State Machines as Composite Structure:
(Onto)Logical State Machines
Part 1

Bjorn Cole
Georgia Institute of Technology

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

- RoadMap
- Motivation
  - Behavior, review
  - Interactions, review
  - State machines, requirements
- State Machines Solution
  1. Stimuli = end of transfers (events)
  2. State and transition behaviors
  3. Matching past events to transitions
- Summary
Behavior as Composite Structure
Presentation Stack

Onto State Machines
(this one)

Onto Interactions
(ad/18-06-11)

Onto OO
(ad/18-09-07)

Onto Behavior Basics
(ad/2018-03-02)
Overview

- RoadMap
- Motivation
  - Behavior, review
  - Interactions, review
  - State machines, requirements
- State Machines Solution
  1. Stimuli = end of transfers (events)
  2. State and transition behaviors
  3. Matching past events to transitions
- 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

**State Machine**

- **Gripping**
- **Slipping**
- **LossOfTraction**
- **RegainTraction**

**Sequence Diagram**

- `detTrkLos()`
- `modBrkFrc()`
- `sendSignal()`
- `modBrkFrc(traction_signal:boolean)`
- `sendAck()`

**Activity Diagram**

- `act PreventLockup [Activity Diagram]`

**Property**

- `d1:Traction Detector`
- `m1:Brake Modulator`

**Connector**

- `act PreventLockup [Activity Diagram]`
- States and transitions in the State Machine
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 10-11pm ET:

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

Shoot
3/15/09 11-12pm ET:
Behavior: What’s in Common?

- 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.
Overview

- RoadMap
- Motivation
  - Behavior, review
  - Interactions, review
  - State machines, requirements
- State Machines Solution
  1. Stimuli = end of transfers (events)
  2. State and transition behaviors
  3. Matching past events to transitions
- Summary
Interactions Problem

Object Flow

Activity

Item Flow

SysML Internal Block Diagram

Message

sd ABS_ActivationSequence

d1: Traction Detector

sendSignal() 

modBrkFrc() 

mc1: Brake Modulator

modBrkFrc(traction_signal:boolean)

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.
Transfers (M1)

Standard Model Library

Model (M1)

User Model

Things Being Modeled (M0)

**Behavior Occurrence**

**Transfer**

**Product Transfer**

**Product**

**Stove234:**

**Store654:**

**John’sHouse:**

Transferred Thing

Source Thing

Target Thing

Involves

{subsets}

{redefines}

[1..*]

{subsets}

[1..*]

Any Thing
Interactions (M2)

- **Metamodel (M2)**
  - **Class**
  - **Behavior**
  - **Interaction**
  - **Property**
  - **Class**

- **Standard Model Library**
  - **Behavior Occurrence**
  - **Transfer**
  - **Product Transfer**
  - **Product**
  - **Any**

- **Model (M1)**
  - **Transfer**
  - **targetThing**
  - **sourceThing**
  - **transferredThing**

- **User Model**
  - **Transfer**
  - **targetThing**
  - **sourceThing**
  - **transferredThing**

M1 property at tail of arrow is value of M2 property at head of the arrow.
*Not instance links*
Transfers Along 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.
Links (M1) & Associations (M2)

Metamodel (M2)

Standard Model Library

Model (M1)

User Model

Things Being Modeled (M0)

Class

Association

Property

Owned Property

Participant Property

Anything

Camera

Controller

Link

Linked Thing

Linked Target

Linked Source

Camera 34

Link 251

Cntrl 12

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

*Not instance links*
Transfers as Links (M1)

Standard Model Library

Any Thing

Model (M1)

Behavior Occurrence

Link

(targetThing \rightarrow [2..*])

(linkedThing \{\text{non-unique}\} \rightarrow [*])

(sourceThing \rightarrow [1..*])

Product Transfer

(Product \rightarrow *)

Product

User Model

Things Being Modeled (M0)

Product Transfer 3/15/09 10-12pmET

(Product Transfer \rightarrow 3/15/09 10-12pmET)

(Product \rightarrow Stove234)

(Stove234 \rightarrow John’sHouse)

(John’sHouse \rightarrow sourceThing)

(sourceThing \rightarrow Store654)

(Store654 \rightarrow targetThing)

