Interactions as Composite Structure:
(Onto)Logical Interaction Modeling

Conrad Bock,
U.S. National Institute of Standards and Technology

Antoine Lonjon
MEGA

Charles Galey
Lockheed Martin

Overview

- **Motivation**
  - **Behavior, review**
  - **Interactions, requirements**
- **Interactions Solution**
  1. **Between long-lived participants**
  2. **Reusable & composable**
  3. **Outputs / inputs**
- **Summary**

Overview

- **Motivation**
  - **Behavior, review**
  - Interactions, requirements
- Interactions Solution
  1. Between long-lived participants
  2. Reusable & composable
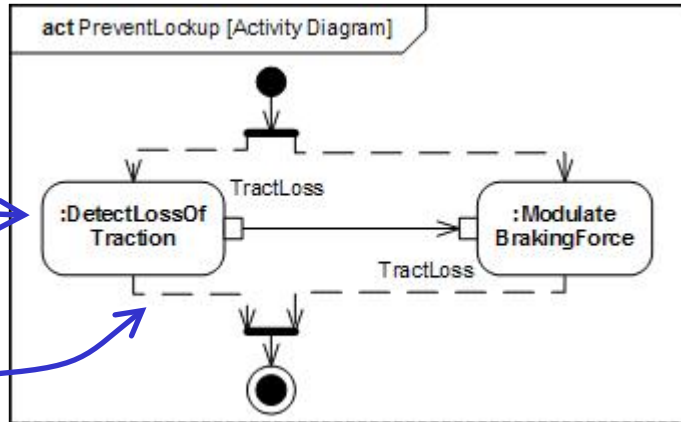 3. Outputs / inputs
- Summary

General Problem

- **UML has three behavior diagrams.**
  - **Activity, state, interaction.**
- **Very little integration or reuse between them.**
  - **Three underlying metamodels.**
  - **Three representations of temporal order.**
- **Triples the effort of learning UML and building analysis tools for it.**

General Solution

- **Treat behaviors as assemblies of other behaviors.**
  - **Like objects are assemblies of other objects.**
- **Assembly = UML internal structure**
  - **Pieces represented by properties.**
  - **Put together by connectors.**
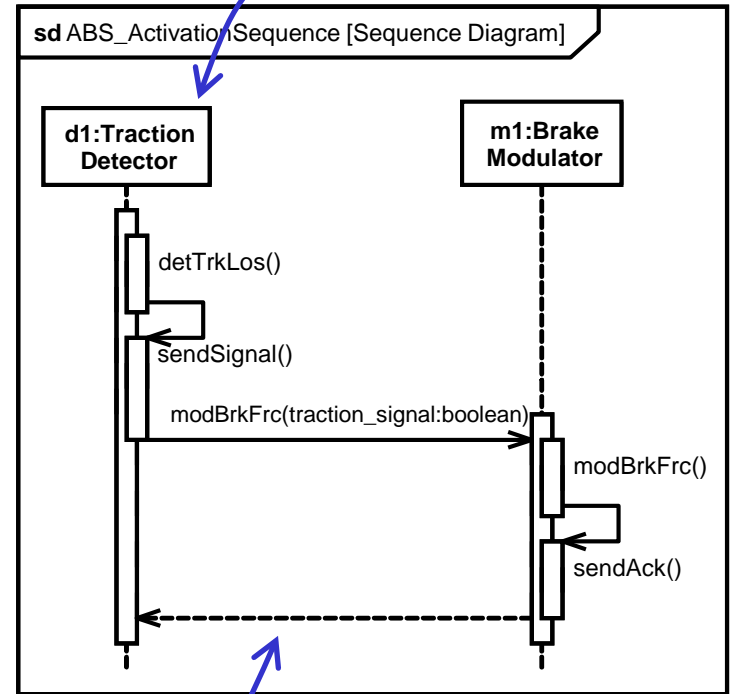- **Put all behavior diagrams on the same underlying behavior assembly model.**

Behaviors as Composite Structure

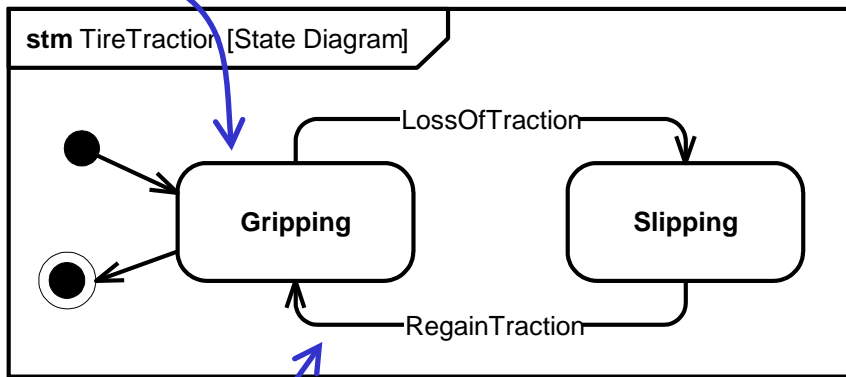**act** PreventLockup [Activity Diagram]

Property

:DetectLossOf Traction

TractLoss

:Modulate BrakingForce

TractLoss

Connector

Activity

stm TireTraction [State Diagram]

Property

Gripping

LossOfTraction

Slipping

RegainTraction

Connector

State Machine

Property

sd ABS_Activation Sequence [Sequence Diagram]

d1:Traction Detector

m1:Brake Modulator

detTrkLos()

sendSignal()

modBrkFrc(traction_signal:boolean)

modBrkFrc()

sendAck()

Connector

Interaction

6

Behavior: What's Being Modeled?

Real, Simulated, or Desired Things Being Modeled (M0)

Not instance specs.

Focus
3/15/09 10-11amET :

TakePicture
3/15/09 10-12pmET :

Shoot
3/15/0911-12pmET :

- ▪ **"Things" that occur in time**
  - – **Eg, taking a picture, focusing, etc.**
  - – **Not "behaviors", "actions", etc.**

Behavior: What's in Common?

Standard Model Library (M1)

happens Before

Behavior Occurrence

happens During^{-1}

Things Being Modeled (M0)

TakePicture
3/15/09 10-12pmET :

happens During^{-1}

Focus
3/15/09 10-11amET :

happensBefore

happens During^{-1}

**Shoot**
3/15/0911-12pmET :
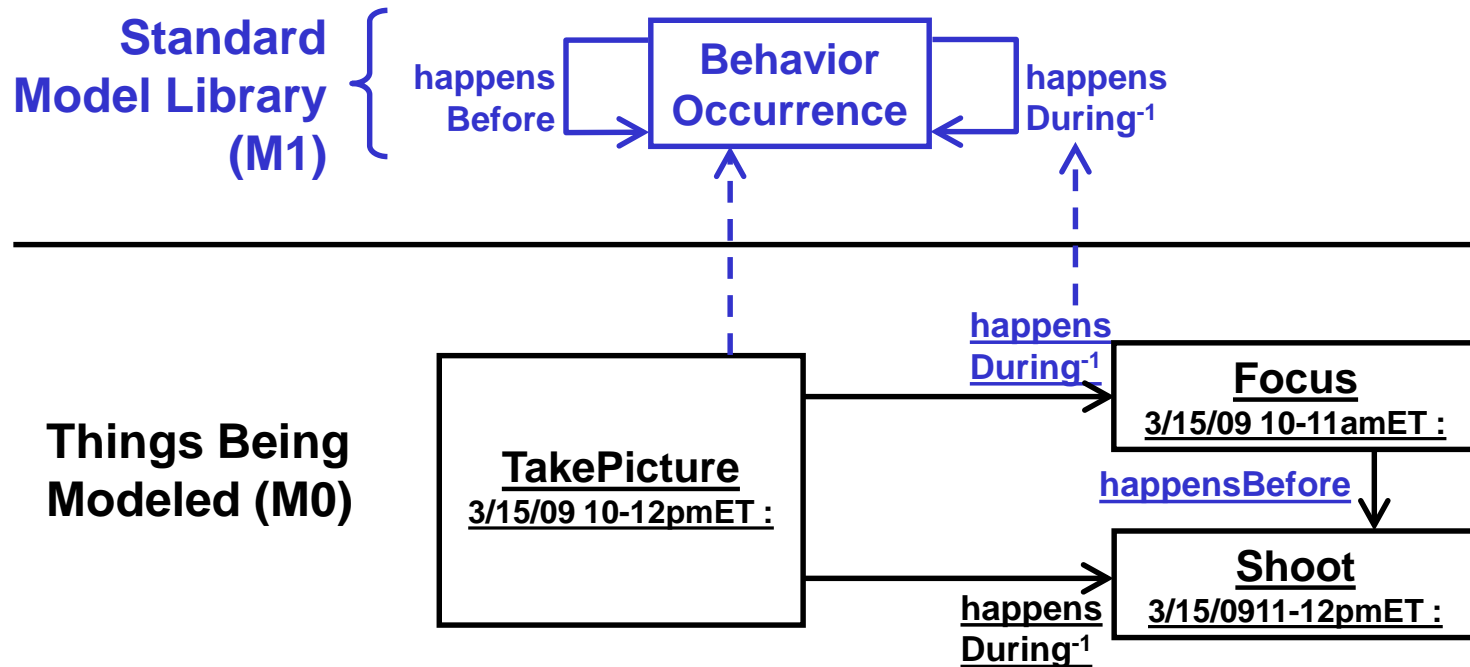
- **They happen before or during each other.**
  - **Construct M1 library for this.**
  - **Use it to classify things being modeled.**

Behavior: Use Library

Standard Model Library (M1)

Behavior Occurrence

happens Before

happens During⁻¹

{subsets}

User Model (M1)

TakePicture

step1 : Focus

: HappensBefore

step2 : Shoot

Things Being Modeled (M0)

TakePicture
3/15/09 10-12pmET :

step1

Focus
3/15/09 10-11amET :

