Ontological Product Modeling for Collaborative Design

Conrad Bock, Xuan Zha

Hyowon Suh, Jeahyun Lee

December 11, 2008

Overview

- Goals and approach.
- Desired capabilities.
- Background
  - Ontology modeling
  - Modeling languages
- Proposed Solution
- Summary

Overall Goal and Challenges

- **Improved support for collaboration in the design process.**
  - **Right knowledge at the right time.**
  - **Avoid backtracking and rework.**
  - **Especially in global economy.**
- **Challenges:**
  - **Combining and refining independently-developed product descriptions.**
  - **Alignment in interpretation of product descriptions.**

General Approach

- Apply ontological techniques …
  - Open world semantics: Multiple product models can describe the same product and be checked for consistency.
  - Rigorously-defined interpretation of ontological languages.
- … and model-driven techniques …
  - Engineering-friendly domain languages specialized from ontological languages.

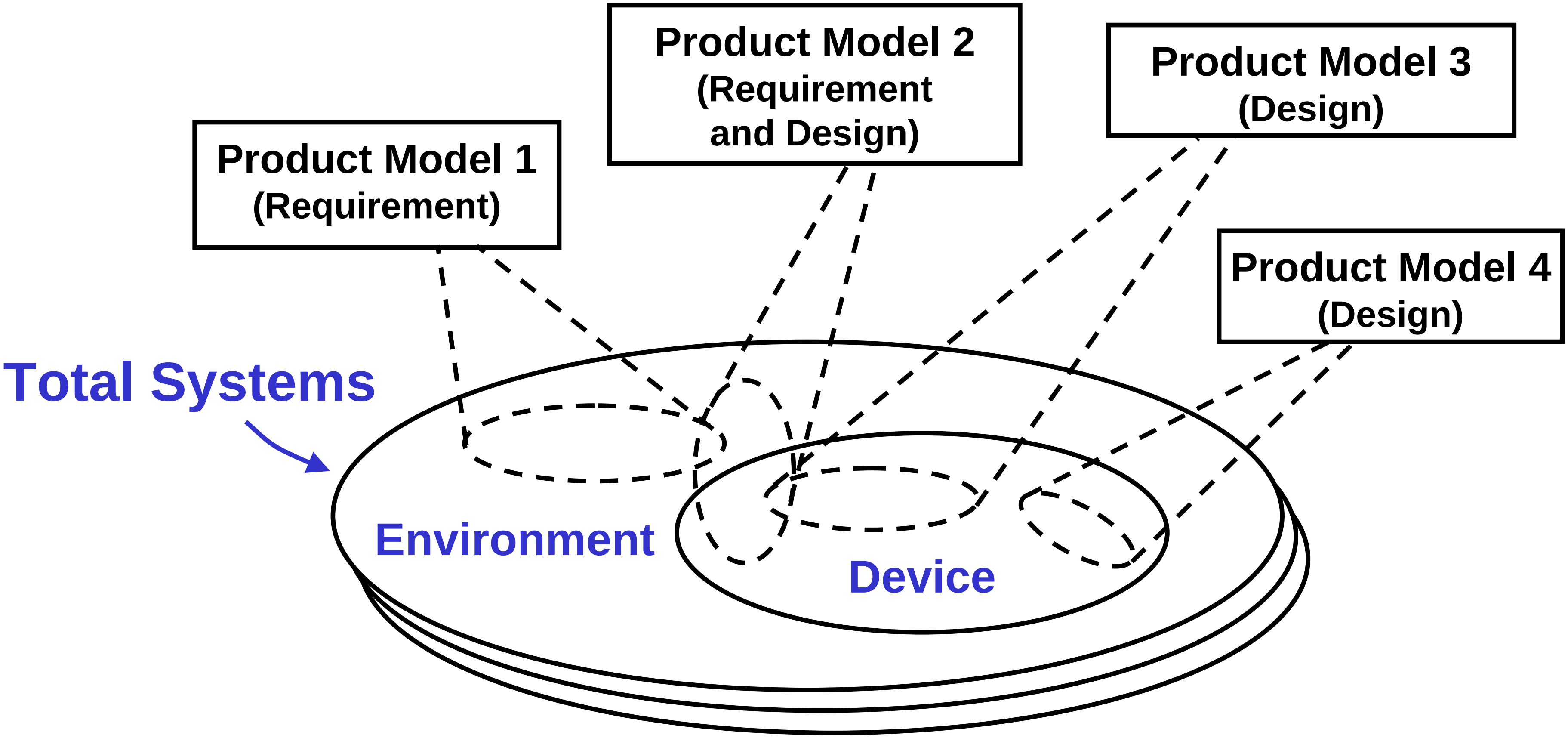
Applied to Product Modeling

- **… to generalized notions in product modeling:**
  - **Product models describe (some portion of) the *total system* of device and environment in which it is used.**
  - **Behaviors include the entities involved in them. Models can describe a portion of a behavior or its entities or both.**
  - **Interconnections between components have same capabilities as components.**

Product Models

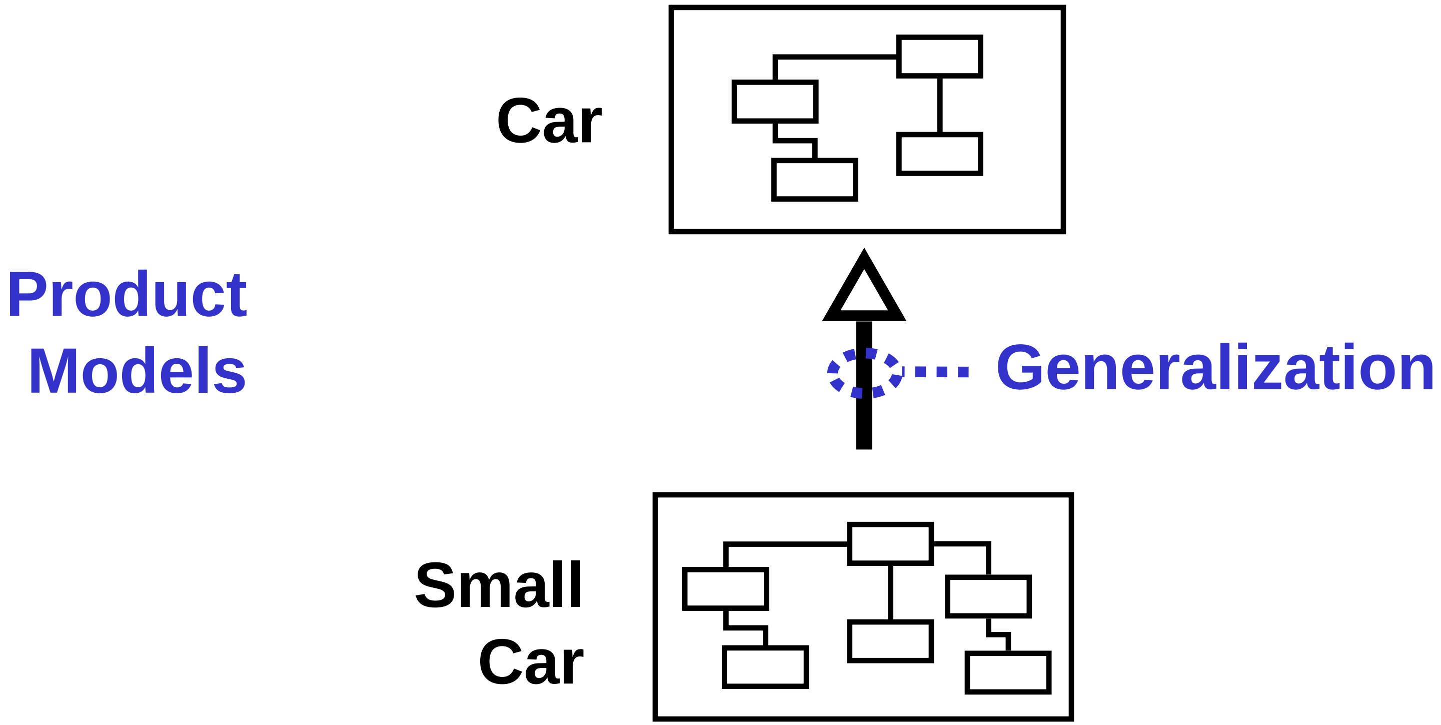
- ***Product models*** **describe any aspect of total systems (environment and device).**
  - **Environment (requirements)**
  - **Device (designs)**
  - **Or both.**
- **No limit on how much or how little of the environment and/or device is described.**

Product Models



- Treat product models as *partial descriptions* of total systems (environment and/or device behavior).

Product Taxonomies



Car

Product Models

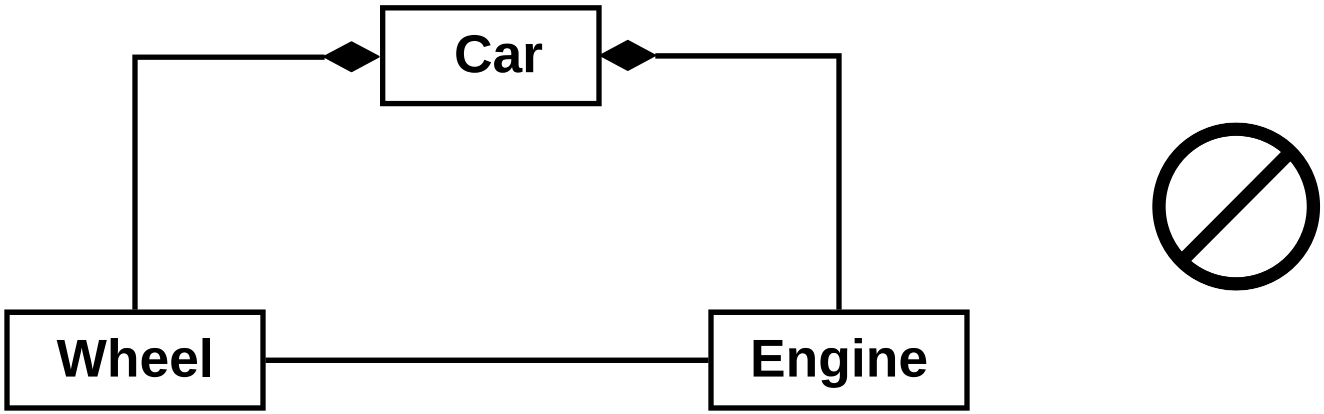
Generalization

Small Car

- **Specialized model includes the general model.**

Interconnections (Not)

Assembly
Model

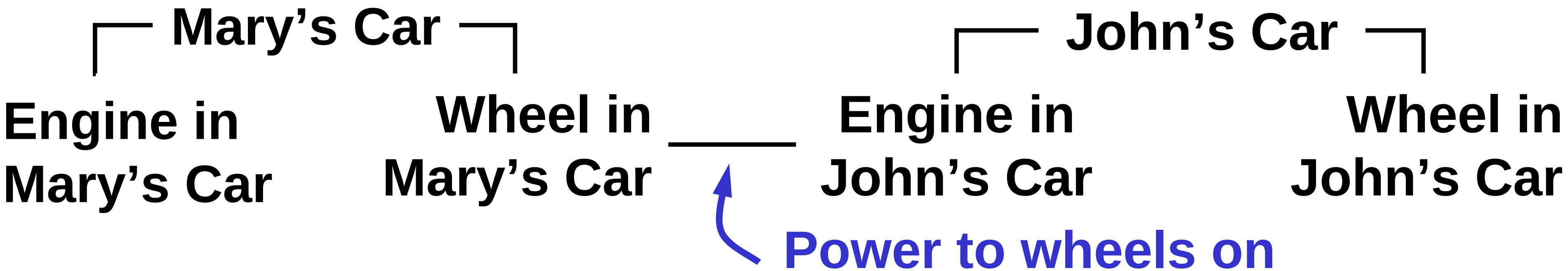


Car

Wheel

Engine

Individuals



Mary’s Car

John’s Car

Engine in
Mary’s Car

Wheel in
Mary’s Car

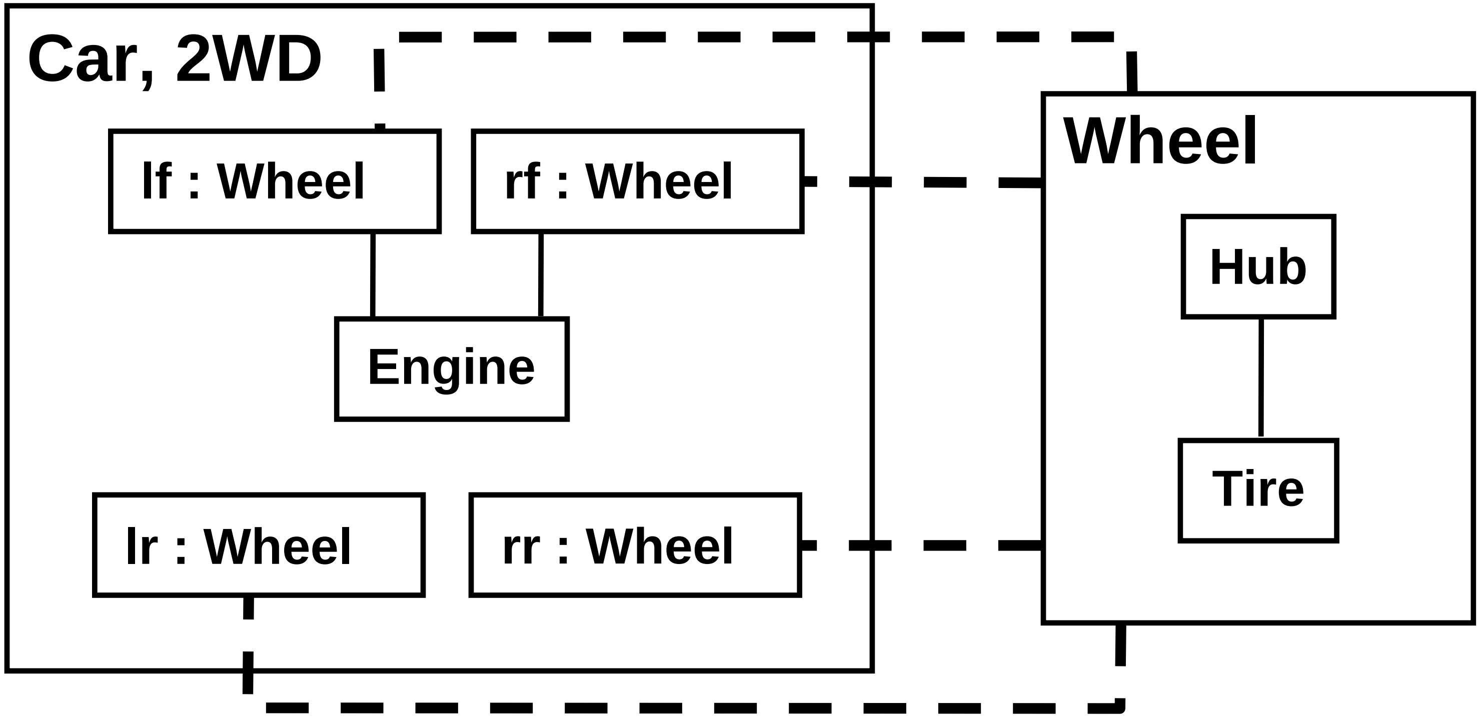
Engine in
John’s Car

Wheel in
John’s Car

Power to wheels on
different car than engine

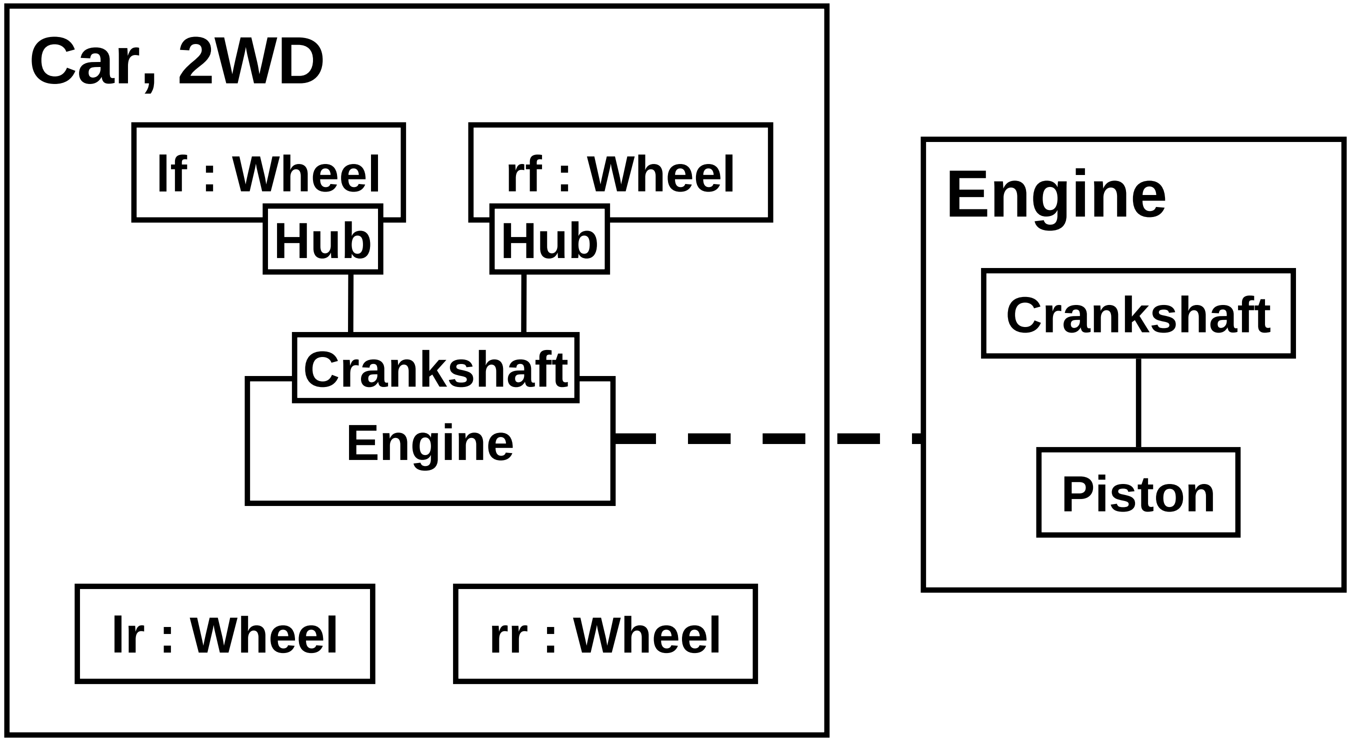
- **Need connections in the context of an individual assembly.**

