Prolegomena of a Multi-Level Modeling Method

Illustrated with the FMML^{x}

Ulrich Frank

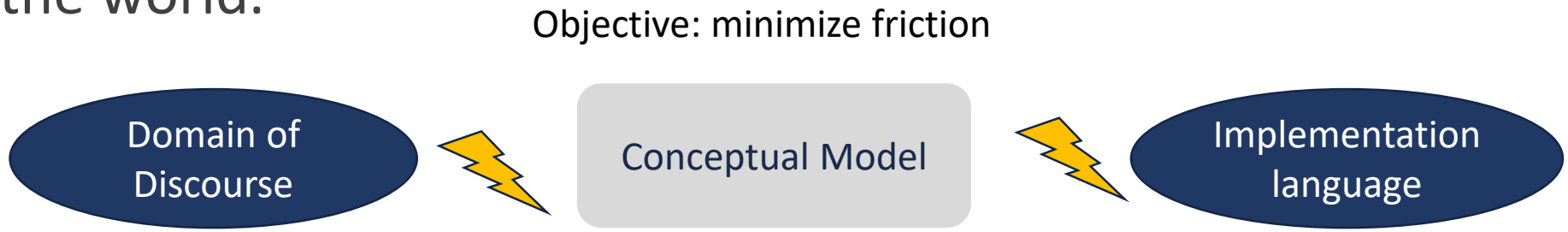
UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

Prof. Dr. Ulrich Frank
Research Group Information Systems and Enterprise Modeling
Institute for Computer Science and Business Information Systems (ICB)
University of Duisburg-Essen, Essen, Germany

Background

- Conceptual modeling is about bridging the gap between software systems and our ideas about the construction of the world.



"Through models we take the reality of the past and the possibilities of the **future** into the present."

Bernd Mahr (translated)

Conceptual modelling is about epistemology, ontology and system design (and implementation).

Motivation

- Multi-level modeling strengthens conceptual modeling through additional abstraction, thus promoting
  - reduction of friction (more „natural“ representations)
  - reuse
  - integrity
  - maintainability
- However, leads to more dependencies (tighter coupling).
- Need for methodical support for appropriate design of multi-level models.

So far, the emphasis of research on MLM was on languages and tools, hardly on modeling methods.

Common Ground: General Design Principles

GDP 1: Commonalities should be captured by an appropriate abstraction.

GDP 2: Commonalities should be captured by an appropriate abstraction only, if the abstraction is likely to be invariant during the lifetime of a system.

GDP 3: If part A of a system depends on part B, B should be more invariant than A.

GDP 4: Dependencies that may change over time should be removed or relaxed.

