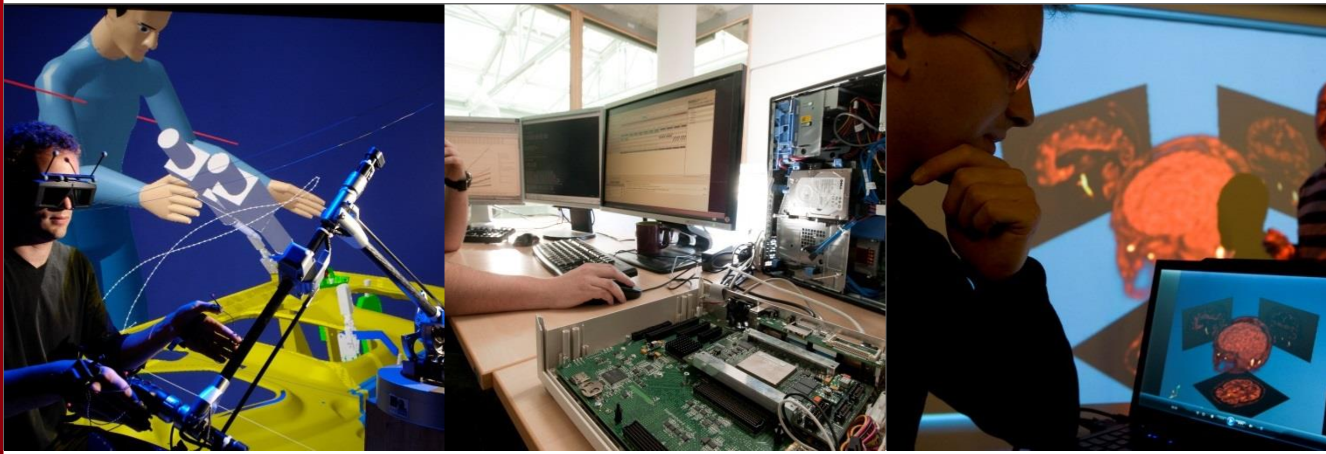
FROM RESEARCH TO INDUSTRY

cea tech

[PSSM] – WEBEX DECEMBER 04TH

Jérémie TATIBOUET (CEA LIST)
Arnaud CUCCURU (CEA LIST)



list

INSTITUT CARNOT CEA LIST

digiteo

SUMMARY

Contributions from last Webex

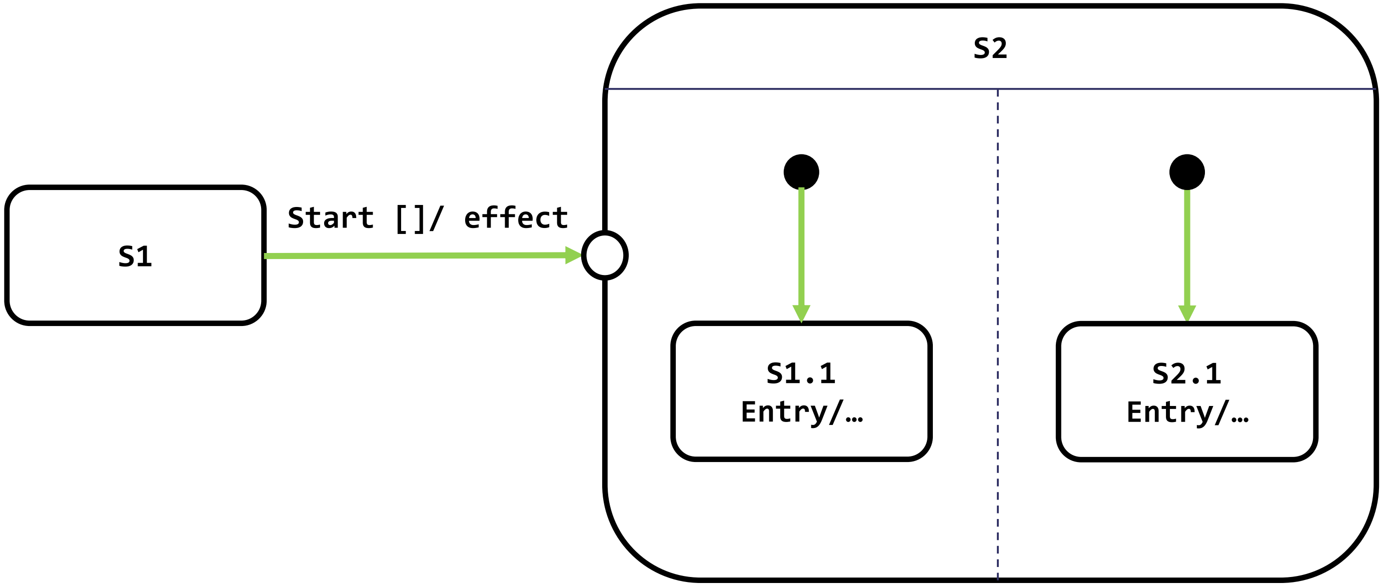
- PSSM repository
  - Presentations realized during Boston meeting
  - Initial semantic model version (based on normative versions of fUML and PSCS semantic models – aligned with fUML 1.2)
- Prototype implementation
  - Handling of explicit of regions
  - Improve EntryPoint and ExitPoint support
- Test suite
  - Test for requirement “Exiting003”
  - Test for requirement “Exiting005”
  - Test for requirement “Event018”
  - Test for requirement “Event015”
  - Test for requirement “Exit002”
  - Test for requirement “Entry002”
  - Test for requirement “Entry002-bis”
- Progress on the “grant of rights”

