Overview

• Recap of Harel statecharts
• State machines
• New modeling constructs in UML2.0 state machines
Harel state charts

- State diagrams

Figure 1
Harel state charts (2)

- State diagrams
- Depth

Figure 2
Harel state charts (3)

- State diagrams
- Depth
- Orthogonality

Figure 3
Harel state charts (4)

- State diagrams
- Depth
- Orthogonality
- Broadcast
Harel state charts (5)

- State diagrams
- Depth
- Orthogonality
- Broadcast
- History

Figure 5
State machines

• A behavioral formalism

Figure 6 from [3]
State machines (2)

• Behavior state machines
  – Express behavior of various model elements (e.g., class instances)
  – Object-based variant of Harel state charts

• Protocol state machines
  – Express usage protocols
  – A convenient way to define a lifecycle for objects, or an order of the invocation of its operation
Abstract syntax

Figure 7 from [1]
New modeling constructs

- State machine extension
- Modularized state machine
- State machine termination
- Protocol state machines
State machine extension

- A state machine is generalizable. A specialized state machine is an extension of the general state machine.
- In a specialized state machine, all elements of the general state machine will be inherited, regions, vertices and transitions can be added/redefined.
- Rationale: Corresponds to classifier generalization. “It should be possible to define the redefined behavior of a special classifier as an extension of the behavior of the general classifier.” [1]
State machine extension (2)

Figure 8 from [3]
State machine extension (3)

- The general state machine for ATM
- The specialized state machine for FlexibleATM

Figure 9 from [3]

Figure 10
Modular Submachines

- Defines a state machine for the submachine.
- Adds entry/exit points (i.e. pseudostates).

Figure 11 from [3]
Modular Submachines (2)

- Inserts the submachine into a state machine as a submachine state.
State machine termination

• When the final state is entered, its containing region is completed.
• If the region is contained in a state machine and all other regions contained in the state machine are also completed, the entire state machine terminates, which implies the termination of the context object of the state machine.
Protocol state machines

- Abstract syntax
Protocol state machines (2)

- **Interface**
  - Specifies conformance characteristics.
  - May own a protocol state machine to specify the legal sequence of the invocation of the behavior features described in the interface.

- **Port**
  - Represents an interaction point between a classifier instance and its environment or between a classifier instance and instances it may contain.
  - Provided/required interfaces
  - Encapsulation

---

Figure 14 from [3]
More Usages on Ports

- Connector: Model the communication channel

- Models a structured class that contains parts and connectors

Figure 15 from [3]
Protocol state machines (3)

- **Protocol state machines**
  - Specify which operations of the classifier can be called in which state and under which condition, thus specifying the allowed call sequences on the classifier’s operations.
  - Provide a global view of the classifier protocol usage.

- **Protocol transition**
  - Pre/post condition and trigger.
  - No effect action.

Figure 16 from [1]
Protocol state machines (4)

- ProtocolConformance
  Used in the protocol state machine redefinition. Protocol state machines can be redefined into more specific protocol state machines, or into behavior state machines.
  - specificMachine:StateMachine
    Specifies the state machine that conforms to the general protocol state machine.
  - generalMachine:ProtocolStateMachine
    Specifies the protocol state machine to which the general state machine conforms.
References