

Modeling Events in Object-Process Methodology and in Statecharts

Iris Reinhartz-Berger, Arnon Sturm, Dov Dori
Technion – Israel Institute of Technology
[ieiris@tx](mailto:ieiris@tx.technion.ac.il), [sturm@tx](mailto:sturm@tx.technion.ac.il), [dori@ie](mailto:dori@ie.technion.ac.il).technion.ac.il



Outline

- Motivation
- Methods and Issues Examined
- Event Expression Comparison
- Event Implementation
- Comparison and Summary

Motivation

- Complex systems are often **reactive**, i.e., they continuously respond to external and internal **stimuli** (events) and may have time constraints.
- When modeling such systems, the designer should be able to **determine** the system's **behavior**, as well as its **flow of control**.
- System modeling methods must express this action semantics in a way that is both formal and intuitive.

Methods and Issues Examined

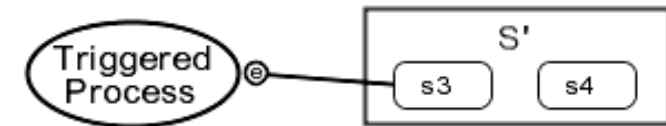
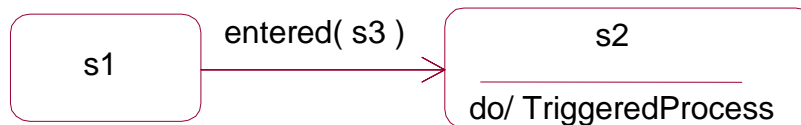
- We focus on two behavior oriented approaches:
 - **Statecharts** – Harel's extension of state machines, used in UML
 - **Object-Process Methodology** – an integrated modeling method that unifies the system function, structure, and behavior within one frame of reference.
- We examine:
 - The **types** of supported events
 - How these event types are **integrated** into complete system specifications
 - What are the potential **implications on the code** derived from each method.

Comparison by Event Types 1/2

Statecharts

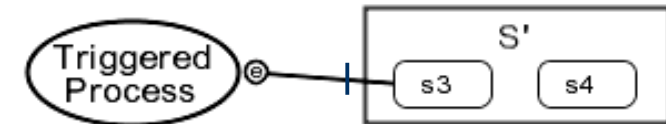
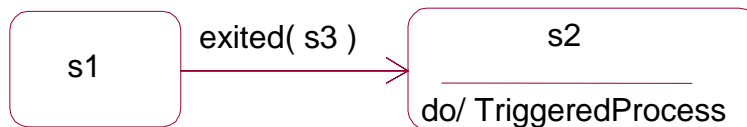
OPM

State Entrance



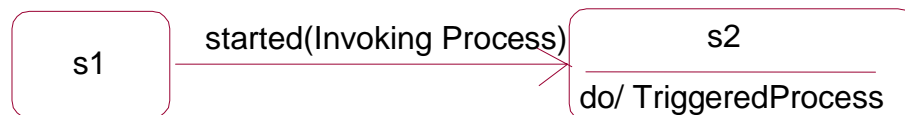
S' triggers Triggered Process when it enters s3.

State Exit



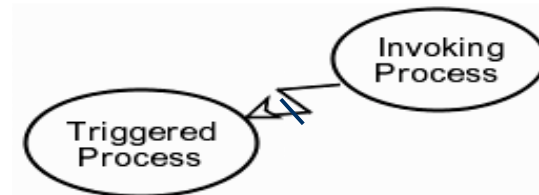
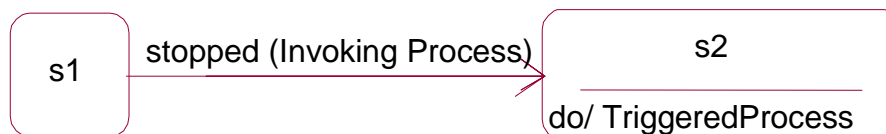
S' triggers Triggered Process when it exits s3.