ENTRY POINTS – 1 [IMPLEMENTED]

Entry point – Case 1

- An entry point without any outgoing transition that is placed on the edge of a composite state lead the state to perform a default entry.
  - Entry behavior is executed if any
  - All regions perform a default entry (i.e., start by Initial state)

- Example:



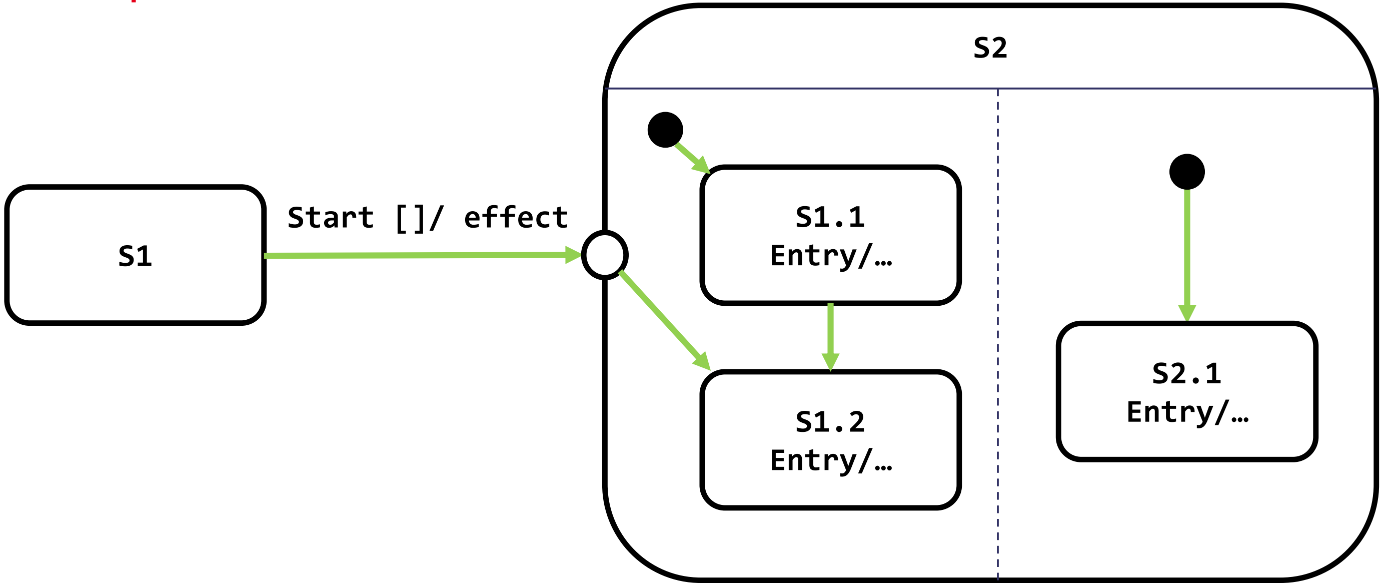
- Reference in UML 2.5: p. 372 of the PDF

ENTRY POINTS – 2 [IMPLEMENTED]

Entry point – Case 2

- An entry point placed on the edge of a composite state with orthogonal regions can have outgoing transitions targeting internal vertices.
  - The entry behavior of the state is executed if any
  - Regions in which a state is targeted by the a transition originating from an entry point performs an explicit entry.