happensBefore

step2

Shoot
3/15/0911-12pmET :
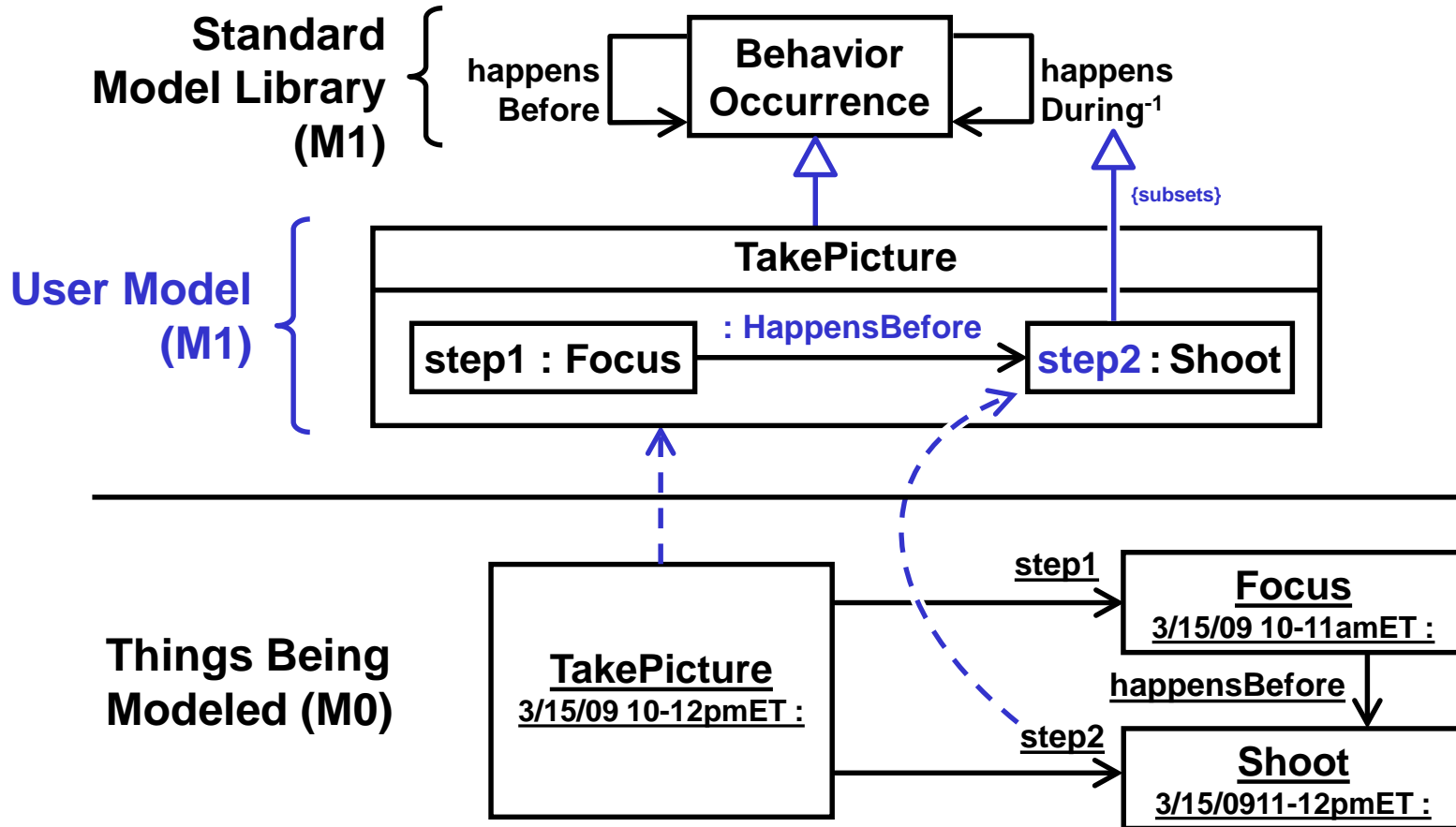
- **Specialize library classes and subset/redefine library properties.**

10

Behavior: Too repetitive at M1?
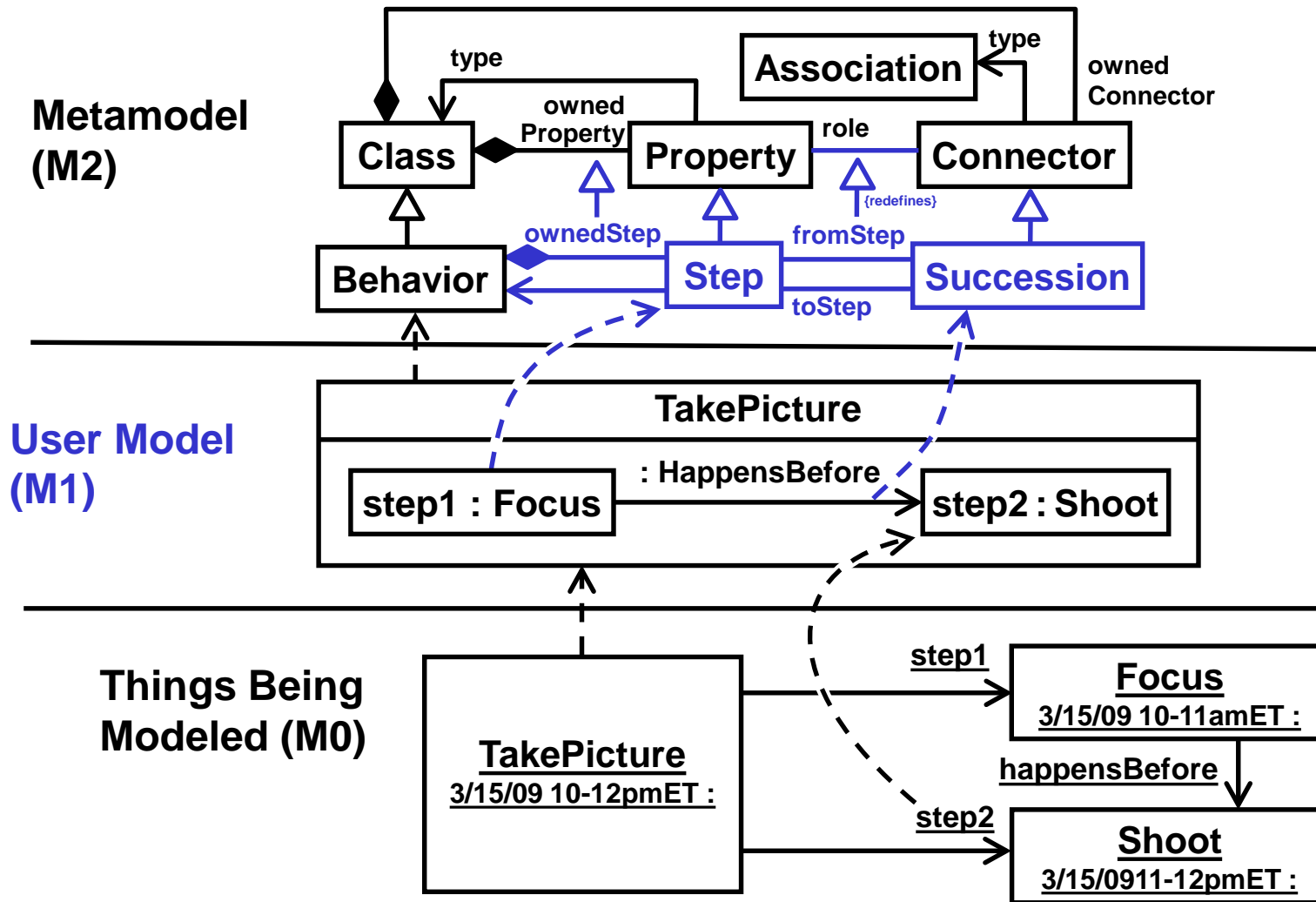
- **Capture M1 patterns in M2 elements.**
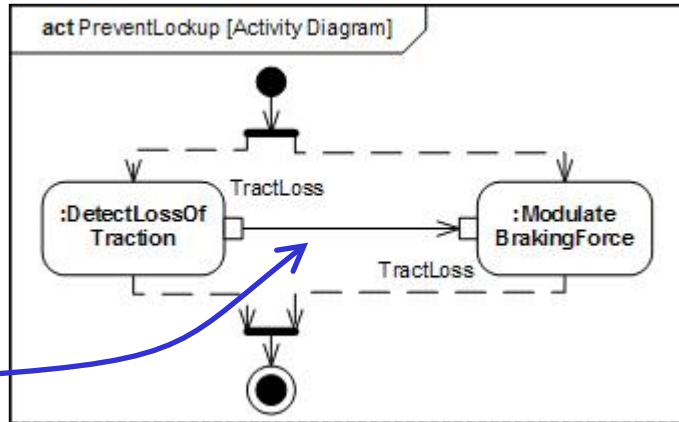 - **Tools apply patterns automatically.**

11

Overview

- **Motivation**
  - **Behavior, review**
  - **Interactions, requirements**
- **Interactions Solution**
  1. **Between long-lived participants**
  2. **Reusable & composable**
  3. **Outputs / inputs**
- **Summary**

Interactions Problem

- **UML/SysML have three ways things can "flow":**
  - Activities have **object flows** between actions.
  - Interactions have **messages** between lifelines.
  - SysML blocks have **item flows** between parts.
    - Via flow properties on each end.

- **Very little integration or reuse.**
  - Three underlying metamodels/profiles.
  - Three representations of things flowing.

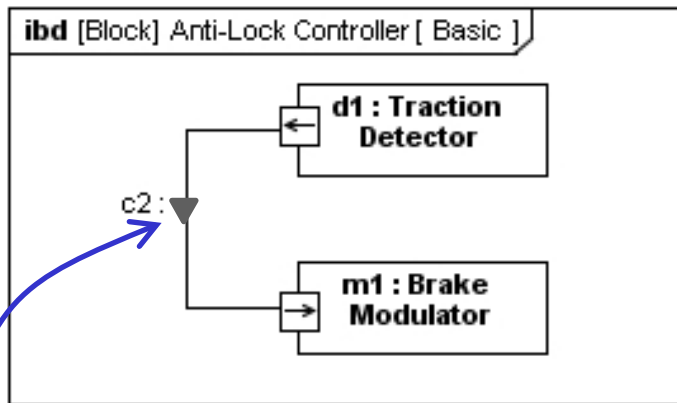- **Triples the effort of learning UML/SysML and building analysis tools for them.**
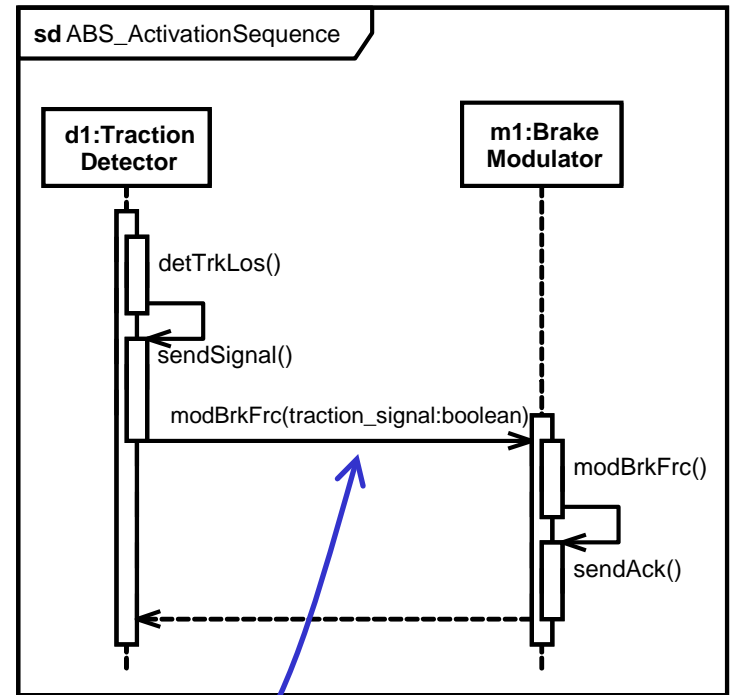
14

Interactions Problem

Activity

act PreventLockup [Activity Diagram]

:DetectLossOf Traction

TractLoss

:Modulate BrakingForce

TractLoss

Object Flow

ibd [Block] Anti-Lock Controller [Basic]

d1 : Traction Detector

c2 :

m1 : Brake Modulator

SysML Internal Block Diagram

Item Flow

sd ABS_ActivationSequence

d1:Traction Detector

m1:Brake Modulator

detTrkLos()

sendSignal()

modBrkFrc(traction_signal:boolean)

modBrkFrc()

sendAck()