Specific MLM Design Principles

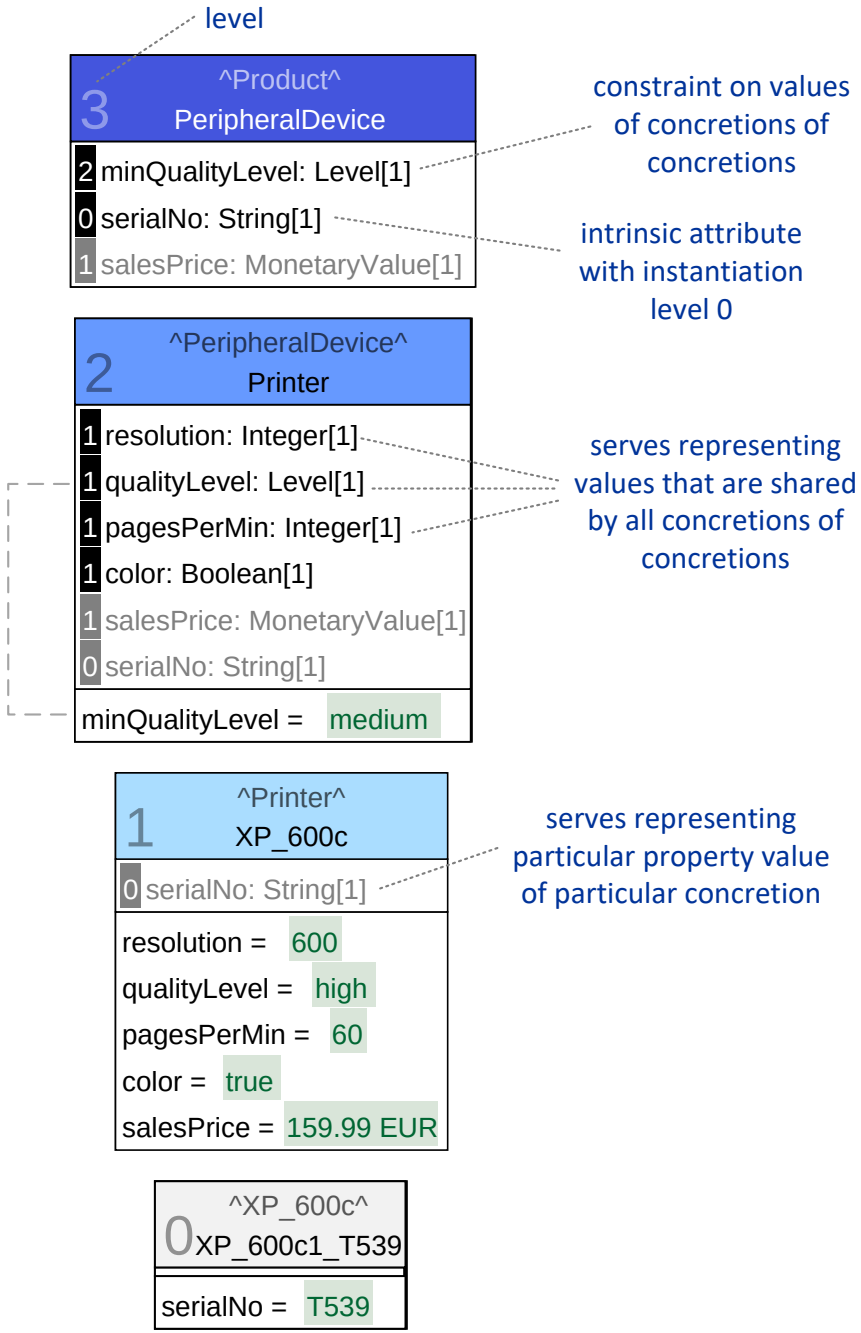
SDP 1: Specify known knowledge on the highest possible level within the scope of your project. (-> GDP 1).

SDP 2: The higher the level of a class, the more invariant it should be. (-> GDP 3)

SDP 3: The design of a class at any level should aim at modification consistency. In other words: concretization relationships between two classes on different levels should be invariant. (-> GDP 2)

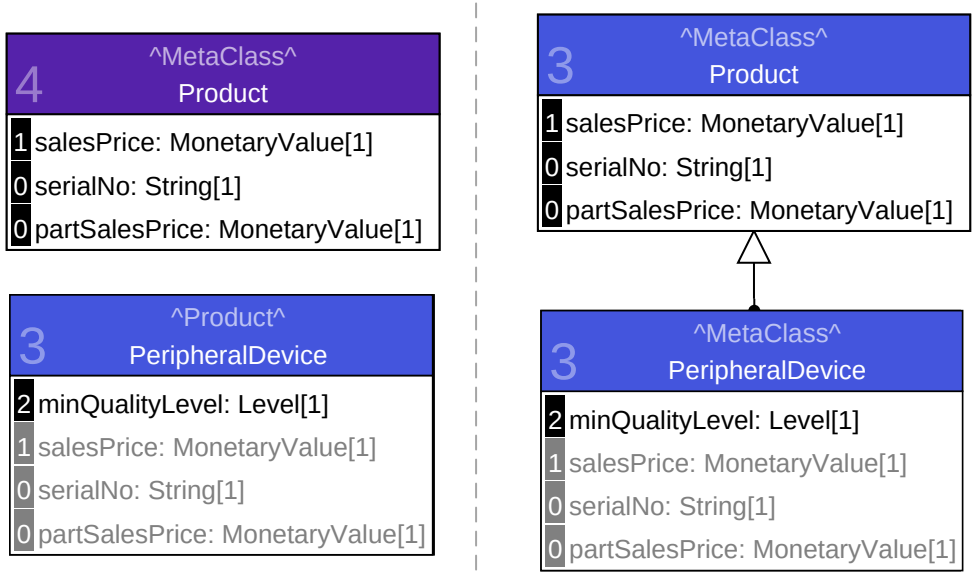
SDP 4: Assign properties of classes on levels higher than 1 to categories that indicate semantic differences.