- Example:



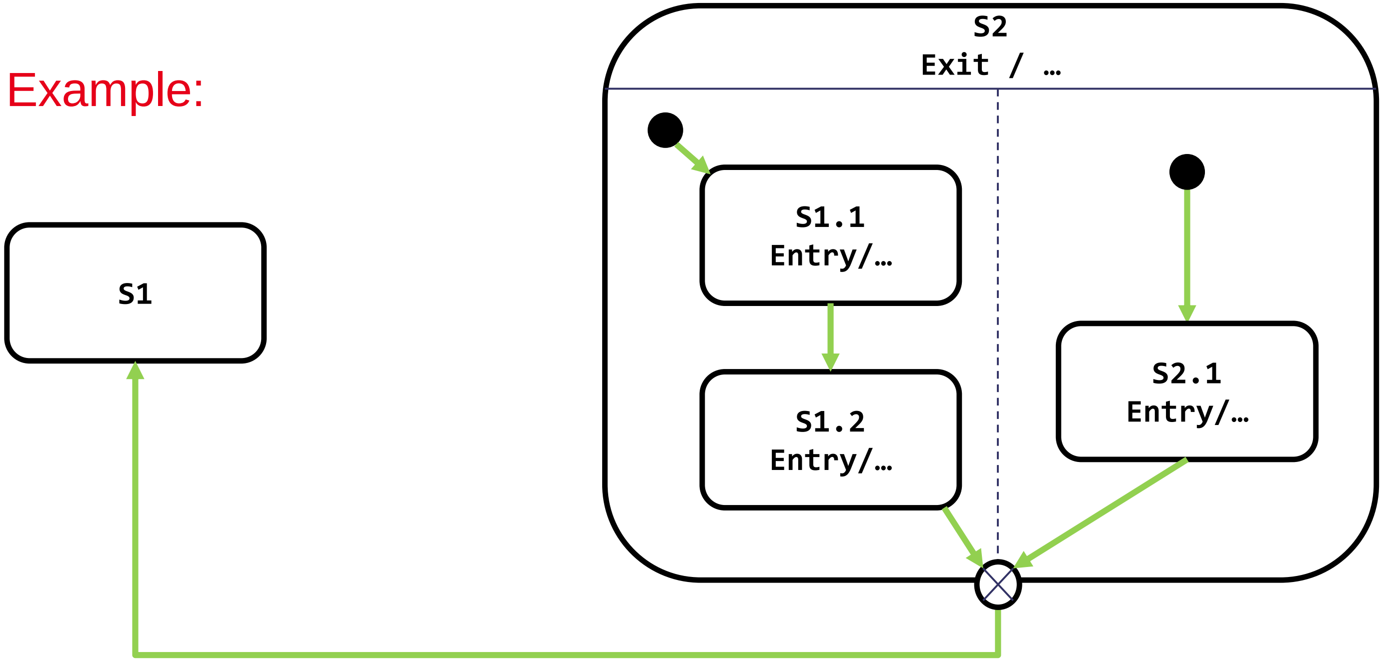
- Reference in UML 2.5: p.369 of the PDF

EXIT POINTS – 1 [IMPLEMENTED]

Exit point – Case 1

- An exit point that is placed on a composite state can be the target of transitions originating from states placed in orthogonal regions
  - The exit point should act like a join (i.e., every incoming transition should have been fired to be able to arrive in S1).