Interaction

Message

15

Interactions Requirements

1. **Between things that outlive interactions.**
   - **Objects have many interactions over time.**
   - **Not just between steps in an activity.**
2. **Interactions are reusable and composable.**
   - **The same kind of interaction might be used in many user models and**
   - **contain many other interactions ordered in time.**
3. **Interacting objects have "mailboxes".**
   - **Things being exchanged leave and arrive at specified places in the interacting objects.**
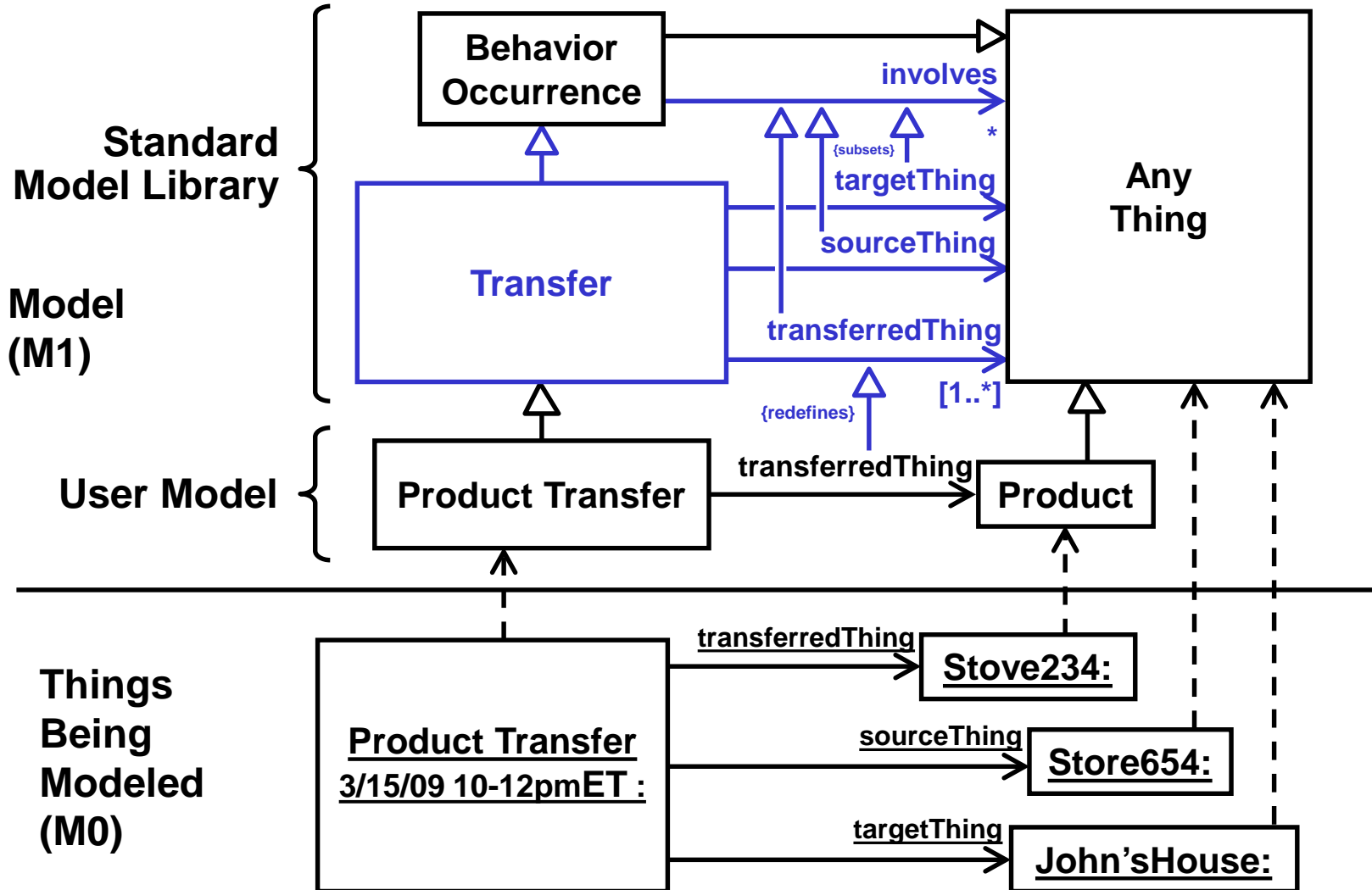   - **Aka, output/inputs.**

Overview

- **Motivation**
  - **Behavior, review**
  - **Interactions, requirements**
- **Interactions Solution**
  1. **Between long-lived participants**
  2. **Reusable & composable**
  3. **Outputs / inputs**
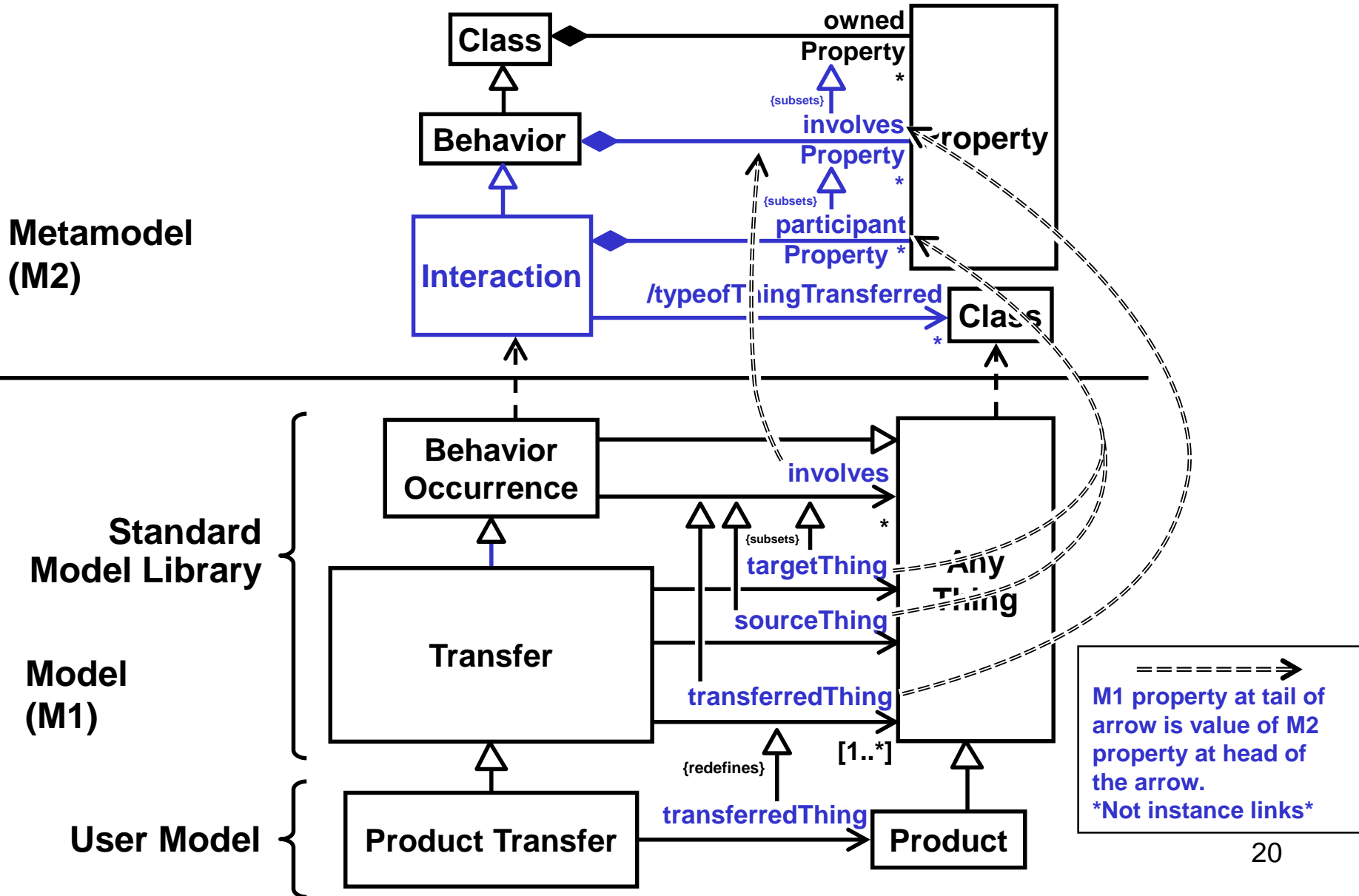- **Summary**

Interactions Solution (Part 1)
(between things that outlive interactions)

- **Flows happen in time.**

  - **They are behaviors.**

- **Start when an entity begins flowing.**

  - **Leaves output pin of an action.**

    … execution on a lifeline.

    … SysML out flow property.

- **End when the entity stops flowing.**

  - **Arrives at input pin of an action.**

    … execution on a lifeline.

    … SysML in flow property.

18

Transfers (M1)

Interactions (M2)

Overview

- **Motivation**
  - **Behavior, review**
  - **Interactions, requirements**
- **Interactions Solution**
  1. **Between long-lived participants**
  2. **Reusable & composable**
  3. **Outputs / inputs**
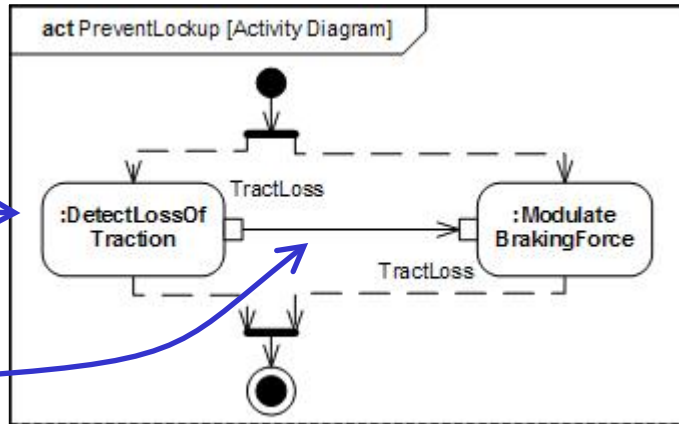- **Summary**

Interactions Solution (Part 2a)
(interactions are reusable)

- **Interactions happen across links between objects.**

- **Links specified by connectors …**

- **… that are typed by associations.**

- **Interactions must be associations reused by connectors.**

Transfers Along Connectors?
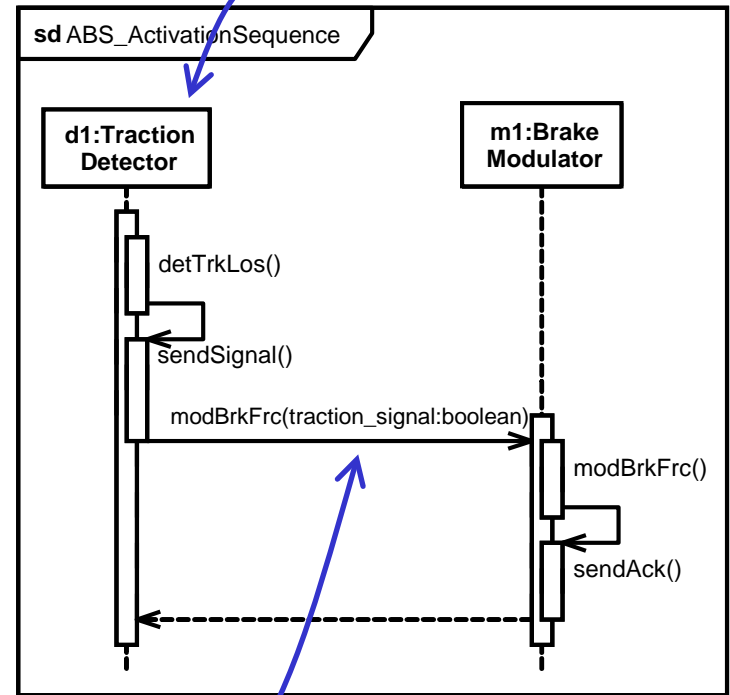

Property

Property

Connector

Activity

Property

Connector
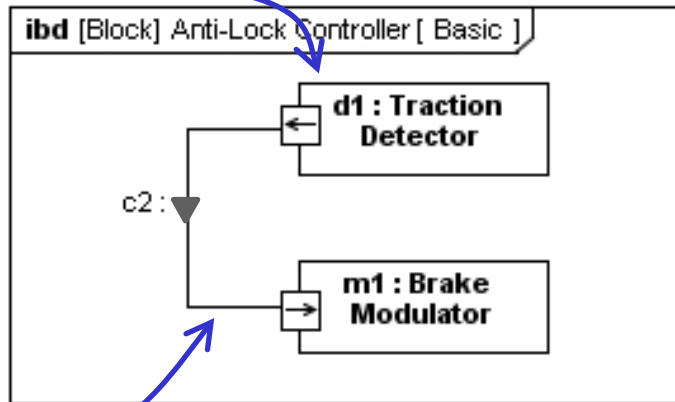
SysML Internal Block Diagram

Interaction

Connector

- **Connectors are typed by associations.**
  - **But transfers are behaviors.**

Interaction = Behavior & Association

- **Associations and behaviors both have objects that participate in them.**
  - **Associations link their participants.**
  - **Behaviors involve their objects.**
    - **Interactions have lifelines.**
    - **Activities have object nodes, partitions, etc.**
    - **Behaviors have parameters.**