Interconnected Elements



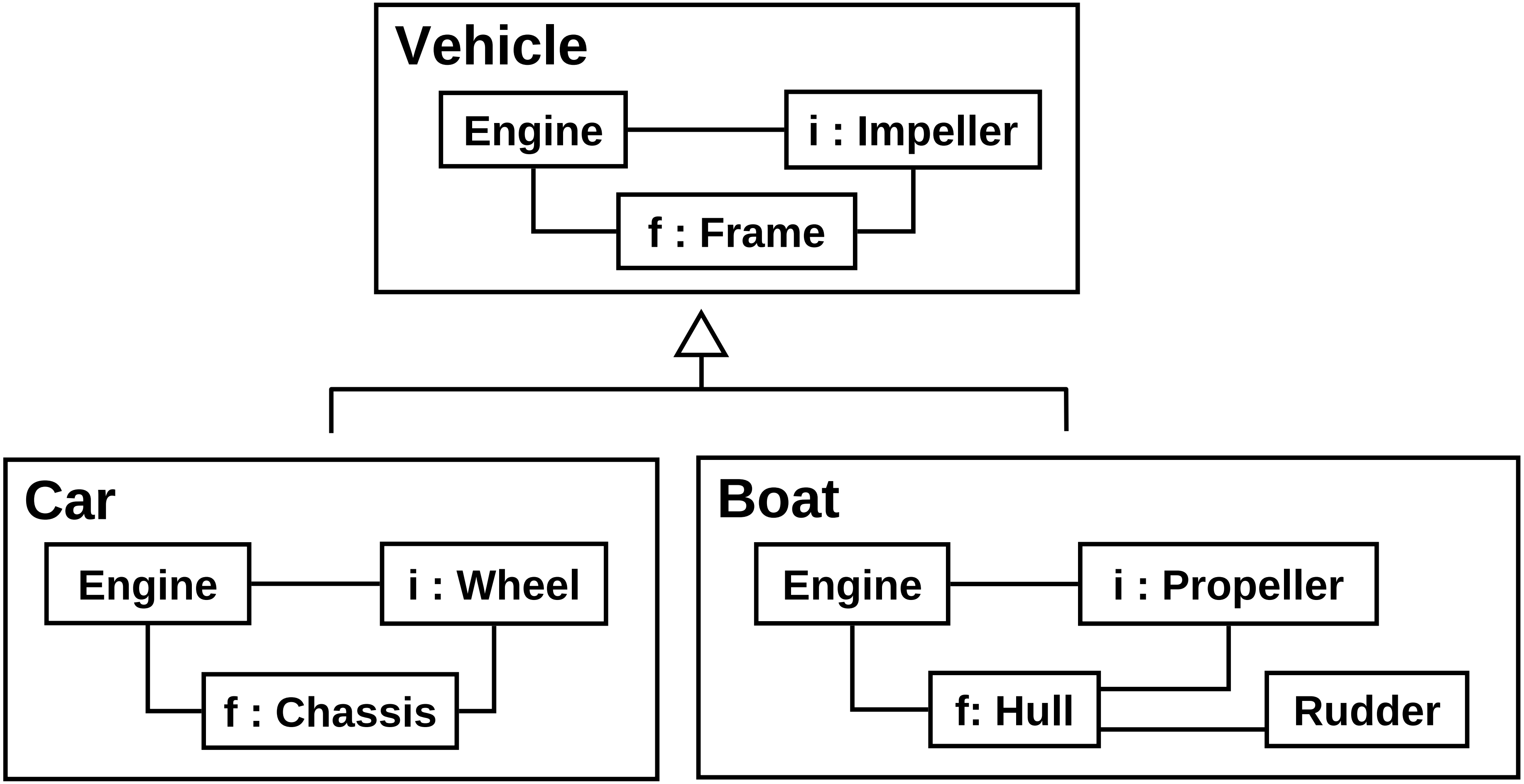
- Connections in context.
- Reuse of other assemblies.

Interconnected Subelements



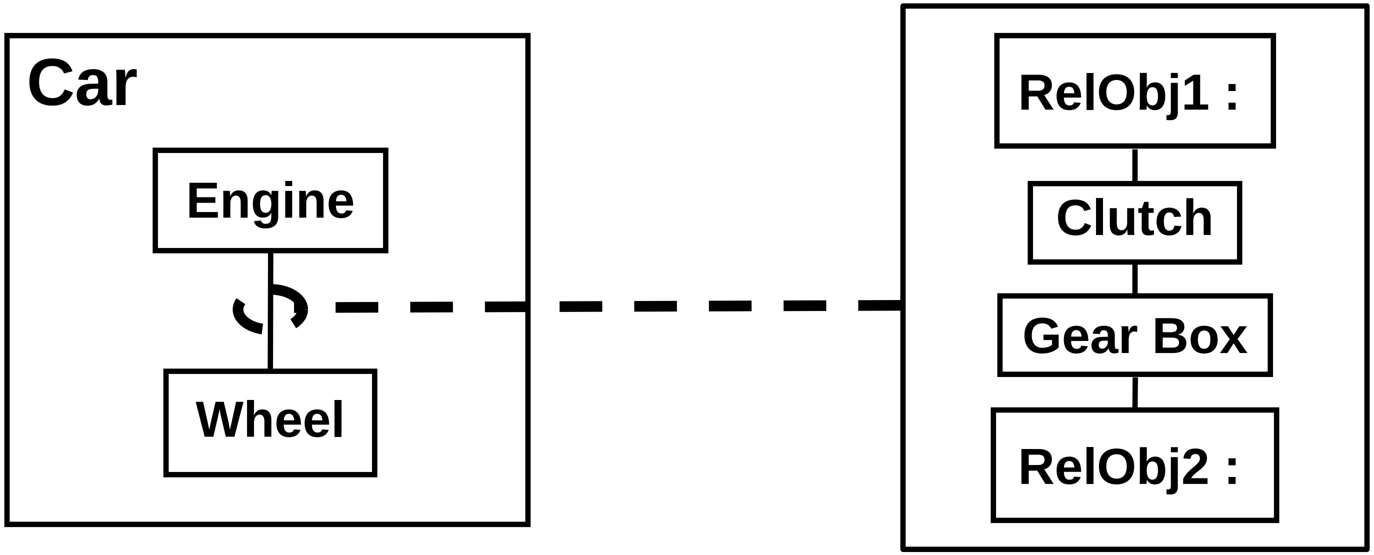
- Interconnections between elements of elements (“ports”).

Interconnection Inheritance



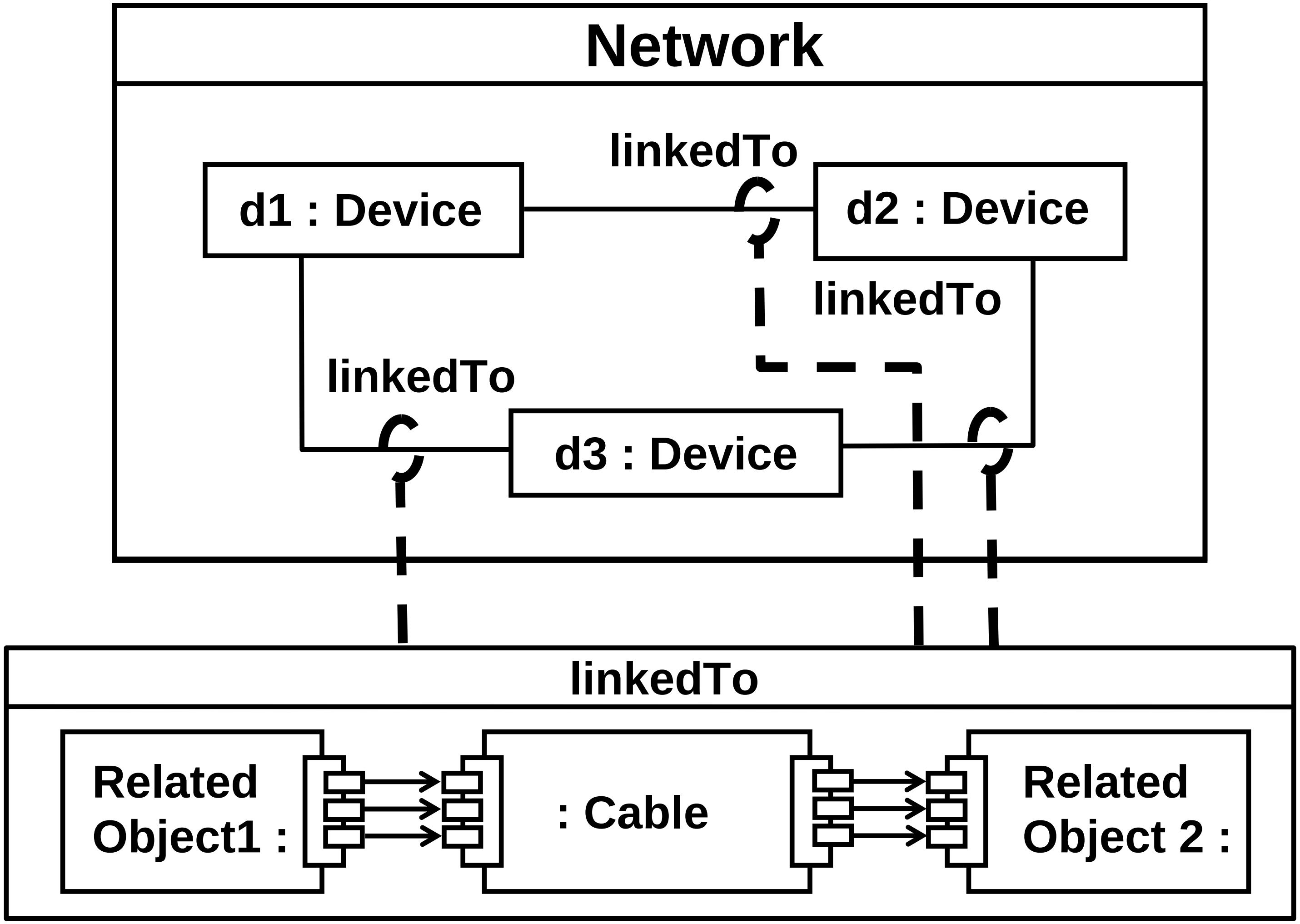
- Inherit, add, specialize interconnections in taxonomy.

Interconnection Decomposition



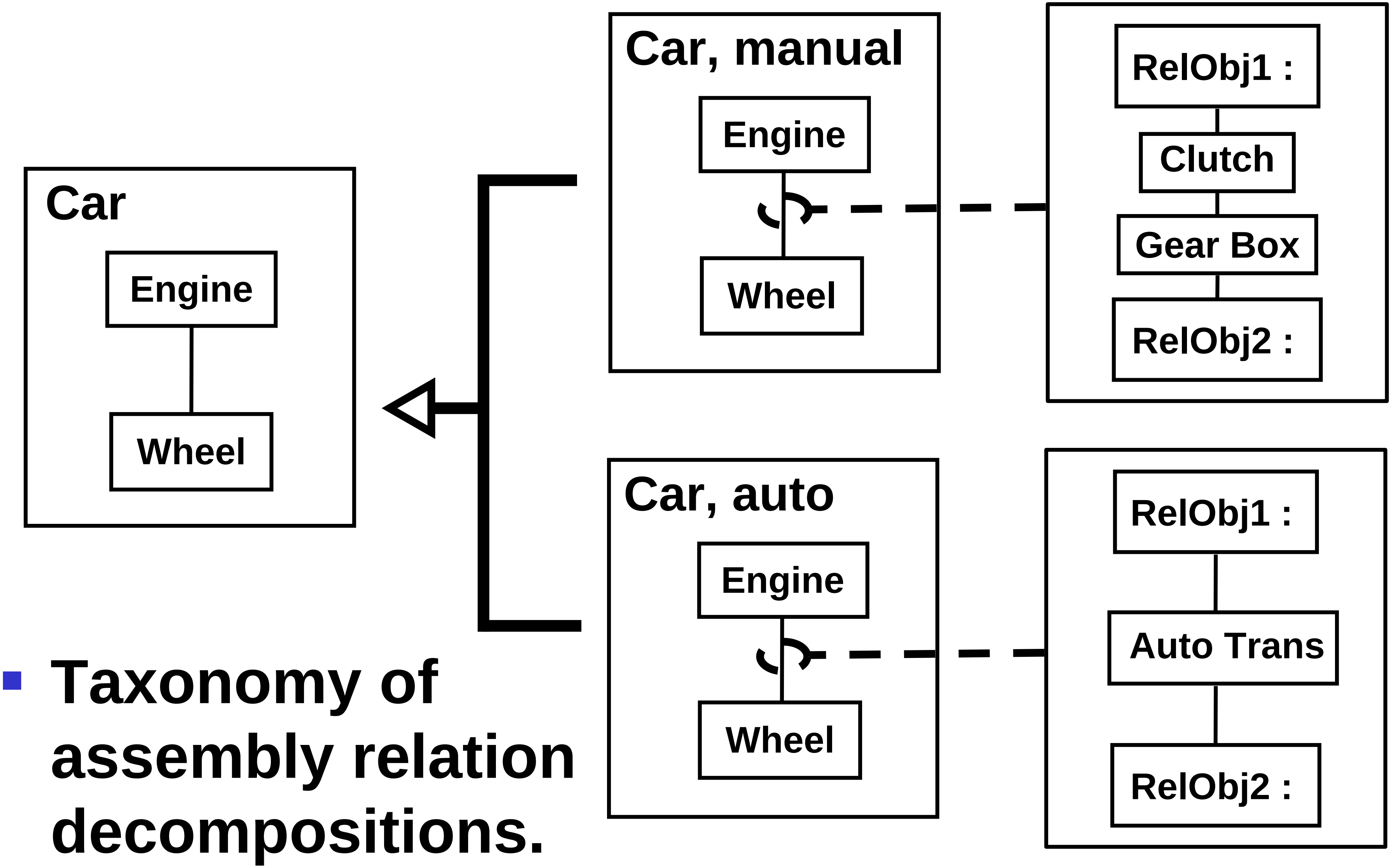
- **Interconnection has subassemblies and interconnections of its own.**

Interconnection Decomposition



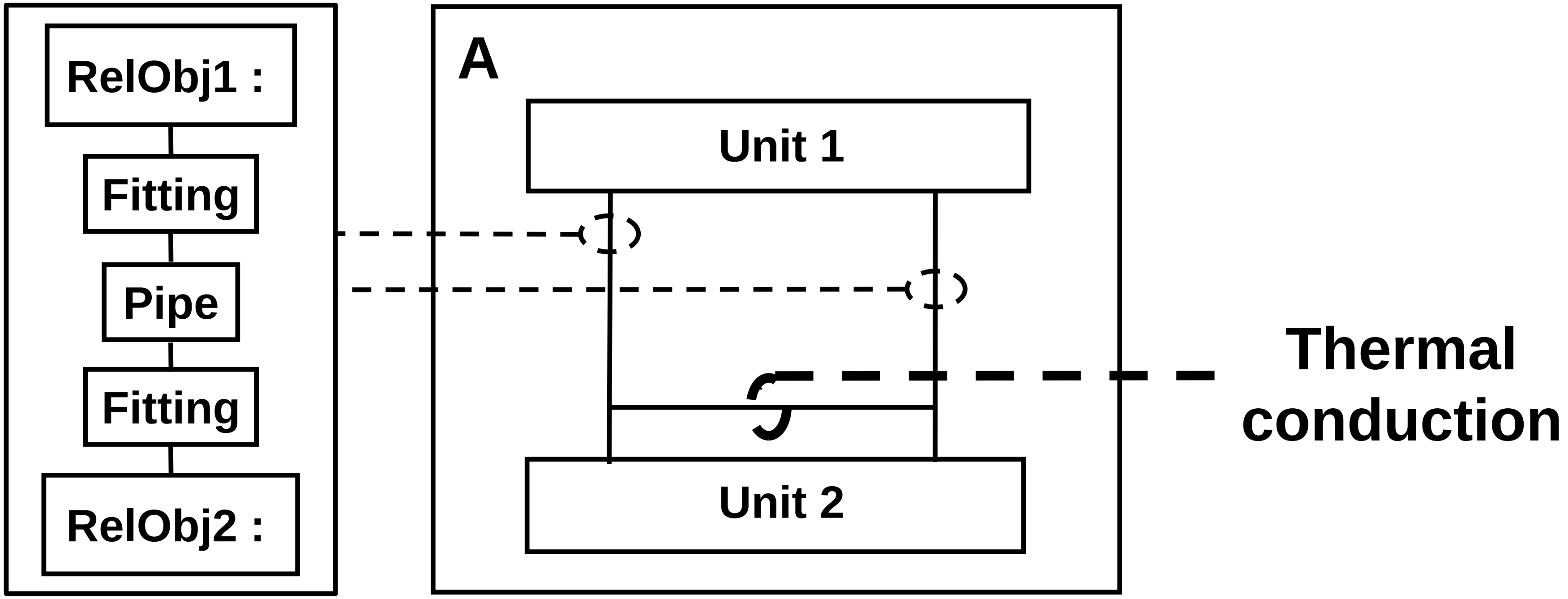
- **Reusing the same relational decomp.**

Alternative Relation Decomp



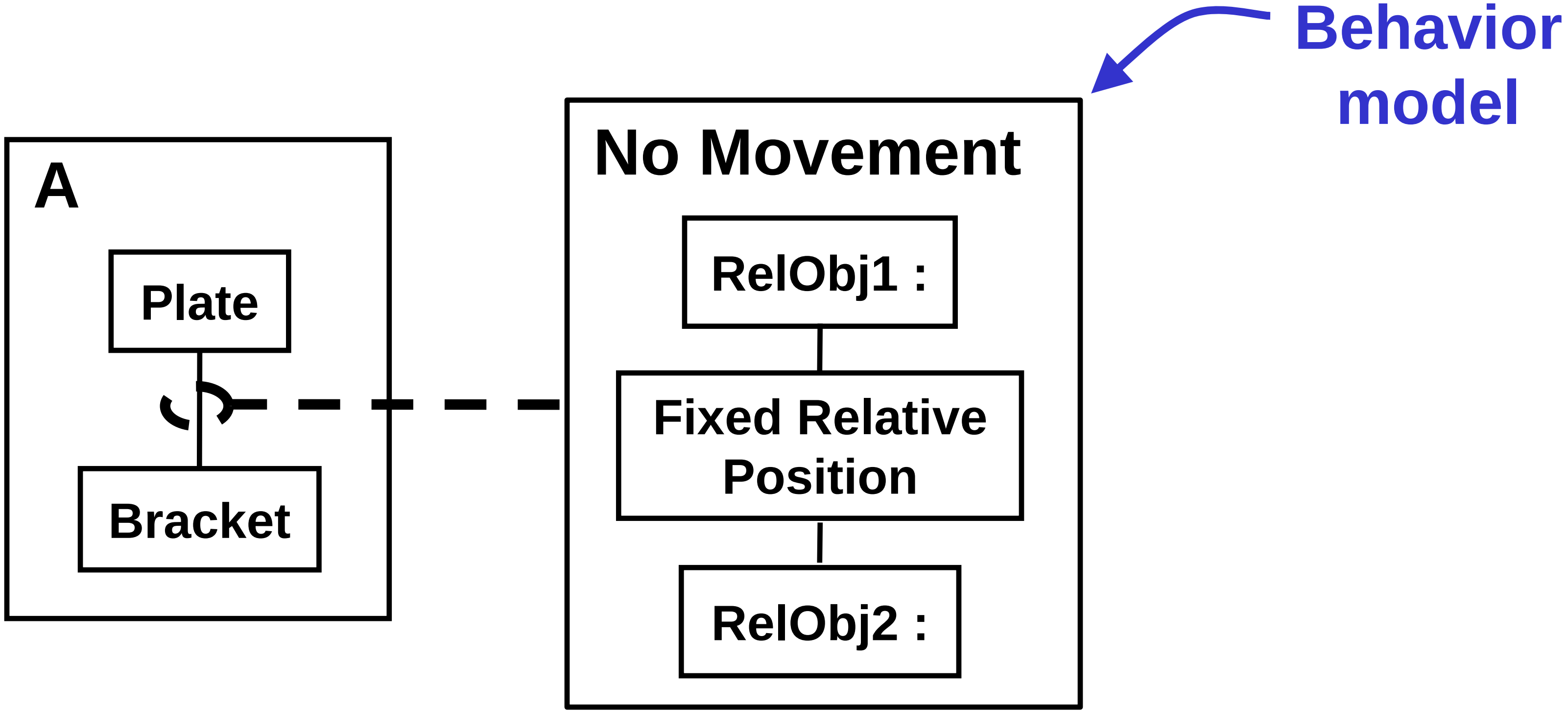
- Taxonomy of assembly relation decompositions.

