
Semantik der UML 2.0



What is UML

The *Unified Model Language* is an industry standard language for

- visualizing,
- specifying,
- constructing, and
- documenting

software-intensive systems. It can be used with many processes, throughout the development lifecycle, and across different implementation technologies.

Why we build models

In order to

- understand complex systems,
- minimize engineering risk,
- communicate with stakeholders,
- drive the implementation.

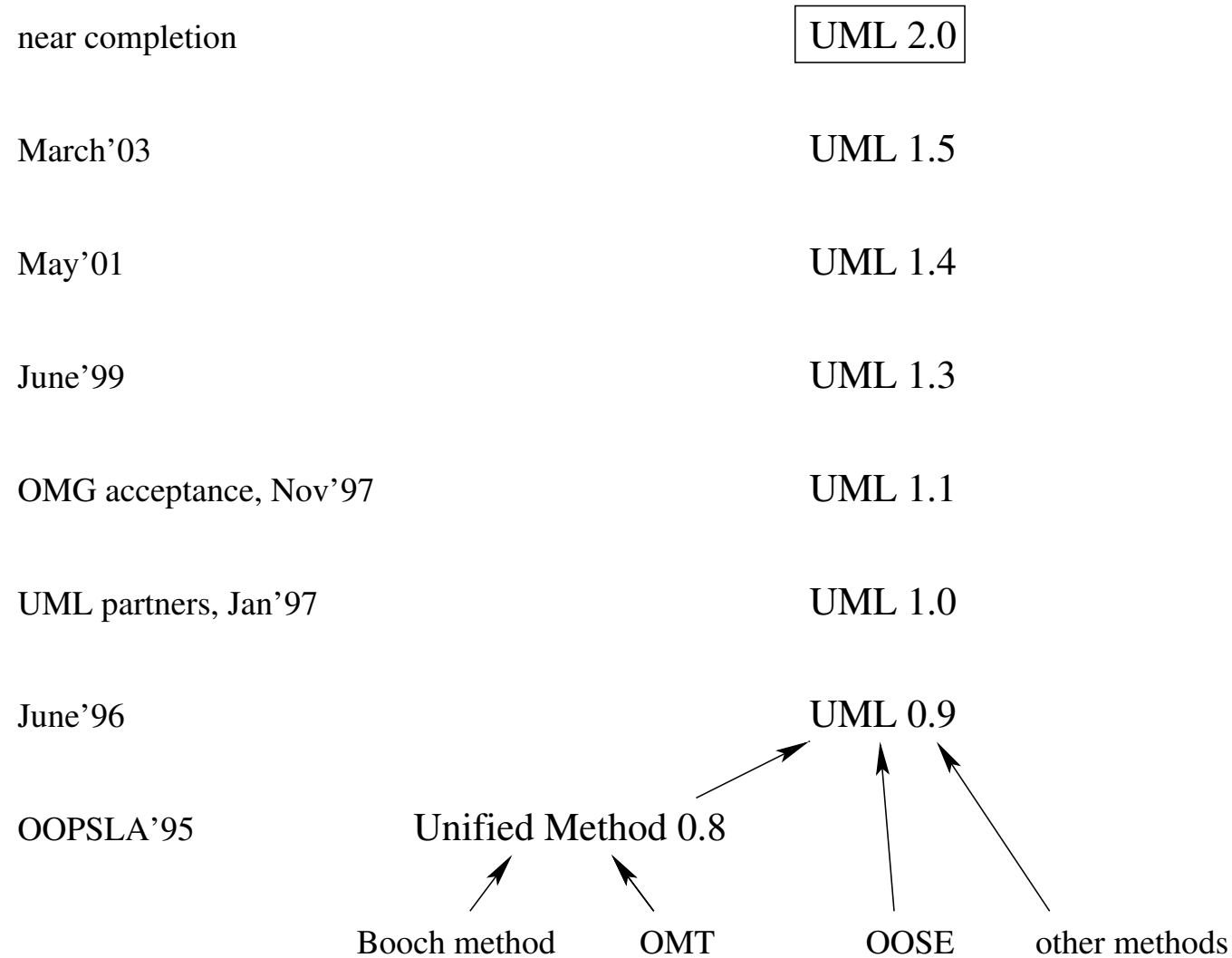
Characteristics of useful models

A models should be

- abstract,
- understandable,
- accurate,
- predictive,
- inexpensive.

Most software models of the past failed on one or more of these aspects.

UML History



UML 1.x: What went wrong?

