Behavior as Composite Structure: (Onto)Logical Behavior Modeling

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

Antoine Lonjon
MEGA

James Odell
Odell & Associates
Overview

- Motivation
- Composite structure
- Behaviors as composites
- UML (lack of) support
- (Onto)logical modeling
- Summary
Overview

- Motivation
- Composite structure
- Behaviors as composites
- UML (lack of) support
- (Onto)logical modeling
- Summary
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.
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 Diagram**
- `act PreventLockup (Activity Diagram)`
- `d1: Traction Detector`
- `m1: Brake Modulator`
- `detTrkLos()`
- `sendSignal()`
- `modBrkFrc(traction_signal:boolean)`
- `sendAck()`

**State Machine Diagram**
- `stm TireTraction (State Diagram)`
- **States:**
  - Gripping
  - Slipping
  - LossOfTraction
  - RegainTraction

**Sequence Diagram**
- `[Sequence Diagram]`
- `sd ABS_ActivationSequence`
- `detTrkLos()`
- `sendSignal()`
- `modBrkFrc(traction_signal:boolean)`
- `sendAck()`
- `modBrkFrc()`

**Diagram Elements:**
- `Property`
- `Connector`
Overview

- Motivation
- Composite structure
- Behaviors as composites
- UML (lack of) support
- (Onto)logical modeling
- Summary
Whole-Part

- Whole-part relationships can be modeled as associations.

---

Whole-part

Model (M1)

Things Being Modeled (M0)
Part-Part, Not

- Part-part relationships cannot be represented by associations.

Model (M1)

- Camera
- Controller
- Satellite

Things Being Modeled (M0)

- Controller in RussianSat
- Camera in RussianSat
- Controller in USSatellite
- Camera in USSatellite

RussianSatellite: ctrl -> Controller in RussianSat
Camera in RussianSat: cam -> Camera in RussianSat

USSatellite: ctrl -> Controller in USSatellite
Camera in USSatellite: cam -> Camera in USSatellite

Controls: Controller in RussianSat -> Camera in RussianSat
Controls: Controller in USSatellite -> Camera in USSatellite

Attempt at part-part
UML Composite Structure

Model (M1)

Property

Connector (part-part)

Things Being Modeled (M0)

RussianSatellite:

Controller in RussianSat: : Controls

Camera in RussianSat:

Controller in USSatellite: : Controls

Camera in USSatellite:

Controlling camera in same satellite
Overview

- Motivation
- Composite structure
- Behaviors as composites
- UML (lack of) support
- (Onto)logical modeling
- Summary
Behavior model “things” happening over time.

- With temporal relations (time constraints) between them.
• Composite structure relations are temporal:
  – Whole-part = happens during.
  – Part-part = happens before.
Behavior as Composite Structure

Model (M1)

Property (whole-part)

Connector (part-part)

class TakePicture

step1: Focus

: HappensBefore

step2 : Shoot

Focus

HappensBefore

Shoot

Things Being Modeled (M0)

TakingPic1:

step1: Focusing

: Happens Before

step2: Shooting

TakingPic2:

step1: Focusing

: Happens Before

step2: Shooting

Focusing before shooting in same taking picture
Generalizing Composite Structure

- Constraints are inherited in UML
  - including temporal constraints.
Overview

- Motivation
- Composite structure
- Behaviors as composites
- UML (lack of) support
- (Onto)logical modeling
- Summary
Behaviors are Classes in UML

- Things being modeled are executions (instances) of behavior.
Behaviors are Classes in UML

- Things being modeled are executions (instances) of behavior.
But No Properties & Connectors

- No links between things being modeled.
Steps:
- Are properties ...
- typed by behaviors at M1...
- that have “suboccurrences” as values at M0.
Whole-part for Behaviors

Steps:
- Are properties ...
- typed by behaviors at M1...
- specialized from happensDuring^{-1} in a standard M1 library...
- that have “suboccurrences” as values at M0.
Part-part for Behaviors

- Successions:
  - Are connectors ...
  - typed by happensBefore from a standard M1 library ...
  - resulting in links between suboccurrences at M0.
Automated M1 Patterns

- Such as
  - Typing properties by behaviors and ...
  - ... subsetting them from happensDuring\(^{-1}\)
  - ... linking them with connectors typed by HappensBefore.

- Specified in a standard M2
  - In M2 Step and Succession ...
  - ... using OCL, etc.

- Applied at M1 during M2 instantiation ...
  - ... automatically.
  - Modelers & API users don’t need to know.
Overview

- Motivation
- Composite structure
- Behaviors as composites
- UML (lack of) support
- (Onto)logical modeling
- Summary
The “O” Word

- Has many meanings
  - Can spend more time defining it than doing it.

- Two meanings used here:
  - Start with the things being modeled (real, desired, imagined, simulated, etc).
  - Group (classify) those things by their commonalities.
(Onto)Logical Modeling

- Start with the things being modeled (M0), and works towards a language.
  - At OMG we normally start immediately with language (metamodels, M2).

- Look for commonalities among the things being modeled
  - Build model libraries (M1) capturing commonalities of those (M0) things.

- When modeling becomes too repetitive, capture M1 patterns in metamodels.
“Things” that occur in time
- Eg, taking a picture, focusing, etc.
- Not “behaviors”, “actions”, etc.
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.
Capture M1 patterns in M2 elements.
- Tools apply patterns automatically.
Overview

- Motivation
- Composite structure
- Behaviors as composites
- UML (lack of) support
- (Onto)logical modeling
- Summary
Summary

- Unify UML’s three behavior models with
  - Composite structure.
  - Model library for temporal relations.
  - Metamodel elements capturing patterns of using library, applied automatically.

- Simplifies metamodel with
  - More common behavior elements, fewer specializations.
  - Standard model library.

- Speeds learning and analysis integration.
More Information

- **Additional slides**
  - Starts with onto, includes interactions.

- **Paper:** [http://dx.doi.org/10.5381/jot.2011.10.1.a3](http://dx.doi.org/10.5381/jot.2011.10.1.a3)

- **Application to BPMN:** [http://conradbock.org/#BPDM](http://conradbock.org/#BPDM)

- **KerML:**
  - Contact Chas Galey [charles.e.galey@lmco.com](mailto:charles.e.galey@lmco.com)