Interconnections between Interconnections



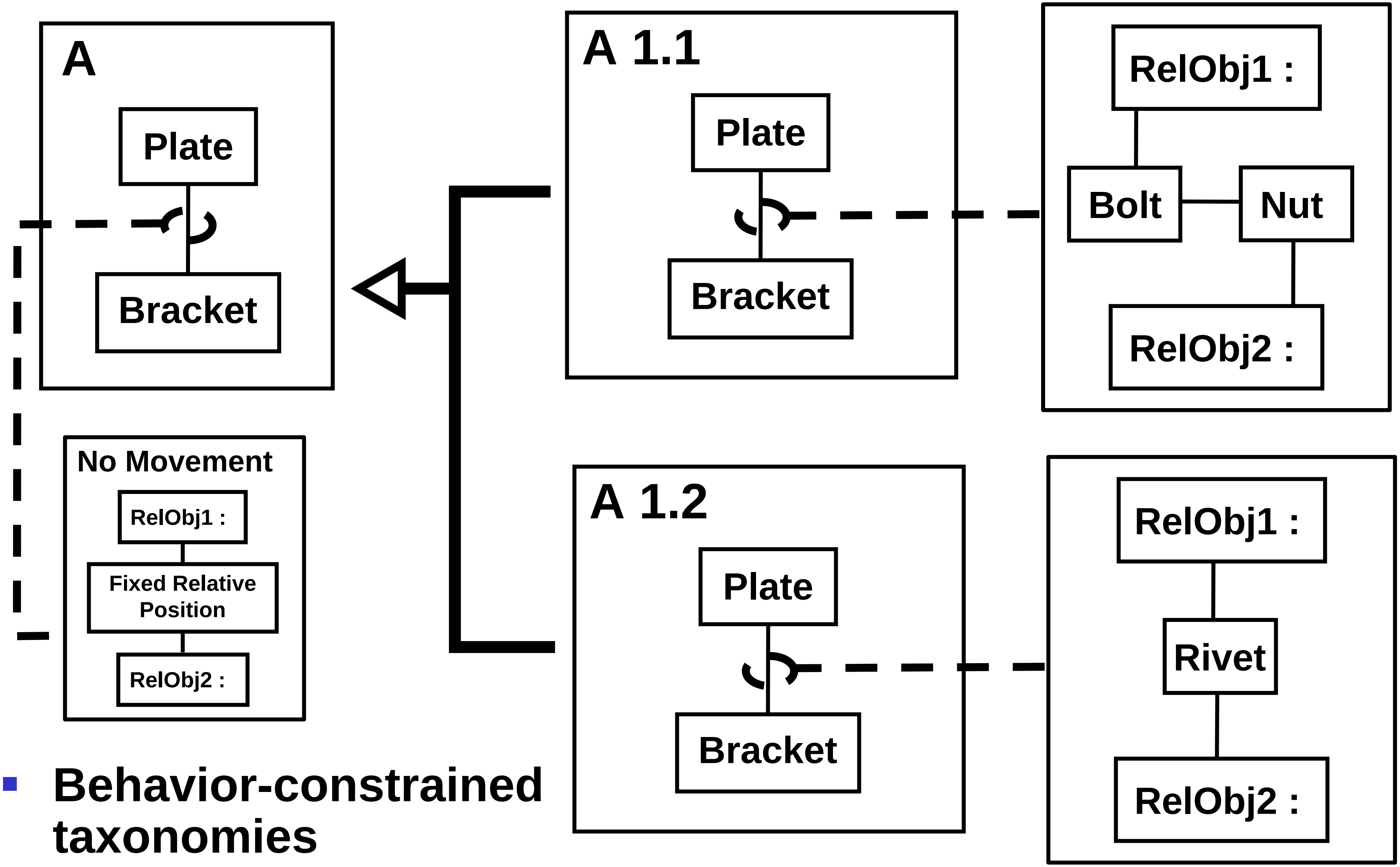
- **Interconnections can be interconnected.**

Behaviors as Interconnections



- **Behaviors relate the objects participating in them.**
- **Plate and bracket participate in a behavior that keeps their relative position constant.**

Alternative Decompositions of Behavior Connections



- **Behavior-constrained taxonomies**

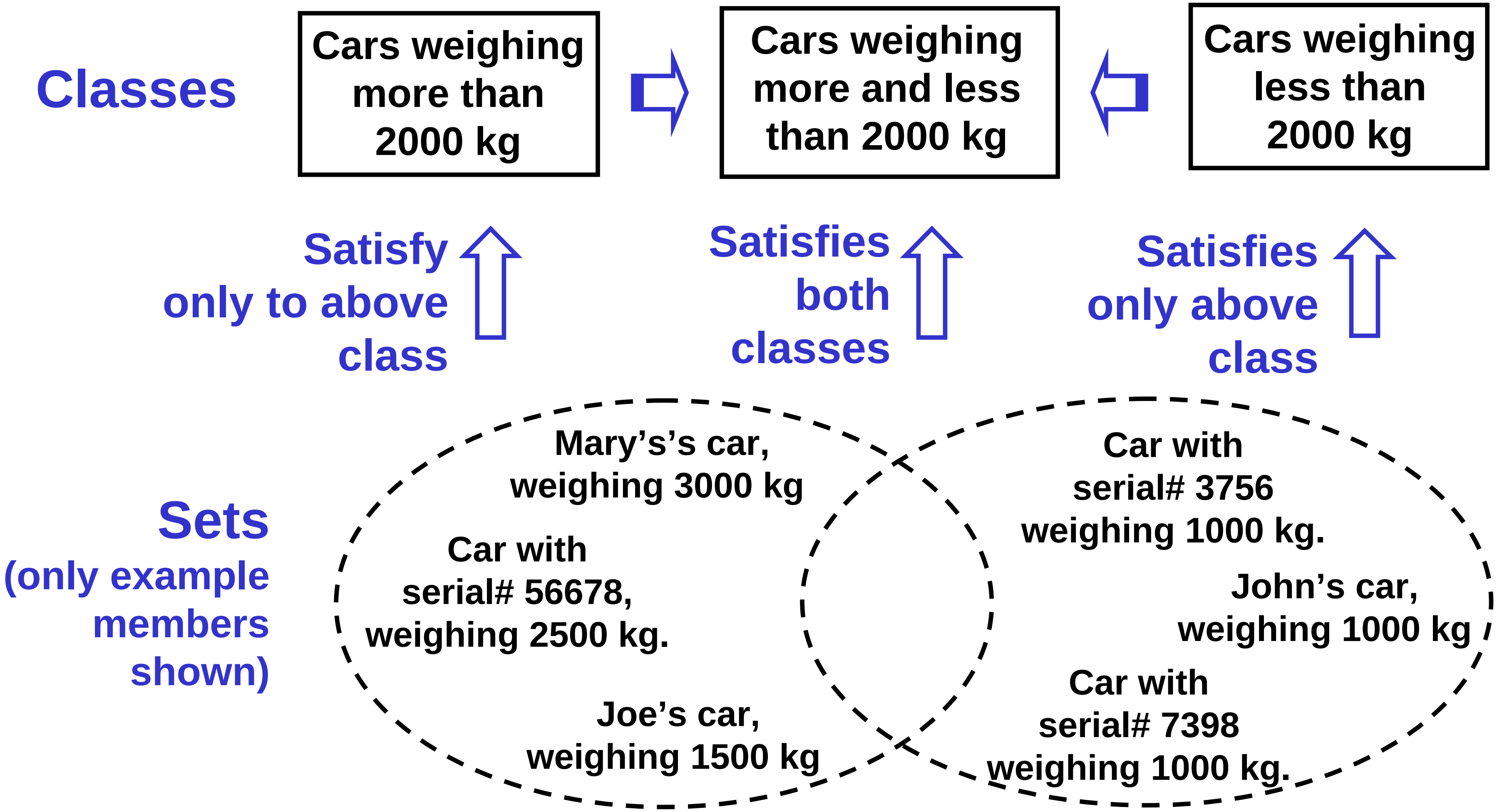
Ontology

- Two kinds of information modeling:
  - Modeling software that carries and manipulates information (*software modeling*).
  - Modeling things that information is about (*ontology modeling*).
- Differ in their styles of classification.
  - Software: classes are “factories” from which software objects are created.
  - Ontology: classes are categories of individuals.

Ontology

- Formalized with set theory.
  - Members of the sets are actual things.
  - *Classes* = rules for membership.
- Rules for membership can be about:
  - One, some, or all aspects of things.
  - Things from the past, present, future.
  - Real, intended, or only imagined things.
  - Physically possible or impossible things.
  - Things with alot or little in common.
- Power from separating membership rules from members themselves.

Ontology



Classes

Cars weighing more than 2000 kg

Cars weighing more and less than 2000 kg

Cars weighing less than 2000 kg

Satisfy only to above class

Satisfies both classes

Satisfies only above class

Sets (only example members shown)

Mary’s’s car, weighing 3000 kg

Car with serial# 56678, weighing 2500 kg.

Joe’s car, weighing 1500 kg

Car with serial# 3756 weighing 1000 kg.

John’s car, weighing 1000 kg

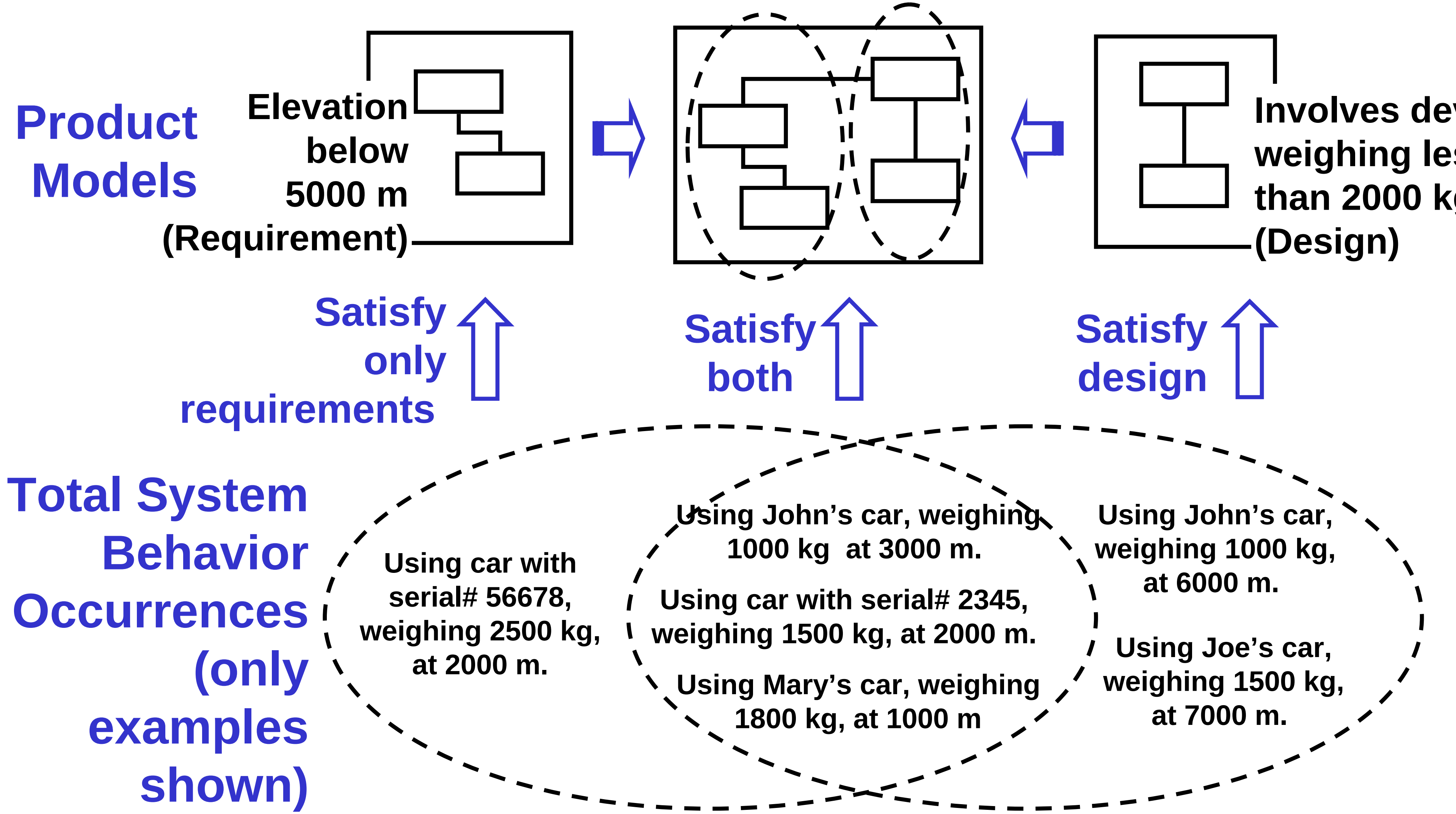
Car with serial# 7398 weighing 1000 kg.

- **Reasoners can operate on classes, without using members.**

Ontological Product Modeling

- Classes of what?
  - Physical things, real or intended (cars with serial numbers)
  - Behavior occurrences (John commuting to work on May 18, 2008).
- Members must be the same kind of thing to support reasoning.
- Behavior occurrences involve individual physical things …
- … but individual things are involved in many behavior occurrences.

Ontological Product Modeling



Product Models

Elevation below 5000 m (Requirement)

Involves dev weighing les than 2000 kg (Design)

Satisfy only requirements

Satisfy both

Satisfy design

Total System Behavior Occurrences (only examples shown)

Using car with serial# 56678, weighing 2500 kg, at 2000 m.

Using John's car, weighing 1000 kg at 3000 m.

Using car with serial# 2345, weighing 1500 kg, at 2000 m.

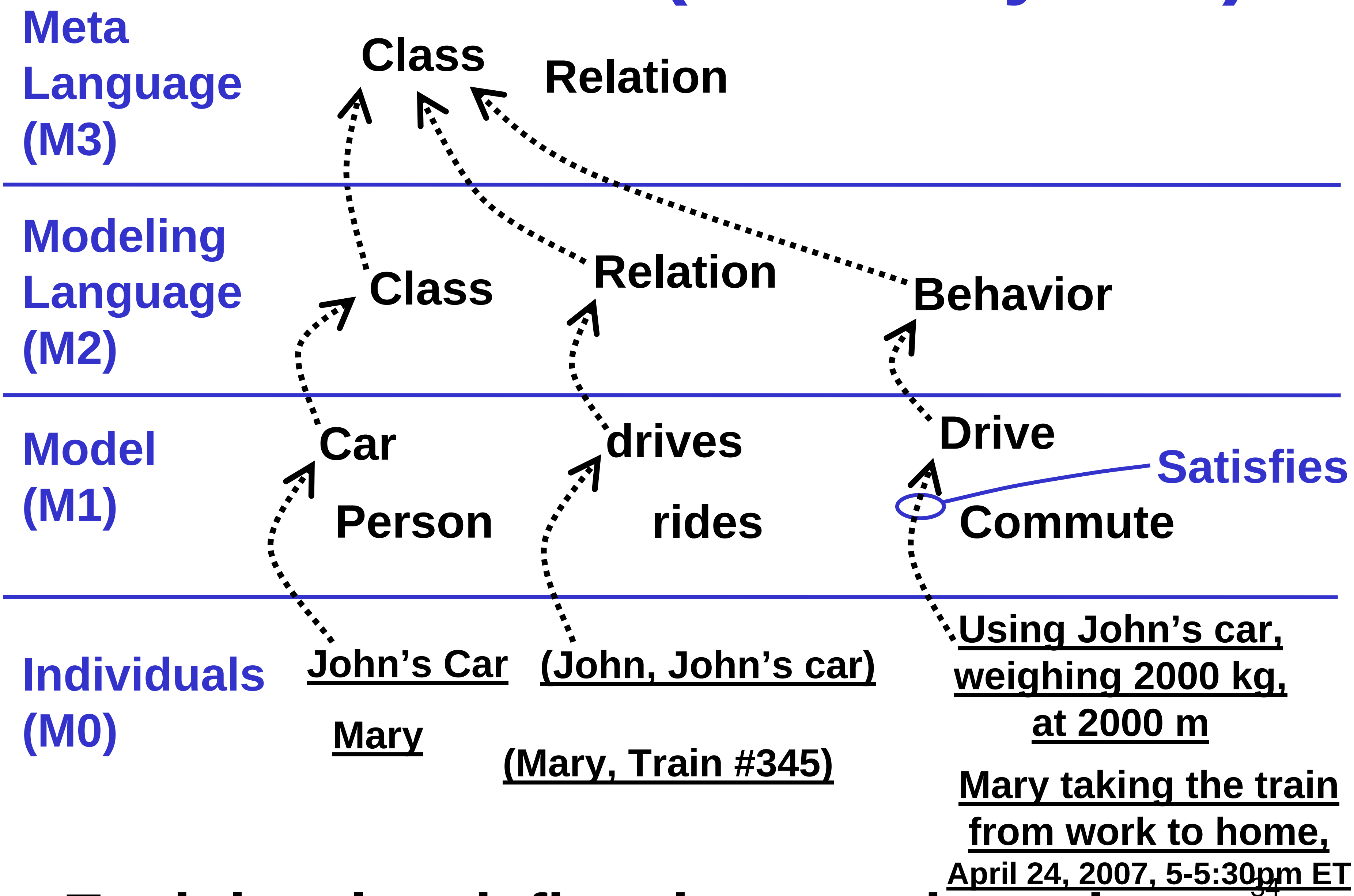
Using Mary's car, weighing 1800 kg, at 1000 m

Using John's car, weighing 1000 kg, at 6000 m.