- Example:



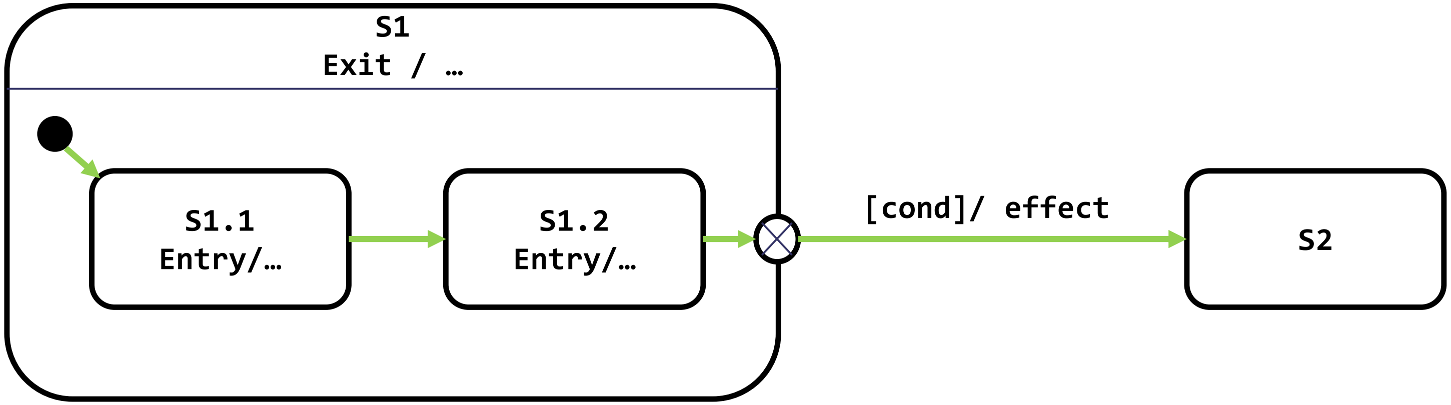
- Reference in UML 2.5: p. 372 of the PDF

ENTRY AND EXIT PSEUDO STATES

EVENTS AND GUARDS [NOT IMPLEMENTED]

A. UML does not place constraints over transitions originating from entry and exit points

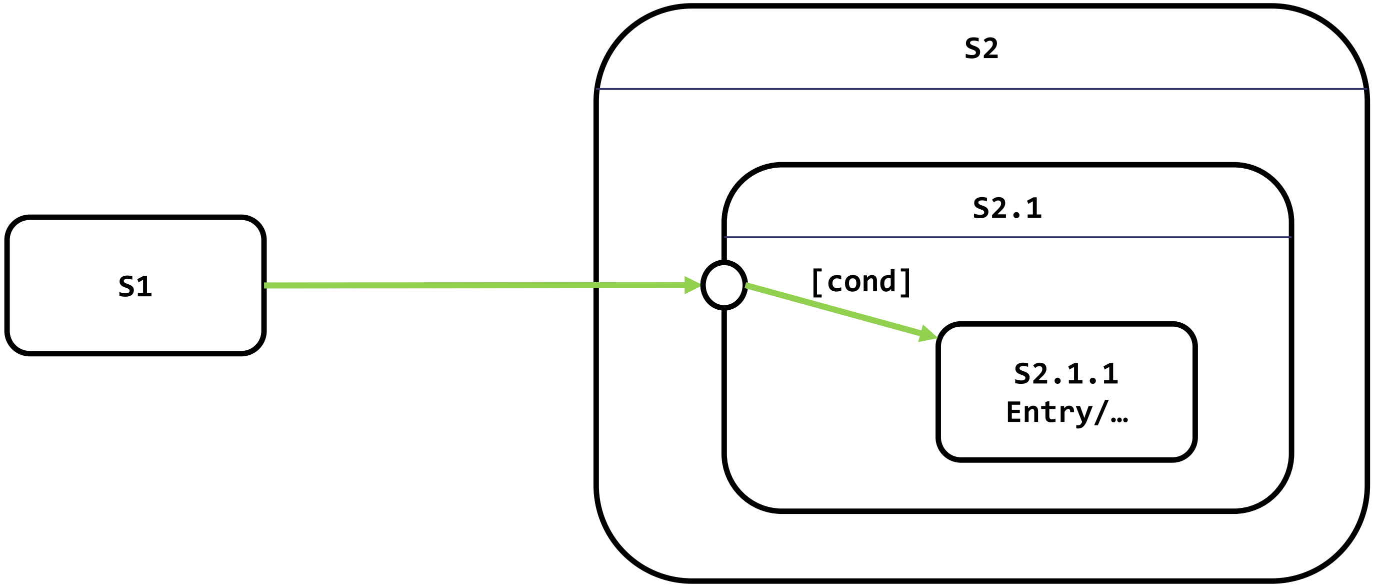
- They can have guards
- They can have triggers
- They can have an effect



- What should occur in the following situation ?
  - S1 remains in the active configuration when the exit point is entered
  - S1 is exited when the exit point is itself exited. This occurs when the transition originating from the exit point fires.