- **Interactions are behaviors that are also associations between their participants.**

Two Kinds of Association Properties

Model (M1)

Camera — conCam

Controls
linkedCon : Controller
linkedCam: Camera

Controller — camCon

Things Being Modeled (M0)

Camera 34 : — linkedCam

Link 251 : — linkedCon

Cntrl 12 :

conCam

camCon
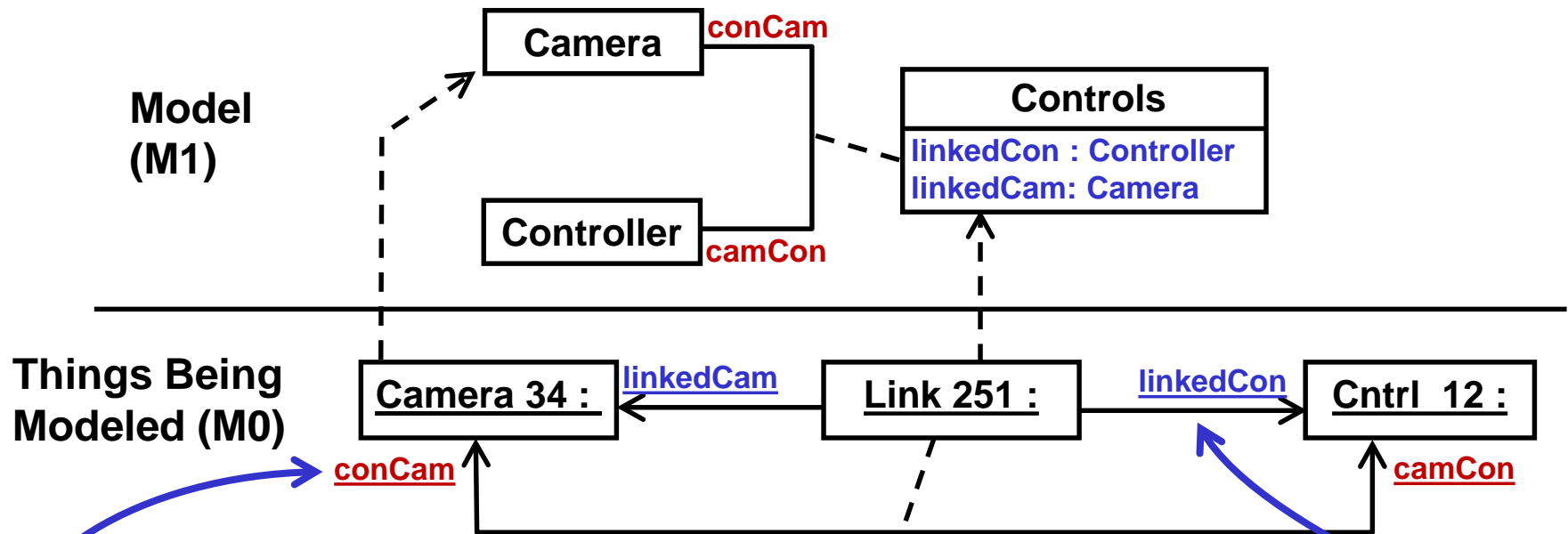
- **Two kinds** of properties, for navigation between:
  - One end object to another (**end properties**).
  - Links and end objects (**link properties**).

25

Link Properties (M1)

Standard Model Library

Model (M1)

User Model

Things Being Modeled (M0)

Anything

linkedThing {non-unique} [2..*]

Link

{subsets}

Camera   conCam

Controls

linkedCon : Controller
linkedCam: Camera

Controller   camCon

Camera 34 :   linkedCam   Link 251 :   linkedCon   Cntrl 12 :

conCam   camCon

26
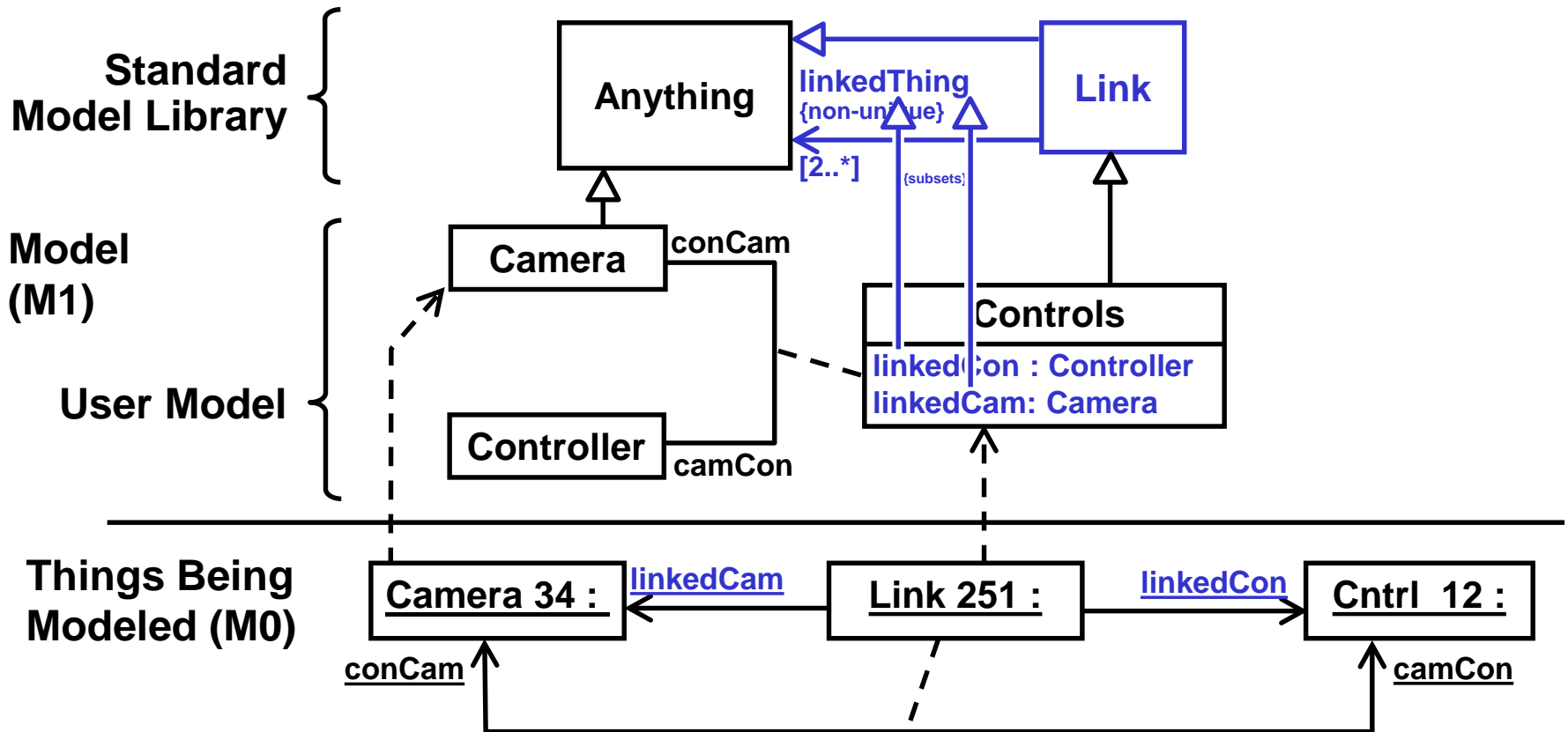
Association Participants (M2)
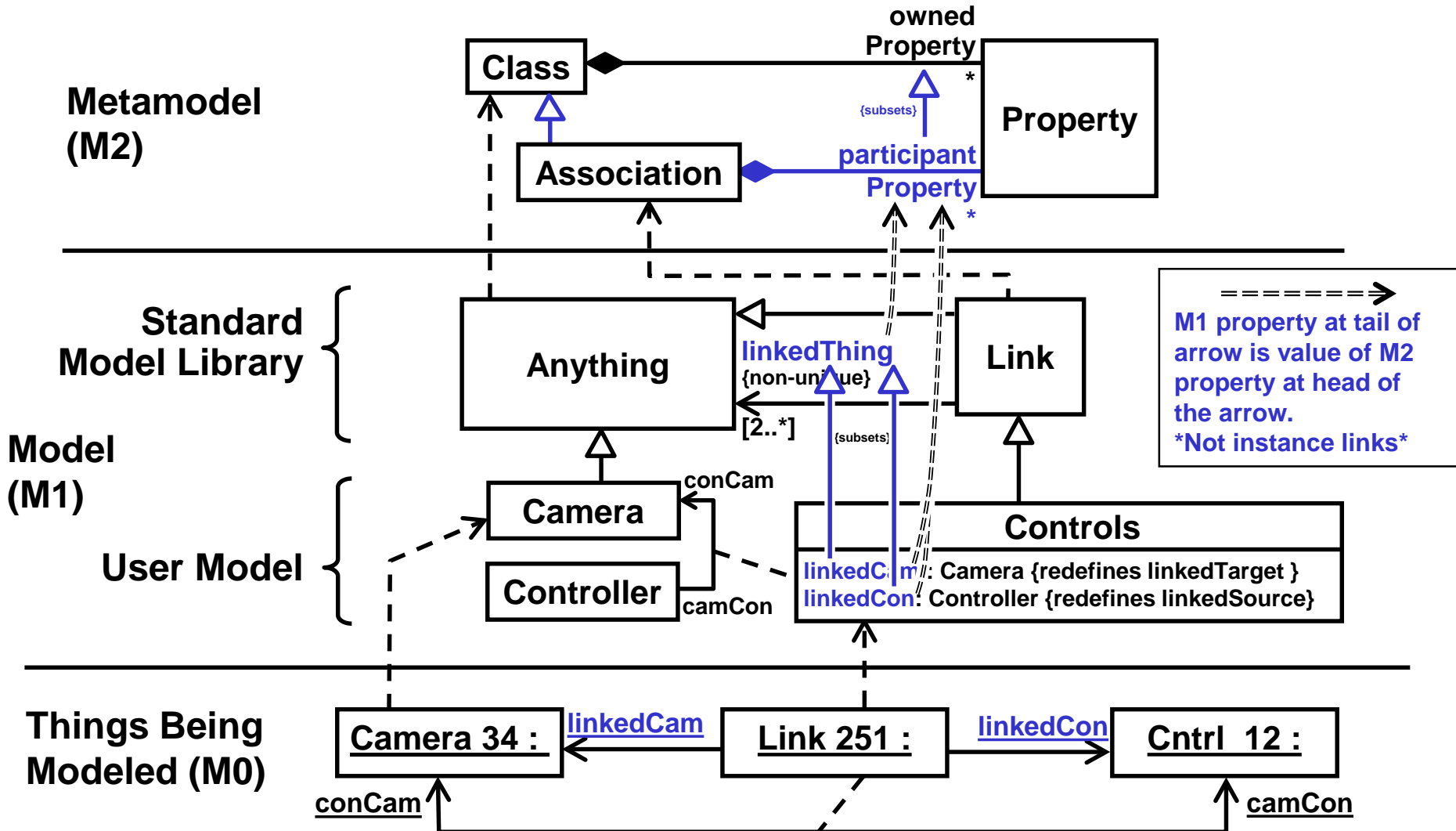
Transfers as Links (M1)

