Manufacturing Engineering

Execution Interoperability

Conrad Bock

U.S. National Institute of Standards and Technology

March 26, 2007

NIST

Overview

- Kinds of interoperability.
- Metamodeling and interoperability.
- Composing execution.
- Execution events.
- Capturing common elements in orchestration and choreography.
- BPMN execution examples.

Kinds of Interoperability

- **Diagrams:**
  - **Bitmaps (PNG, etc), Shapes (SVG, etc).**
  - **Receiver's *screen* same as Sender's.**
- **Repository (metamodels):**
  - **Orchestrations, choreographies.**
  - **Receiver's *repository* same as Sender's.**
- **Execution (runtime):**
  - **Performance or enactment of orchestration and choreography.**
  - **Receiver's *execution* same as Sender's.**

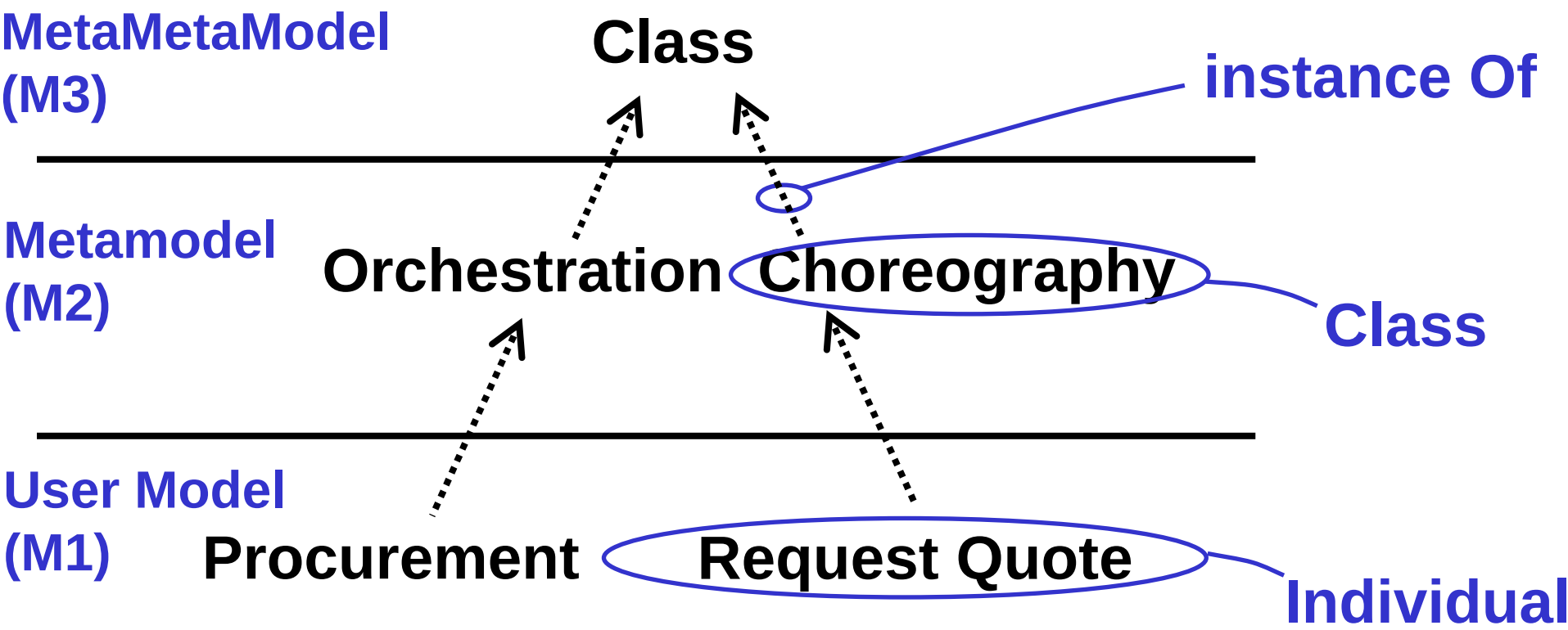
Uniform Execution

- **Large organizations have many kinds of execution tools from many vendors.**
- **Not enough to exchange diagrams and repository contents.**
- **Orchestrations and choreographies must execute the same way before and after interchange.**
- **Otherwise: cost overruns due to rework and managing different versions of the same process for different platforms.**