Using Joe's car, weighing 1500 kg, at 7000 m.

- Classes of behavior occurrences.

Model Levels (“metalayers”)



Meta Language (M3)

Class Relation

Modeling Language (M2)

Class Relation Behavior

Model (M1)

Car drives Drive

Satisfies

Person rides Commute

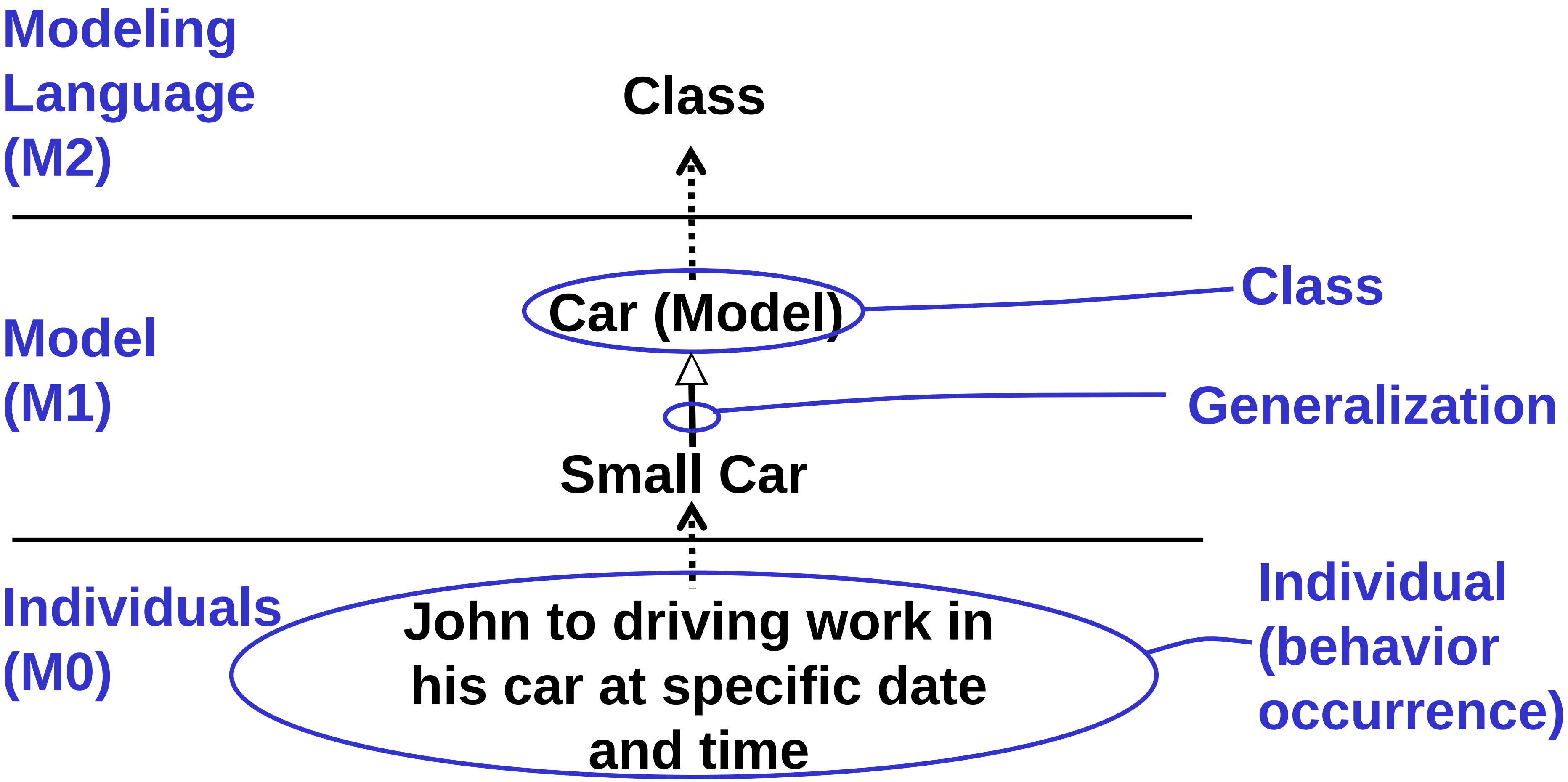
Individuals (M0)

John’s Car (John, John’s car) Using John’s car, weighing 2000 kg, at 2000 m

Mary (Mary, Train #345) Mary taking the train from work to home, April 24, 2007, 5-5:30pm ET

- Each level satisfies the one above it.

Ontology, No Modeling Language



Modeling Language (M2)

Class

Model (M1)

Car (Model)

Class

Generalization

Small Car

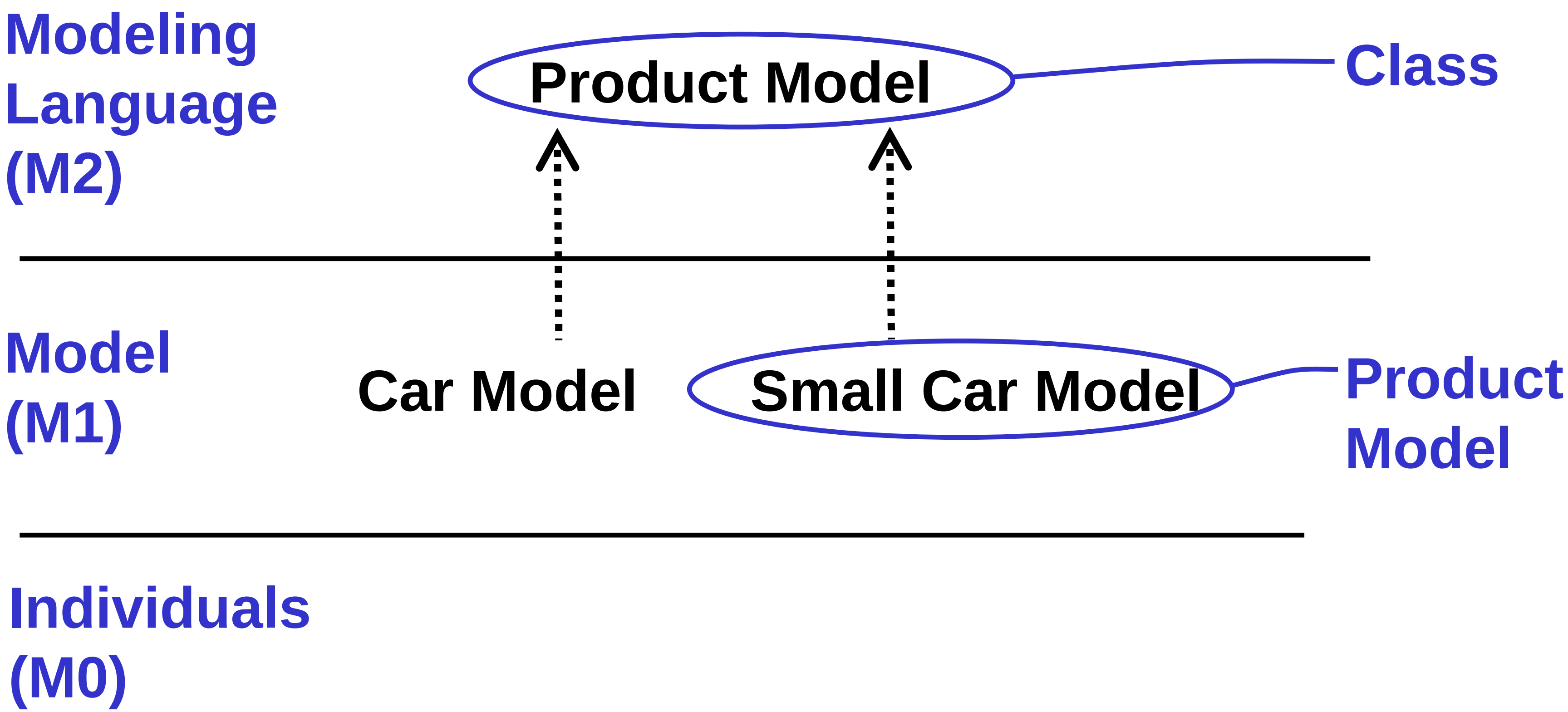
Individuals (M0)

John to driving work in his car at specific date and time

Individual (behavior occurrence)

- Engineer uses ontology language directly.
- M1 product models are classes, can be specialized in M1 and instantiated at M0.

Modeling Language, No Ontology



Modeling Language (M2)

Product Model

Class

Model (M1)

Car Model

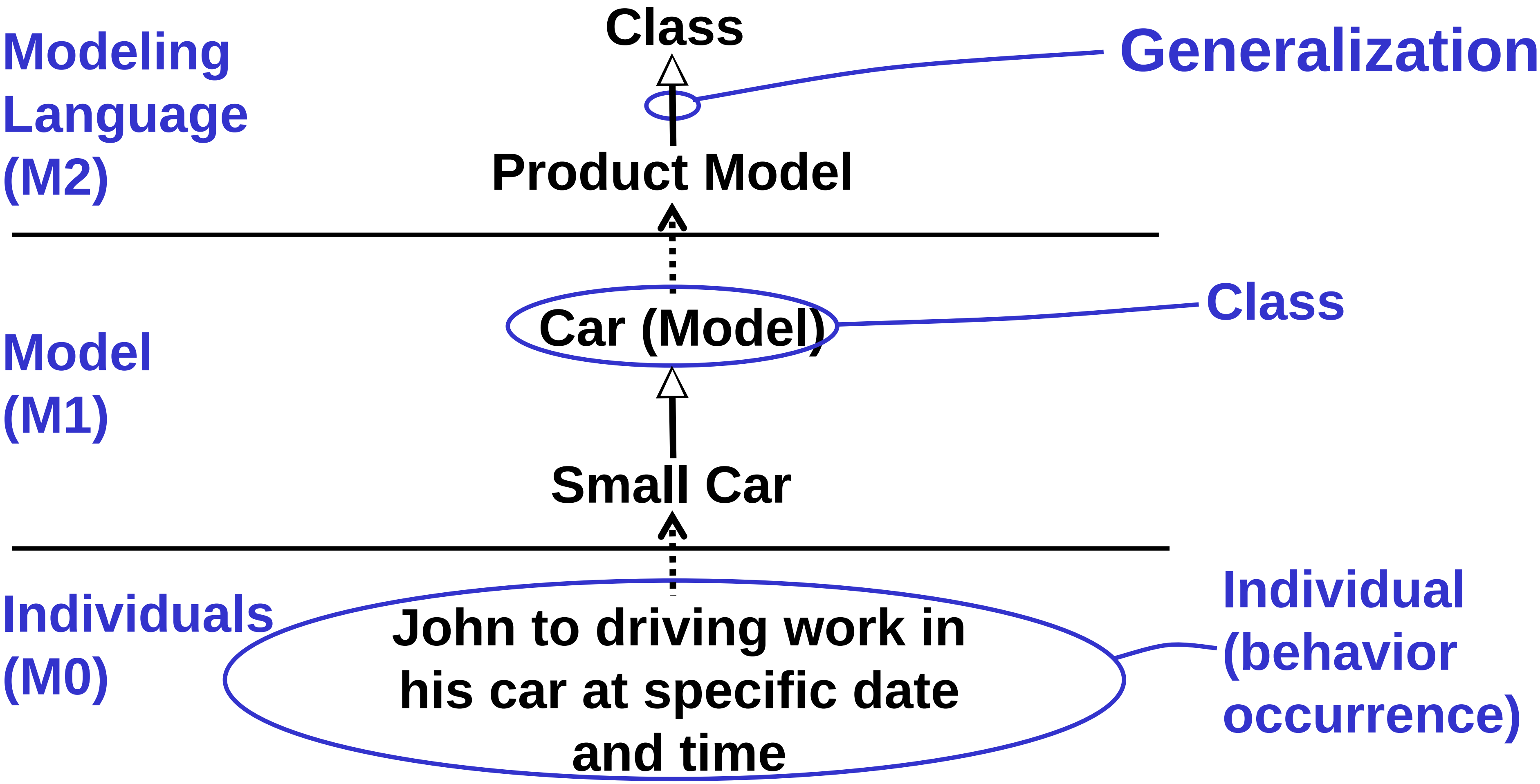
Small Car Model

Product Model

Individuals (M0)

- Engineer uses familiar language.
- Cannot instantiate and specialize M1 product models (they are individuals, not classes).

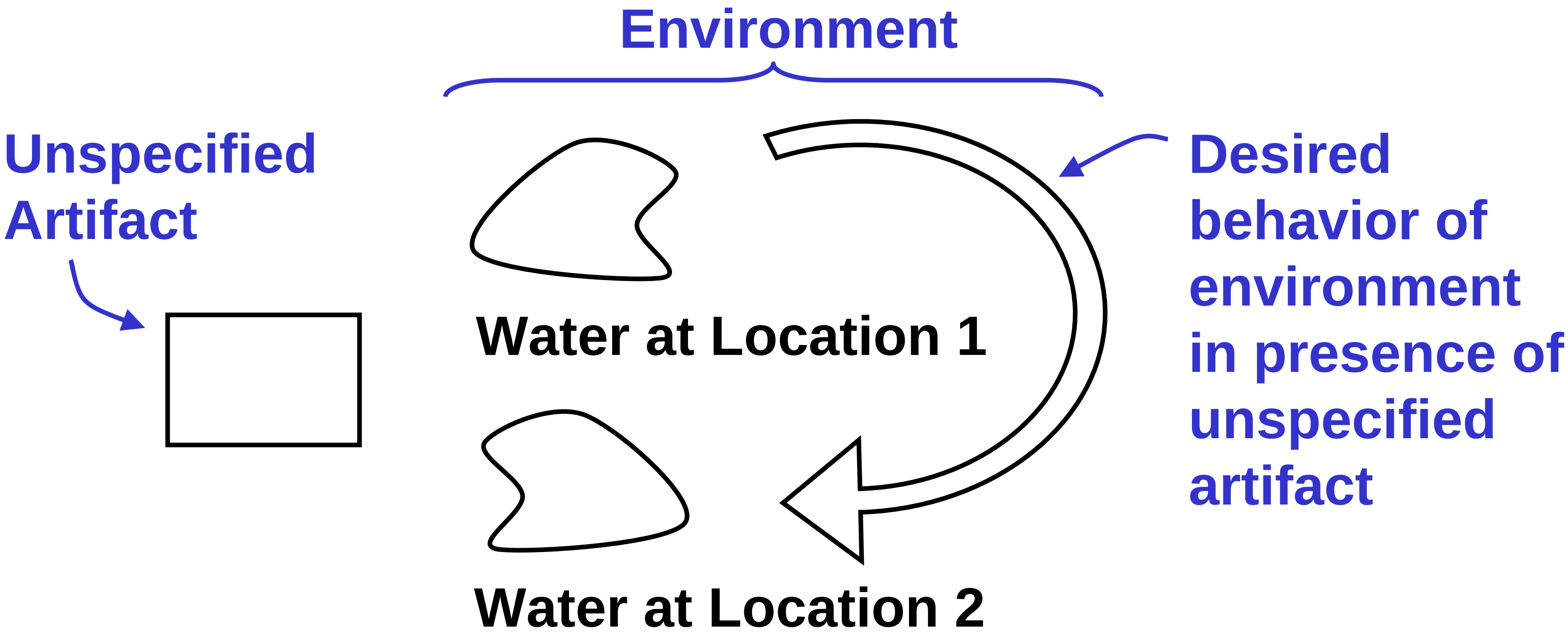
Ontology and Modeling Language



- Engineer uses familiar language.
- M1 product models are classes, can be specialized in M1 and instantiated at M0.