Activity Start



Invoking Process triggers Triggered Process when it starts.

Activity Stop



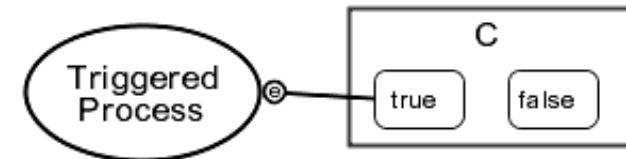
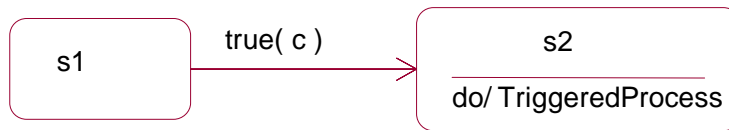
Invoking Process triggers Triggered Process when it stops.

Comparison by Event Types 2/2

Statecharts

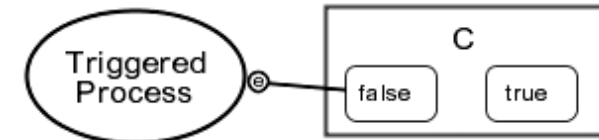
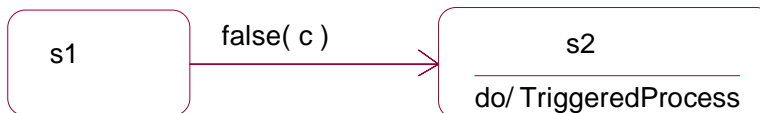
OPM

Condition
Fulfillment



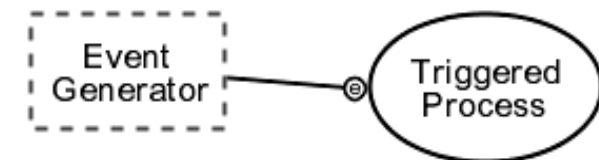
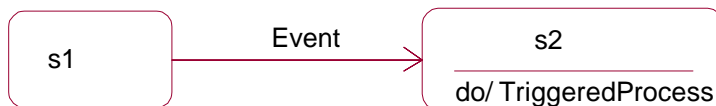
C triggers **Triggered Process** when it enters **true**.

Condition
Violation



C triggers **Triggered Process** when it enters **false**.

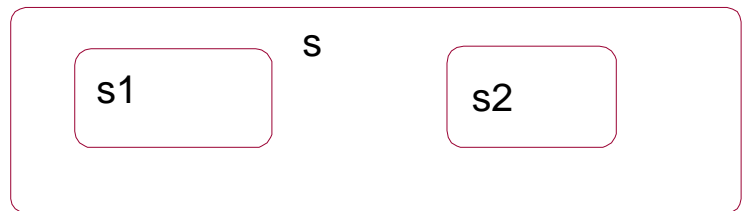
External
Event



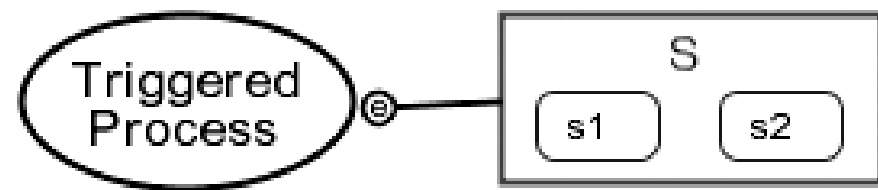
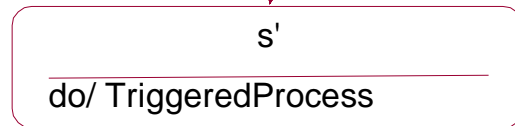
Event Generator, which is environmental, triggers **Triggered Process**.