Metamodels and Execution

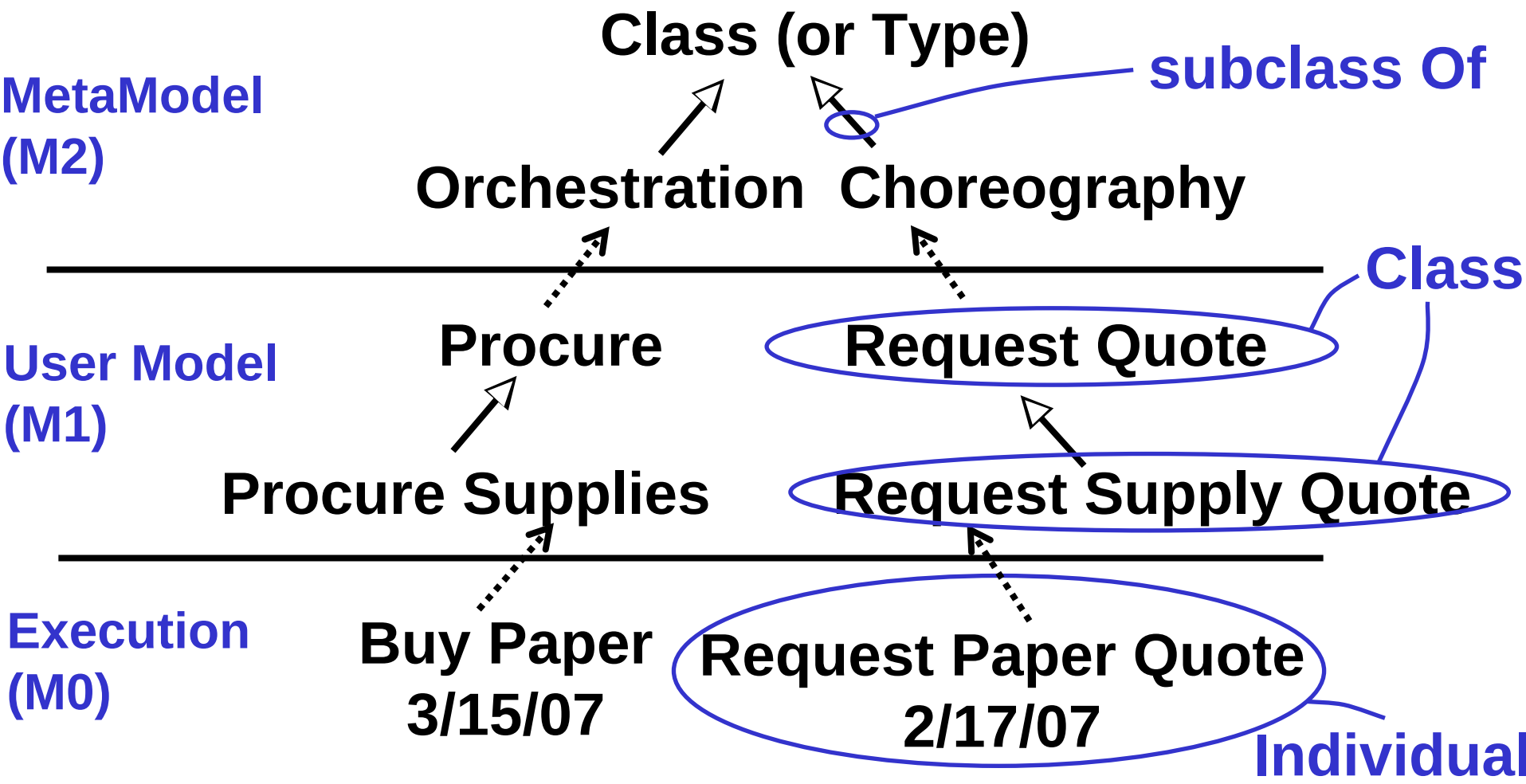
- **Not all metamodels are created equal.**
- **Some carry only modeling terminology, with runtime behavior relegated to text.**
- **Result: nonuniform execution, higher cost, lower ROI, fragile assets.**
- **Others account for runtime behavior, depending less on text.**
- **Result: more uniform execution, lower cost, higher ROI, assets hold value.**

