Interactions as Composite Structure:
(Onco)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

Property
Connector
Property
Connector

Activity

State Machine

Interaction

Property
Connector
Behavior: What’s Being Modeled?

“Things” that occur in time
- Eg, taking a picture, focusing, etc.
- Not “behaviors”, “actions”, etc.

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

Not instance specs.

Focus
3/15/09 10-11am ET:

Take Picture
3/15/09 10-12pm ET:

Shoot
3/15/09 11-12pm ET:
They happen before or during each other.
- Construct M1 library for this.
- Use it to classify things being modeled.
Specialize library classes and subset/redefine library properties.
Behavior: Too repetitive at M1?

- Capture M1 patterns in M2 elements.
  - Tools apply patterns automatically.

Things Being Modeled (M0)

- TakePicture
  - 3/15/09 10-12pmET:
  - Focus
    - 3/15/09 10-11amET:
  - Shoot
    - 3/15/09 11-12pmET:

User Model (M1)

- Step1: Focus
- Step2: Shoot
  - HappensBefore

Metamodel (M2)

- Class
- Property
- Connector
- Association
- Behavior
- Succession
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.
Interactions Problem

Activity

Object Flow

Item Flow

SysML Internal Block Diagram

Message

Interaction

PreventLockup [Activity Diagram]

sd ABS_ActivationSequence

d1: Traction Detector

m1: Brake Modulator

detTrkLos()
sendSignal()
modBrkFrc(traction_signal:boolean)
modBrkFrc()
sendAck()
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.
Transfers (M1)

Standard Model Library

Model (M1)

User Model

Things Being Modeled (M0)

Transferred Thing

Source Thing

Target Thing

Behavior Occurrence

Transfer

Any Thing

Product Transfer

Product

Stove234:

Store654:

John’s House:

Product Transfer

3/15/09 10-12pm ET:
Interactions (M2)

Metamodel (M2)

Model (M1)

User Model

Standard Model Library

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

- **Two kinds** of properties, for navigation between:
  - One end object to another (end properties).
  - Links and end objects (link properties).
Link Properties (M1)

Standard Model Library

Model (M1)

User Model

Things Being Modeled (M0)
Association Participants (M2)

Metamodel (M2)

Standard Model Library

Model (M1)

User Model

Things Being Modeled (M0)

Association

Class

Property

owned Property

(subsets)

participant Property

[2..\star] {subsets}

Anything

Camera

Controller

Camera

Controller

Link

LinkedThing {non-unique}

linkedThing

[2..\star]

linkedCon

linkedTarget

linkedSource

linkedCam

linkedCon

\{redefines linkedTarget\}

\{redefines linkedSource\}

Owned Property

\{non-unique\}

\{subsets\}

M1 property at tail of arrow is value of M2 property at head of the arrow.

*Not instance links*

Camera 34 : linkedCam

Link 251 : linkedCon

Cntrl 12 : camCon

conCam

camCon

ConCam

linkedCon

\{redefines linkedTarget\}

\{redefines linkedSource\}
Transfers as Links \((M1)\)

**Standard Model Library**

- **Link**
- **Behavior Occurrence**
- **Transfer**

**Model \((M1)\)**

- **Product Transfer**

**User Model**

- **Product**

**Things Being Modeled \((M0)\)**

- **Product Transfer**
  - **Stove234:**
  - **Store654:**
  - **John’s House:**

**Transfer Details**

- **3/15/09 10-12pm ET:**

**Links and Relations**

- **transferredThing**
- **sourceThing**
- **targetThing**
- **linkedThing**
- **{non-unique}**
- **{subsets}**
- **involves**
- **{redefines}**
- **{subsets}**

**Occurrence Details**

- **Behavior Occurrence**
  - **[1..*]**
  - **[2..*]**

**Model Library Details**

- **Product Transfer**
  - **Stove234:**
  - **Store654:**
  - **John’s House:**
Interaction Participants (M2)

Metamodel (M2)

Class

Association

Behavior

Property

Interaction

typeofThingTransferred

participant

involves

Property

M1 property at tail of arrow is value of M2 property at head of the arrow.

