Software Engineering
Model Driven Architecture
Applications of Metamodeling

Prof. Dr. Peter Thiemann, Stefan Wehr

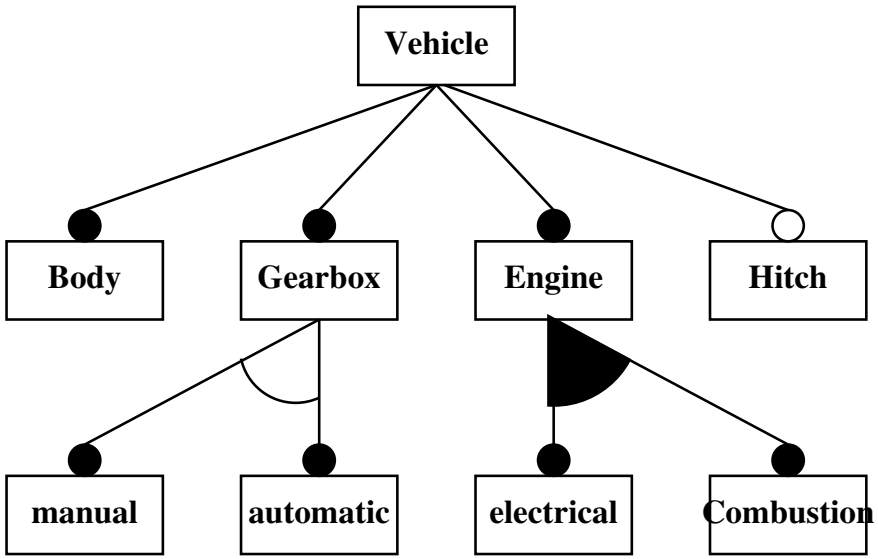
Universität Freiburg

18.07.2008

Applications of Metamodeling
Feature Modeling

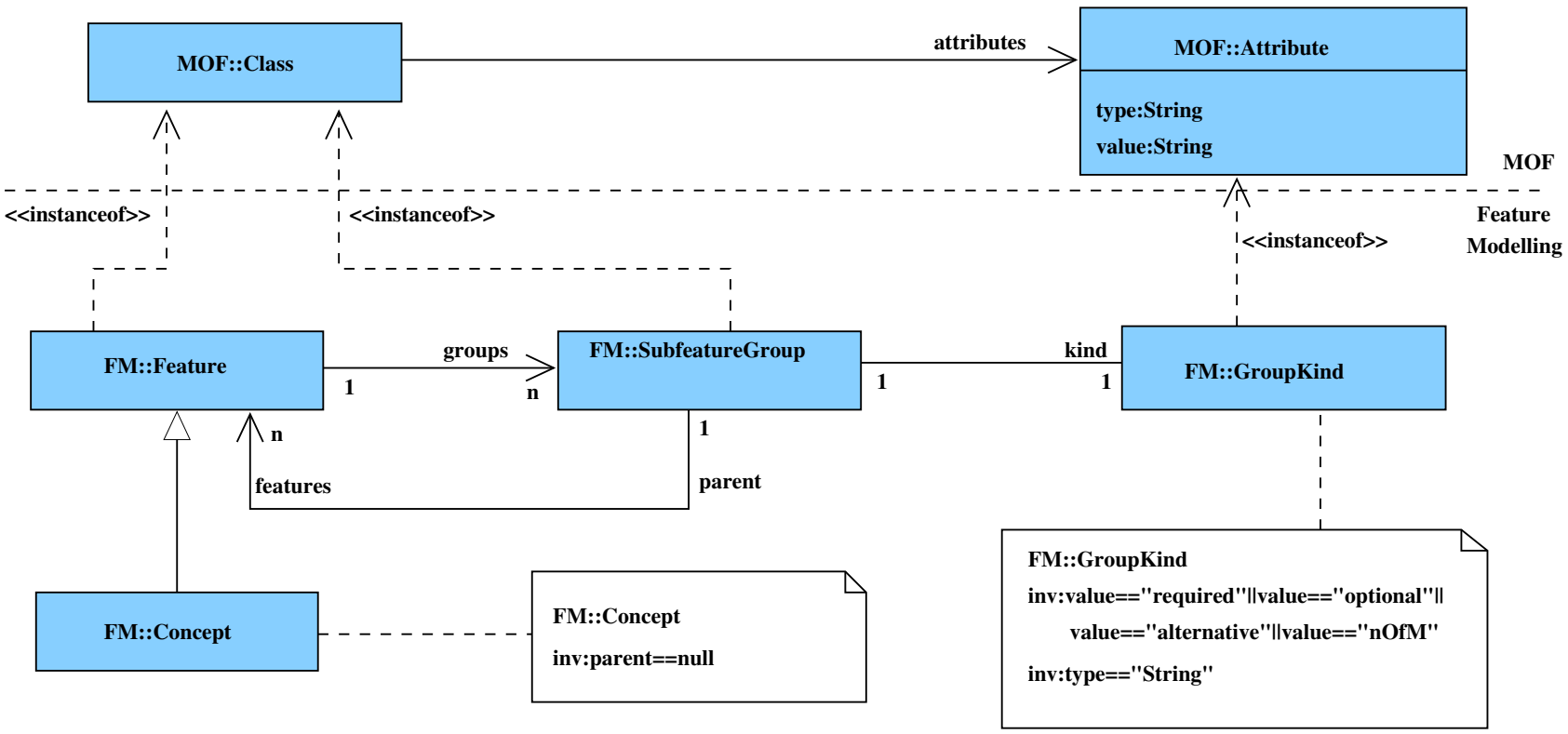
- Feature models are a tool for domain analysis
  - Provide a hierarchical view of features and their dependencies
  - Establish an ontology for categorization
- Visualized by feature diagrams
- Conceived for software domain analysis: Kang, Cohen, Hess, Novak, Peterson. Feature-Oriented Domain Analysis (FODA) Feasibility Study. Technical report CMU/SEI-90-TR-21. 1990.
- Popularized for Generative Programming by Czarnecki and Eisenäcker
- Also for analyzing other domains