ENTRY AND EXIT PSEUDO STATES

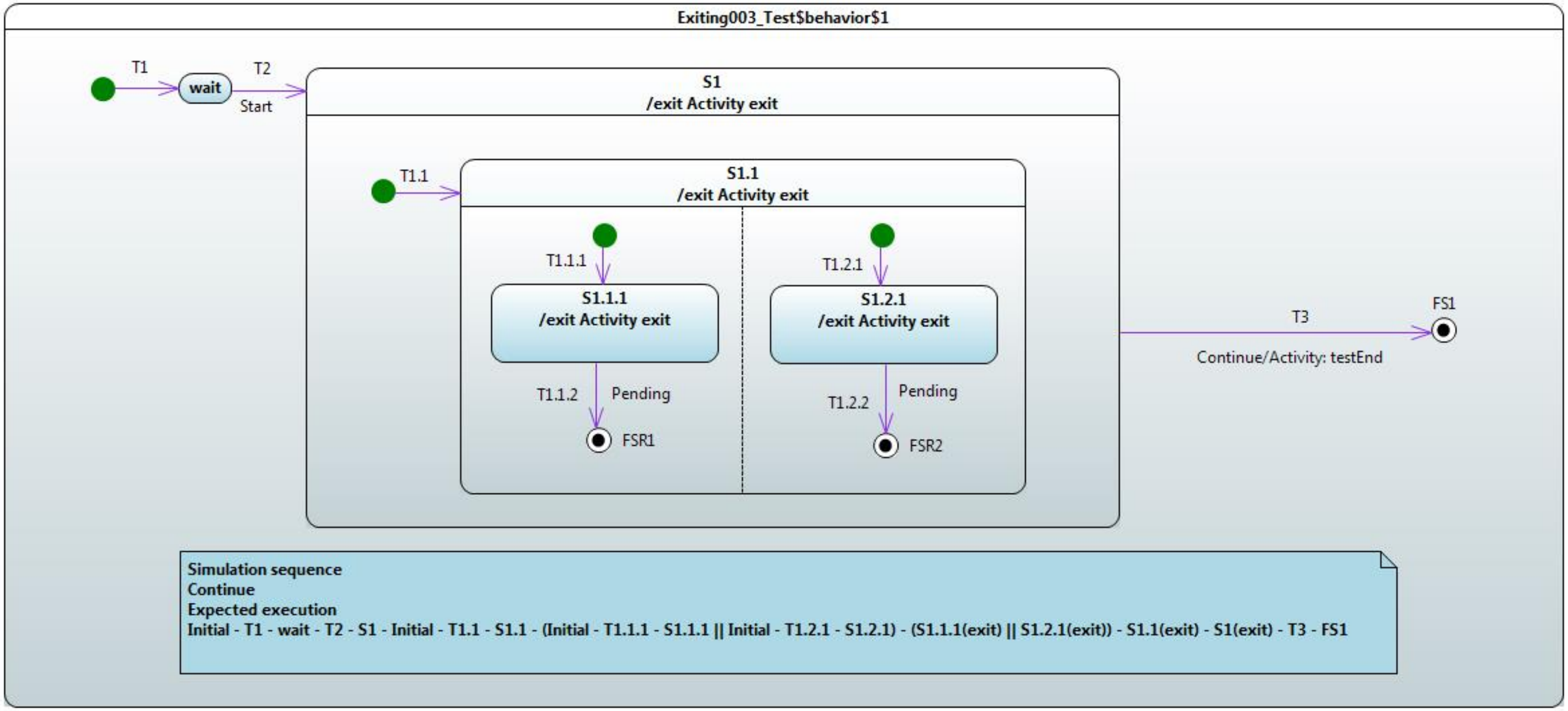
EVENTS AND GUARDS [NOT IMPLEMENTED]



- What should occur in the following situation ?
  - S2 is entered
  - S2.1 is entered
  - S2.1.1 is entered only when Start is received

