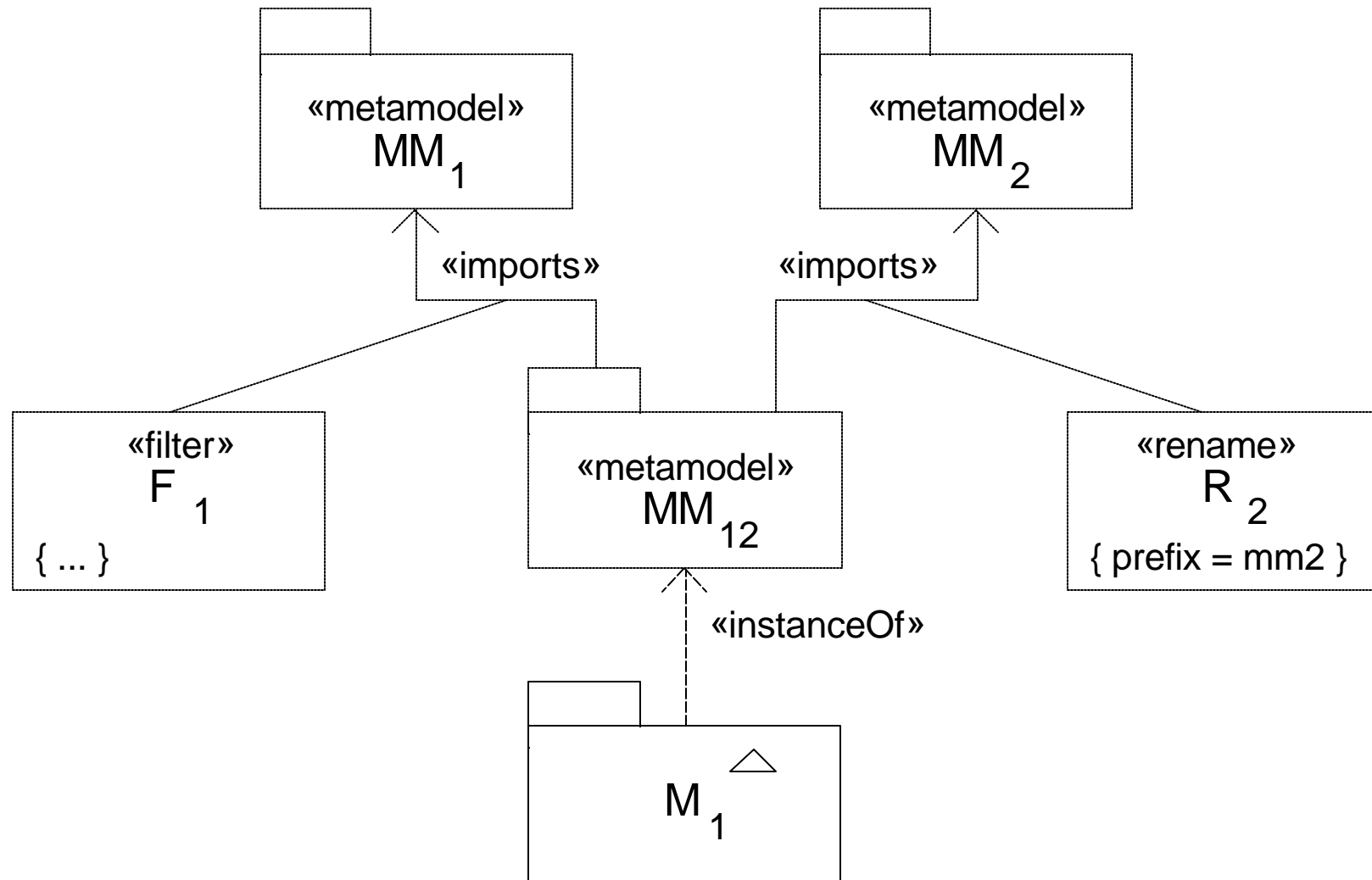
4.4.1. Solutions for extending meta-models

- For implicit meta-model extensions
 - Stereotypes should always be contained in a profile (package with the stereotype “profile”)
 - Multiple stereotypes per model element should be allowed
 - Profile should act as namespace for stereotypes and tags
 - Tagged values should have type
- For explicit meta-model extensions
 - Global meta-model registry to facilitate interoperability between tools