- MDD potential of models not fully exploited
- Inadequate modeling capabilities
- Too complex
- Inadequate semantics definition
- No diagram interchange capability
- No fully alignment with MOF

Formal RFP requirements

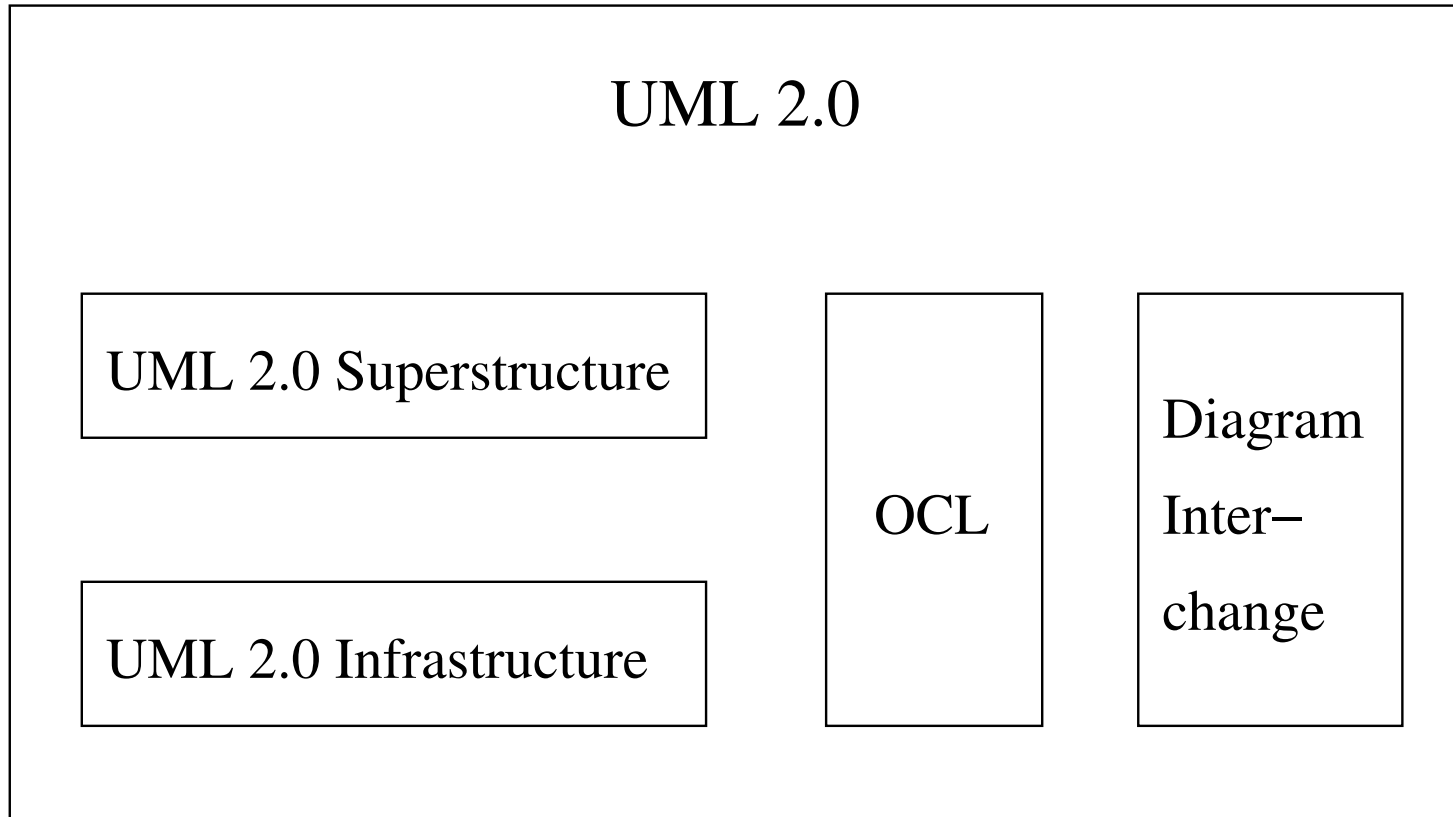
1. Infrastructure - UML internals
more precise conceptual base for better MDA support
2. Superstructure - user-level features
new capabilities for large-scale software systems
consolidation of existing features
3. OCL - constraint language
full conceptual alignment with UML
4. Diagram interchange standard
for exchanging graphic information

Superstructure requirements

The language has been restructured and modularized.