Requirements

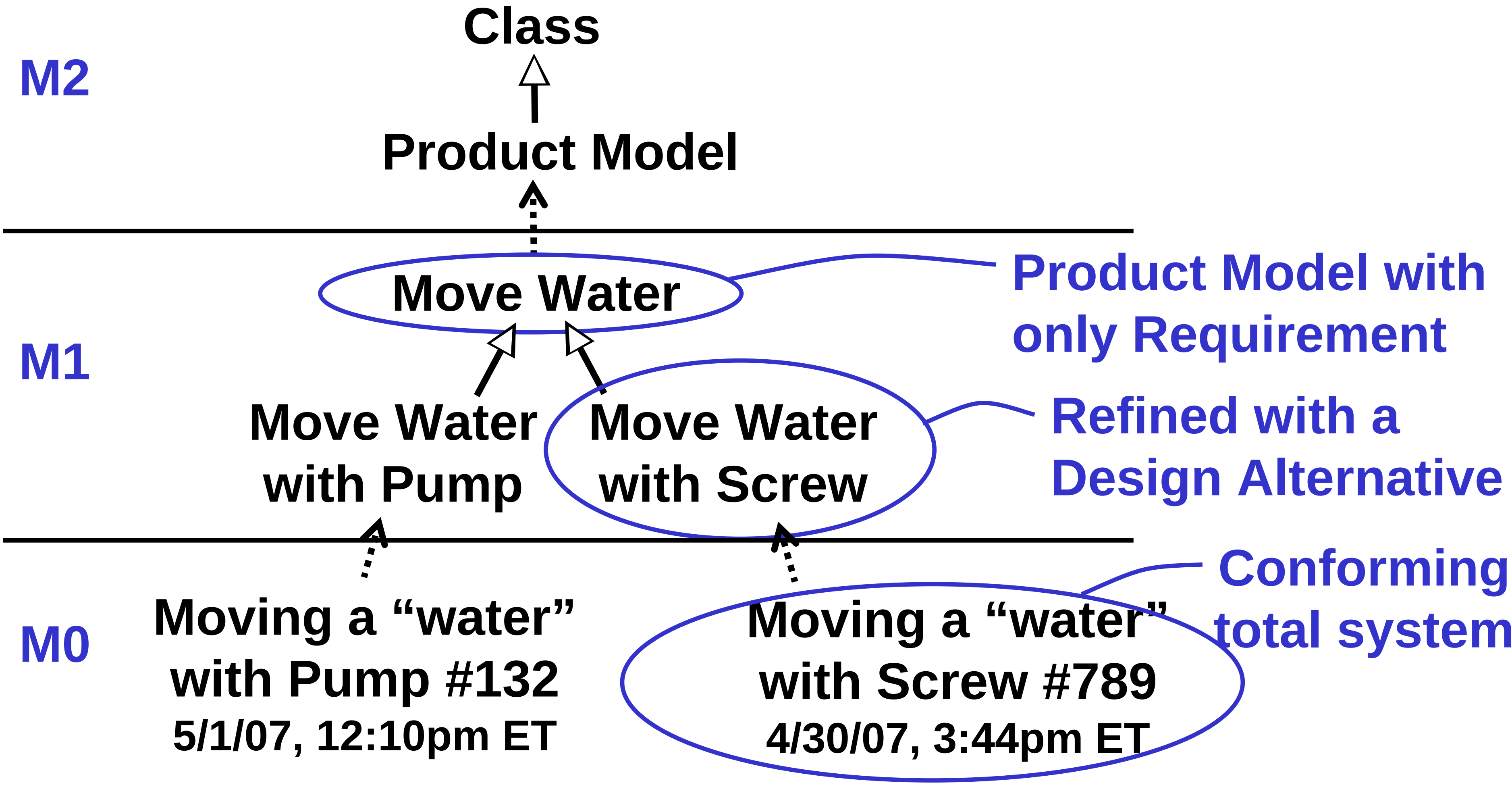
- A product model might be only requirements, no designs (only about the environment, nothing about the artifact).



Alternative Designs

- Different artifact designs satisfy requirements in different ways.
- Example: pump uses pressure to move water, Archimedes screw moves containers of water.
- The above behaviors (putting water under pressure, containing water) are specializations of the desired behavior (moving water) that specify more about the participants.

Alternative Designs



M2

Class

Product Model

M1

Move Water

Product Model with only Requirement

Move Water with Pump

Move Water with Screw

Refined with a Design Alternative

M0

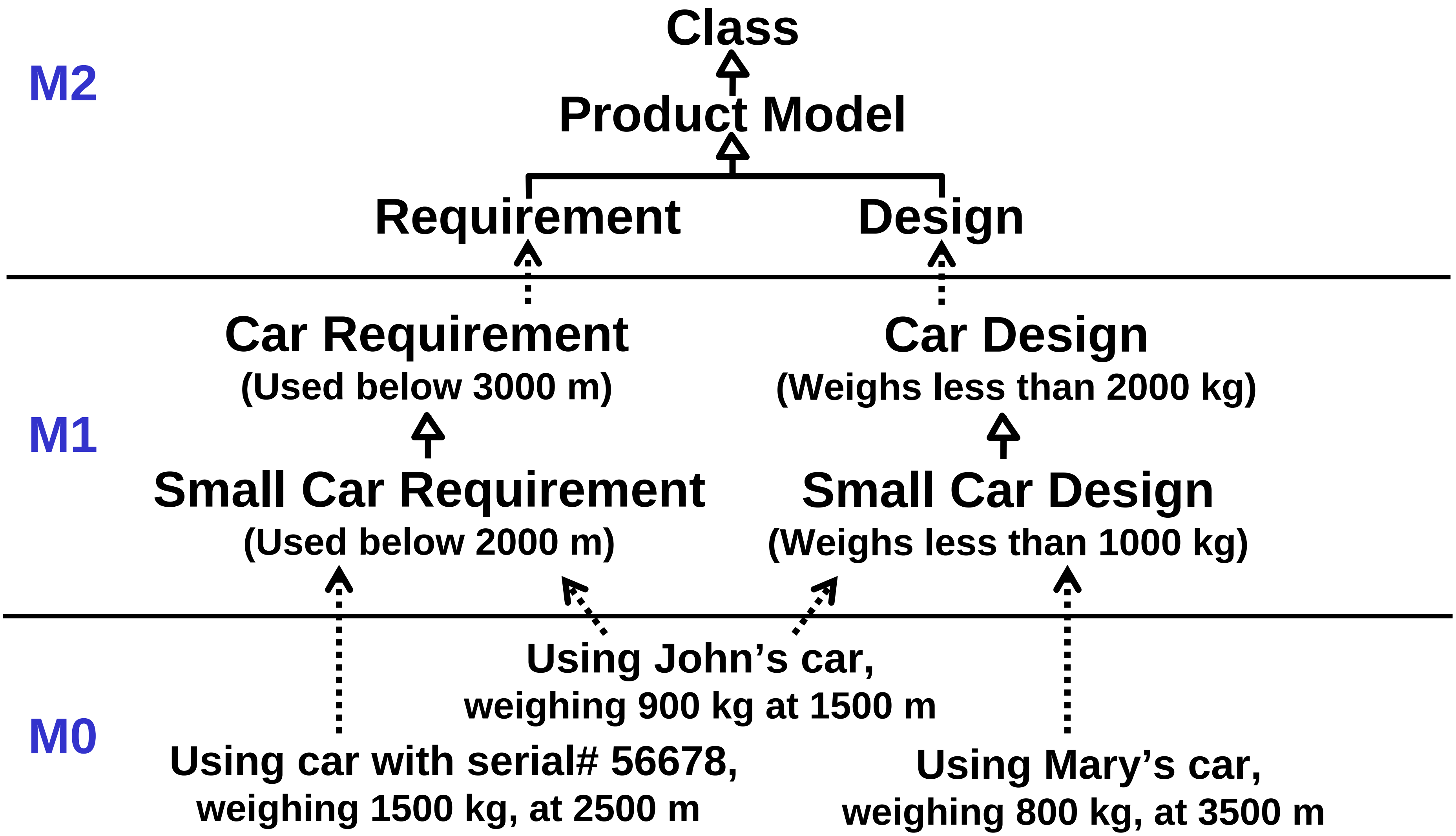
Moving a “water” with Pump #132
5/1/07, 12:10pm ET

Moving a “water” with Screw #789
4/30/07, 3:44pm ET

Conforming total system

- M1 product model that only has requirements is refined to include alternative designs.
- Constrains total systems at M0.

Requirements and Designs



M2

Class

Product Model

Requirement

Design

M1

Car Requirement
(Used below 3000 m)

Car Design
(Weighs less than 2000 kg)

Small Car Requirement
(Used below 2000 m)

Small Car Design
(Weighs less than 1000 kg)

M0

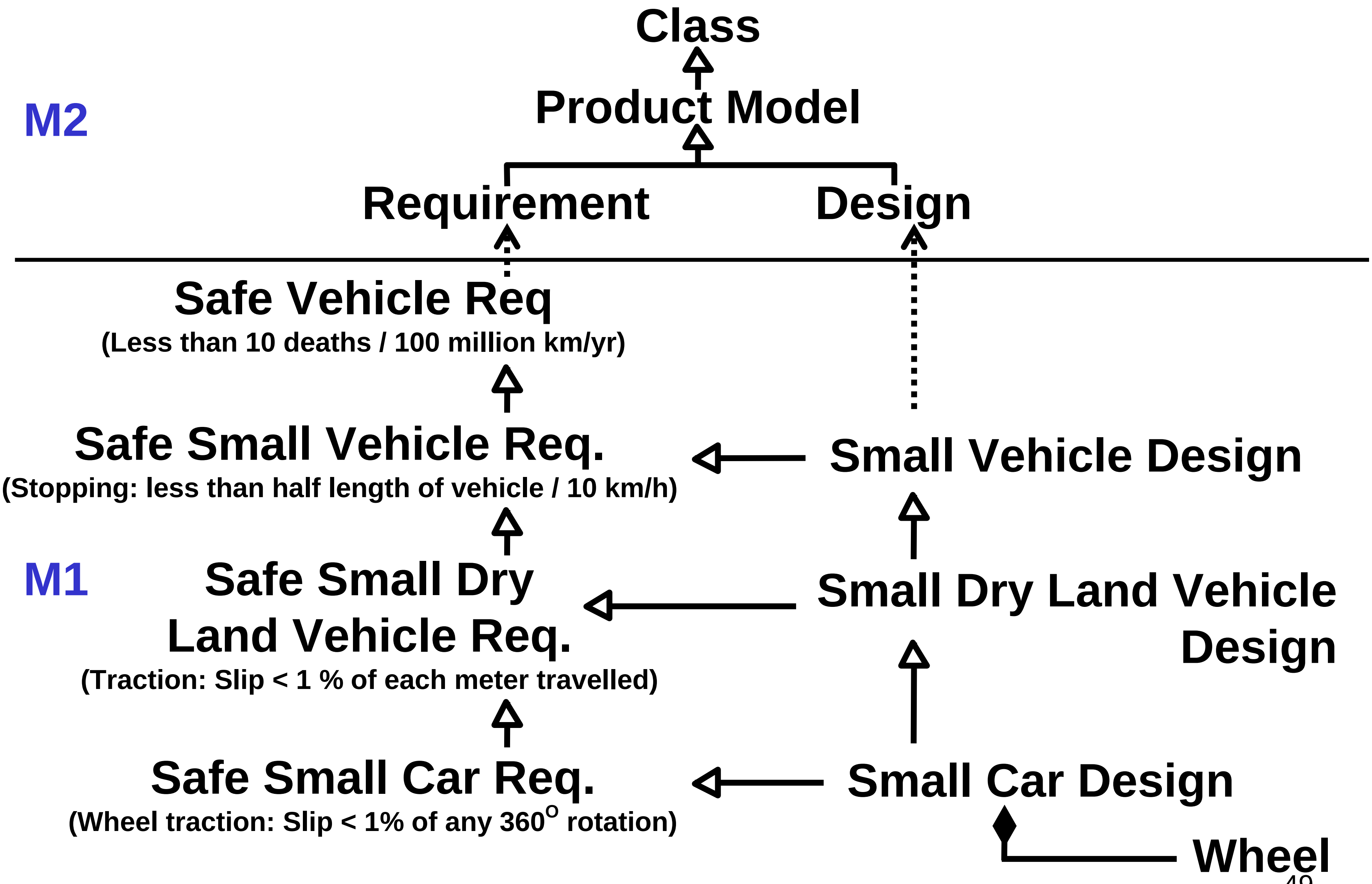
Using John’s car,
weighing 900 kg at 1500 m

Using car with serial# 56678,
weighing 1500 kg, at 2500 m

Using Mary’s car,
weighing 800 kg, at 3500 m

- Product model can be requirements and / or design.
- M0 total system conforms to the M1 product model (consistency checking).

Requirements “roll down”



Class

Product Model

M2

Requirement

Design

Safe Vehicle Req

(Less than 10 deaths / 100 million km/yr)

Safe Small Vehicle Req.

(Stopping: less than half length of vehicle / 10 km/h)

Small Vehicle Design

M1

Safe Small Dry Land Vehicle Req.

(Traction: Slip < 1 % of each meter travelled)

Small Dry Land Vehicle Design

Safe Small Car Req.

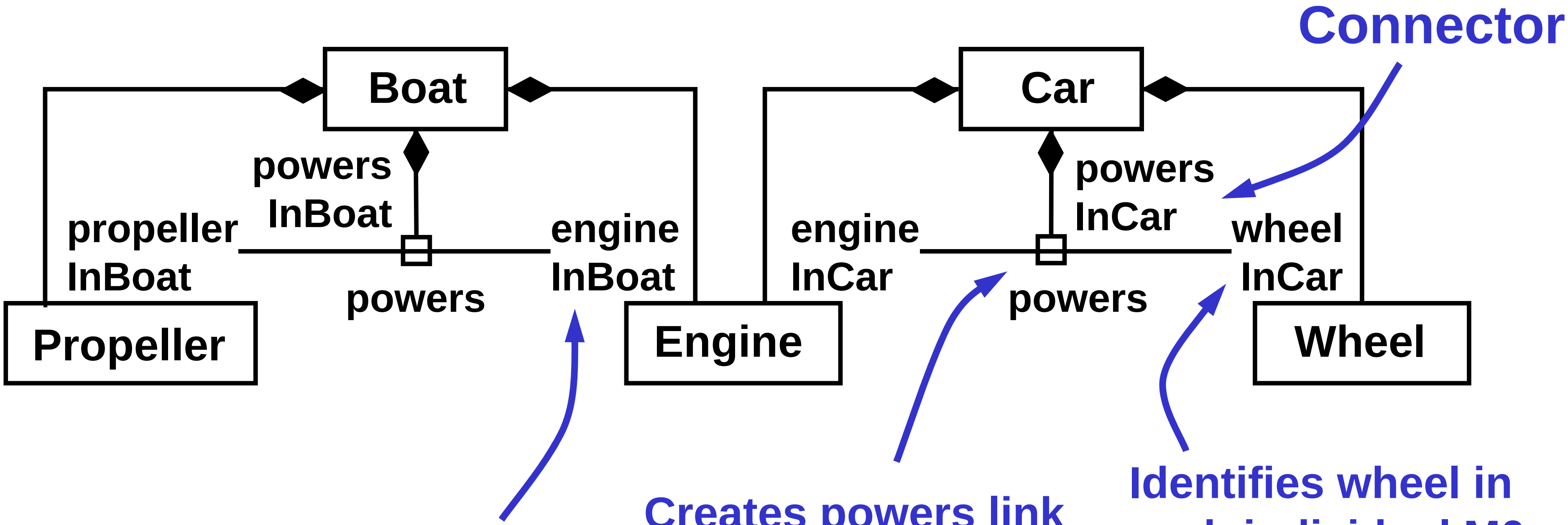
(Wheel traction: Slip < 1% of any 360^{o} rotation)

Small Car Design

Wheel

- Requirements refined along with design.

Interconnections



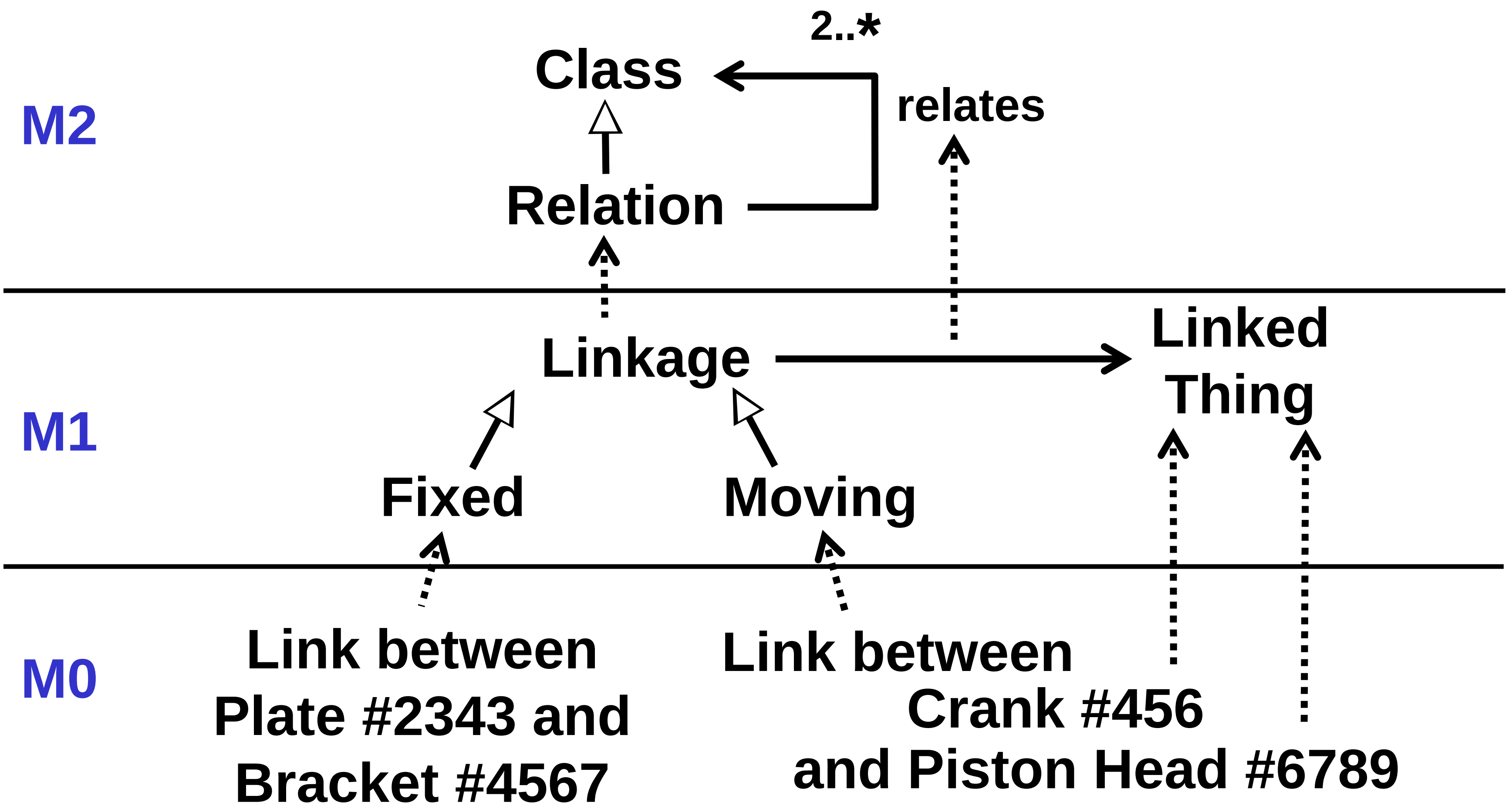
Identifies engine in each individual M0 boat.