Feature Modeling
Example



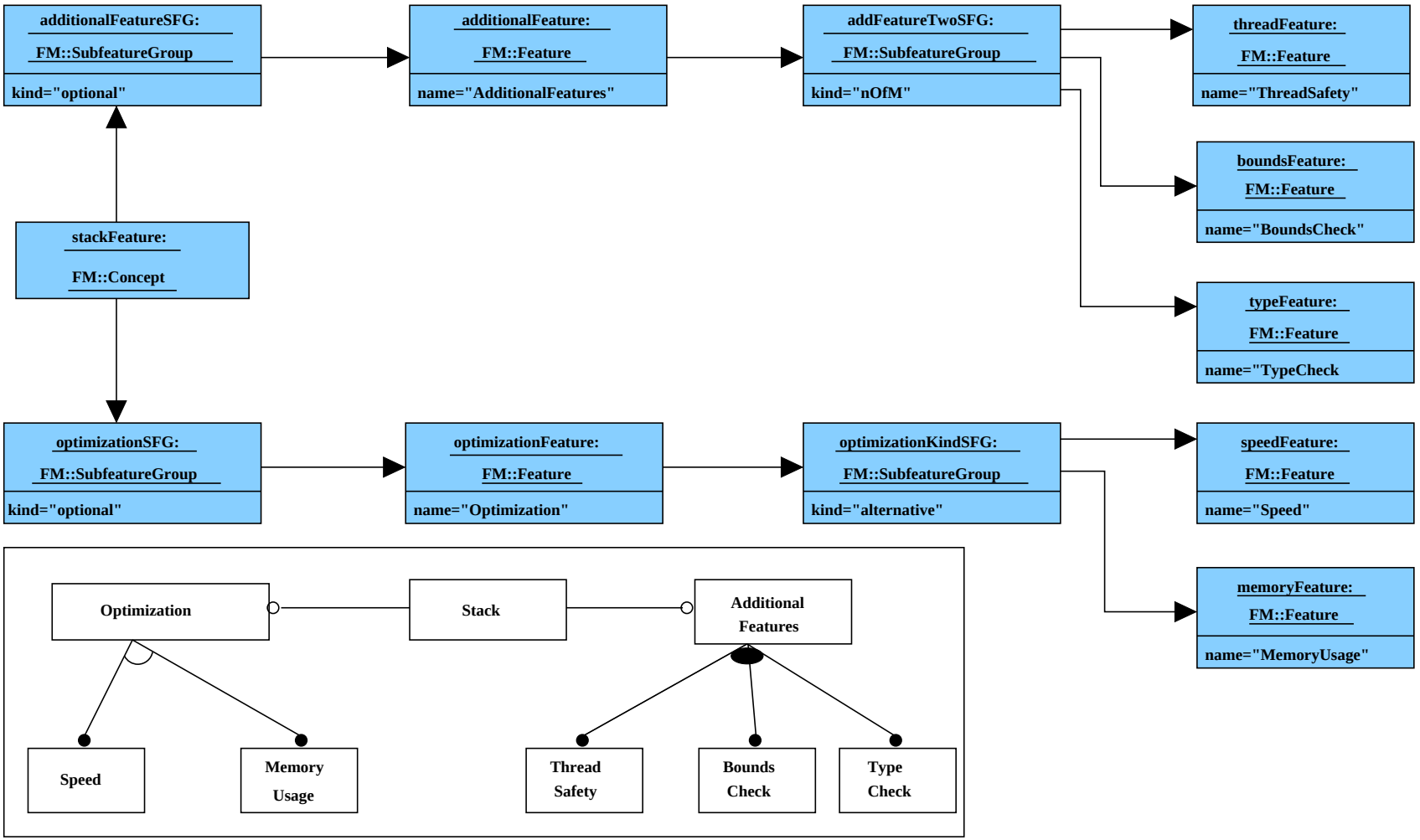
- Hierarchical, but **not** is-a relation (as in a class diagram)
- Features may be qualified as
  required, optional, alternative, or *n*-of-*m* (selection)