State Change Event

State
Change



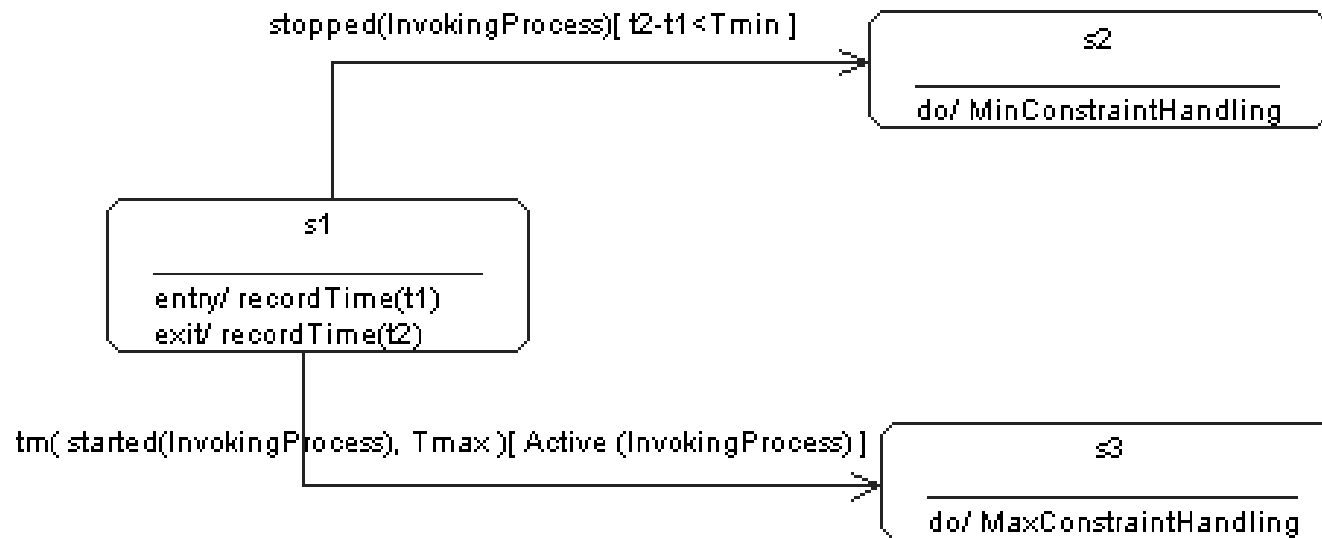
entered (s1) or entered (s2)



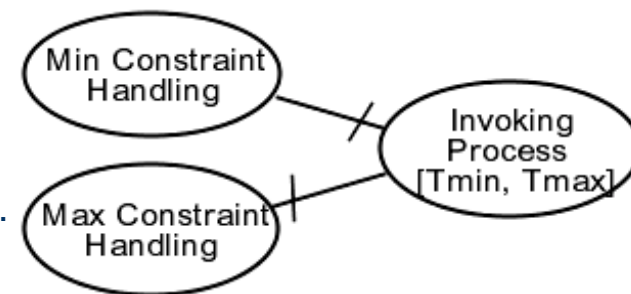
S triggers **Triggered Process** when its state changes.