TEST CASE – EXITING 003 [IMPLEMENTED]

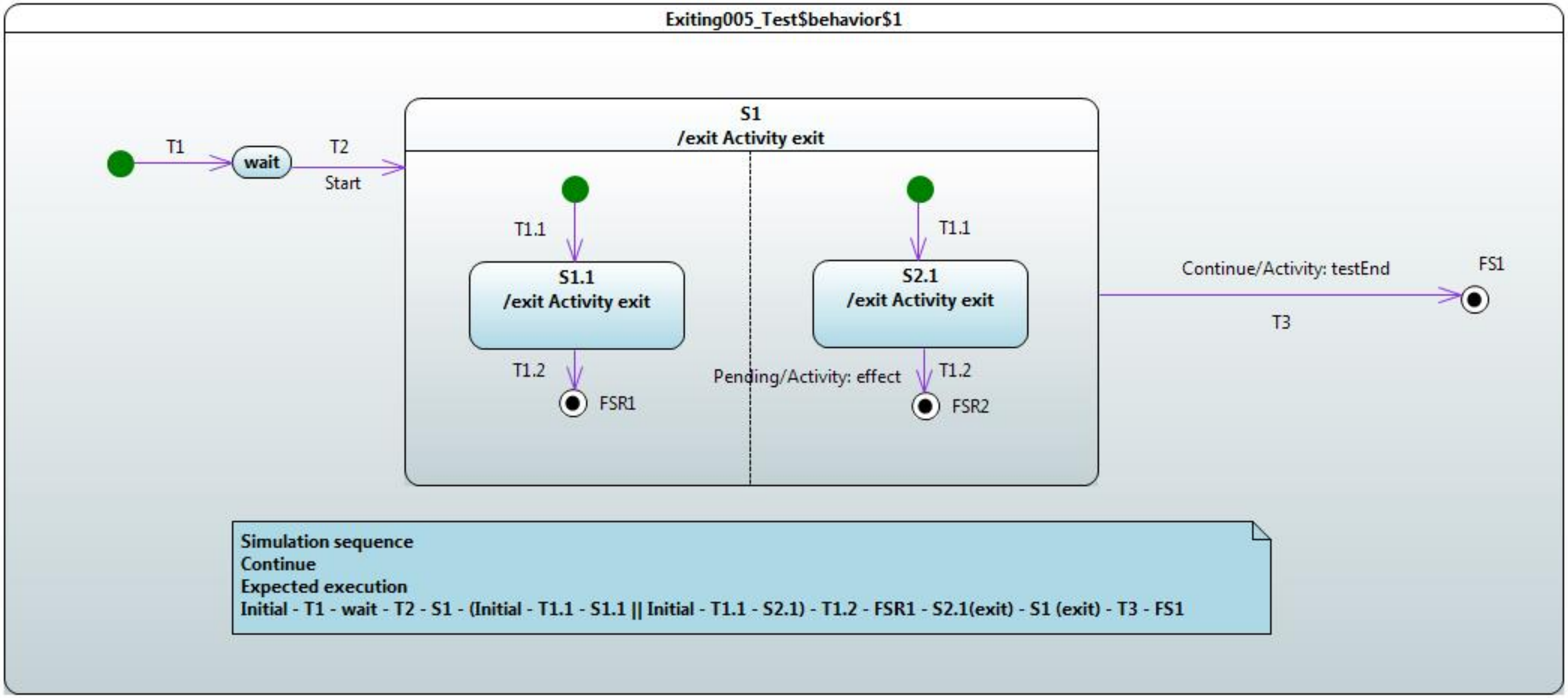
When exiting from a composite State, exit commences with the innermost State in the active state configuration. This means that exit Behaviors are executed in sequence starting with the innermost active State. (p.324)



Simulation sequence
Continue
Expected execution
Initial - T1 - wait - T2 - S1 - Initial - T1.1 - S1.1 - (Initial - T1.1.1 - S1.1.1 || Initial - T1.2.1 - S1.2.1) - (S1.1.1(exit) || S1.2.1(exit)) - S1.1(exit) - S1(exit) - T3 - FS1

TEST CASE – EXITING 005 [IMPLEMENTED]