4.4.2. Solutions for combining meta-model extensions

- To resolve name conflicts introduce
 - Import and inheritance mechanisms for meta-models should allow massive renaming
 - A global name registry could be established (e.g. similar for Internet domain names)
- To resolve conflicting meta-model elements introduce
 - Overriding of meta-model elements
 - Replacing of meta-model elements
 - Deferring of meta-model elements
- To resolve meta-model clutter introduce
 - Selective import and inheritance

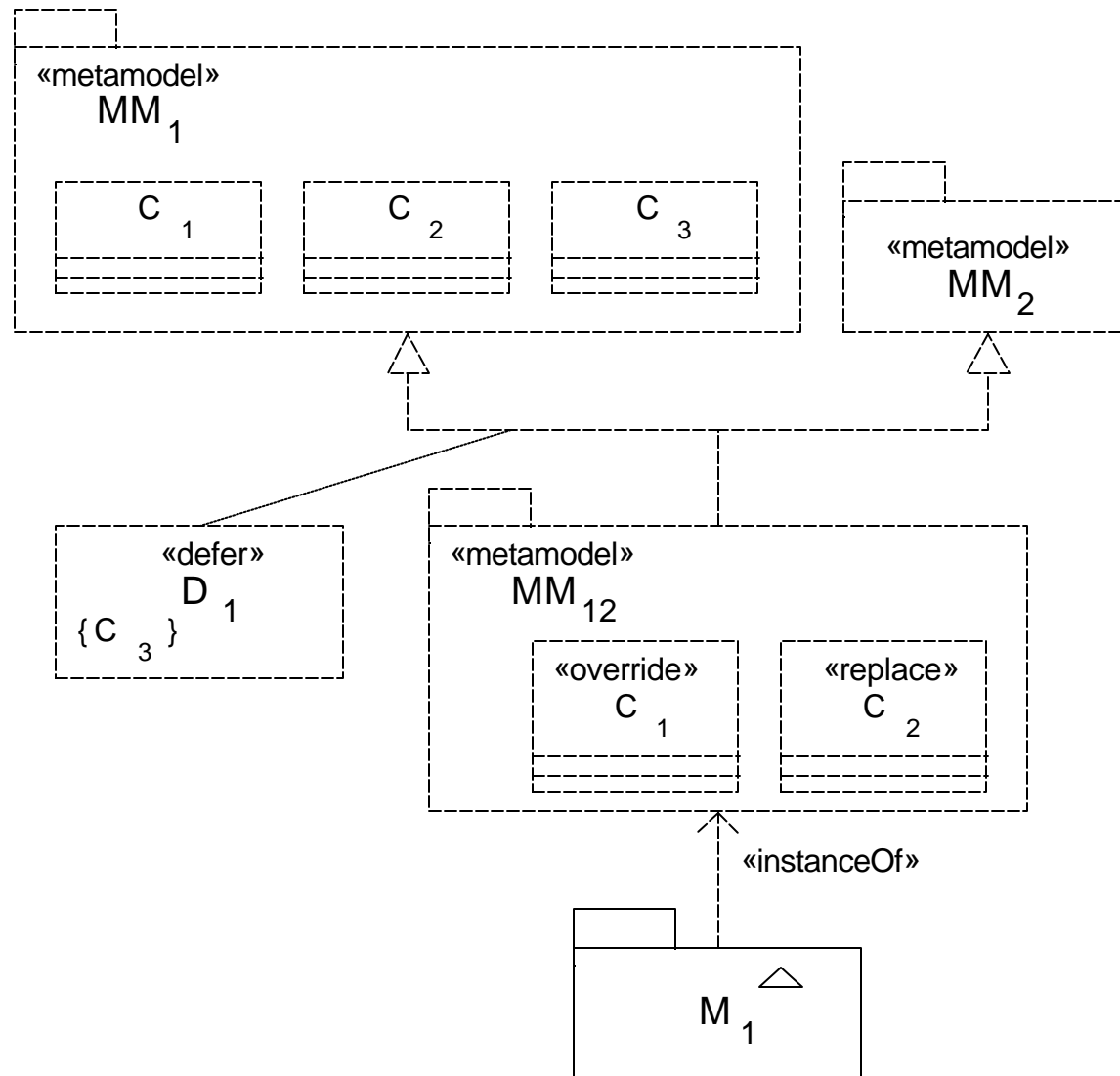
4.4.3. Combining of meta-models (modified importing)