*Not instance links*

transferredThing

sourceThing

targetThing

linkedThing

[1..*]

[2..*]

transferredThing

[1..*]

Product Transfer

Product

Model (M1)

Standard Model Library

User Model
Connectors Reusing Interactions

Metamodel (M2)

User Model (M1)

Things Being Modeled (M0)

Product Delivery
3/15/09 9-1pm ET:

PickupFrom:

DeliverTo:

Associations

Connections

Flow

Classes

Interactions

Product Transfer
3/15/09 10-12pm ET:

John's House:

Stove 234:
Connectors as Properties

Things Being Modeled (M0)

Product Delivery
3/15/09 9-1pm ET:

pickupFrom:

deliverTo:

User Model (M1)

DeliverProduct:

pickupFrom:

pt : ProductTransfer

deliverTo:

Association

Things Being Modeled (M0)

Product Transfer
3/15/09 10-12pm ET:

pickupFrom:

pt

deliverTo:

User Model (M1)

DeliverProduct:

pickupFrom:

pt : ProductTransfer

deliverTo:

Association

Metamodel (M2)

Class

owned Property

* {subsets}

owned Connector

* {subsets}

owned ItemFlow

* redefines

Property

Connector

Interaction

Association

Class

owned Property

* {subsets}

owned ItemFlow

* redefines

Flow

Type
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.**
  - Successions = connectors typed by HappensBefore from standard M1 library.
IBD UML Interaction Diagrams

Sequence diagram

Communication diagram

1. Command

3. Confirmation

2. Picture
Connectors Reusing Interactions

Model (M1)

FlightOperation : Interaction

cntl : GroundControl

sc : Satellite

db : TelemetryDB

cntl : Flight Control
{subsets linkedThing}

fdb : Flight Database
{subsets linkedThing}

sc: Spacecraft
{subsets linkedThing}

Command

Confirmation

: happens Before

Picture

: happensBefore
IBD UML Collaborations

Model (M1)

FlightOperation : Interaction

cntl : GroundControl

sc : Satellite

fdb : Flight Database

sc : Spacecraft

cntl : Flight Control

db : TelemetryDB

Command

Picture

: CapturePicture

: happensBefore

: happensBefore

: Confirmation
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
  - Class
  - Property
  - sourceOutputProperty
    - [1..*] {ordered, non-unique}
  - targetInputProperty
    - [1..*] {ordered, non-unique}
  - typeOfThingFlowing

Model (M1)

- TakePicture Activity
  - step1: Focus
    - out xrsi: Exposure
  - step2: Shoot
    - in xfs: Exposure

Instances (M0)

- Focus Occ 1:
  - out xrsi = Exp123
- Shoot Occ 1:
  - in xfs = Exp123

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

TakePicture 3/15/09 10-12pmET:
Flows & Out/Inputs (FP)

Metamodel (M2)

Flow

- \text{sourceOutputProperty} \in [1..*] \{\text{ordered, non-unique}\}
- \text{targetInputProperty} \in [1..*] \{\text{ordered, non-unique}\}

Class

Property

Model (M1)

CapturePicture : Interaction

- \text{fcntl : Flight Control}
  - in \text{confRec : Confirmation}
- \text{fdb : Flight Database}
  - out \text{confSend : Confirmation}
- \text{sc : Spacecraft}
  - \text{Command}
  - \text{Picture}

\text{M1 property at tail of arrow is value of M2 property at head of the arrow.}
*Not instance links*
Flows & Out/Inputs (FPP)

Metamodel (M2)

Flow

\[\text{sourceOutputProperty} \rightarrow \text{targetInputProperty} \]

\[\text{sourceOutputProperty} \rightarrow \text{targetInputProperty} \]

Class

Property

\[\text{typeOfThingFlowing} \rightarrow \text{Class} \]

\[\text{Property} \rightarrow \text{M1 property at tail of arrow is value of M2 property at head of arrow.} \]

Capture Picture : Interaction

Model (M1)

fcntl : Flight Control

rc : RC

in confRec: Confirmation

Confirmation

fdb : Flight Database

sc: SC

out : Confirmation

Command

sc: SC

Picture

sc: SC

Confirmation

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.

- 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