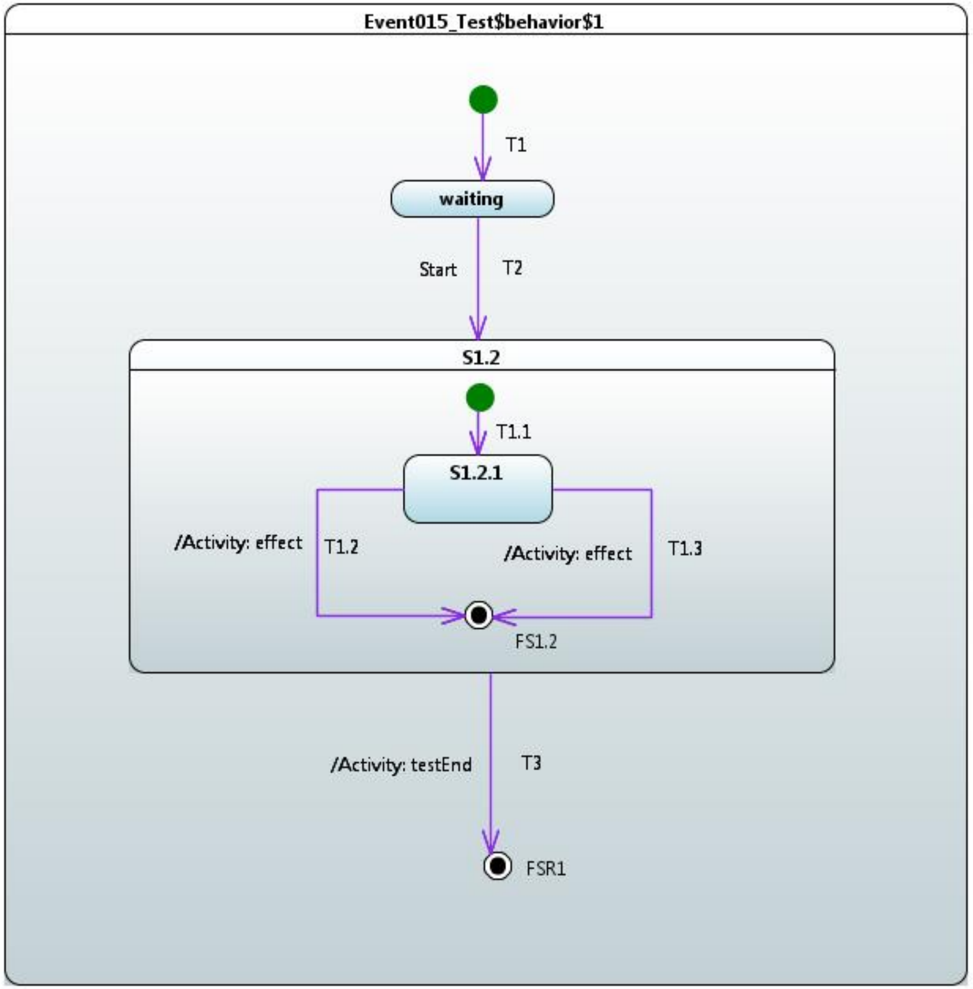
```
When exiting from an orthogonal State, each of its Regions is exited.
After that, the exit Behavior of the State is executed. (p.324)
```



Simulation sequence
Continue
Expected execution
Initial - T1 - wait - T2 - S1 - (Initial - T1.1 - S1.1 || Initial - T1.1 - S2.1) - T1.2 - FSR1 - S2.1(exit) - S1 (exit) - T3 - FS1