SDP 4: Illustration



Focus on Things to Avoid

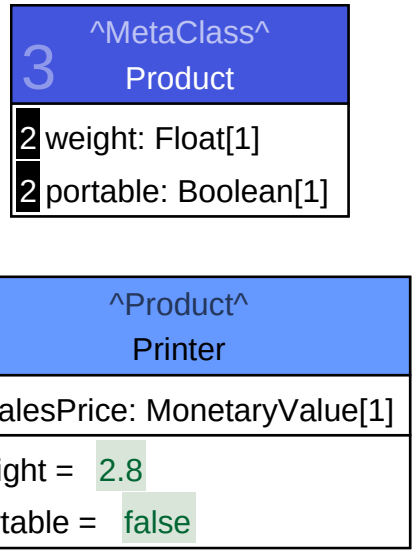
SDP 5: Avoid the introduction of “fake” levels, that is, of levels that could be expressed through generalization/specialization.



Focus on Things to Avoid

SDP 5: Avoid the introduction of “fake” levels, that is, of levels that could be expressed through generalization/specialization.

SDP 6: Treat “pure” instantiations of classes on levels > 1 with care.

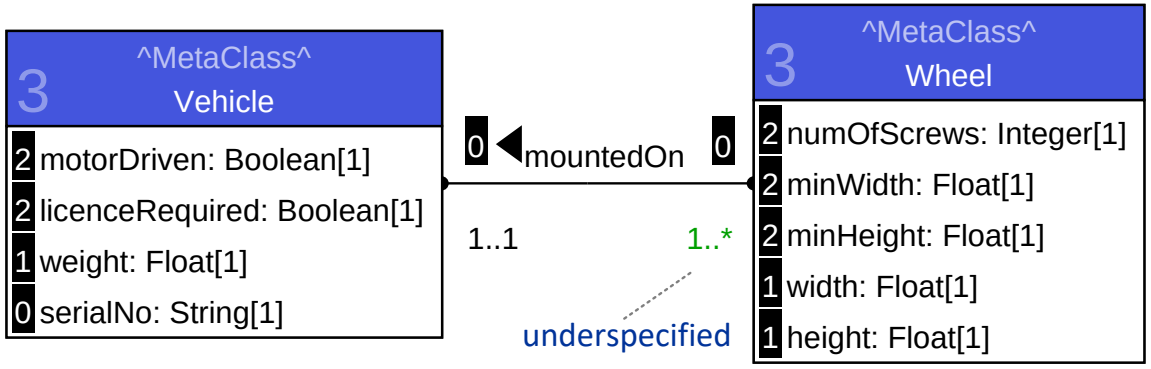


SDP 7: Avoid the use of "dummy" classes.

Coping with Contingency

SDP 8: If the level of a class is not the same for all classes concretized from it on lower levels, mark it as “contingent”. (-> GDP 1)

SDP 9: Apply general design principle GDP 1 also in specific cases of incomplete knowledge. These cases comprise the specification of associations, the instantiation of which is to be deferred.



Epistemological Questions & Guidelines

EQ1: What is the appropriate level of a class?

EQ2: How can one determine
to be represented?

EQ3: How can one determine

EQ4: How can one determine

EQ5: How can one determine
attribute that is intended for deferred instantiation?

Heuristic: start with objects at the bottom that cannot be instantiated. Check against design principles.

Epistemological analysis: Extend use scenarios with questions like:
“Is there any other kind of product conceivable?”
“If that is the case, will that ever be relevant for the system under development?”

Epistemological Questions & Guidelines

EQ1: What is the appropriate level of a class?

EQ2: How can one determine the highest level on which the knowledge is to be represented?

EQ3: How can one determine whether a class is invariant?