Metamodeling Without Execution



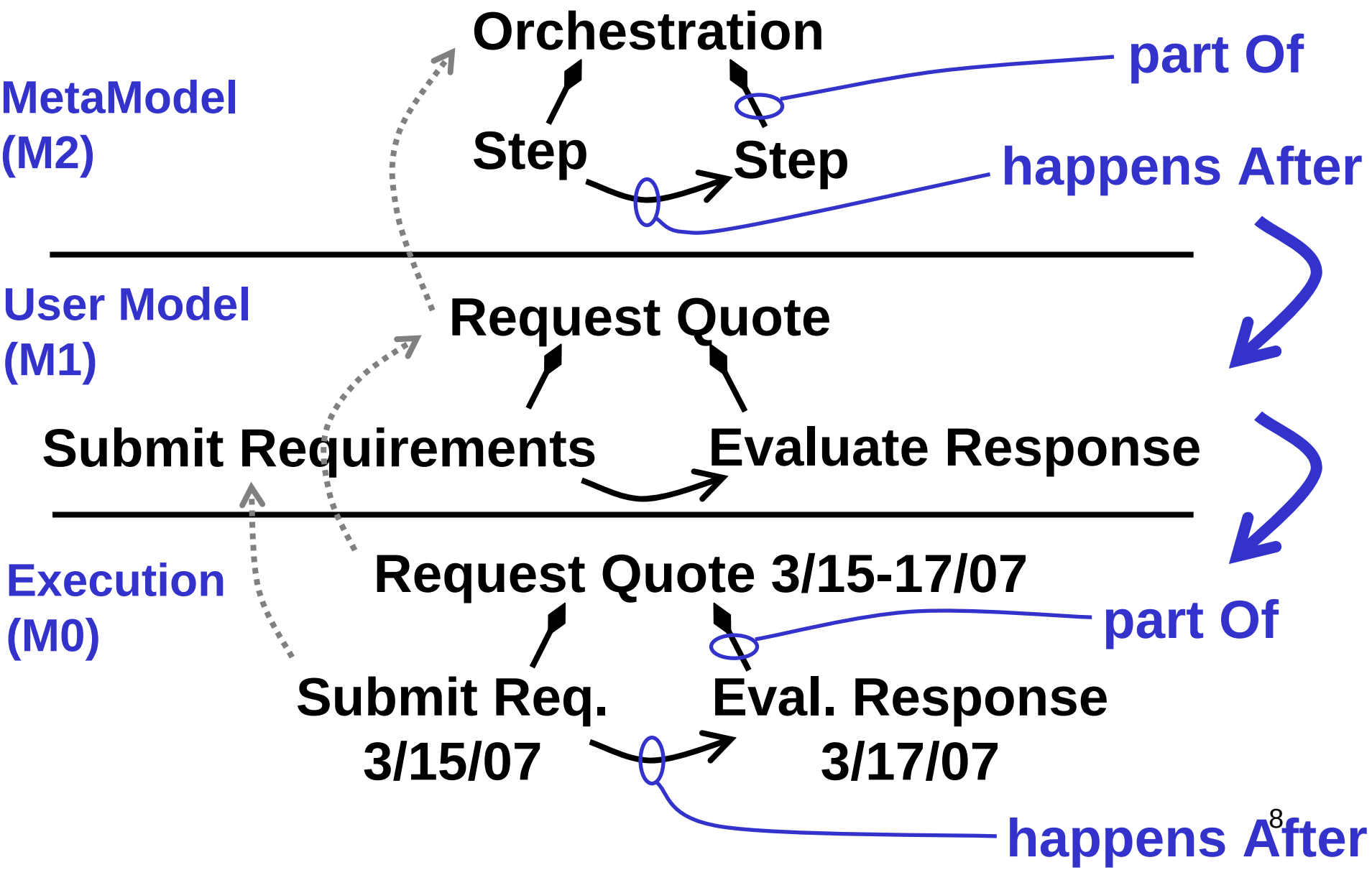
- **Cannot instantiate and specialize user models (they are individuals, not classes).**
- **No runtime execution (M0).**