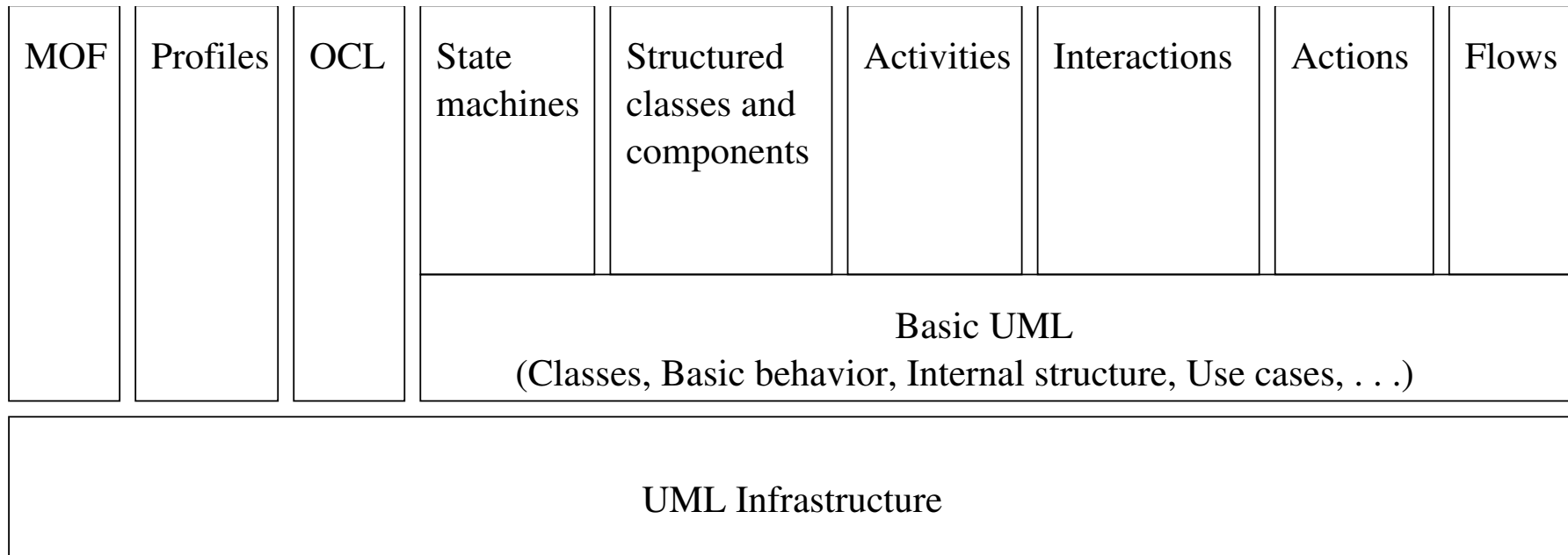
There have been significant changes to the UML metamodel.