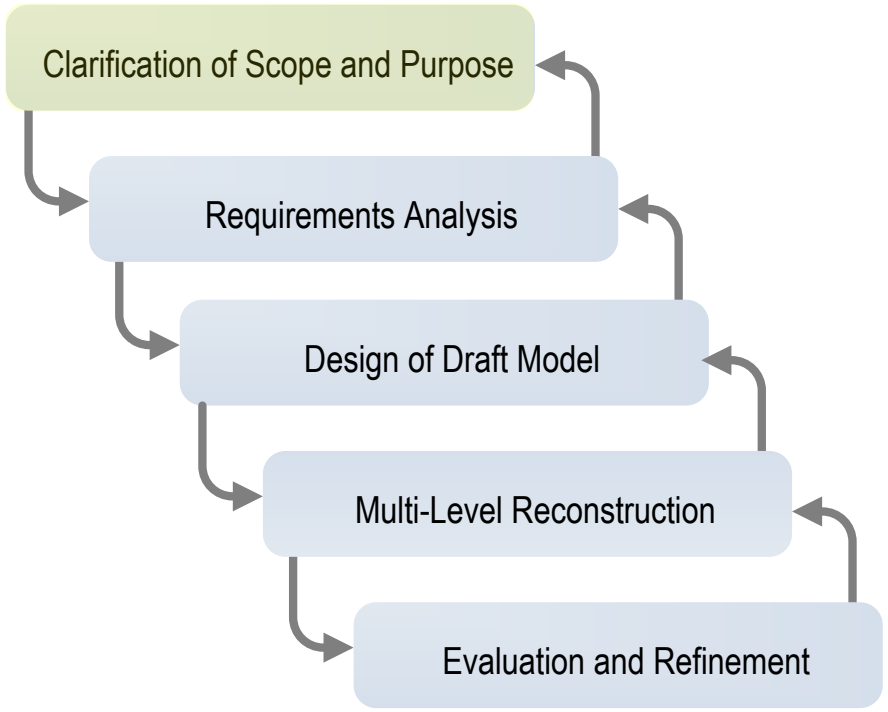
EQ4: How can one determine w

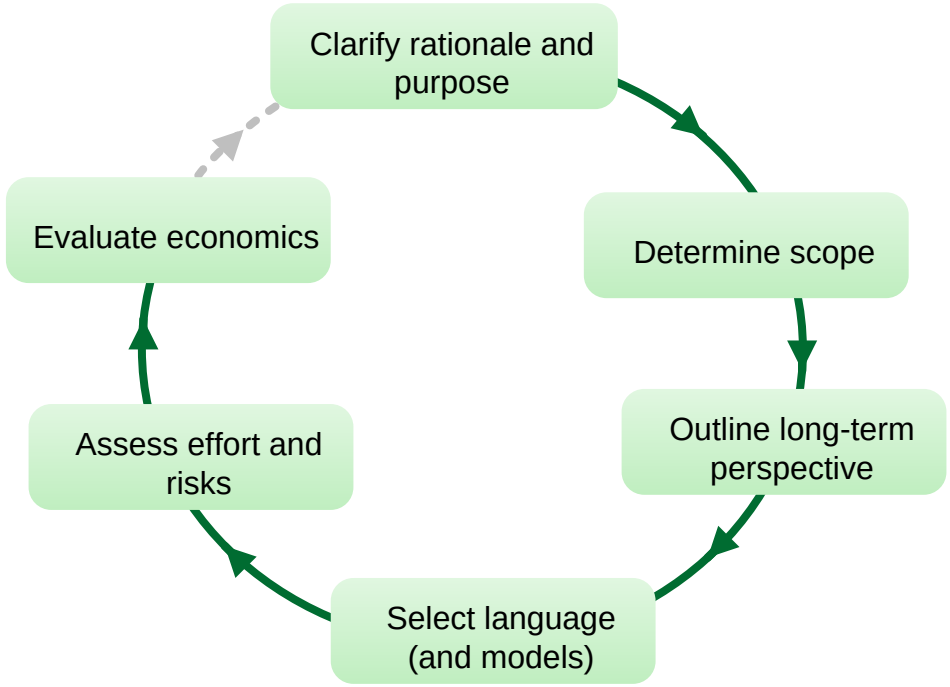
EQ5: How can one determine th
attribute that is intended for de

create scenarios that not only address possible constellations within the scope of a project, but also beyond that scope, including possible future worlds.