Activity Timeout Event

Activity Timeout

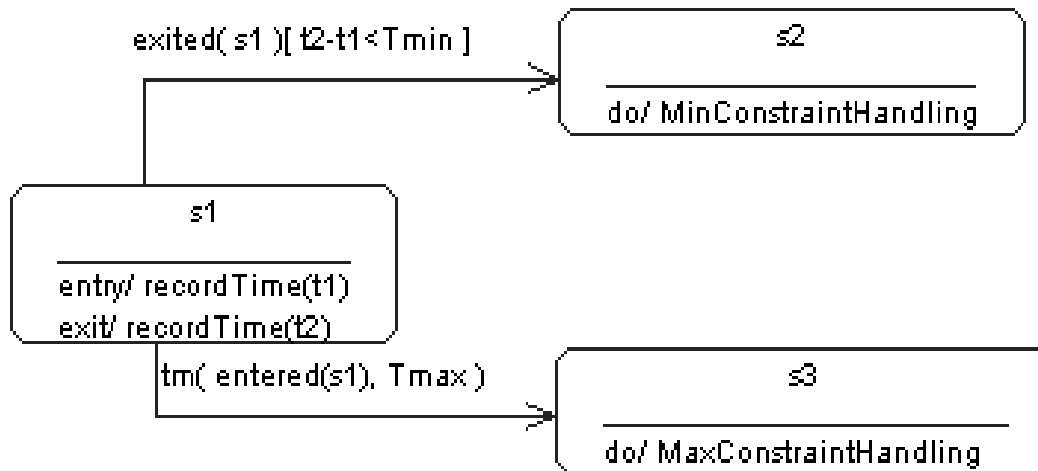


Invoking Process triggers **Min Constraint Handling** when it lasts less than **Tmin** and **Max Constraint Handling** when it lasts more than **Tmax**.

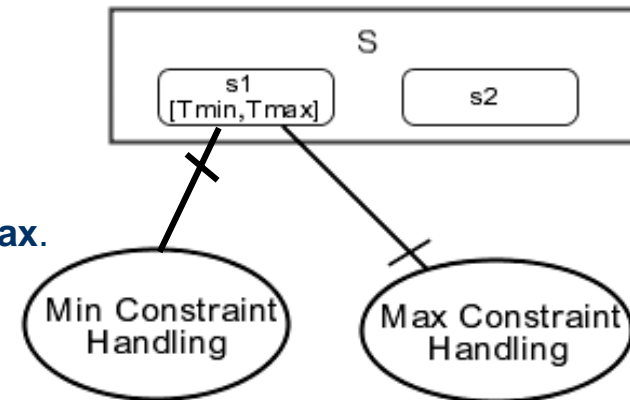


State Timeout Event

State Timeout

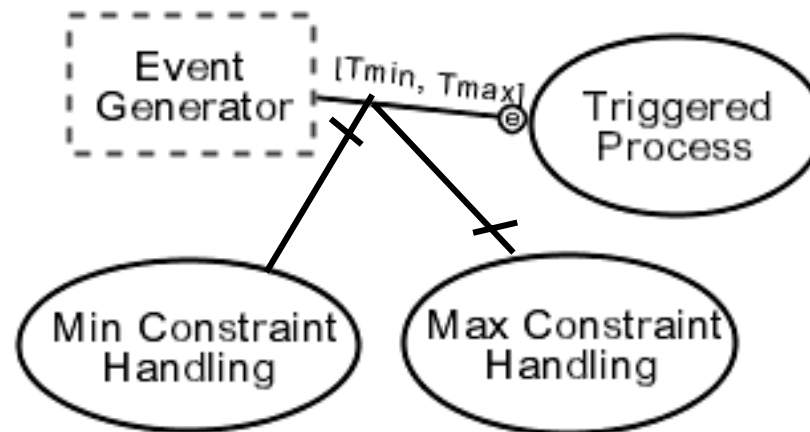


S triggers **Min Constraint Handling** when **s1** lasts less than **Tmin** and **Max Constraint Handling** when **s1** lasts more than **Tmax**.