Feature Modeling
MOF-based Metamodel

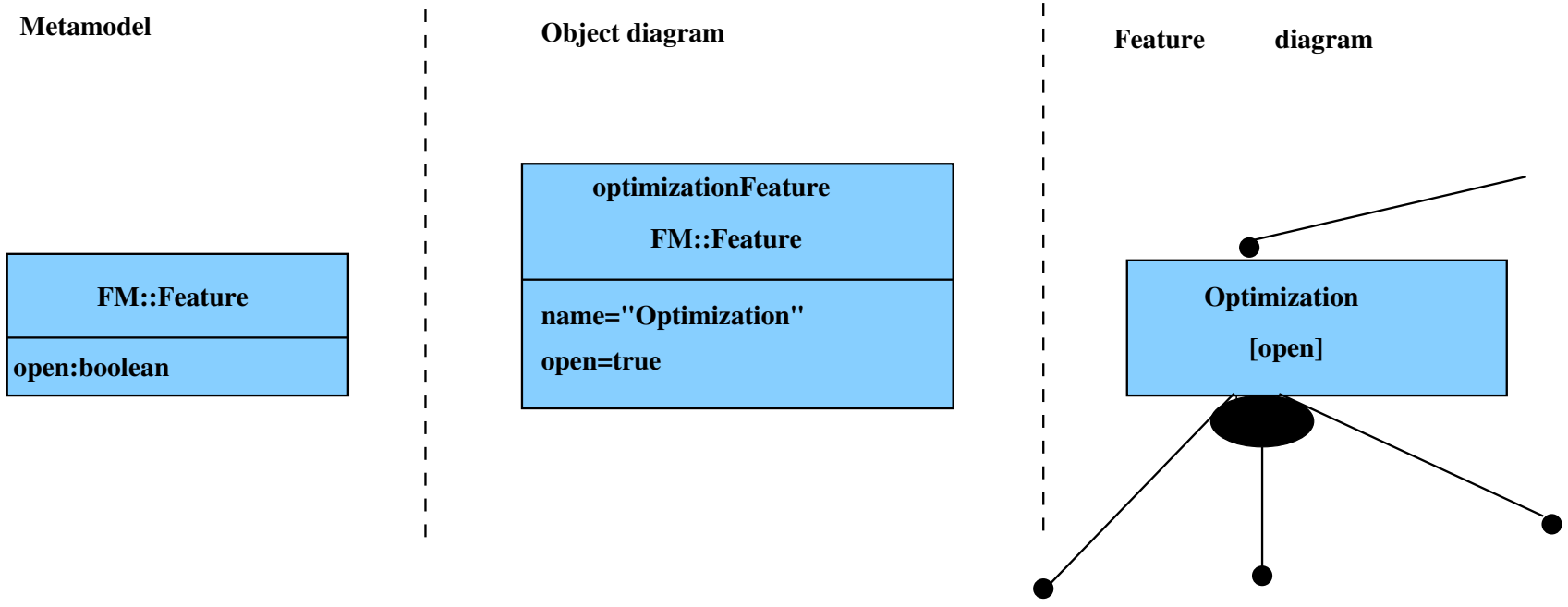


Feature Modeling

Feature Model in Abstract Syntax



Feature Modeling