Two main specifications

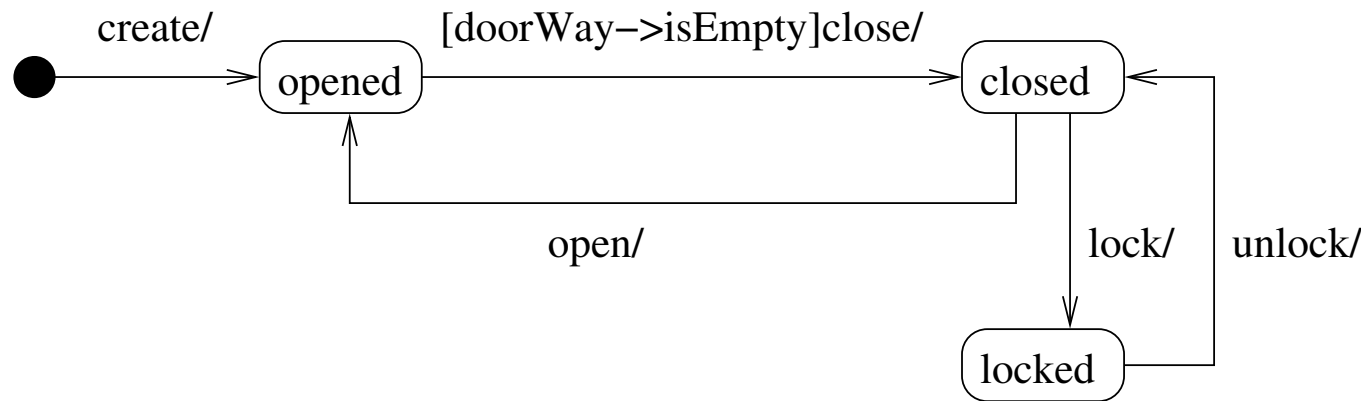


Language Structure

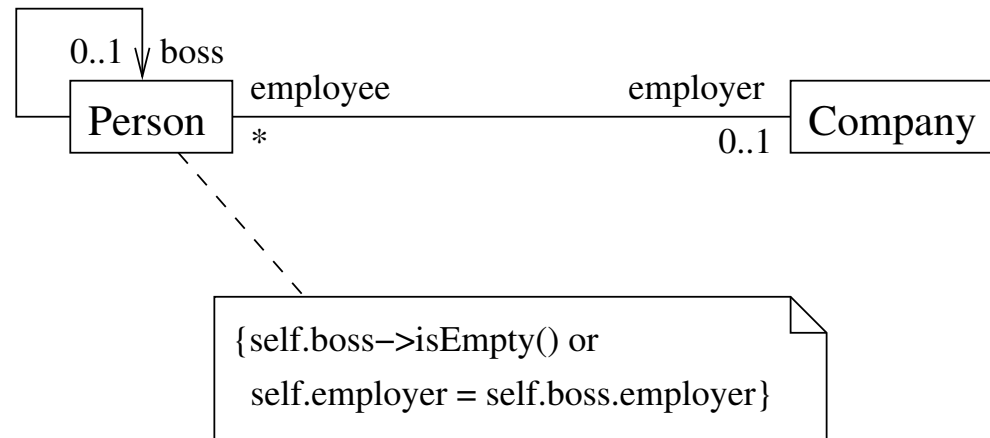
A core language plus a set of optional “sub-languages”