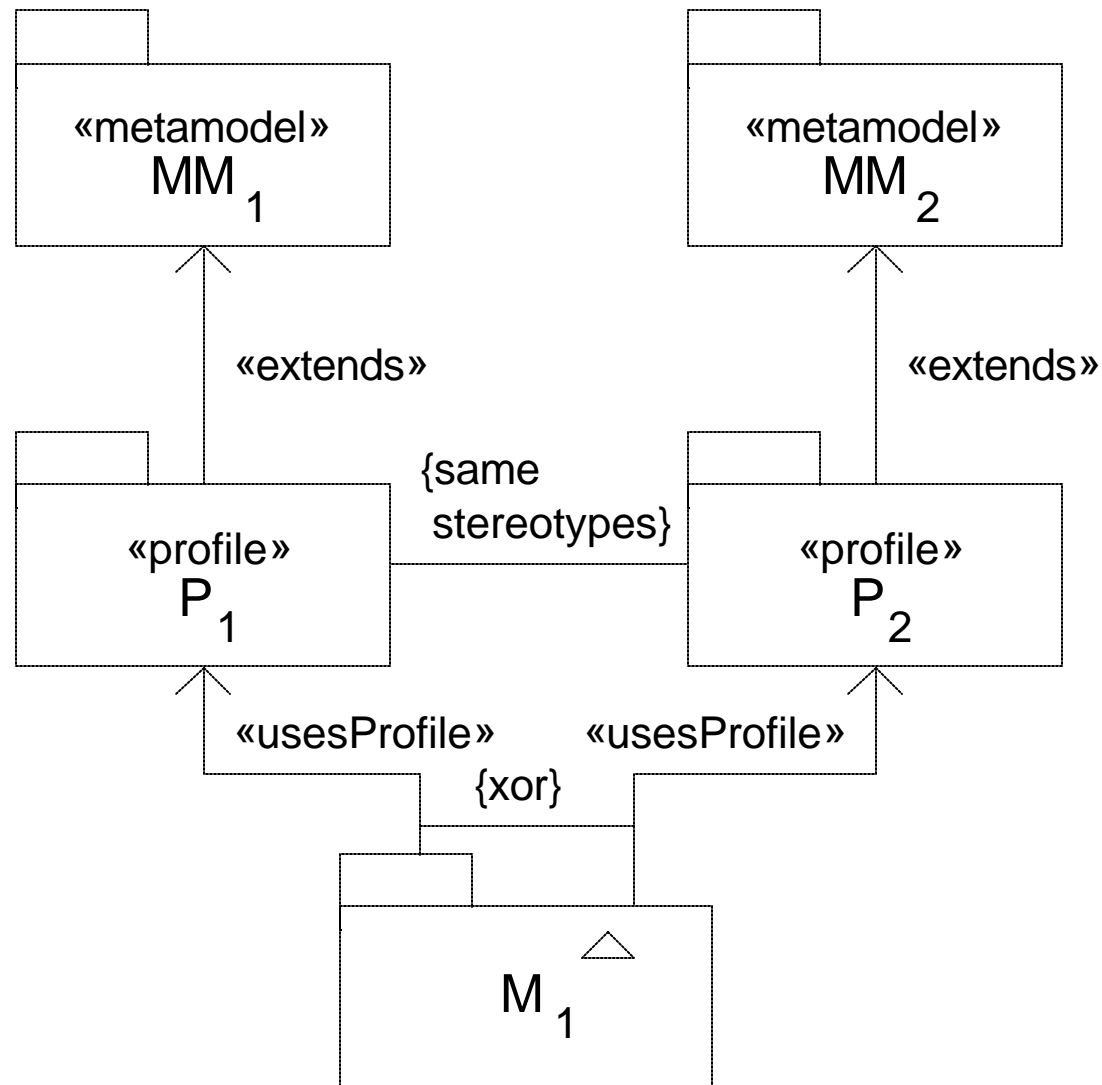
4.4.4. Resolving conflicts between meta-model elements

- Inheritance of meta-models
 - all public or protected elements in owned or imported by the ancestor are also available in the specialized model under the same name and interrelated as in the ancestor
- Modified inheritance of meta-models
 - *Overriding* – meta-model element in ancestor is masked by the meta-model element in child
 - *Replacing* – meta-model element in ancestor is replaced by the meta-model element in child (for instantiations of child)
 - *Deferring* – meta-model element in ancestor is removed/suppressed in child