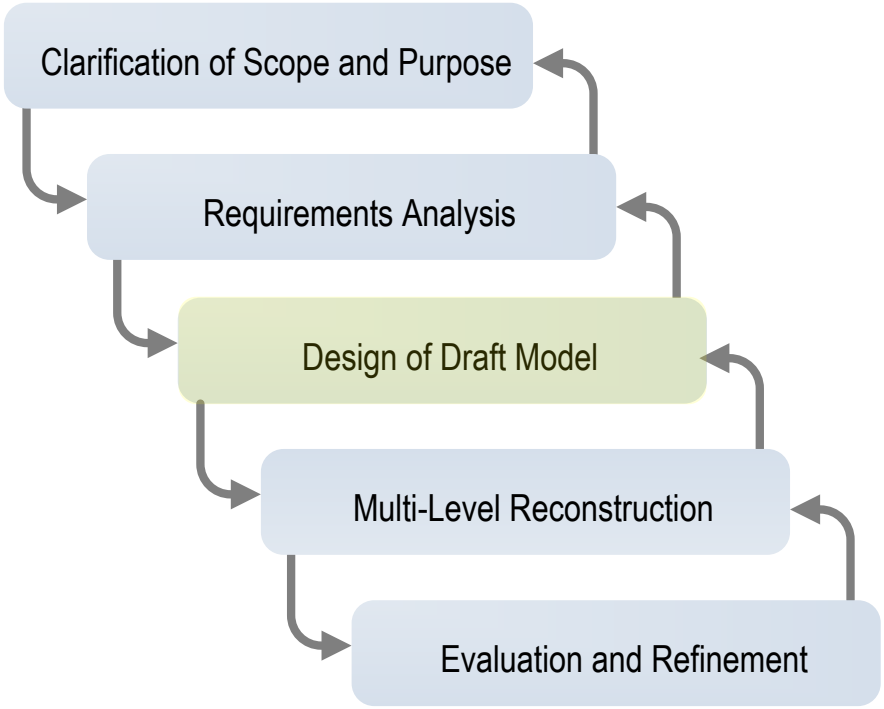
Get people with different backgrounds/viewpoints involved.