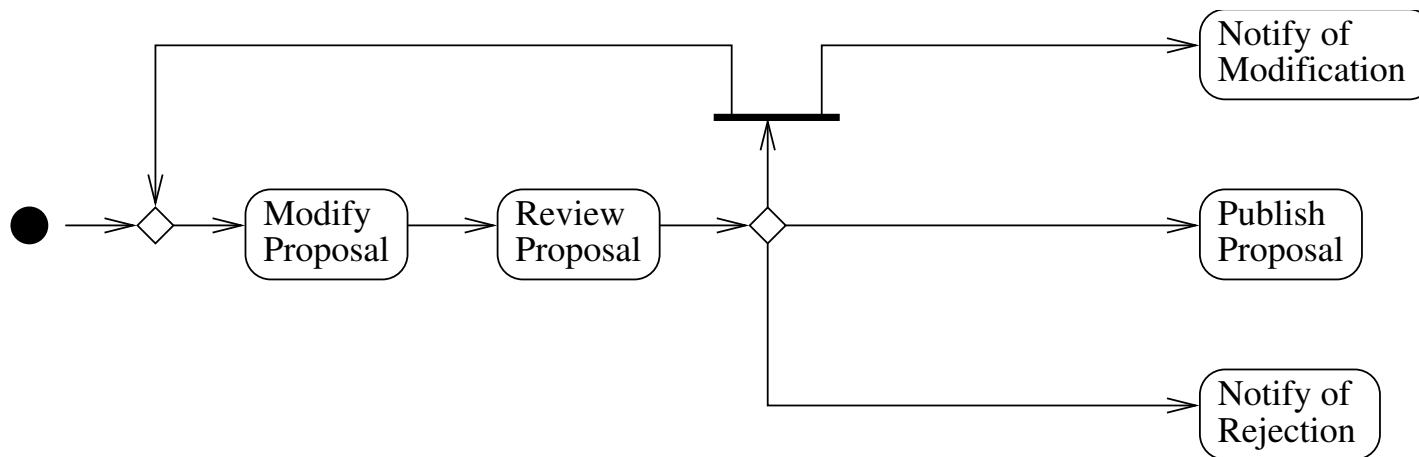
UML state machines



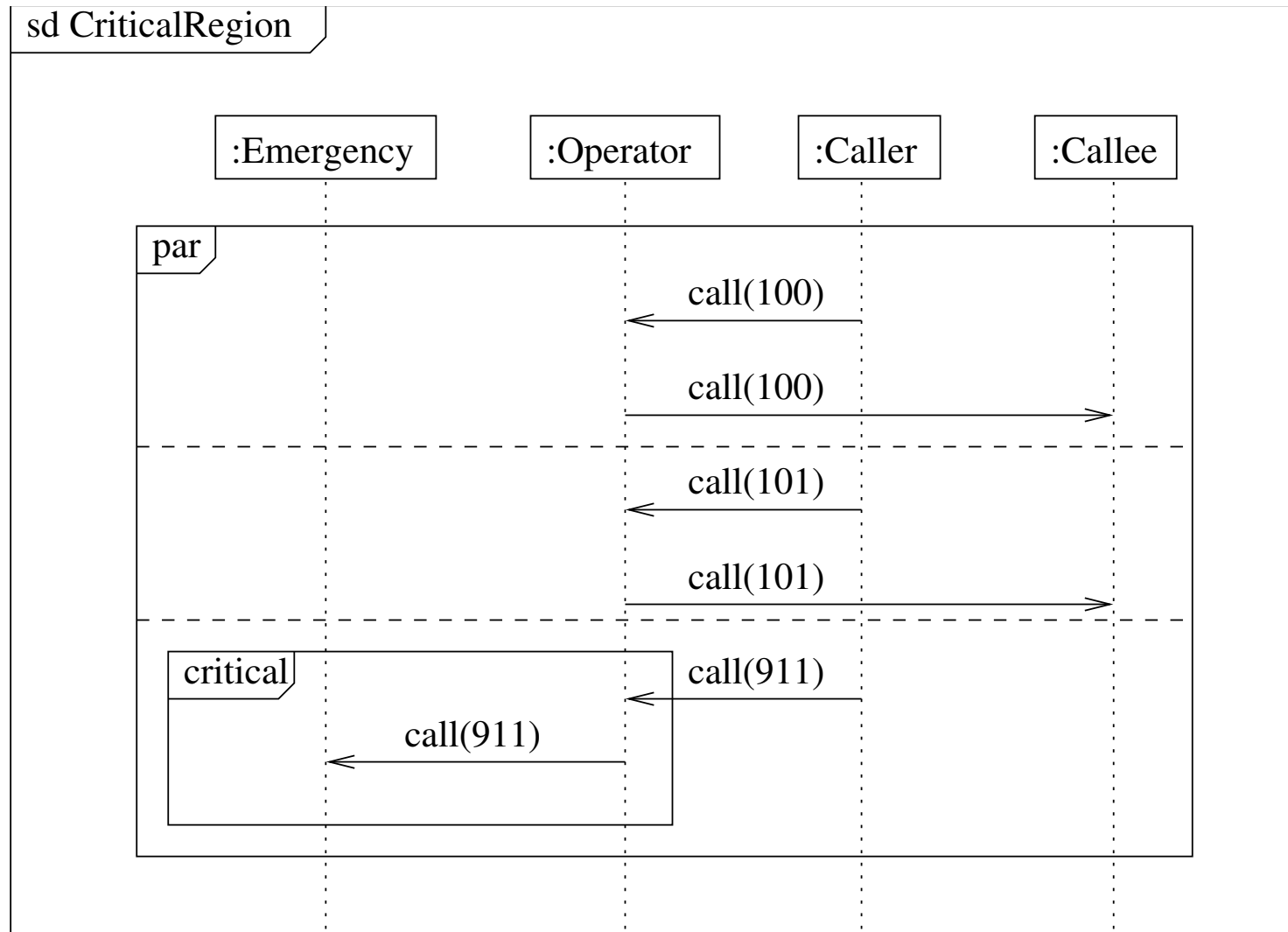
UML class diagrams



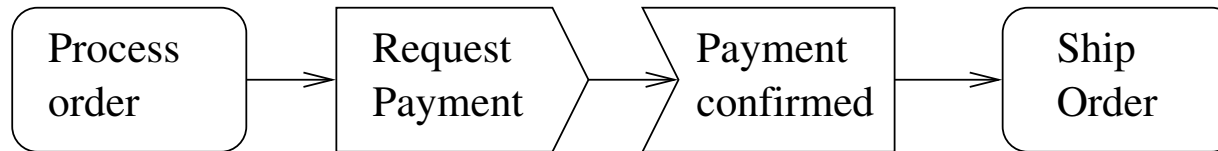
UML activities



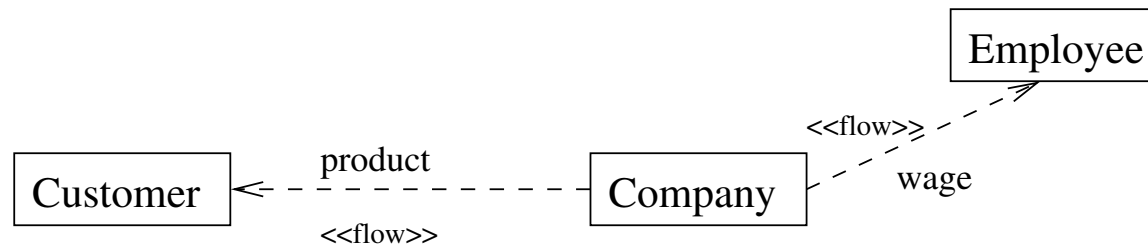
UML interactions



UML actions



UML flows



Goal of the seminar

Study of

- foundations,
- concepts,
- syntax and semantics, and
- implementation

of the sublanguages of basic UML put in context.