Extended Metamodel and Concrete Syntax



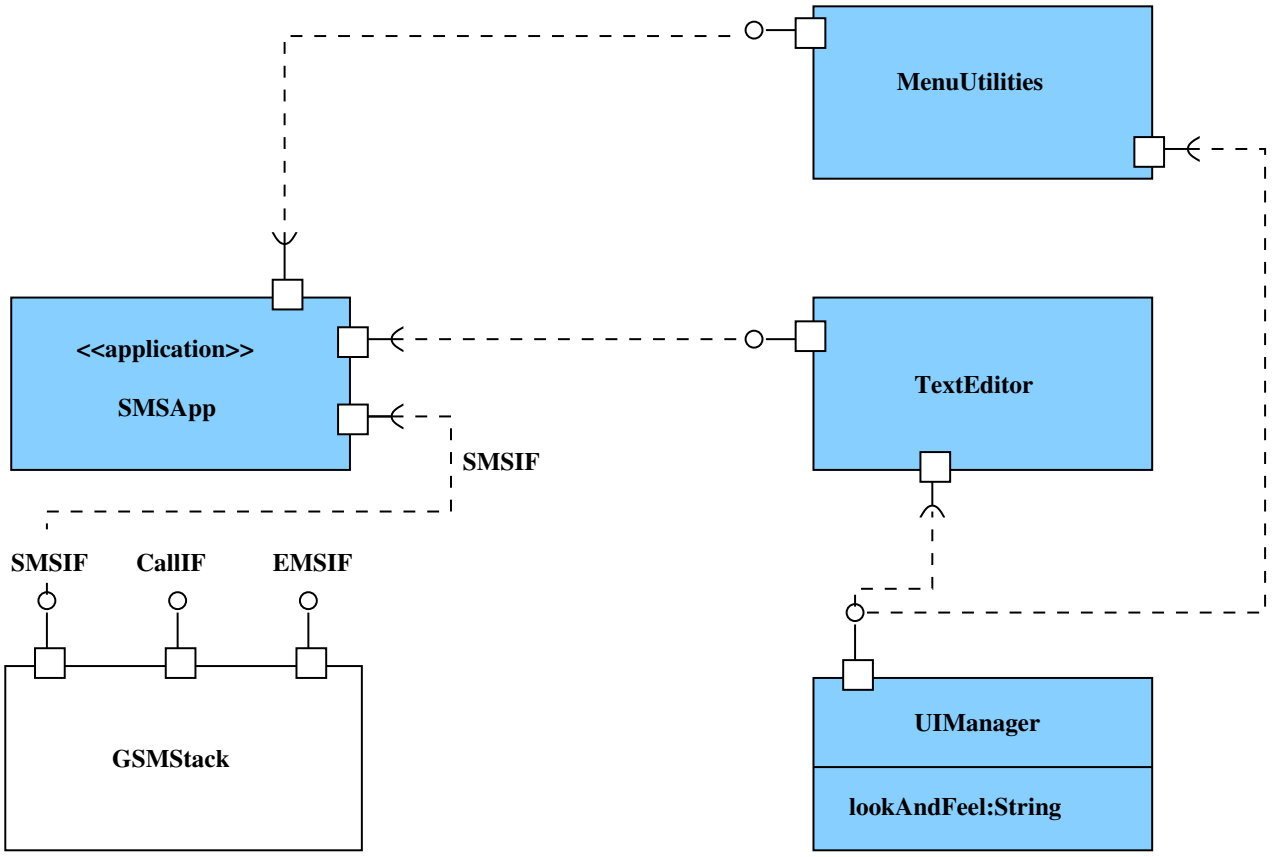
New feature $\Rightarrow$

- new attribute in metamodel