(targetThing \rightarrow Product Transfer 3/15/09 10-12pmET)
Connectors Reusing Interactions

Metamodel (M2)

User Model (M1)

Things Being Modeled (M0)

DeliverProduct:

pickupFrom: pt : ProductTransfer

deliverTo:

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

Store654:

John’sHouse:

Stove234:

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

pickupFrom

deliverTo

pt

transferredThing

owned Property

owned Connector

owned Flow

* (subsets)

{redefines}

{subsets}

{subsets}

Property

Connector

Flow

Class

Association

Interaction

Class
Flow Steps

Metamodel (M2):
- Class
- Property
- Connector
- Association
- Step
- Interaction
- Flow
- CapturePicture
- Behavior
- Succession

Model (M1):
- fcntl: Flight Control
- fdb: Flight Database
- sc: Spacecraft
- Command
- Picture
- Transfer

User Model

Standard Model Library
Flows & Out/Inputs (OF)

Metamodel (M2)

Model (M1)

Instances (M0)

TakePicture Activity

step1 : Focus
out xrls : Exposure

step2 : Shoot
in xfs : Exposure

Focus Occ 1:
out xrls = Exp123

Shoot Occ 1:
in xfs = Exp123

HappensBefore

ExposureTransfer

M1 property at tail of arrow is value of M2 property at head of the arrow.
*Not instance links*
Flows & Out/Inputs (FP)

Metamodel (M2)

Model (M1)

CapturePicture : Interaction

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

- RoadMap
- Motivation
  - Behavior, review
  - Interactions, review
  - State machines, requirements
- State Machines Solution
  1. Stimuli = end of transfers (events)
  2. State and transition behaviors
  3. Matching past events to transitions
- Summary
States of What?

- **Objects**, based on properties
  - Person in married state = has a spouse.

- **Behaviors**, based on past behavior
  - Vending machine in dispensing state after receiving selection and money states.

- **UML states** are mostly behaviors ...
  - ... tied to objects.
  - Weakly include object state invariants.

- Both kinds can be in “machines” that react to external stimuli.
State Machine Problem

- UML has **two ways** things can react to external stimuli:
  - State Machines have **transitions**.
  - Activities have **accept event actions**.

- **Very little integration** or reuse.
  - Two underlying metamodels/profiles.
  - Two representations of reactions.
  - Slightly different temporal semantics.

- **Doubles the effort** of learning UML and building analysis tools for them.
State Machine Problem

Transition Trigger

Accept Event Action

Call Behavior Action
State Machine Requirements

1. **Must selectively react to stimuli** ("events").
   - Based on kind of stimulus and …
   - … current & previous stimuli/reactions ("states")

2. **Must simplify reaction behaviors**, splitting them up …
   - by state and between states (transitions).
   - within states.

3. **Must react to past events**
   - Can have complicated reaction rules to events in the past.
Overview

- RoadMap
- Motivation
  - Behavior, review
  - Interactions, review
  - State machines, requirements
- State Machines Solution
  1. Stimuli = end of transfers (events)
  2. State and transition behaviors
  3. Matching past events to transitions
- Summary
State Machine Solution (Part 1)  
(Reacting to stimuli)

- Reaction **depends on current state**.
  - **Change** states (leave current one, enter another).
  - **Re-enter** current state.
  - **Do nothing**.

- Events can **arrive during or before** states expecting them.
  - Addressed separately.
State Machine Solution (Part 1)
(Reacting to stimuli)

- **UML events = things “arriving” at objects**
  - Signals, operation calls
  - Not events happening externally
    - Except unmodeled “changes” to anything.