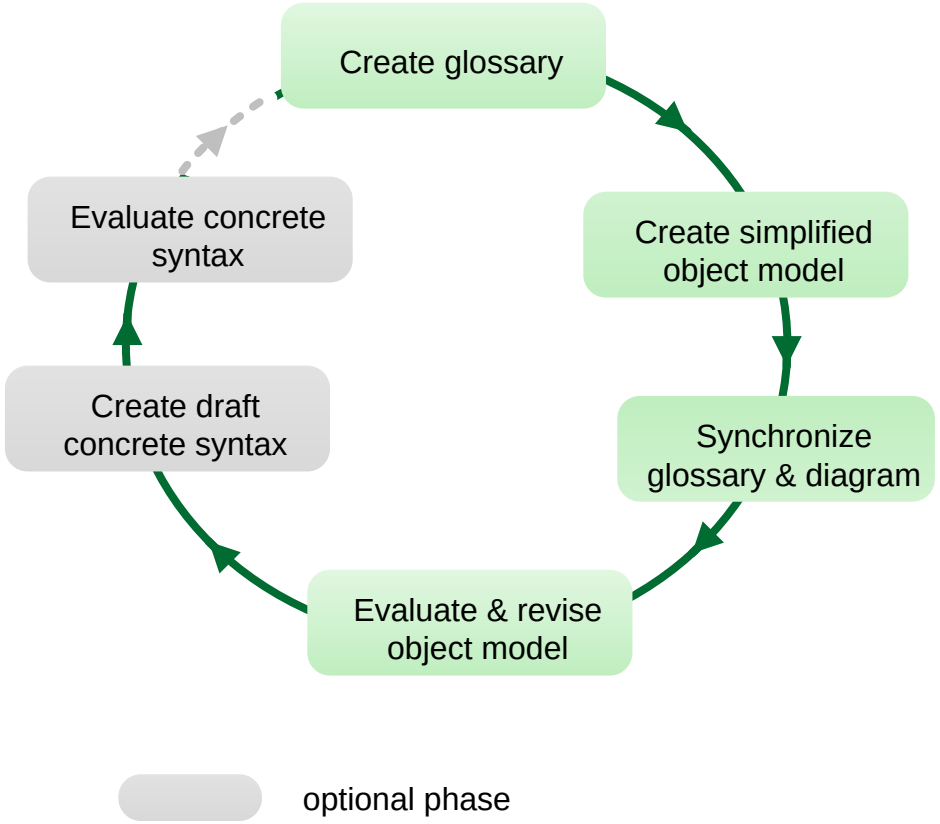
Process Model



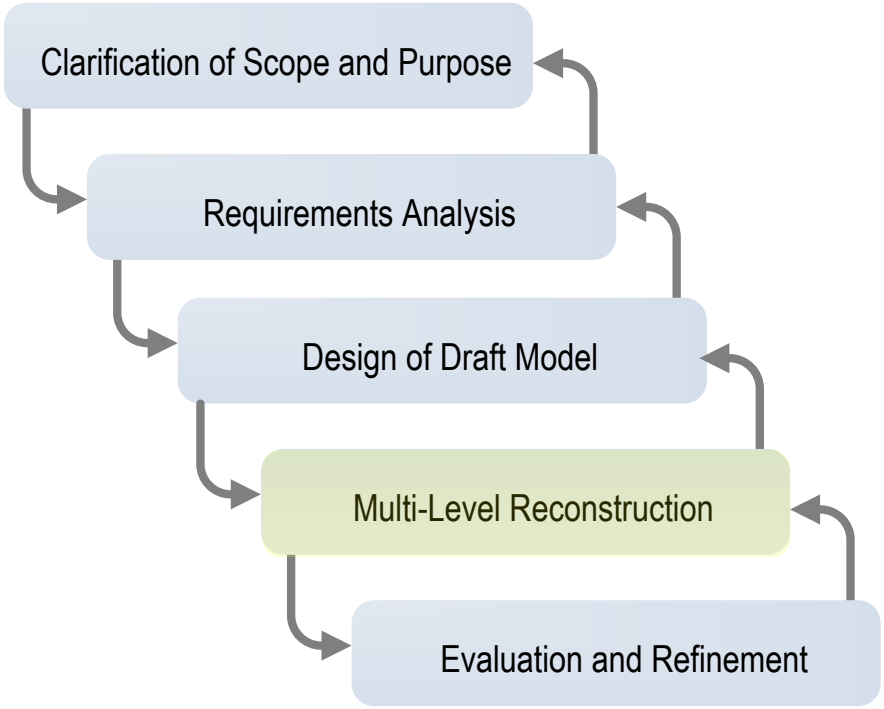


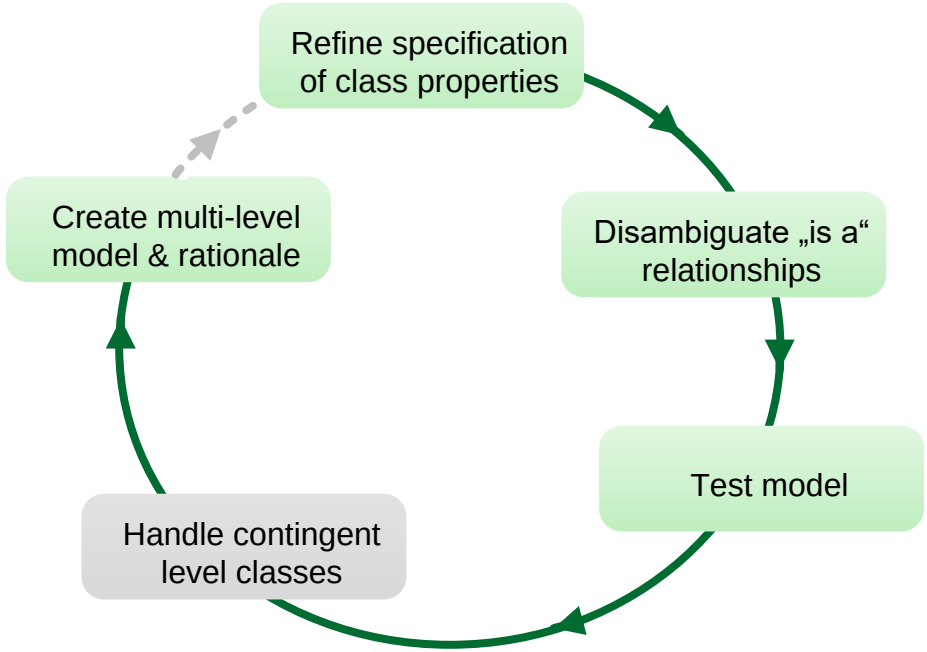
Process Model





Process Model





Final Remarks

- (Multi-level) modeling method requires accounting for ontological and epistemological aspects of domain (like conceptual modeling in general).
- Design principles and guidelines intended to serve as orientation, not suited as „cookbook“.
- Epistemological questions demand for reflection of what is – and what could be.
- Process model reflection of experience with development of multi-level DSML.

presented prologomena of multi-level modeling method intended as input to discussion and further development