28
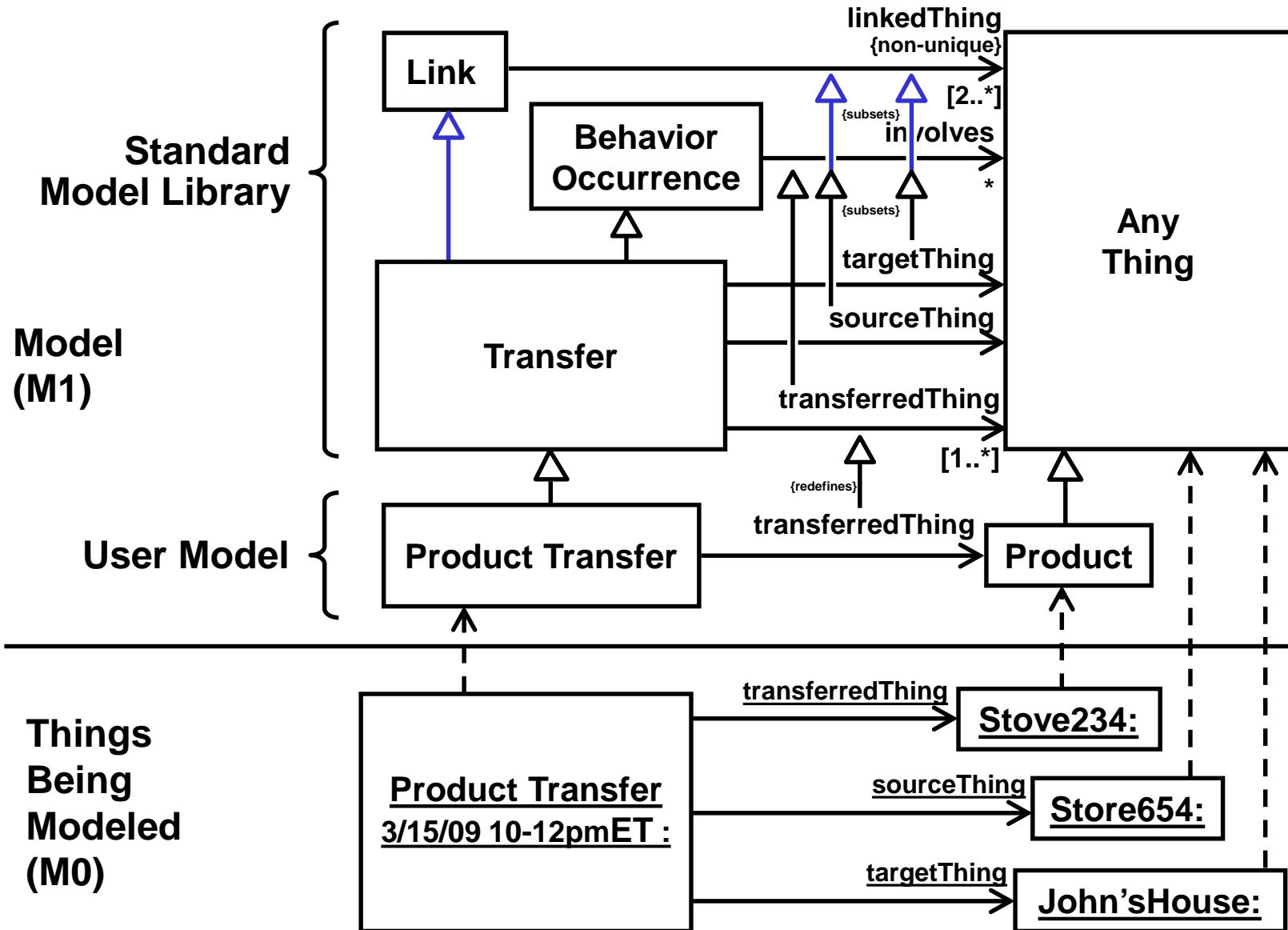
Interaction Participants (M2)

29
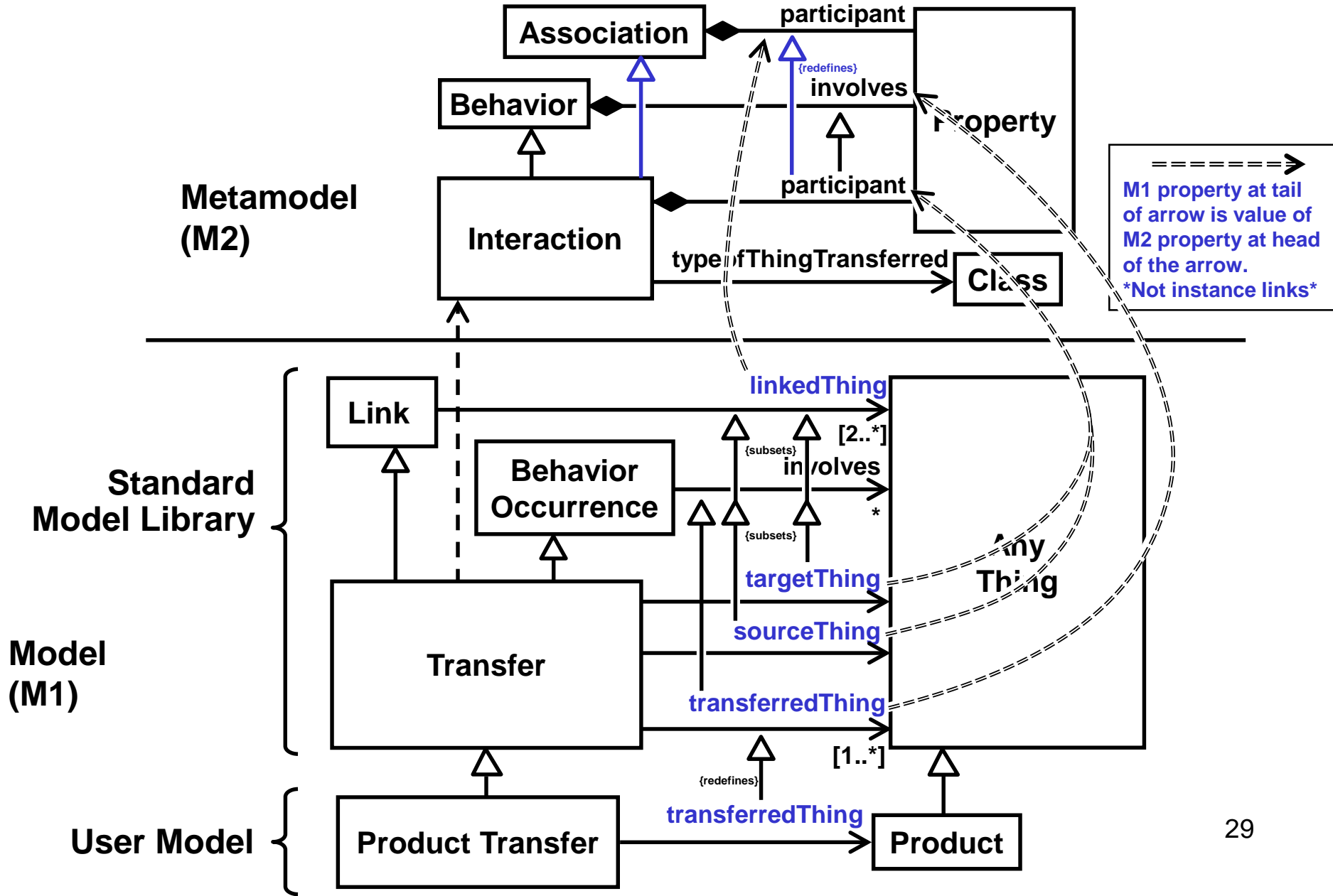
Connectors Reusing Interactions
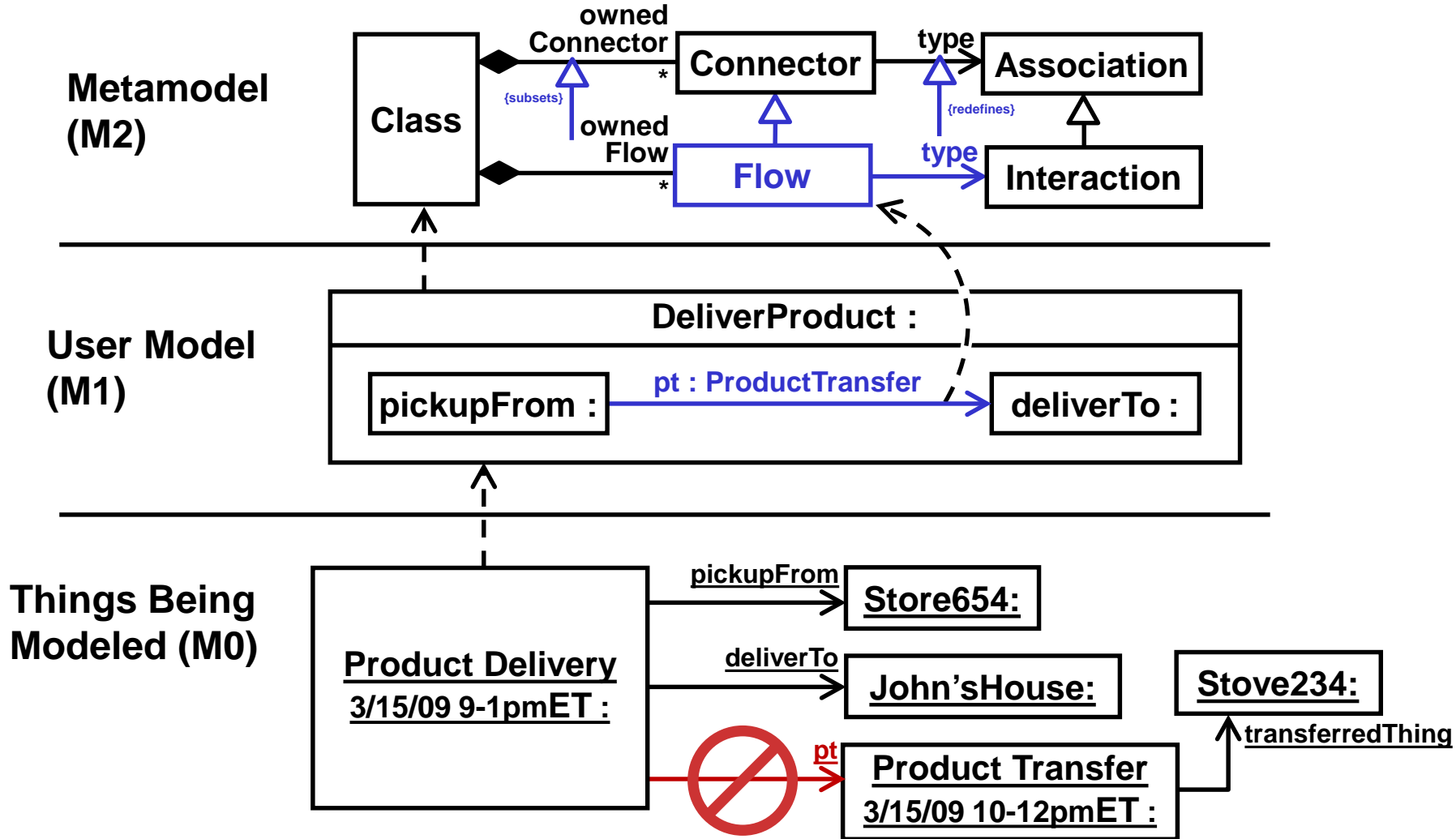
Connectors as Properties
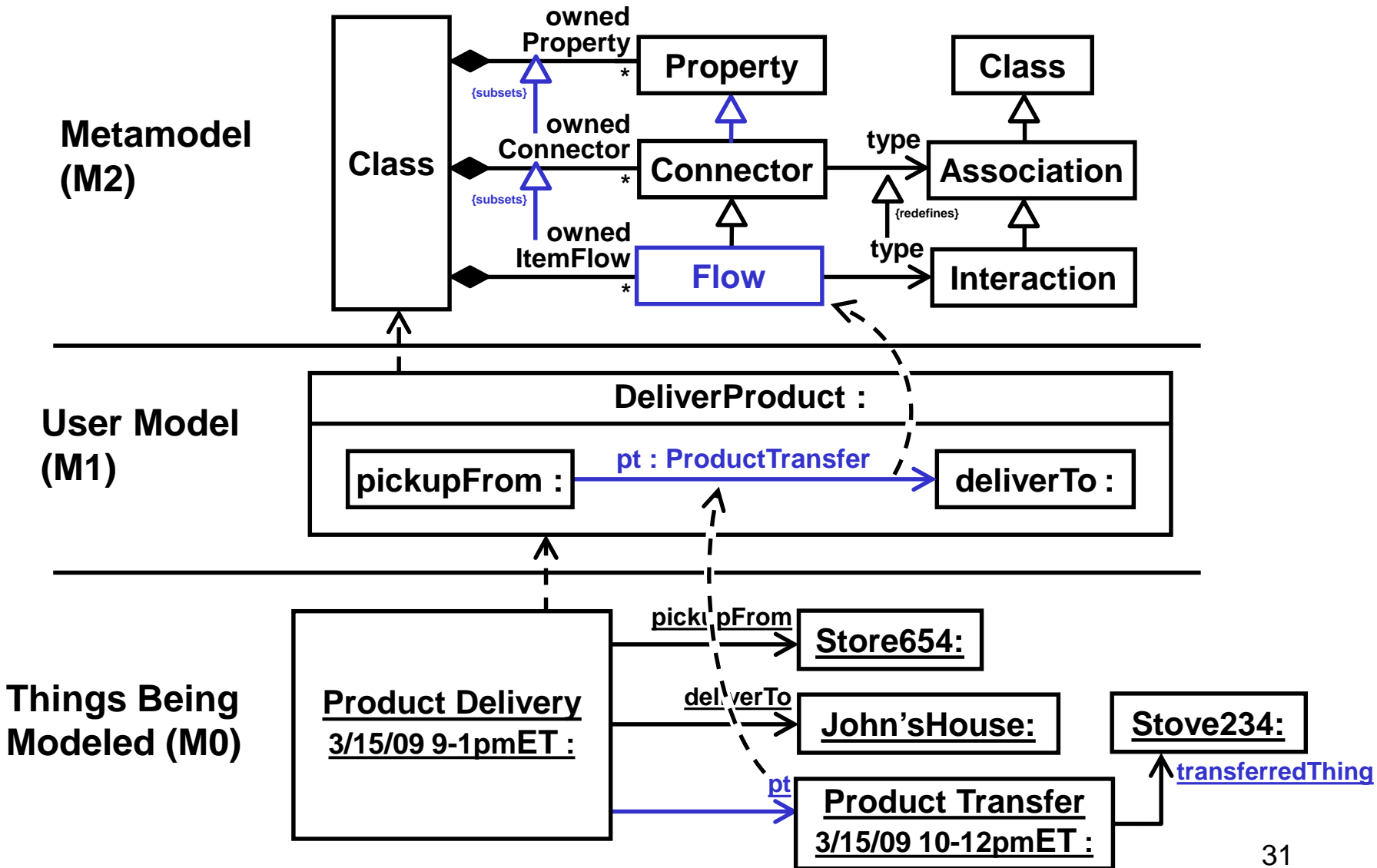
Metamodel (M2)