Reaction Timeout Event

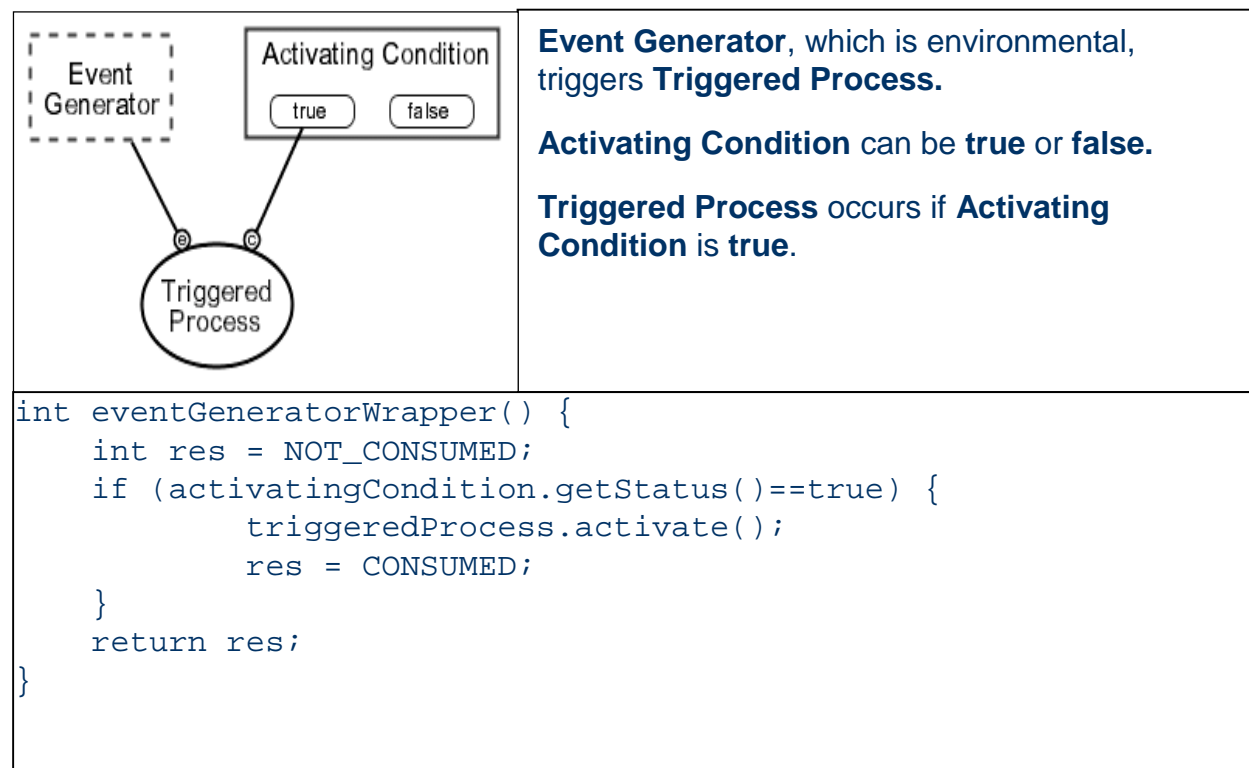
Reaction Timeout



Event Generator, which is environmental, triggers **Triggered Process** with a reaction time of **Tmin** to **Tmax**. If **Triggered Process** starts within less than **Tmin** then **Min Constraint Handling** is triggered. If **Triggered Process** does not start within **Tmax** then **Min Constraint Handling** is triggered.

Event Implementation

- Similar concepts of Statecharts and OPM.



Methods Comparison 1/2

- Where Does the Behavior Occur?
 - In **Statecharts** the behavior occurs **in the states**.
 - In **OPM** the behavior is executed **in the processes**, which act to change the states of objects.
- State-Activity Coupling
 - In Statecharts there is a clear coupling between a **state of an object** and the **activity** performed within it, so it is easy to detect the system behavior within a single state.
 - In OPM, system state may be a combination of several objects' states, making it more difficult to detect the the system behavior within a single state.
 - Since in Statecharts each object gets its own “chart” it is not clear how the entire system state is modeled.

Methods Comparison 2/2

- **Event Source**

- In Statecharts it is difficult to follow the event sources.
- In OPM, the event generating source is explicitly specified, enabling the designer to trace the events associated with a specific entity (object, state, or process).

- **Accessibility, Legibility, Expressivity**

- In OPM, OPDs use less text and more graphics, and the semantics is made clear by the corresponding natural OPL sentence, an element that does not exist in Statecharts

Summary

- OPM's visual and textual representations provide a means for expressing various event types in a formal yet intuitive way, which compares favorably with Statecharts.
- OPM supports modeling aspects (other than behavior) which are not modeled in the stand-alone version of Statecharts (e.g., communication between objects, system structure, and architecture).