Metamodeling With Execution

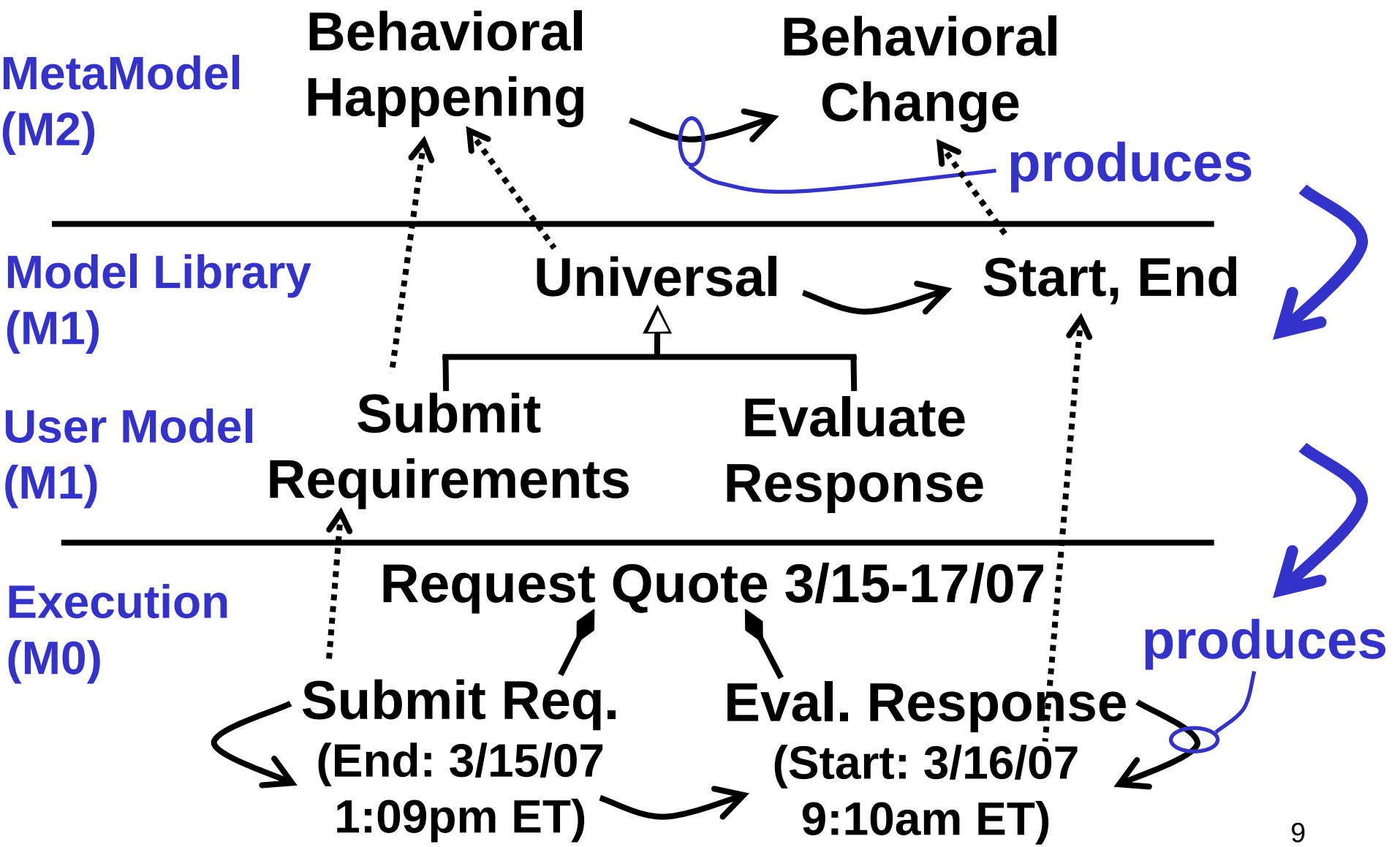


- M1 orchestrations and choreographies are classes, can be specialized in M1 and instantiated at M0.
- M1 constraints apply to M0 executions.