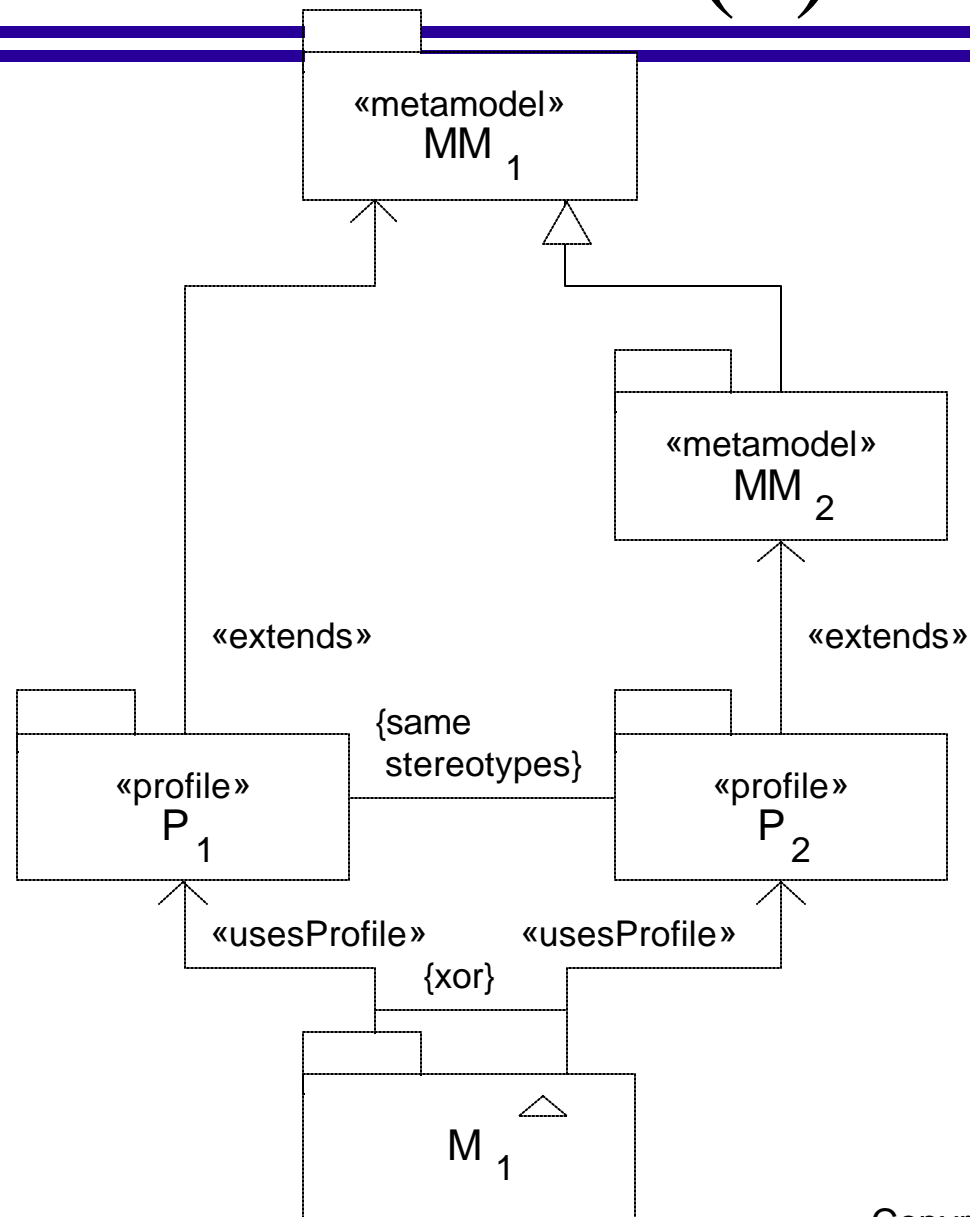
4.4.5. Combining of meta-models (modified inheritance)



4.4.6. Using profiles as interfaces to meta-model extensions (1)



4.4.6. Using profiles as interfaces to meta-model extensions (2)



5. Conclusions

- Meta-model extensions are useful in various phases of software development
- There is need to combine meta-model extensions
- Extended (meta-)model import facility in UML has been proposed
- Extended semantics of (meta-)model inheritance in UML has been proposed
- A method of combining meta-model extensions in UML has been proposed

A vertical grey bar is on the left side of the slide. Two horizontal blue lines cross the top of the slide.

Thank You



Discussion

- Questions
- Suggestions