Creates powers link between engine and wheel identified in each individual car

Identifies wheel in each individual M0 car.

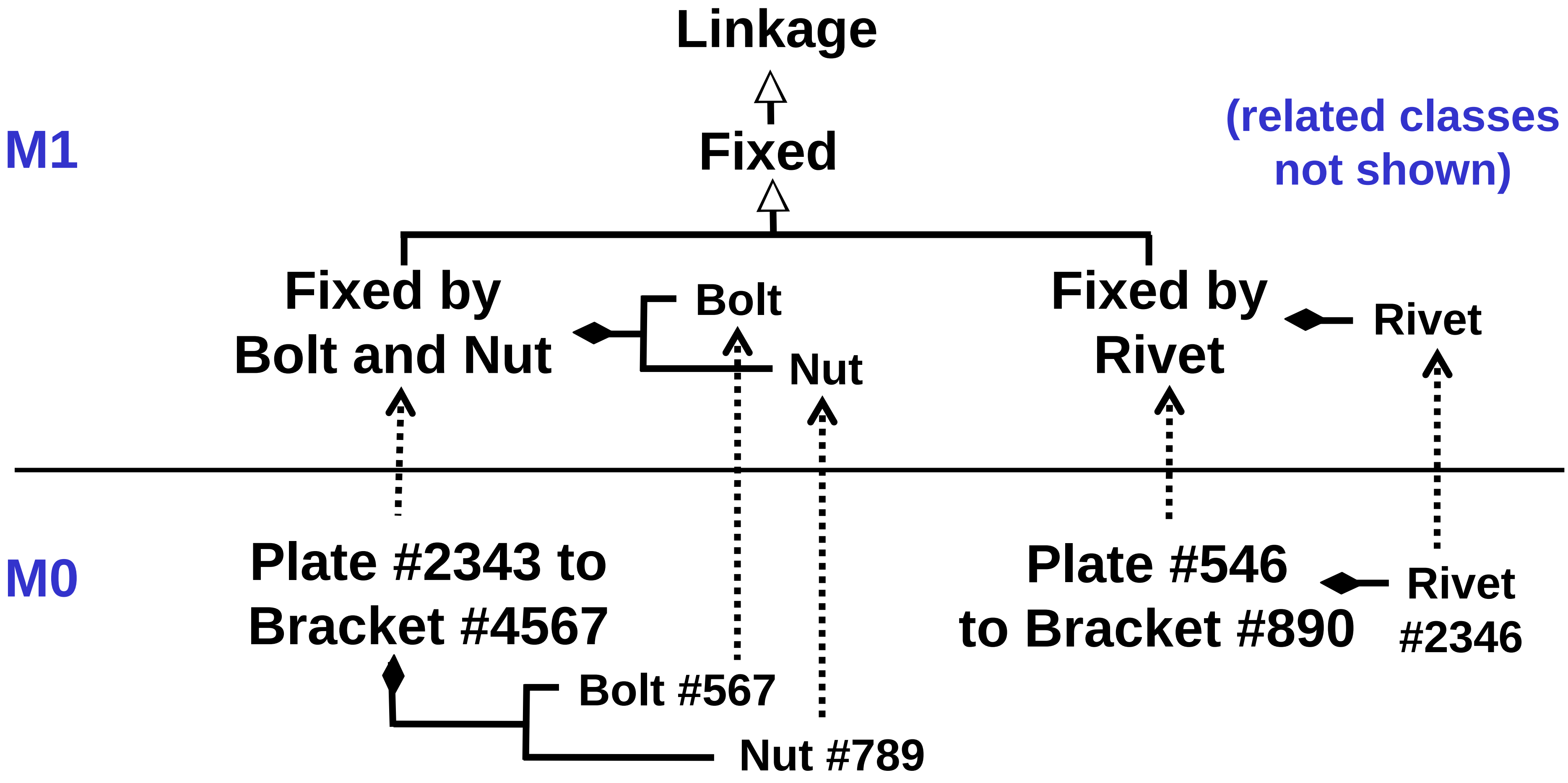
- Connectors relate part-whole relations:
  - to identify parts/subassemblies in each individual M0 whole
  - and link with another relation (powers).

Modeling Relations


2..*
Class
relates
M2
Relation
Linkage
Linked Thing
M1
Fixed
Moving
M0
Link between Plate #2343 and Bracket #4567
Link between Crank #456 and Piston Head #6789

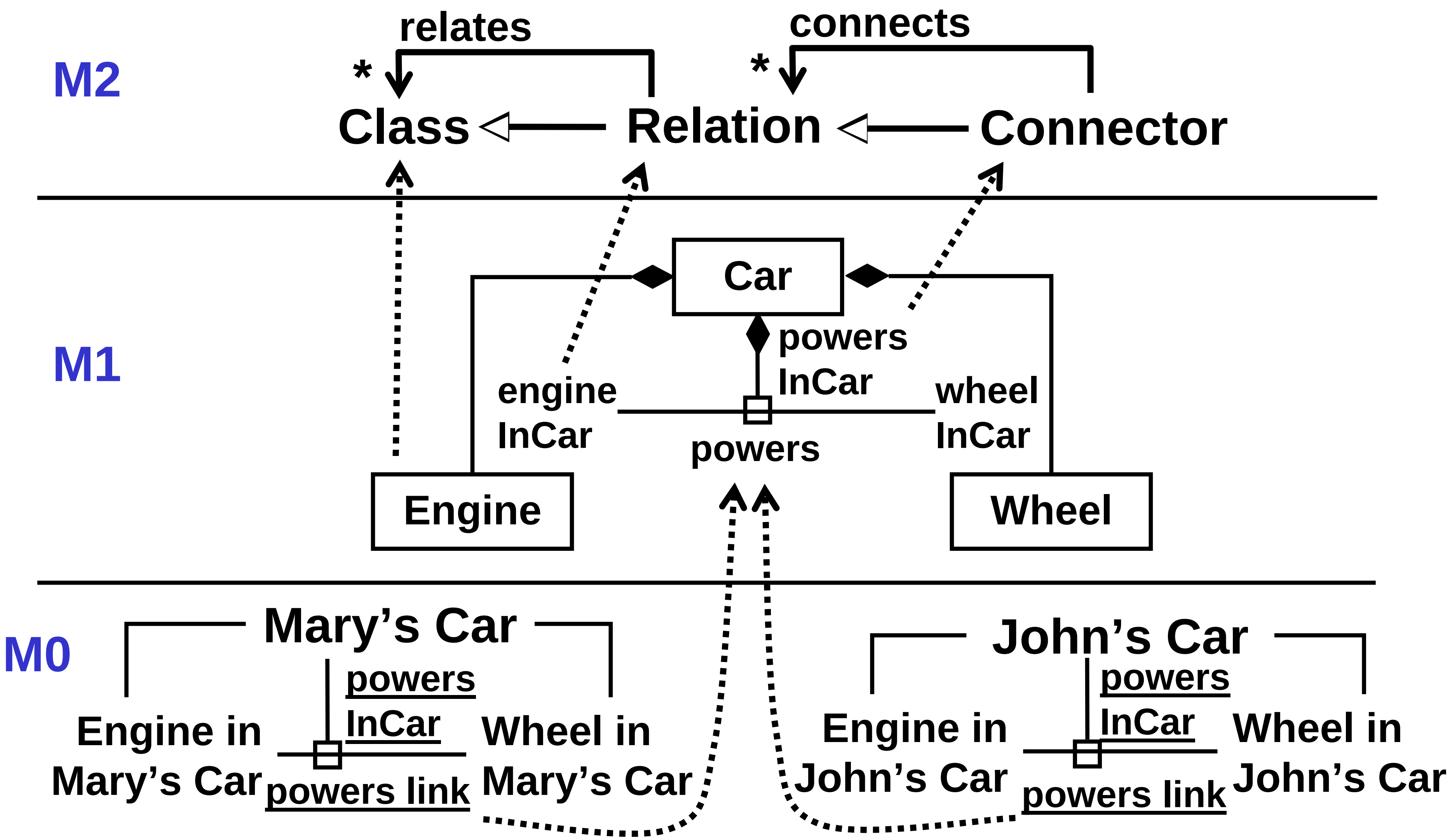
- Relations are classes (M1) of M0 links.
- Can be specialized at M1 and have conforming M0 links between M0 entities.

Decomposing Relations



- Relations (classes) can have parts (M1).
- Conforming M0 links have M0 parts (interconnection decomposition).

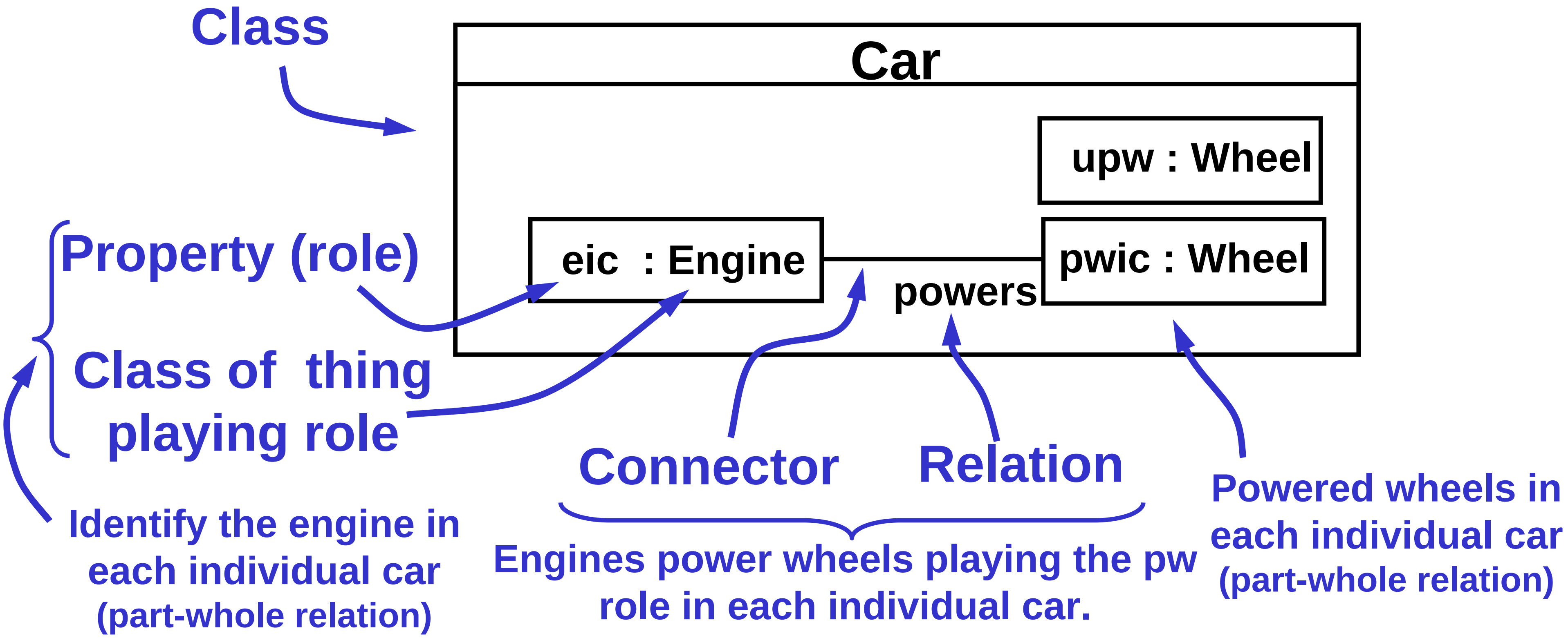
Relations and Connectors



- Connectors establish M0 links within instances of the containing class.

Connectors

UML Notation



Class

Property (role)

Class of thing playing role

Identify the engine in each individual car (part-whole relation)

Connector

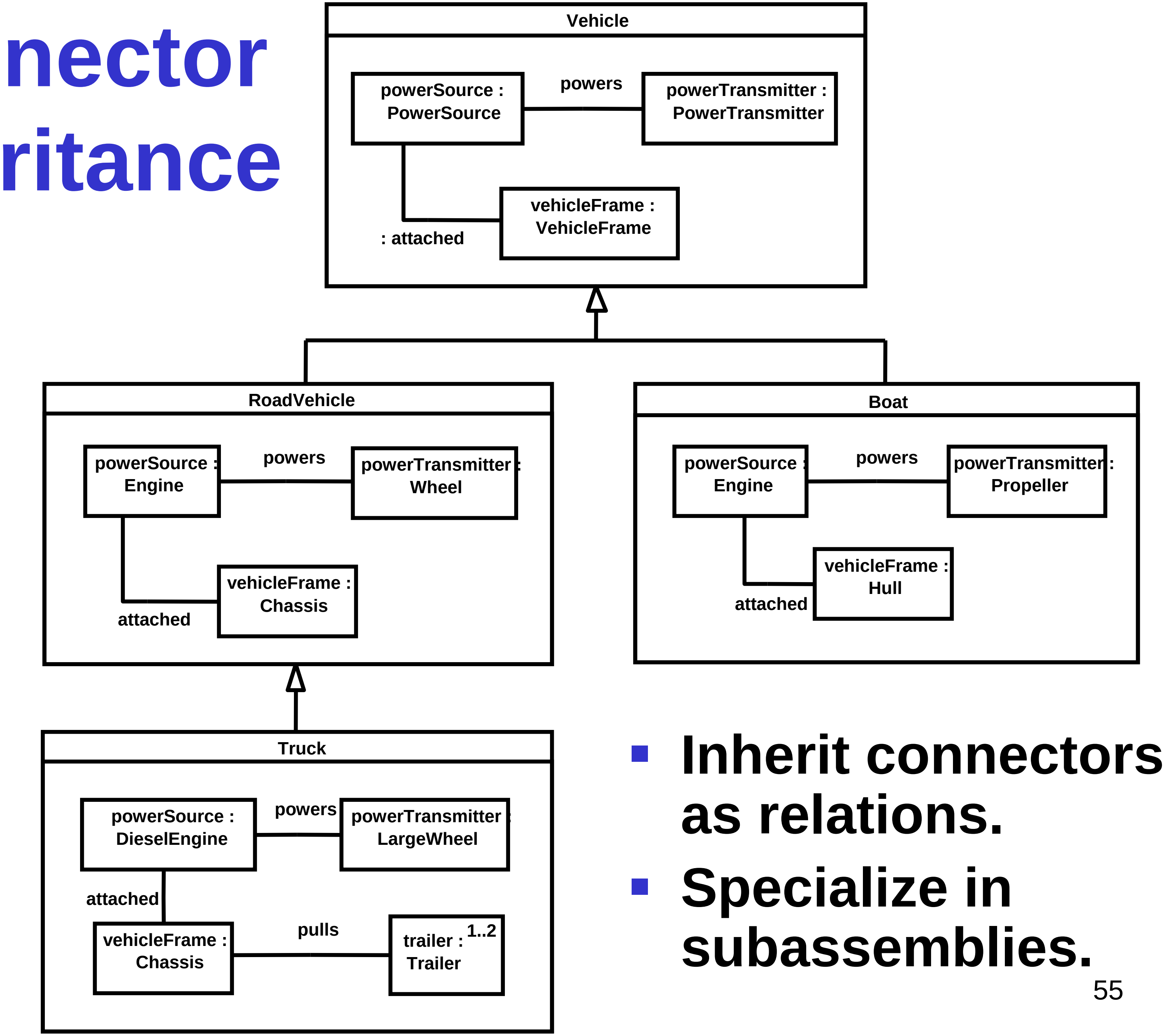
Relation

Engines power wheels playing the pw role in each individual car.

Powered wheels in each individual car (part-whole relation)

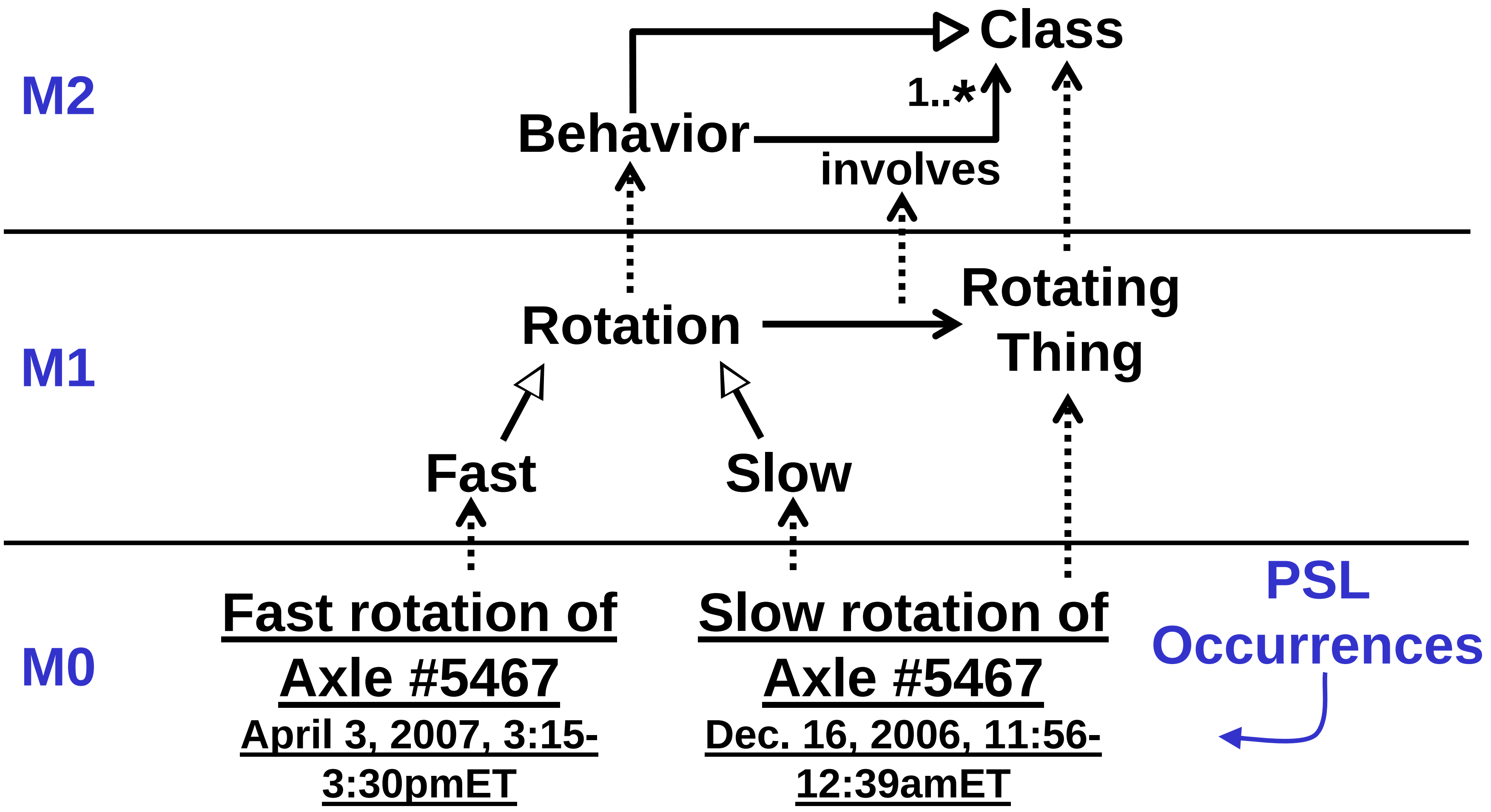
- Properties are “usages” limited to each individual car:
  - Part-whole relations use engines and wheels.
  - Connector uses powers relation.
- Multiple usages of the same subassembly.