- new slot in model
- extension of concrete syntax

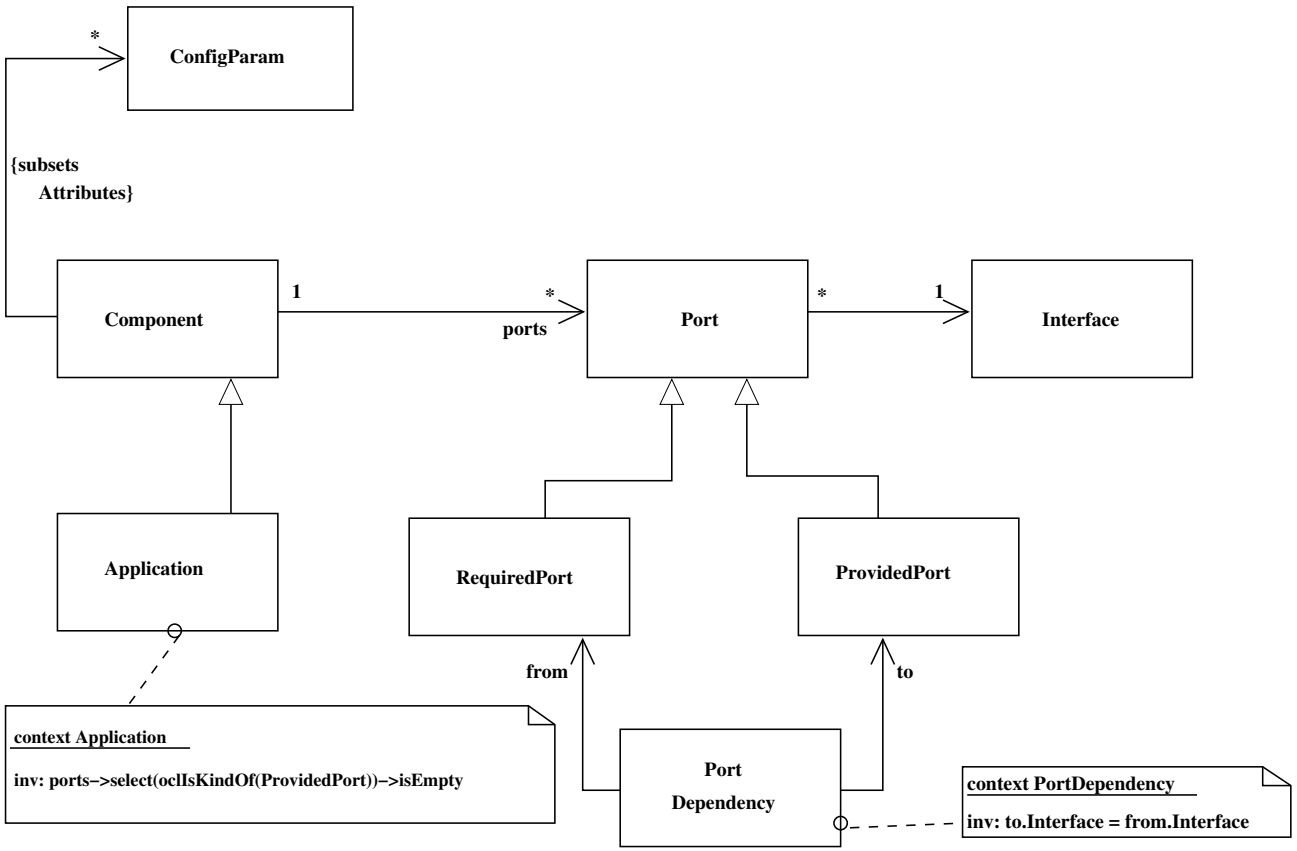
Applications of Metamodeling
Component Modeling