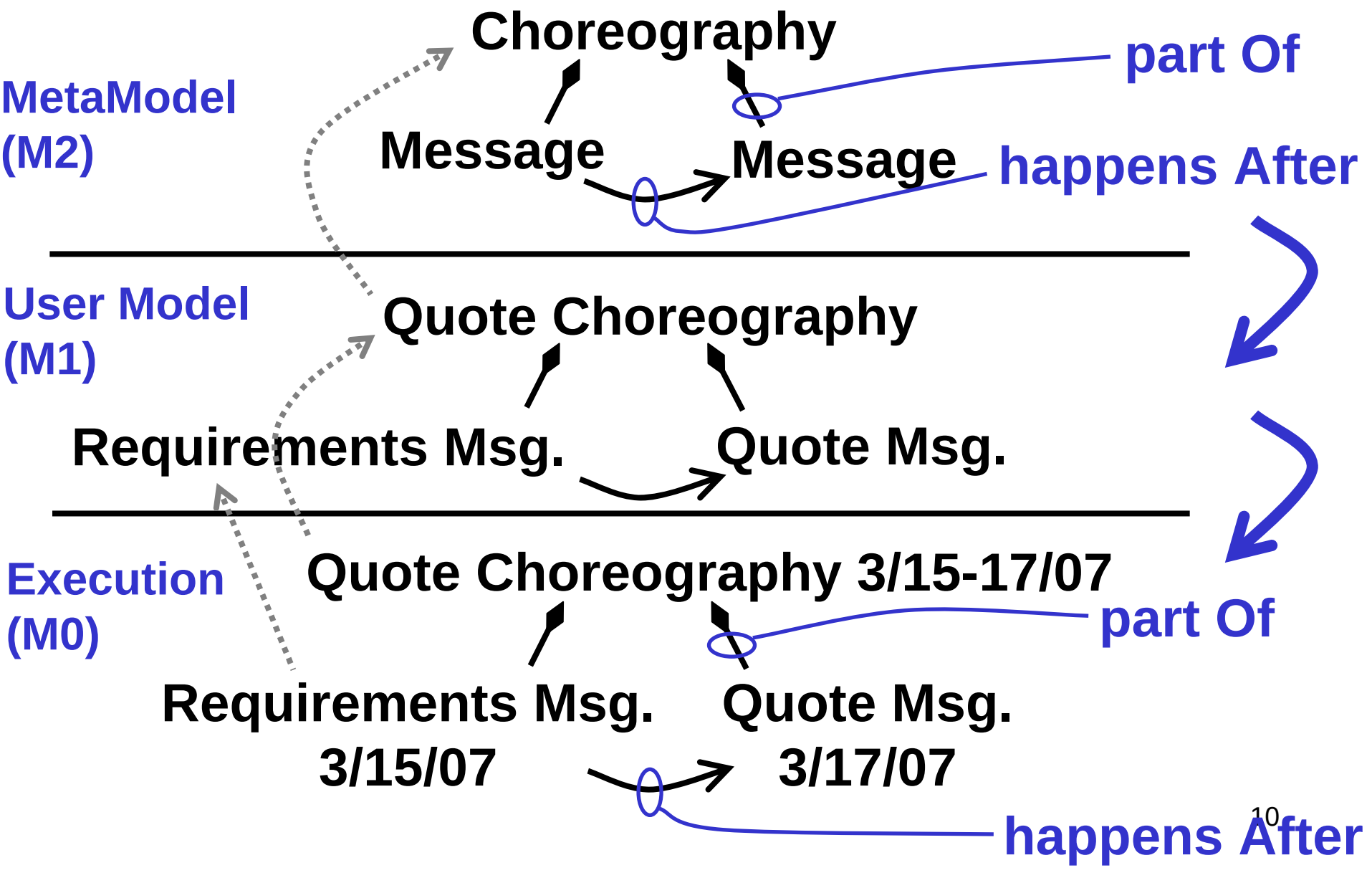
Composing Execution (Orch.)