owned Property

Class

{subsets}

Property

Class

owned Connector

{subsets}

Connector

type

Association

owned ItemFlow

Flow

type

{redefines}

Interaction

User Model (M1)

DeliverProduct :

pickupFrom :

pt : ProductTransfer

deliverTo :

Things Being Modeled (M0)

Product Delivery 3/15/09 9-1pmET :

pickupFrom

Store654:

deliverTo

John'sHouse:

pt

Product Transfer 3/15/09 10-12pmET :

Stove234:

transferredThing

31

Overview

- **Motivation**
  - **Behavior, review**
  - **Interactions, requirements**
- **Interactions Solution**
  1. **Between long-lived participants**
  2. **Reusable & composable**
  3. **Outputs / inputs**
- **Summary**

Interactions Solution (Part 2b)
(interactions are composable)

- **Multiple flows in one interaction.**
  - **Interactions with 1 flow = transfer.**
- **Some flows happen before others**
  - **UML interactions order messages and interaction uses.**
- **Requires successions between flows.**
  - **Sucessions = connectors typed by HappensBefore from standard M1 library.**

Flow Steps

Metamodel (M2)

Class — type → Class

Class — owned Property → Property ◀ Connector — type → Association

Behavior ◆ — ownedStep → Step ← earlierStep — Succession

Behavior ← type — Step ← laterStep — Succession

{redefines}

role

Model (M1)

CapturePicture

fcntl : Flight Control ◀ Confirmation fdb : Flight Database

Command ▶ sc: Spacecraft Picture ▶

: happens Before

: happensBefore

User Model

Interaction

Flow

happens Before

Behavior Occurrence

Transfer

Standard Model Library

IBD UML Interaction Diagrams

Sequence diagram

ibd CapturePicture

gcntl : Ground Control — Command ▶ — **sat : Satellite** — Picture ▶ — **gdb : Ground Database**

◀ Confirmation

Communication diagram

ibd CapturePicture

gcntl : Ground Control — 1. Command → **sat : Satellite**

3. Confirmation ↑

2. Picture ↓

gdb : Ground Database

35
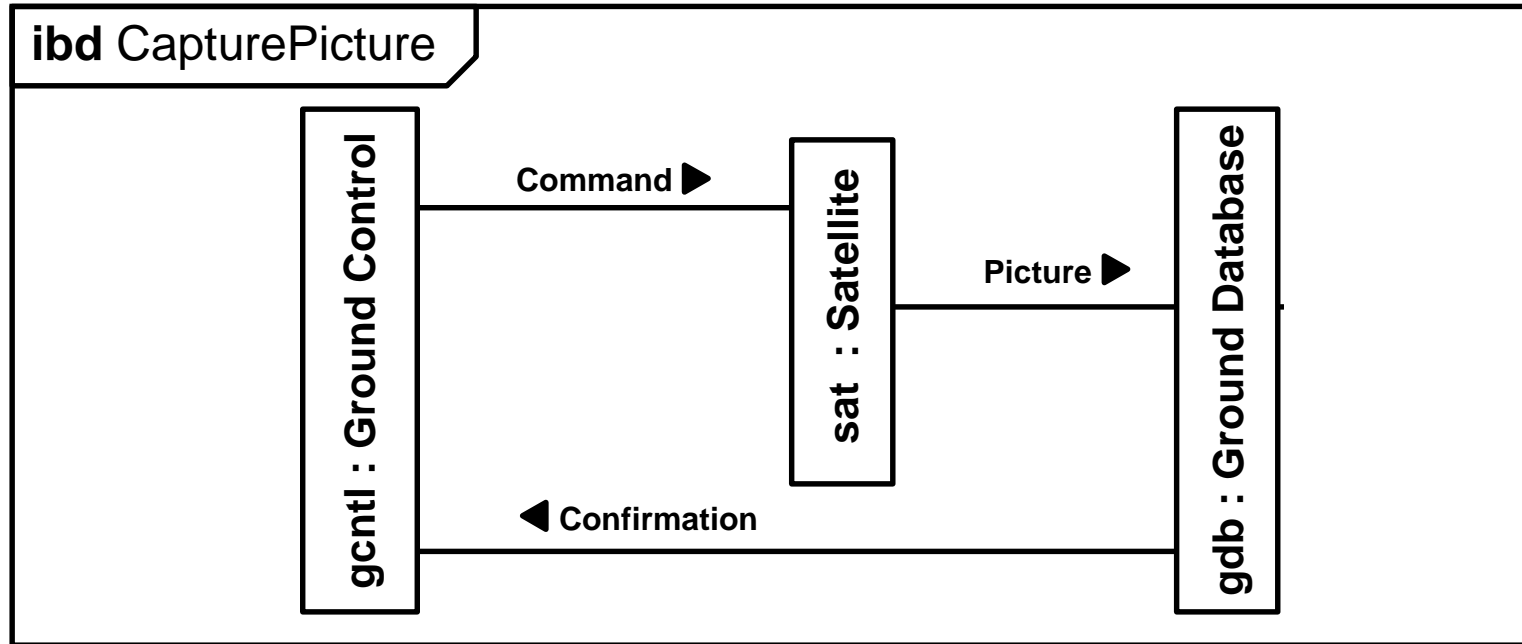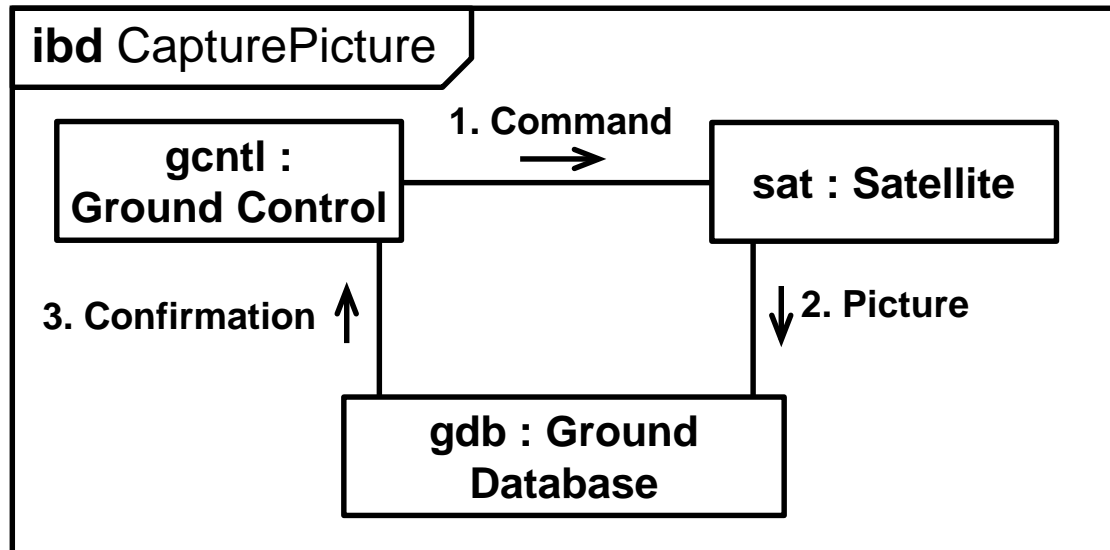
Connectors Reusing Interactions