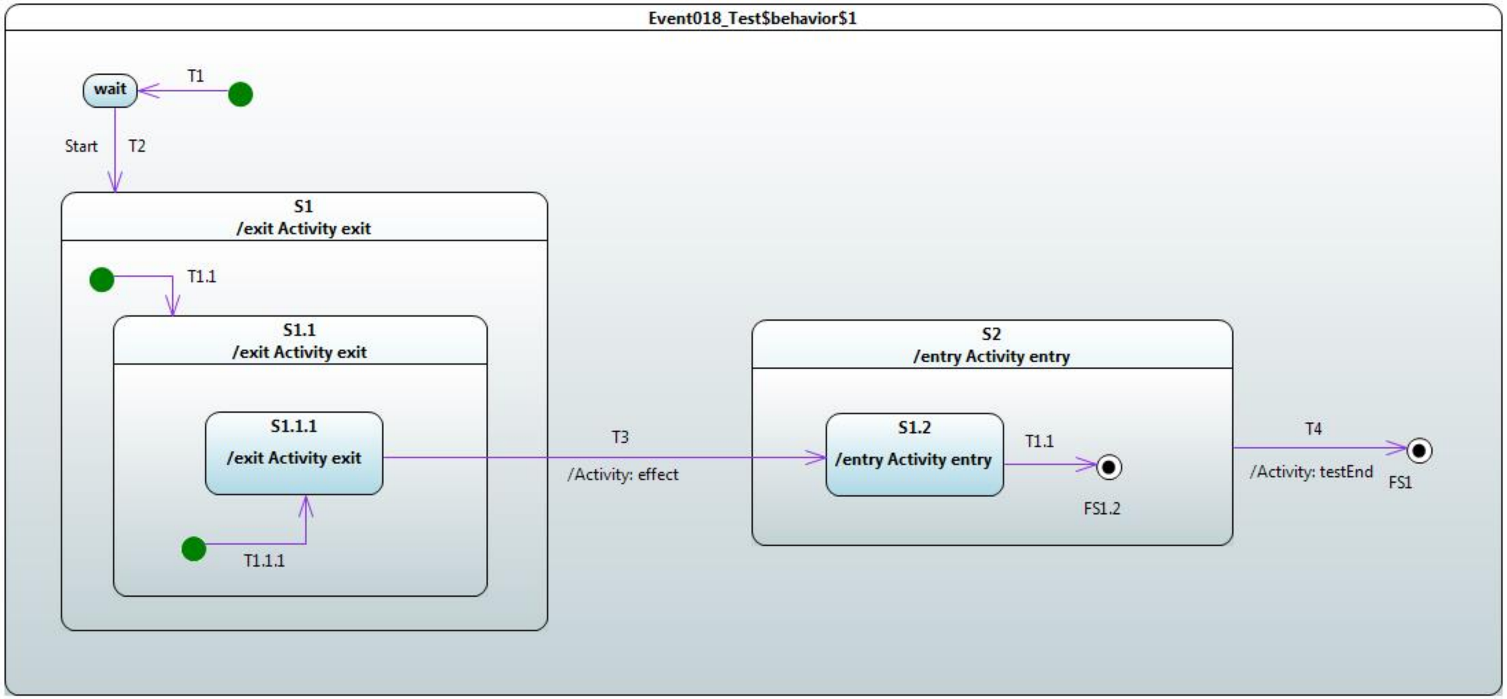
TEST CASE – EVENT015 [IMPLEMENTED]

It is possible for more than one Transition to be enabled within a StateMachine. If that happens, then such Transitions may be in conflict with each other. For example, consider the case of two Transitions originating from the same State, triggered by the same event, but with different guards. If that event occurs and both guard conditions are true, then at most one of those Transition can fire in a given run-to-completion step. (p.331)



TEST CASE – EVENT015 [IMPLEMENTED]

Once a Transition is enabled and is selected to fire, the following steps are carried out in order: 1. Starting with the main source State, the States that contain the main source State are exited according to the rules of State exit (or, composite State exit if the main source State is nested) as described earlier. 2. The series of State exits continues until the first Region that contains, directly or indirectly, both the main source and main target states is reached. The Region that contains both the main source and main target states is called their least common ancestor. At that point, the effect Behavior of the Transition that connects the sub-configuration of source States to the sub-configuration of target States is executed. (A "sub-configuration" here refers to that subset of a full state configuration contained within the least common ancestor Region.) 3. The configuration of States containing the main target State is entered, starting with the outermost State in the least common ancestor Region that contains the main target State. The execution of Behaviors follows the rules of State entry (or composite State entry) described earlier. (p.331)


Event018_Test$behavior$1
wait
T1
Start
T2
S1
/exit Activity exit
T1.1
S1.1
/exit Activity exit
S1.1.1
/exit Activity exit
T1.1.1
T3
/Activity: effect
S2
/entry Activity entry
S1.2
/entry Activity entry
T1.1
FS1.2
T4
/Activity: testEnd
FS1