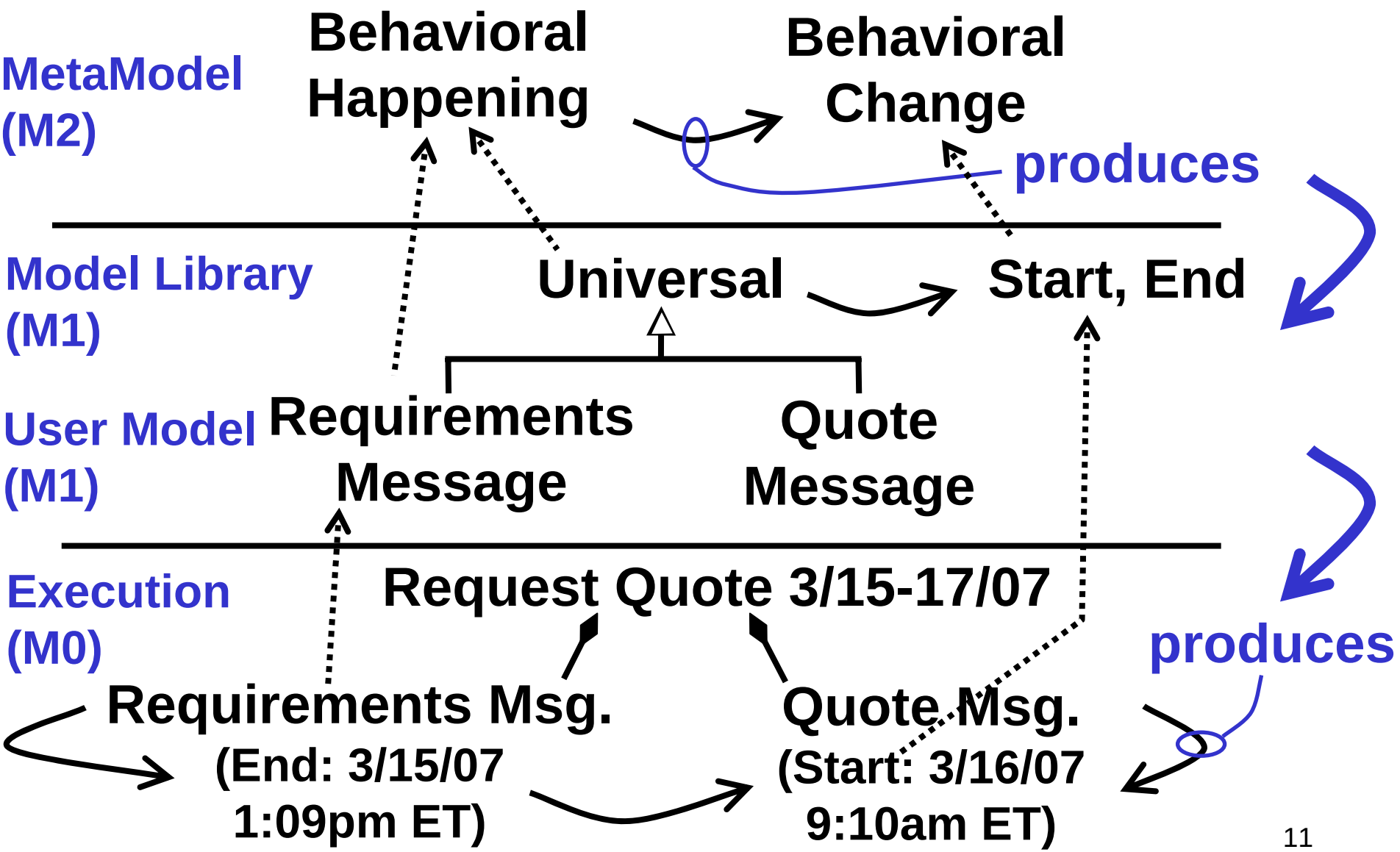
Execution Lifecycle Events



Composing Execution (Chor.)