- Domain specific modeling language for small and embedded systems
- Main abstraction: component
- A component may
  - *provide* services via *interfaces*
  - *require* services via *interfaces*
  - have *configuration* parameters
  - be an application (does not provide services)

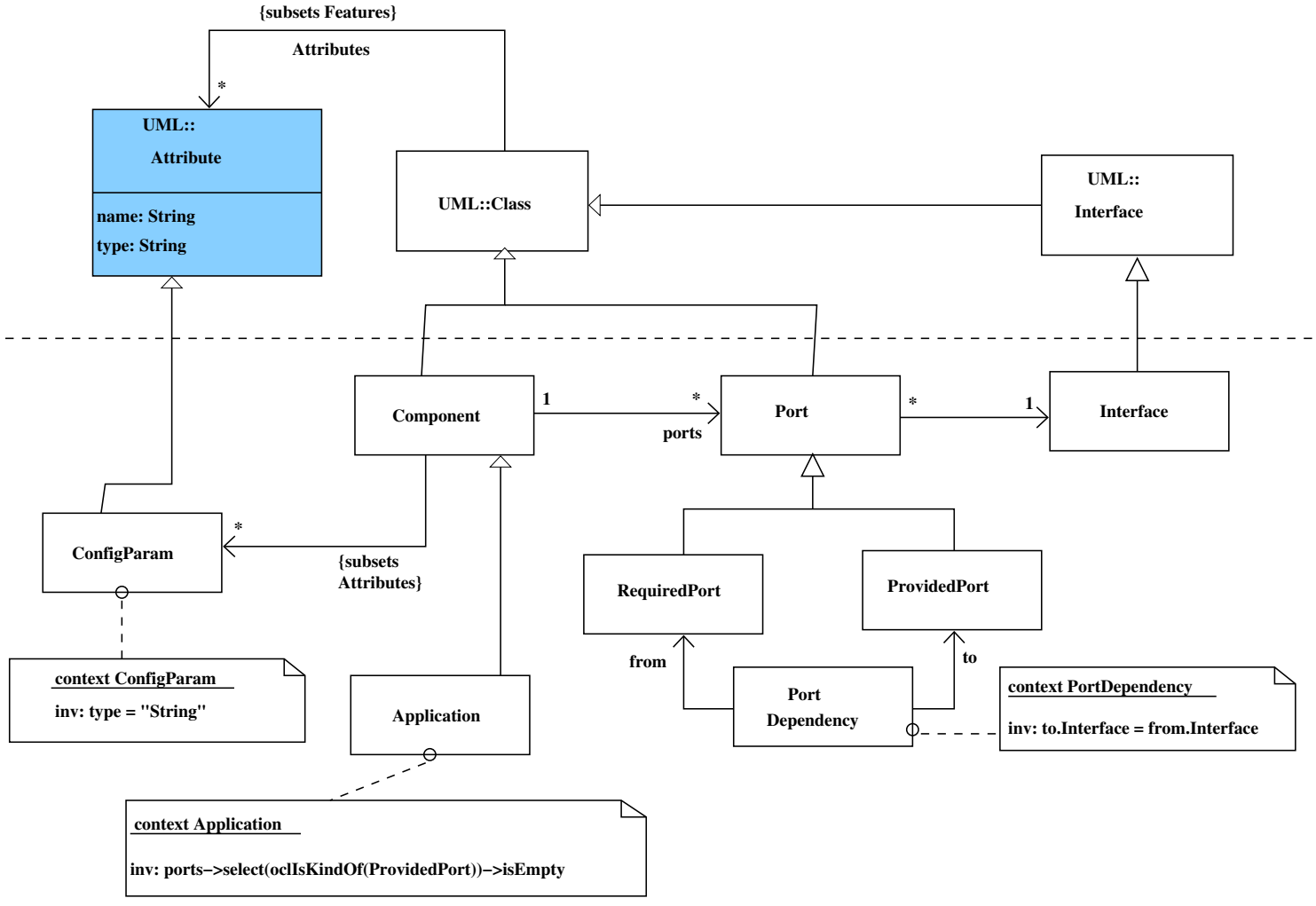
Component Modeling
Example



Component Modeling
Simple Component Metamodel



Component Modeling
MOF-based Simple Component Metamodel



Pitfalls in Metamodeling