Connector Inheritance



- Inherit connectors as relations.
- Specialize in subassemblies.

Modeling Behaviors



M2

Class

1..*

Behavior

involves

M1

Rotation

Rotating Thing

Fast

Slow

M0

Fast rotation of Axle #5467

April 3, 2007, 3:15-3:30pmET

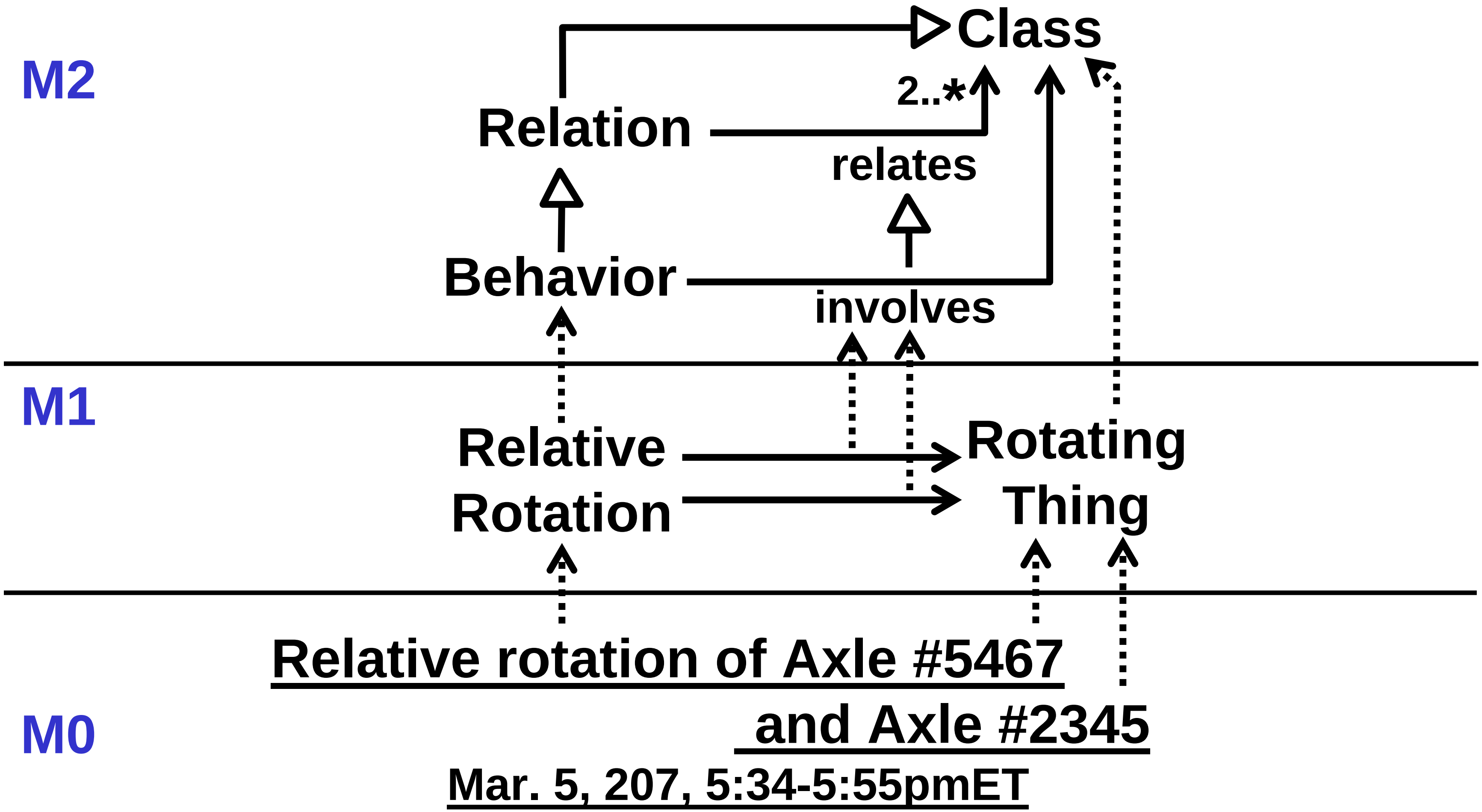
Slow rotation of Axle #5467

Dec. 16, 2006, 11:56-12:39amET

PSL Occurrences

- Behaviors are classes (M1) of M0 “executions”.
- Can be specialized at M1 and have conforming M0 links (modeling behavior occurrences).

Behaviors as Relations



M2

Class

2..*

Relation

relates

Behavior

involves

M1

Relative Rotation

Rotating Thing

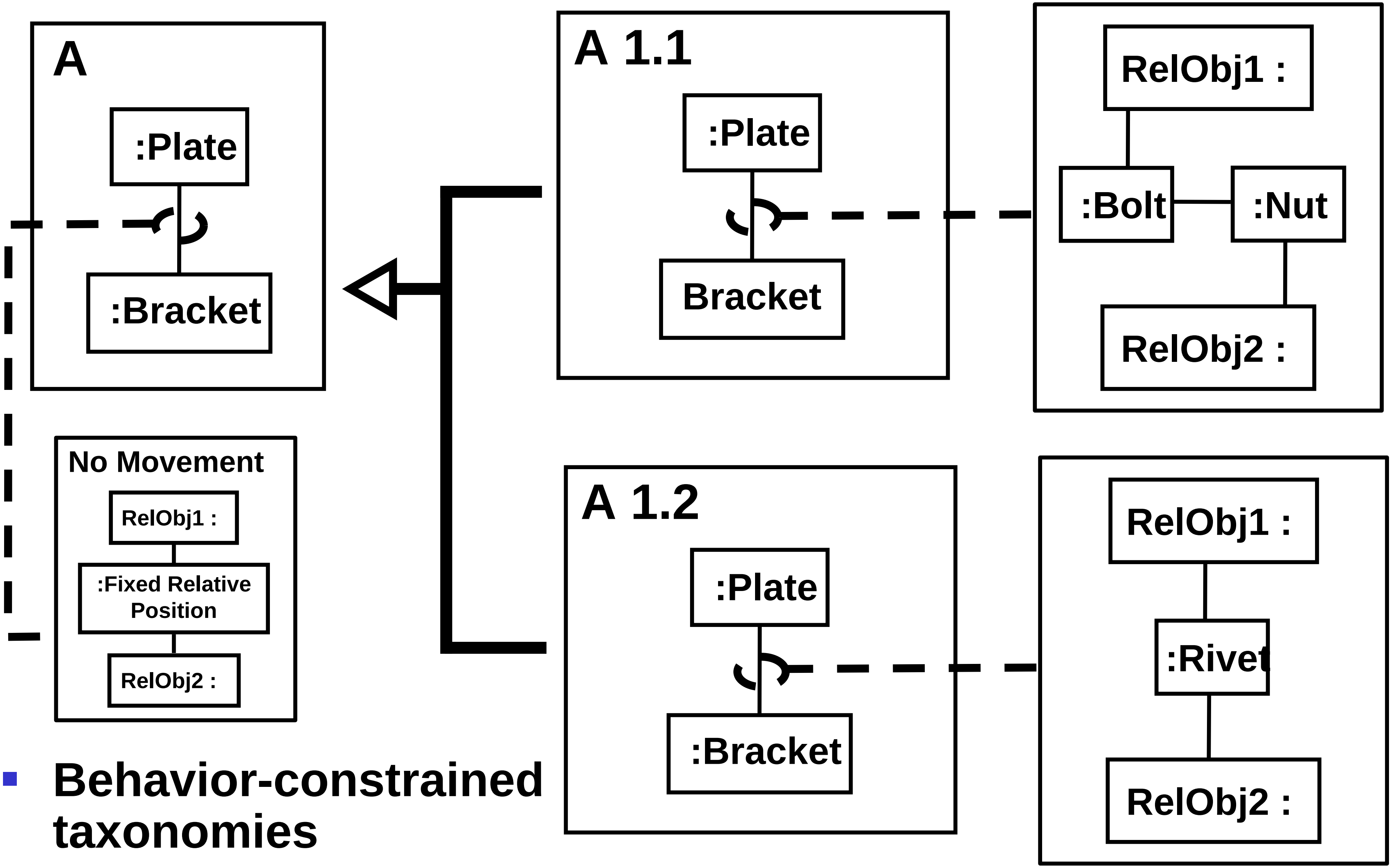
M0

Relative rotation of Axle #5467 and Axle #2345

Mar. 5, 207, 5:34-5:55pmET

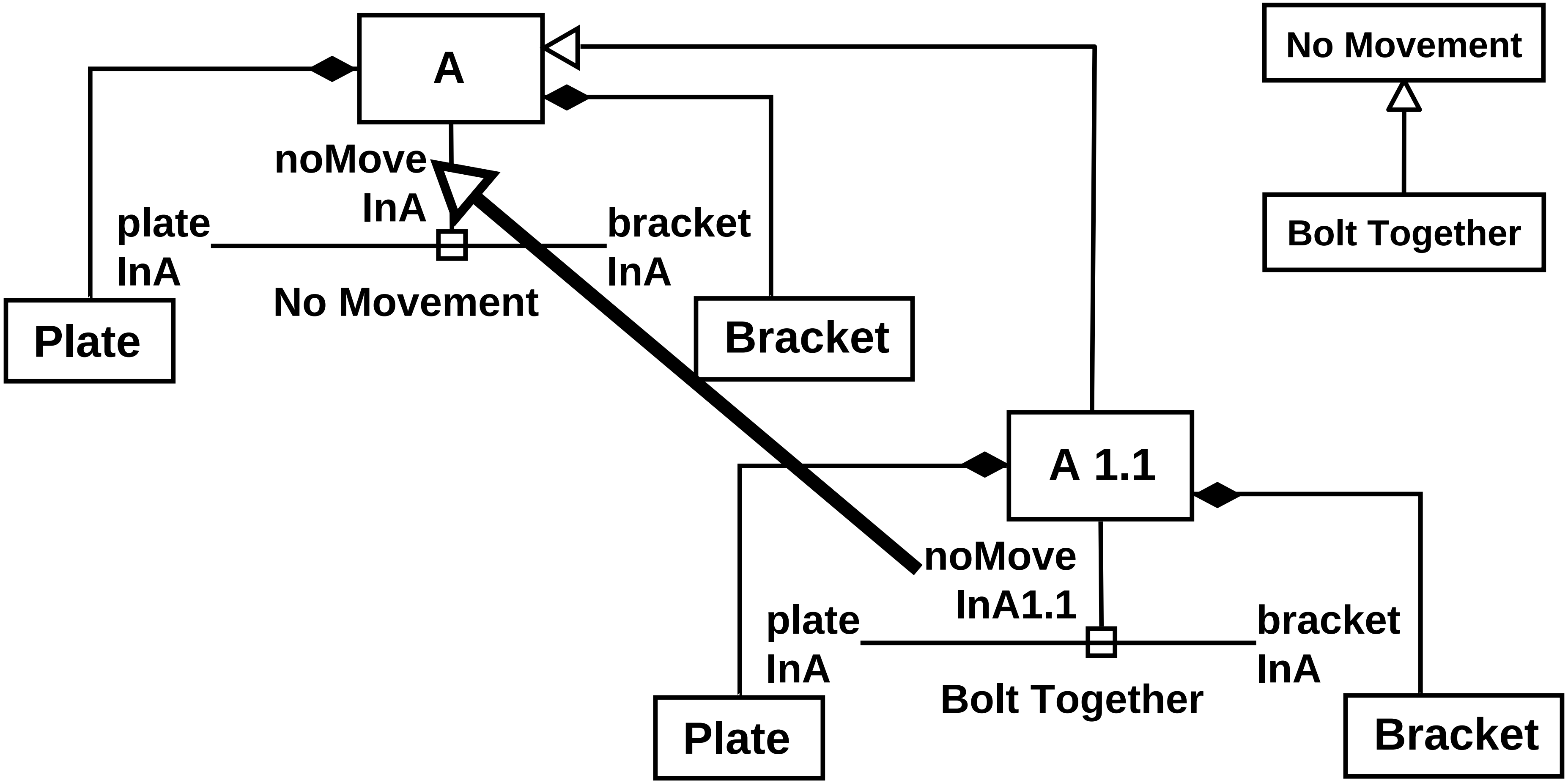
- **Behaviors are relations between things participating in them (M1), conforming at M0.**
- **Applicable to kinematic assemblies.**

Alternative Conforming Decompositions of Assembly Relation



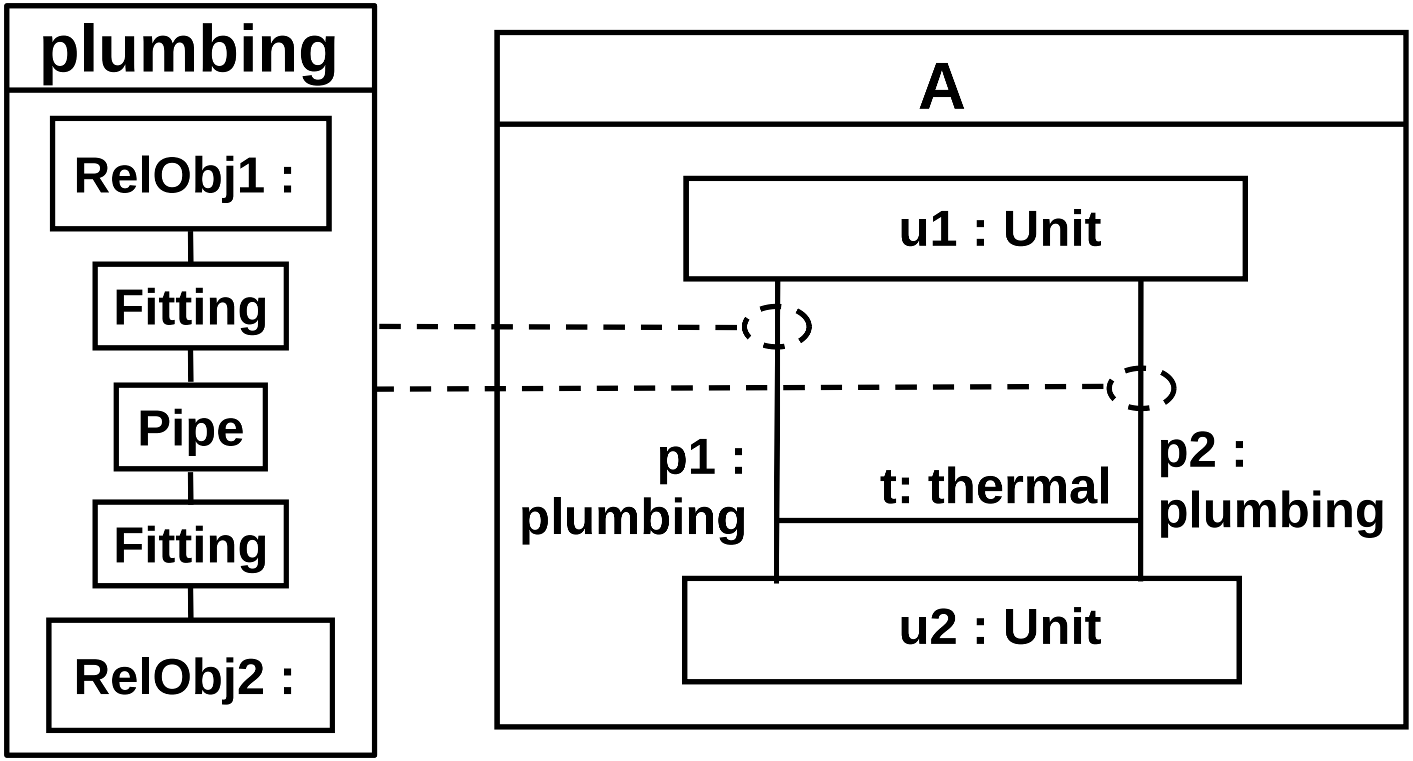
- Behavior-constrained taxonomies

Alternative Conforming Decompositions of Assembly Relation



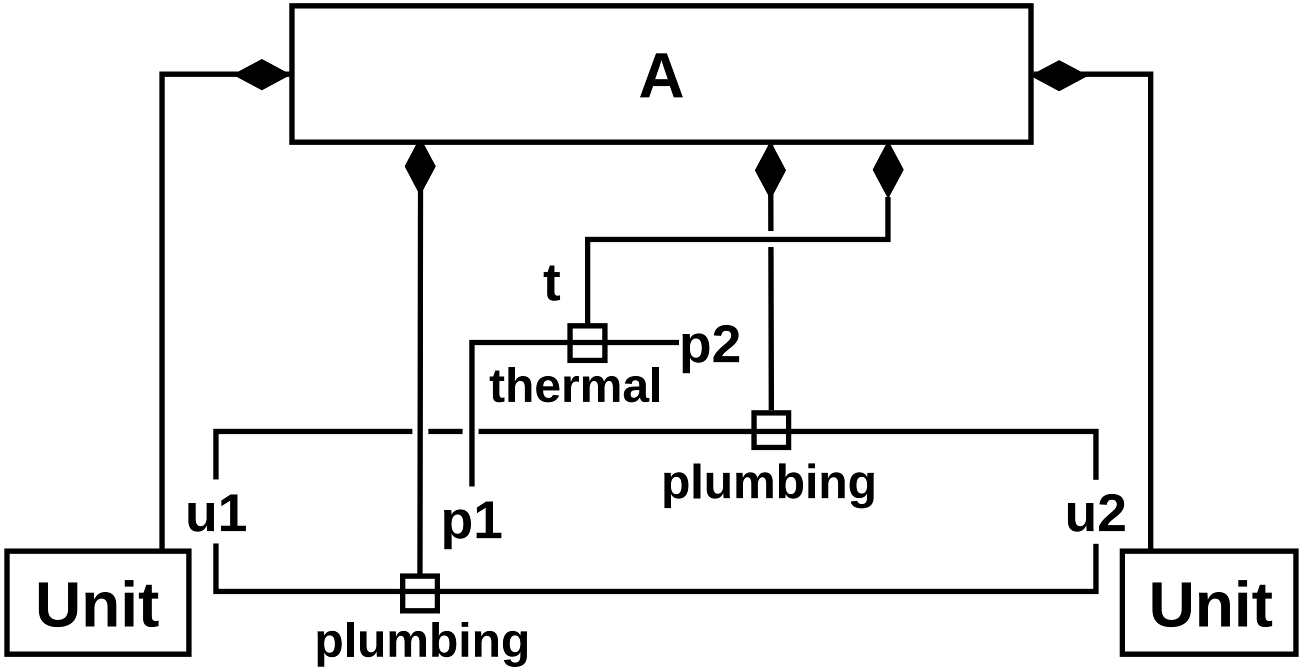
- Connector specialized by restriction.
- From No Movement to BoltTogether.