Execution Events (Messages)



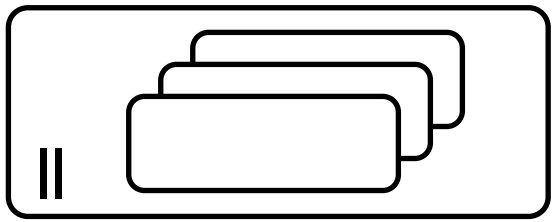
“Happens After” (Succession)

- **One step or message happens *sometime* after another.**
- **Enables orchestration and choreography to:**
  - **be partially defined (say only what you need to).**
  - **form taxonomies (subtyping).**
- **Semantics can be expressed as constraints on execution *derived* from M1 models (PSL).**
- **Compare to token movement:**
  - **Happens immediately.**
  - **Total definitions.**
  - **No taxonomies.**
  - **Semantics *overlaid* on models.**

BPMN Execution:

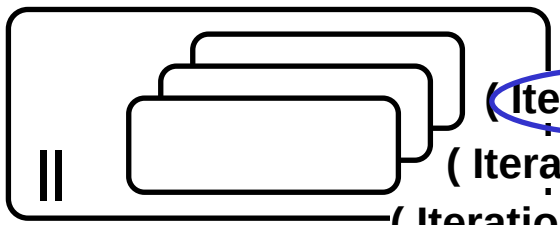
- **Multi-instance loops have one notation (M1, above), but four execution patterns (M0):**

1. End of all iterations:



(End: 3/16/07 9:13am ET)

2. End of each iteration:



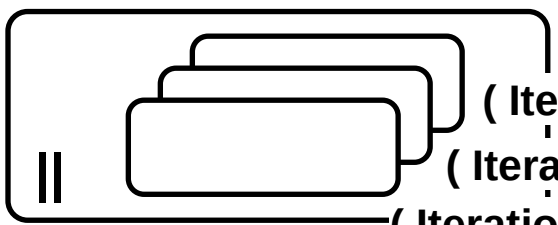
(Iteration End: 9:13am ET)

(Iteration End: 9:12am ET)

(Iteration End: 9:11am ET)

Behavioral Change (BPDM M1)

3. End of some iterations:



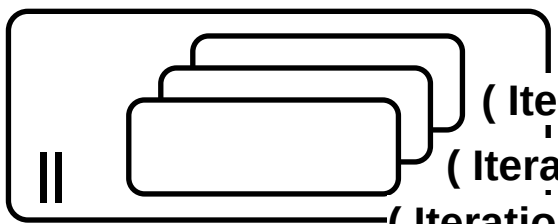
(Iteration End: 9:13am ET , condition succeeds)

(Iteration End: 9:12am ET , condition fails)

(Iteration End: 9:11am ET , condition succeeds)

Condition (User M1)

4. End of first iteration:



(Iteration End: 9:13am ET , “first” fails)