Model (M1)

FlightOperation : Interaction

cntl : GroundControl — fcntl — ◇ — sc — sc : Satellite

◇ — fdb — db : TelemetryDB

: CapturePicture

fcntl : Flight Control {subsets linkedThing}

◀ Confirmation

fdb : Flight Database {subsets linkedThing}

sc: Spacecraft {subsets linkedThing}

: happens Before

Command ▶

Picture ▶

: happensBefore

36
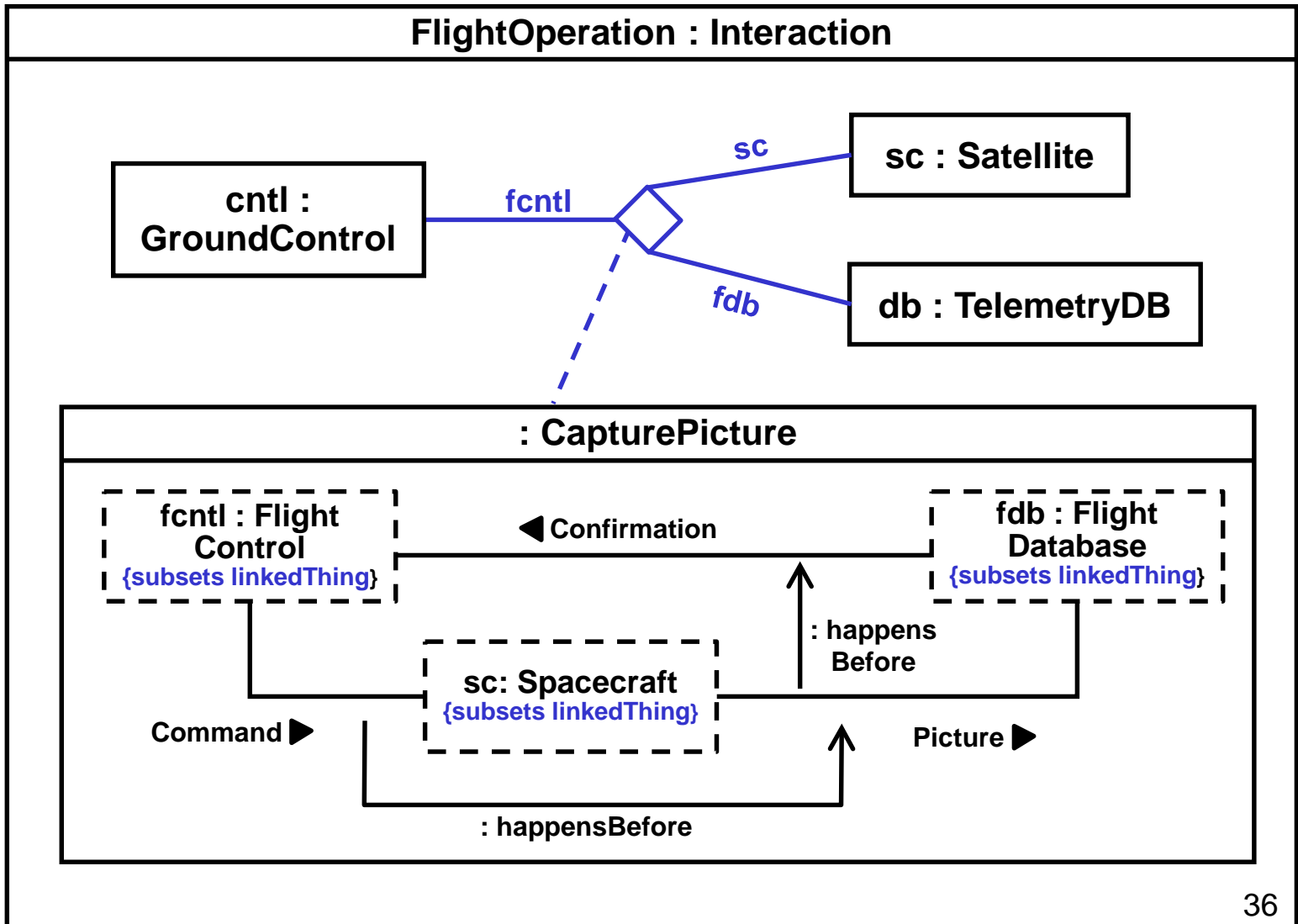
IBD UML Collaborations

Model (M1)
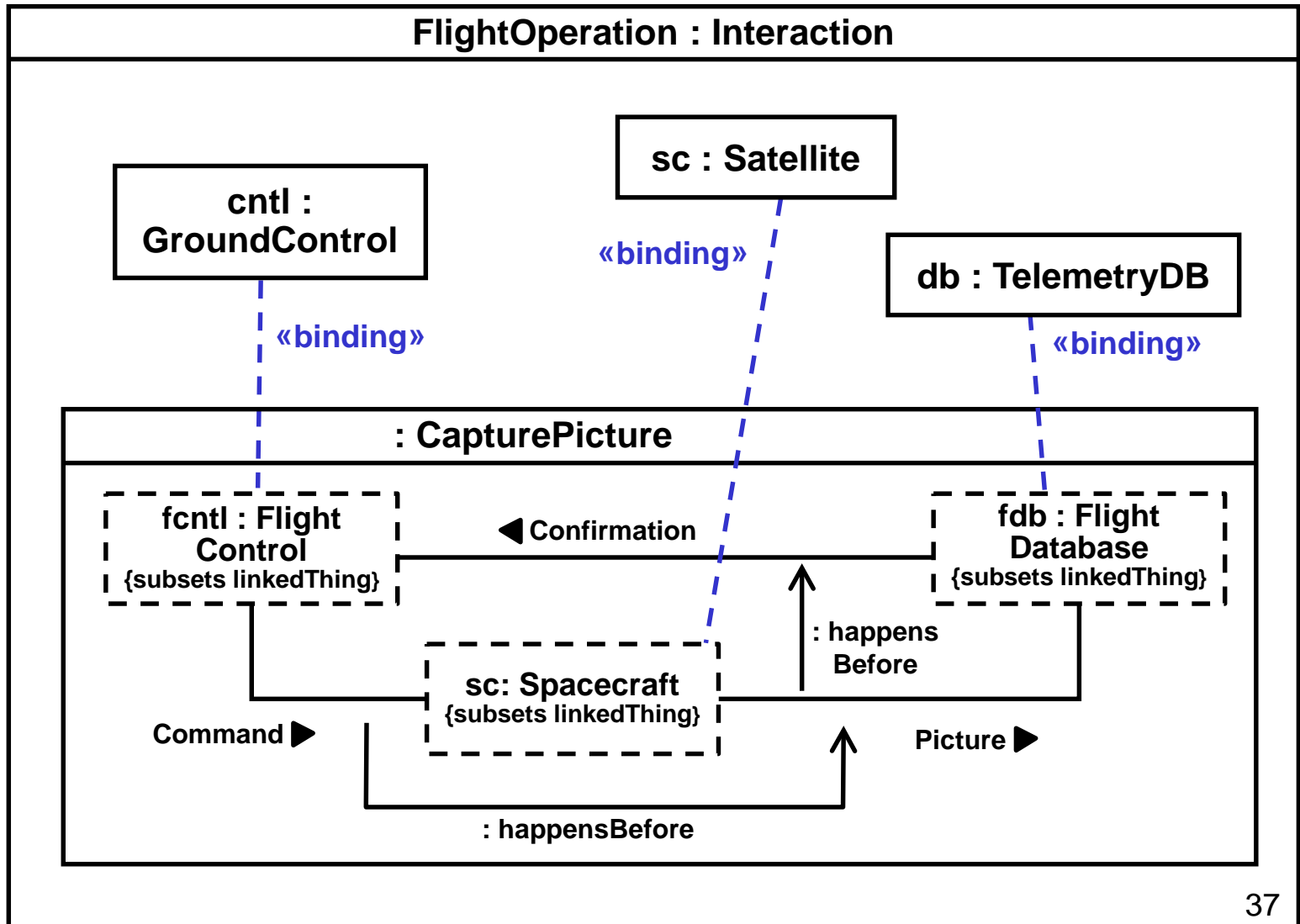
FlightOperation : Interaction

sc : Satellite

cntl : GroundControl

db : TelemetryDB

«binding»

«binding»

«binding»