How to avoid

- confusion with UML notation
- mixing metalevels

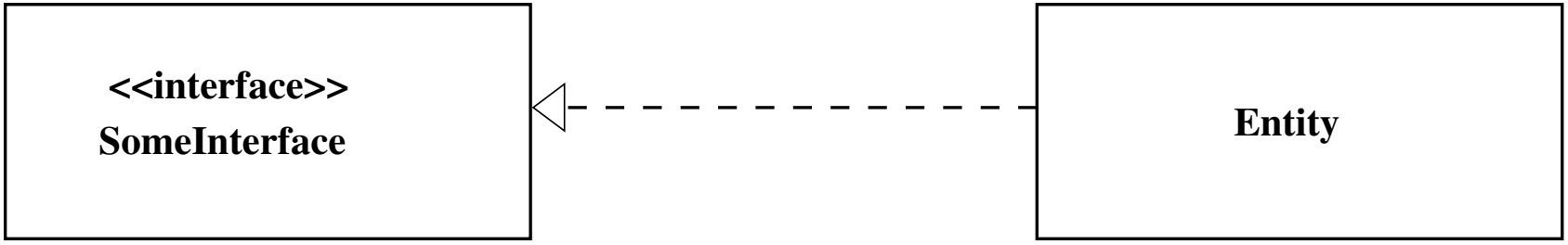
Central question

- what is the mapping to a programming language?

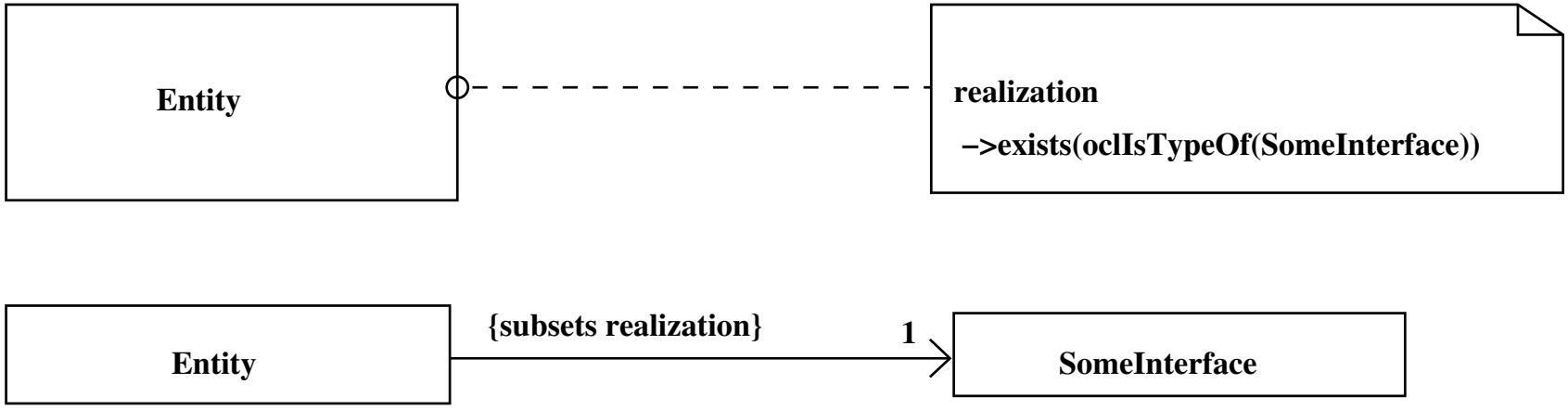
Interfaces

Every instance of **Entity** should implement **SomeInterface**

- wrong approach



- book solution use OCL or subsetting of metaassociation



Interfaces/2

Every instance of **Entity** should implement **SomeInterface**

- correct solution use OCL