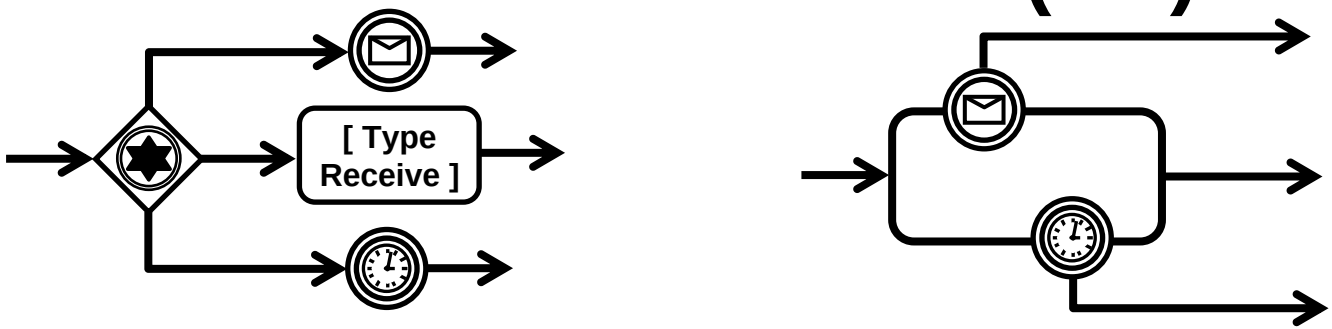
(Iteration End: 9:12am ET , “first” fails)

(Iteration End: 9:11am ET , “first” succeeds)

Condition (BPDM M1)

BPMN Execution:

- **Event-based decisions and attached events have two notations (M1):**

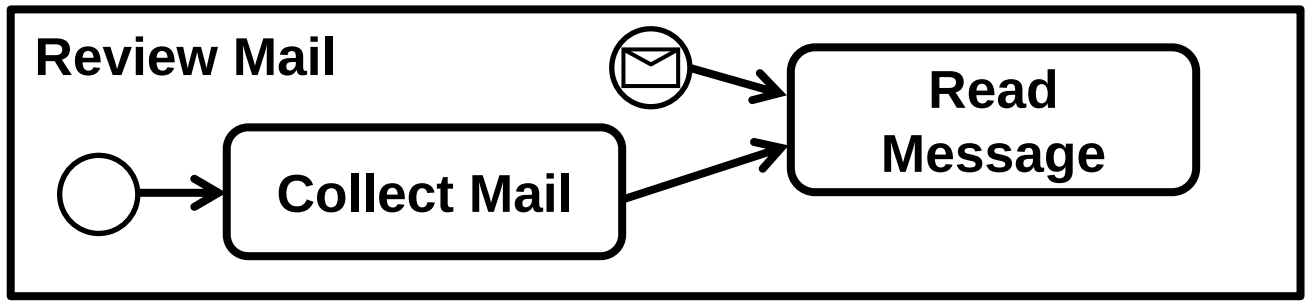


- **but one execution pattern (M0):**

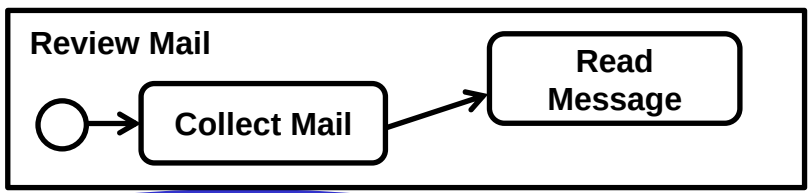
Racing Behavior (BPDM M1)

BPMN Execution:

- **Diagram can have multiple start events (M1):**

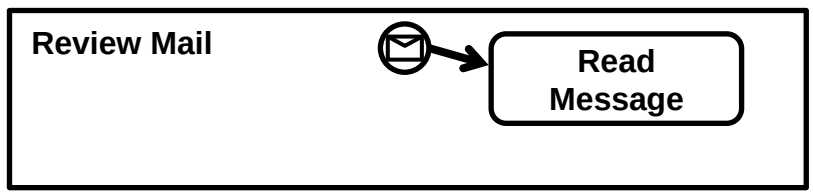


- **but only one is used per execution (M0):**



(startFromSequence: 9:13am ET)

Behavioral Change
(BPDM M1)



(finish: 2:40pm ET)

(start: 2:40pm ET)

happens immediately after

Summary

- **Diagram, modeling, and execution interoperability.**
- **Metamodeling for execution.**
- **Executions and their lifecycles events.**
- **Common execution patterns in orchestration and choreography.**
- **Capturing BPMN execution semantics.**
- **Benefits: significantly improved communication, implementation, and interoperability.**