Interconnections between Interconnections



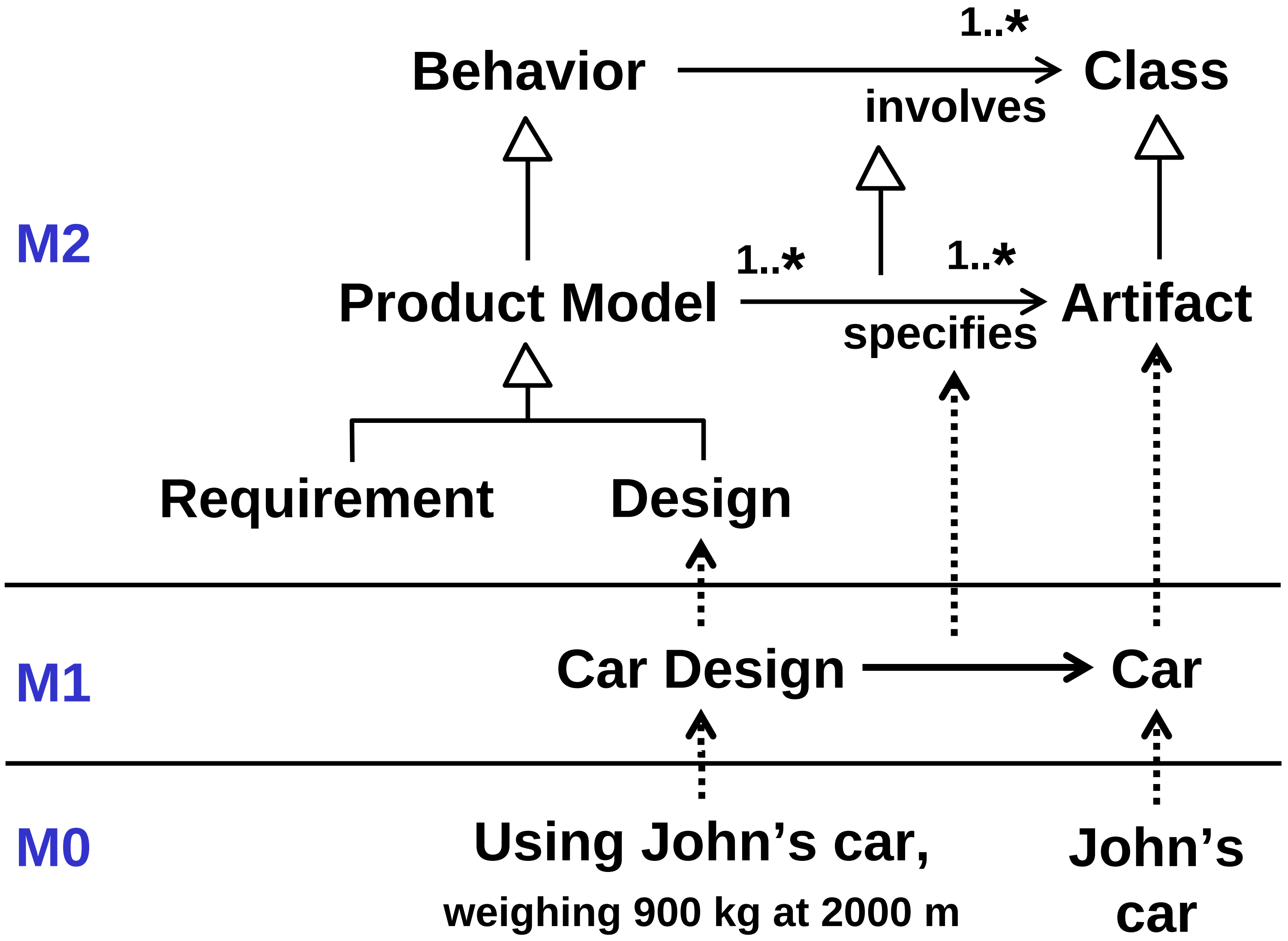
- Connectors are part-whole relations.
- Can be connected.

Interconnections between Interconnections



- Two connectors between same part-whole relations.
- Connectors (as part-whole relations) connected by thermal connector.

Product Model and Artifact



1..*

Behavior

Class

involves

M2

1..*

1..*

Product Model

specifies

Artifact

Requirement

Design

M1

Car Design

Car

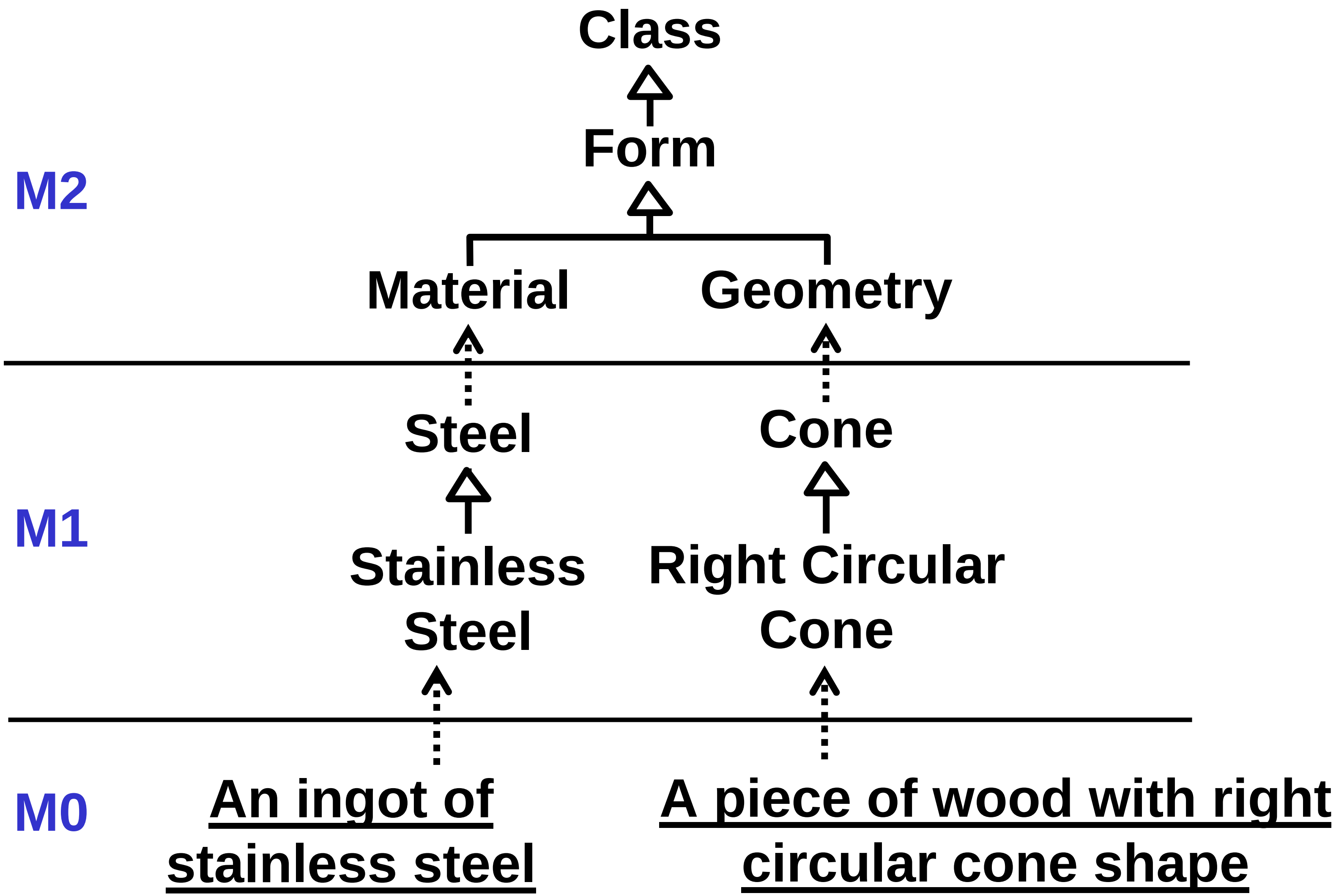
M0

Using John’s car,

weighing 900 kg at 2000 m

John’s car

Form Metamodel



M2

Class

Form

Material

Geometry

M1

Steel

Cone

Stainless Steel

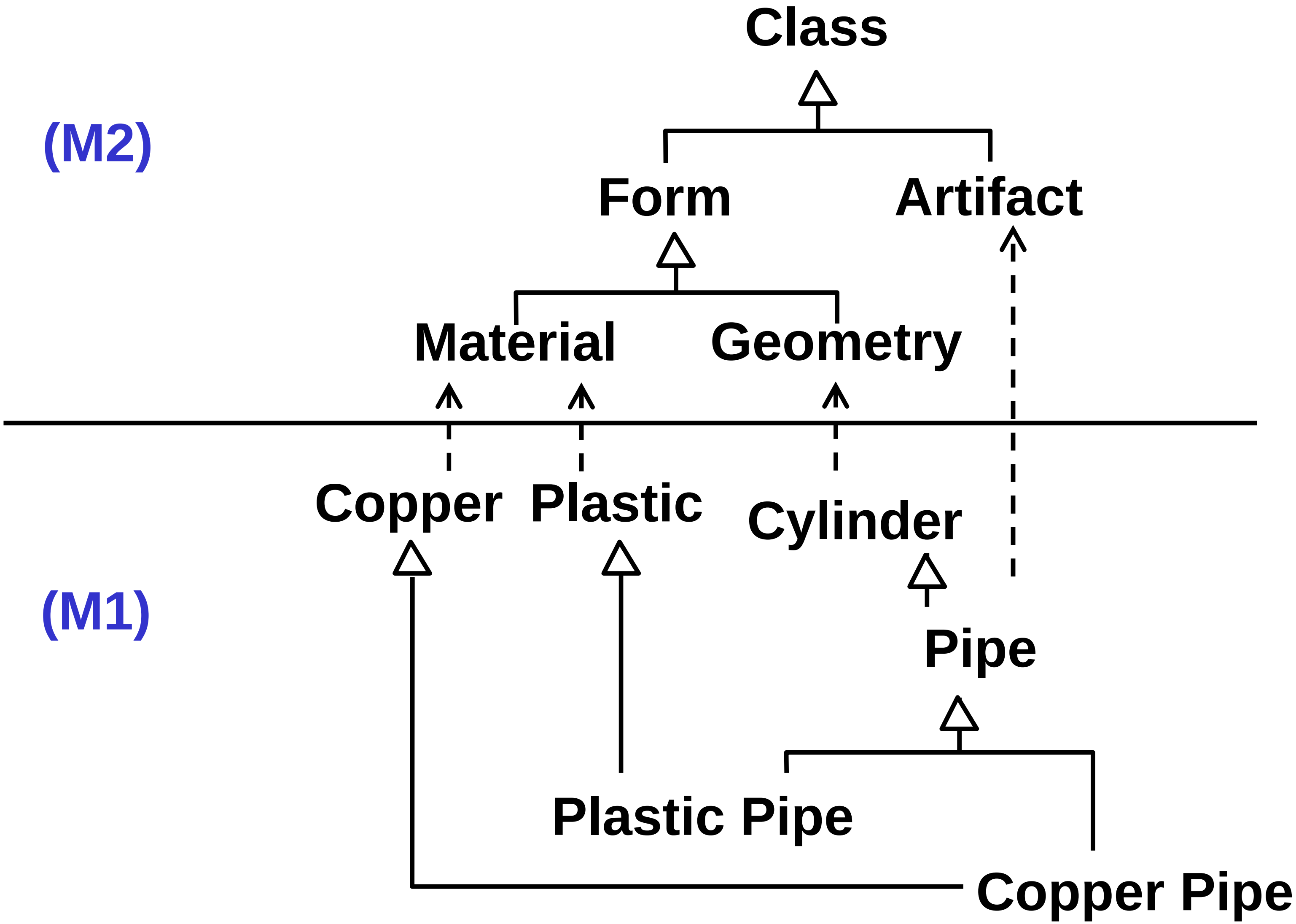
Right Circular Cone

M0

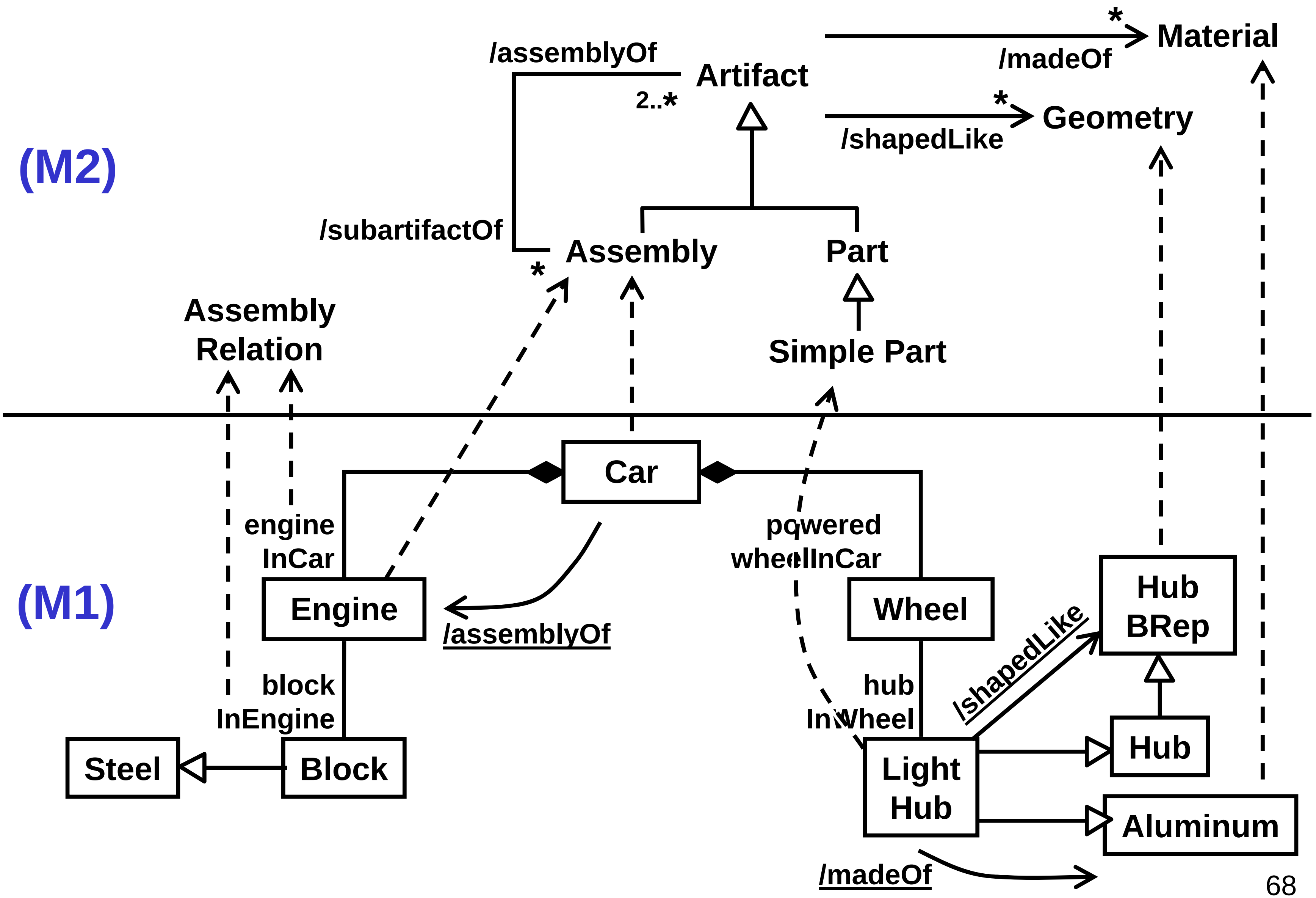
An ingot of stainless steel

A piece of wood with right circular cone shape

Form and Artifact



Material and Geometry Language



Comparison

	STEP	UML 2 / SysML	CPM 2 / OAM 2	MOKA	OPML
Total system (device / environment)	X	X+	X	X	S
Full interoperability	X	X+	X	X+	S
Enables consistency checking / reasoning	X	X+	X	X	S
Composition / assembly					
Interconnection of elements	S-	S	S-	X	S
Multiple usages of the same kind	X+	S	v1: X v2 : S-	X	S
Generalization / refinement	X+	S	X	X+	S
Relation / connector decomposition	X+	UML 2 : X SysML : S-	X+	X	S
Interconnections of interconnections	X	UML 2 : X SysML : S-	X	X	S
Behaviors as constraints on M0	X	X	X	X+	S
Behaviors as relations / connectors	X	X	X	X	S

- S : Full support
- S- : Support with exceptions
- X+ : Does not support, with exceptions
- X : Does not support

Summary

- Combine ontology and modeling languages:
  - Open world for combining partial product models and consistency checking.
  - Modeling for engineering-friendly languages.
  - Taxonomies at M2 and M1.
- Product models describe (a portion of)
  - Total systems (environment and/or device).
  - Behavior occurrences (including objects involved)
- Relations are Classes, Connectors and Behaviors are Relations.