- Treat as **ends of transfers** targeting objects.
  - Receiver doesn’t specify sender.
UML Events = Ends of Transfers

Standard Model Library

Model (M1)

User Model

Things Being Modeled (M0)

TakePicture

step1 : Focus

: HappensBefore

step2 : Shoot

stepX : ExposeCmdXfer 0..1

: HappensDuring

end :

self

Exposure CmdXfer

Transferred

ExposeCmd

Focus

3/15/09 10-11pm ET:

ExpCmdXfer #7453 :

end = 3/15/09 10:45 pm ET:

: HappensDuring

TakePicture

3/15/09 10-12pm ET:

Shoot

3/15/09 11-12pm ET:

HappensBefore

3/15/09 10-12pm ET :

HappensBefore

HappensDuring
Transitions are successions that …
- go out of steps …
- that interactions (triggers) end during …
- that target the machine.
Overview

- RoadMap
- Motivation
  - Behavior, review
  - Interactions, review
  - State machines, requirements
- State Machines Solution
  1. Stimuli = end of transfers (events)
  2. State and transition behaviors
  3. Matching past events to transitions
- Summary
State Machine Solution (Part 2)
(Simplifying reaction behaviors)

- States have **entry, do, and exit behaviors**
  - Happen **going into, being in, going out of states**.
  - No other state behaviors, simplifies behavior modeling.
  - Only do behaviors can be stopped by events.

- Transitions have **effect behaviors**
  - Happen **after source state exit and before target state entry**.
State Behaviors (M1)

- State occurrences:
  - Are behavior occurrences typing state properties...
  - with exactly three step properties ordered in time
Exit behaviors happen after triggers end.

- Exit behaviors happen after triggers end.
Aborting Do Behaviors

- Do behaviors stop before event arrives
  - Even if they aren’t finished.
  - Assumes do behaviors are abortable.

Model (M1)

User Model

Standard Model Library

Model Library

StateOccurrence

TakePicture

state1 : Focusing
  do : Focus

state2 : Shooting

During
state1

Before
stepX : ExposeCmdXfer 0..1
end :

User

Focus

Focus

{redefines do}

state1

40
Transitions can specify behaviors to happen in between states.
State Machine Problem (#2)

Competing Transitions

Interruptible Region

Interrupting Edge
Competing Transitions (M1)

Standard Model Library

StateOccurrence

Transfer

{ Accepted interaction ends before the other acceptable interactions do. }

{ Must have value (link) iff state1.accepted has a value & = step1T1. }

{ Must have value (link) iff state1.accepted has a value & = step1T2. }

Model (M1)

User Model

TakePicture

state1 : Focusing

state2 : Shooting

state3 : Setting WhitePoint

During

Before

exit :

accepted :

acceptable :

subsets

self
target

step1T1 : ExposeCmdXfer 0..1

step1T2 : WB&ExposeCmdXfer 0..1

Must have value (link) iff state1.accepted has a value & = step1T1.

Must have value (link) iff state1.accepted has a value & = step1T2.

tends

{ Accepted interaction ends before the other acceptable interactions do. }
Competing Transitions (M1Lib/M2)

- Library constraints inherited or reused
  - Acceptable/exit timing moved to library.
  - Transition constraints use M2.
  - Commonly used acceptance constraints.

For all models

For models to use as needed
Overview

- RoadMap
- Motivation
  - Behavior, review
  - Interactions, review
  - State machines, requirements
- State Machines Solution
  1. Stimuli = end of transfers (events)
  2. State and transition behaviors
  3. Matching past events to transitions
- Summary
State Machine Problem (Part 3)  
(Reacting to past events)

- So far, states are only triggered by events that arrive during the state.
- Want to enable states to be triggered by events that arrive before the state.
Past Events (M1)

- Events arriving before state are acceptable.
  - But can only be accepted once.
HappensDuring redefined to apply as indicated by boolean.
State Machine TBD

- Concurrent regions.
- Multiple machines and activities using the same events.
  - Objects with multiple behaviors.
- More complex event handling.
- Pulling from buffer, rather than matching (maybe).
Overview

- RoadMap
- Motivation
  - Behavior, review
  - Interactions, review
  - State machines, requirements
- State Machines Solution
  1. Stimuli = end of transfers (events)
  2. State and transition behaviors
  3. Matching past events to transitions
- Summary
Summary

- Unify reacting to events using
  - Transfer ends as events
  - Properties for state behaviors.

- Model of event processing
  - Matching events by constraints …
  - … easier for end user than event handling procedures.

- Speeds learning and analysis integration.
More Information

- Intro to Behavior as Composite Structure
- Interaction as Composite Structure
- Object-orientation as Composite Structure
  - [http://doc.omg.org/ad/18-09-07](http://doc.omg.org/ad/18-09-07)
- Earlier slides (more 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)