: CapturePicture

fcntl : Flight Control {subsets linkedThing}

◄ Confirmation

fdb : Flight Database {subsets linkedThing}

: happens Before

sc: Spacecraft {subsets linkedThing}

Command ▶

Picture ▶

: happensBefore

Overview

- **Motivation**
  - **Behavior, review**
  - **Interactions, requirements**
- **Interactions Solution**
  1. **Between long-lived participants**
  2. **Reusable & composable**
  3. **Outputs / inputs**
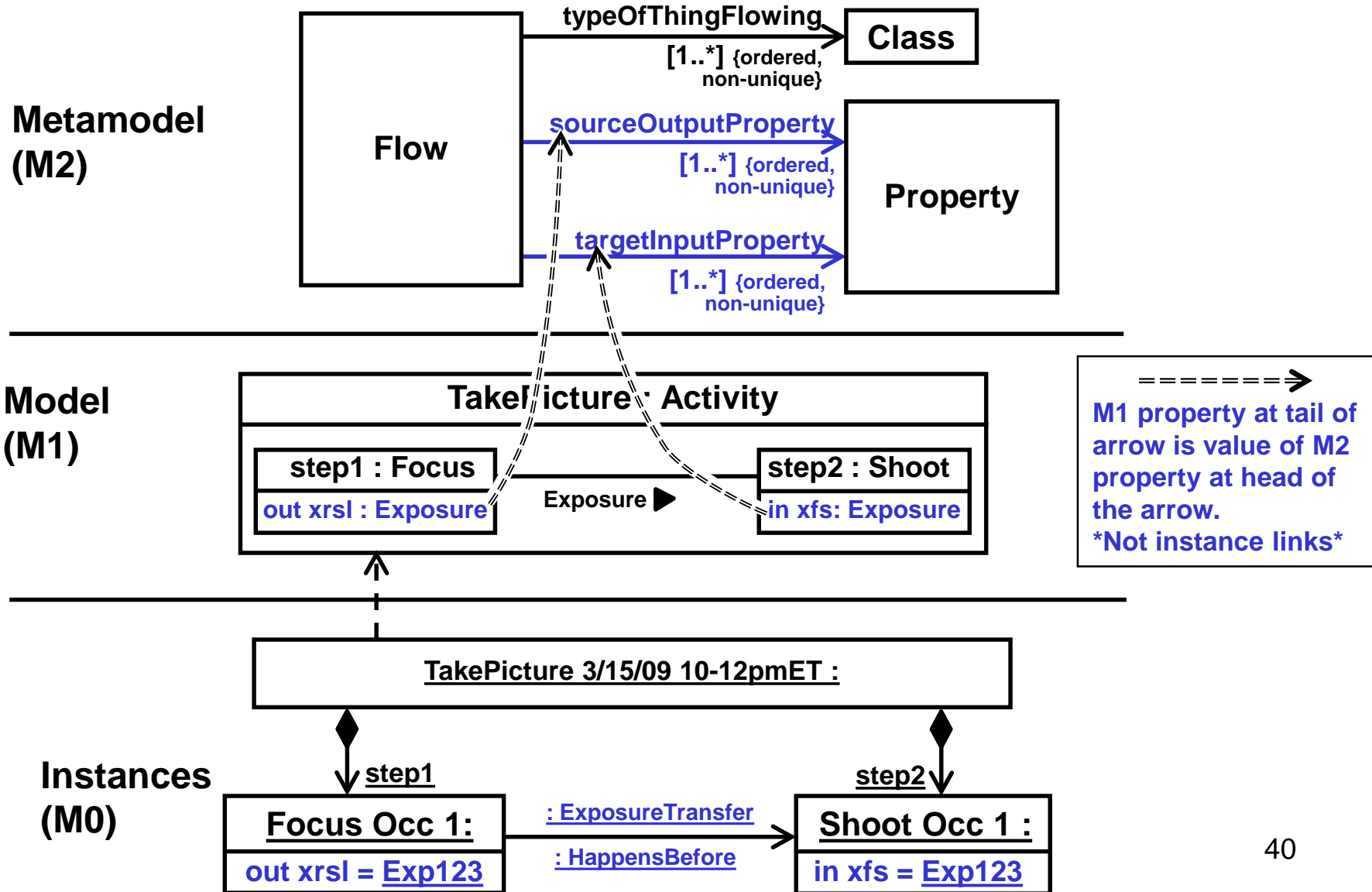- **Summary**
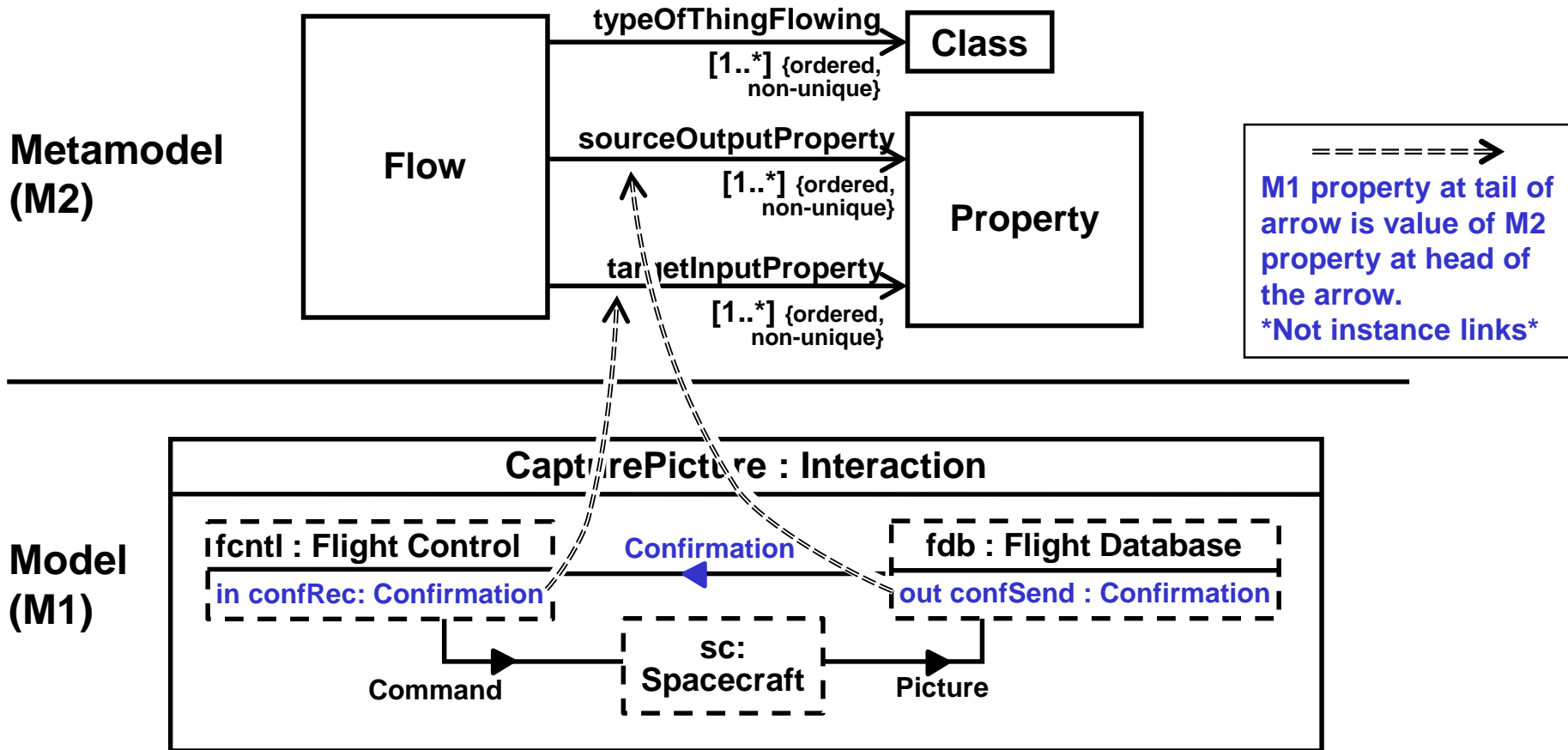
Interactions Solution (Part 3)
(output/input "mailboxes")

- **Outputs**
  - **Output pins of actions, out parameters of behaviors.**
  - **Executions on lifelines.**
  - **SysML out flow properties on parts.**
- **Inputs**
  - **Input pins of actions.**
  - **Executions on lifelines.**
  - **SysML in flow property on parts.**

Flows & Out/Inputs (OF)

Metamodel (M2)

Flow —typeOfThingFlowing→ Class
[1..*] {ordered, non-unique}

Flow —sourceOutputProperty→ Property
[1..*] {ordered, non-unique}

Flow —targetInputProperty→ Property
[1..*] {ordered, non-unique}

Model (M1)

TakePicture : Activity

step1 : Focus
out xrsl : Exposure

Exposure ▶

step2 : Shoot
in xfs: Exposure

M1 property at tail of arrow is value of M2 property at head of the arrow.
Not instance links

Instances (M0)

TakePicture 3/15/09 10-12pmET :

step1
Focus Occ 1:
out xrsl = Exp123

: ExposureTransfer
: HappensBefore

step2
Shoot Occ 1 :
in xfs = Exp123

40

Flows & Out/Inputs (FP)

Metamodel (M2)

Flow

typeOfThingFlowing → Class
[1..*] {ordered, non-unique}

sourceOutputProperty → Property
[1..*] {ordered, non-unique}

targetInputProperty → Property
[1..*] {ordered, non-unique}

=======➔
M1 property at tail of arrow is value of M2 property at head of the arrow.
Not instance links

Model (M1)

CapturePicture : Interaction

fcntl : Flight Control
in confRec: Confirmation

Confirmation

fdb : Flight Database
out confSend : Confirmation

Command

sc: Spacecraft

Picture

Flows & Out/Inputs (FPP)

Metamodel (M2)

Model (M1)

IBD UML Sequence Diagram
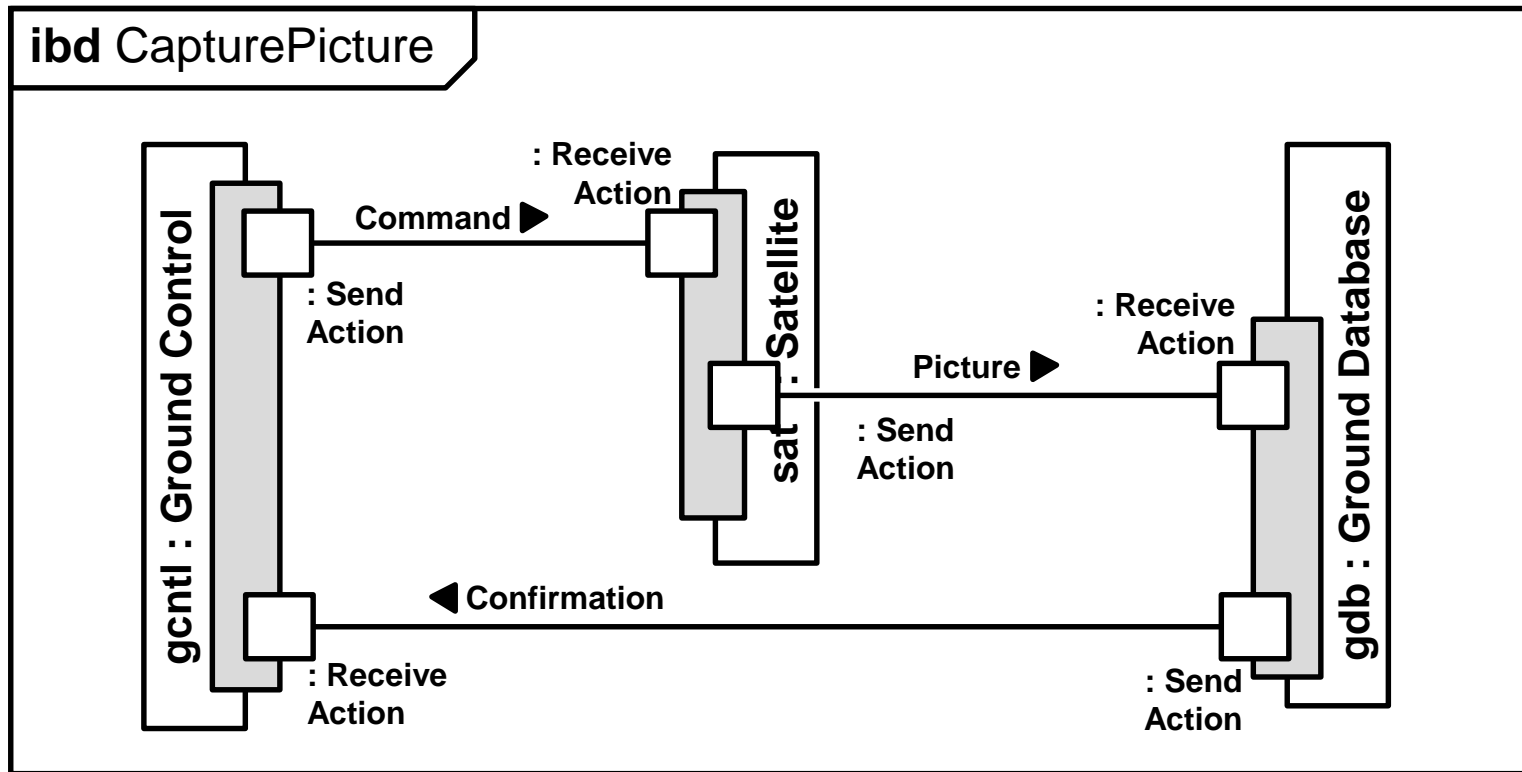
Overview

- **Motivation**
  - **Behavior, review**
  - **Interactions, requirements**
- **Interactions Solution**
  1. **Between long-lived participants**
  2. **Reusable & composable**
  3. **Outputs / inputs**
- **Summary**

Summary

- **Unify UML's models of interaction with**
  - Composite **structure.**
  - Model **library for transfer of things.**
  - Metamodel elements **capturing patterns of using library, applied automatically.**
- **Simplifies metamodel with**
  - More **common interaction elements, fewer specializations.**
  - Standard model library.
- **Speeds learning and analysis integration.**

More Information

- **Intro to Behavior as Composite Structure**
  - **http://doc.omg.org/ad/2018-03-02**
- **Additional slides**
  - **Starts with onto, includes interactions.**
  - **http://conradbock.org/bock-ontological-behavior-modeling-jpl-slides.pdf**
- **Paper: http://dx.doi.org/10.5381/jot.2011.10.1.a3**
- **Application to BPMN:http://conradbock.org/#BPDM**
- **KerML:**
  - **Contact Chas Galey charles.e.galey@lmco.com**