

For Updated version, please click on this
<http://ocw.ump.edu.my>

BCS3323 – Software Testing and Maintenance

Test Case Design Black Box

Editors

Dr. AbdulRahman A. Alsewari
Faculty of Computer Systems & Software
Engineering
alswari@ump.edu.my

Decision Testing & State Transition Testing

Aims is students be able to discover

- The purpose of decision table testing

- The purpose of state transition testing

- How to use these techniques to design the test cases

Expected Outcomes

- Students be able to show how to design the test cases based on these techniques

- Students be able to show when to use these techniques .

References

- ISTQB

- MSTB/GTB

- <http://www.softwaretestingclass.com/software-testing-tools-list/>

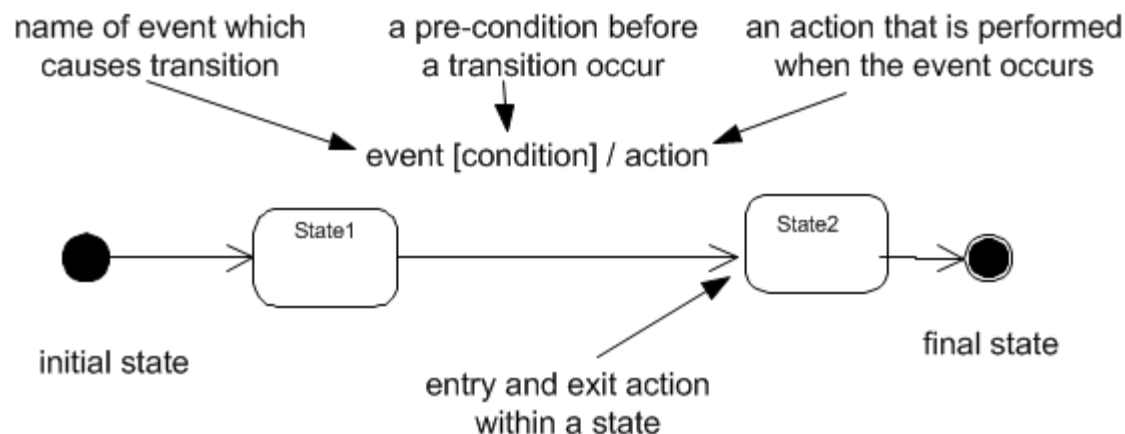
- http://www.softwaretestinggenius.com/articalDetails.php?qry=572#comment_sList

State Based Testing



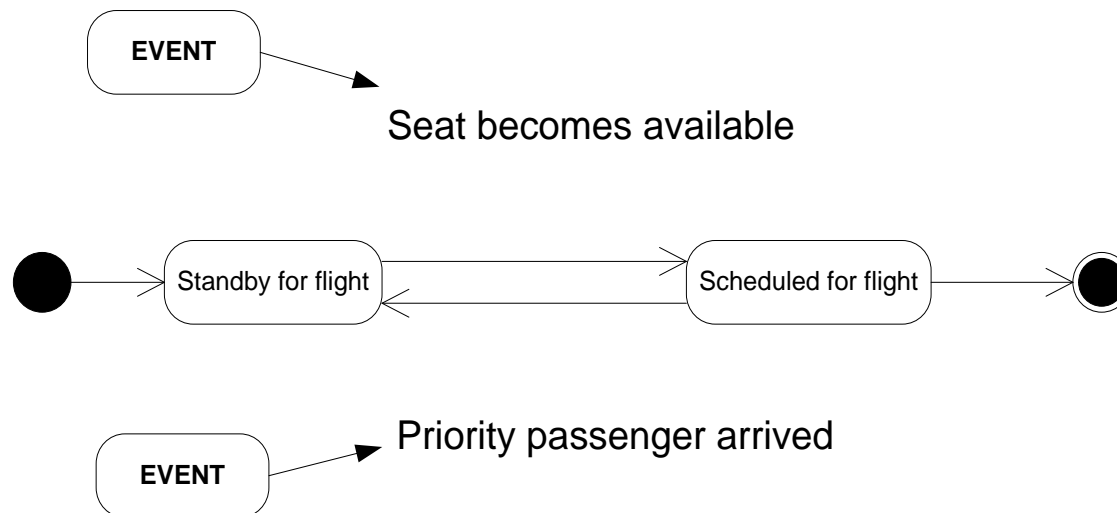
State transition diagram

- A state transition diagram (or *state charts*) is used to show the sequence events and actions in life history of a given class, the events that cause a transition from one state to another, and the actions that result a state to change.
- State space of a given class is the enumeration of all the possible states of an object
- The state of an object is one of the possible conditions in which an object may exist
- In a given state, an object always react in the same manner
- Note: nested states are not covered in this course



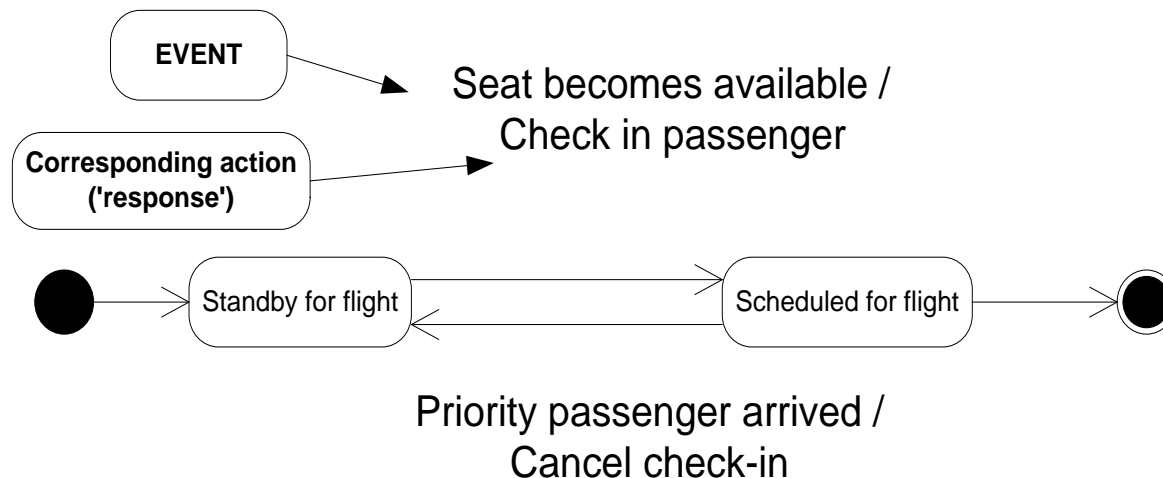
Events

- Event causes a state change
- Here no actions are defined as yet
- An event is sometimes called a stimulus or a trigger
- Example: State transition of an airline passenger system



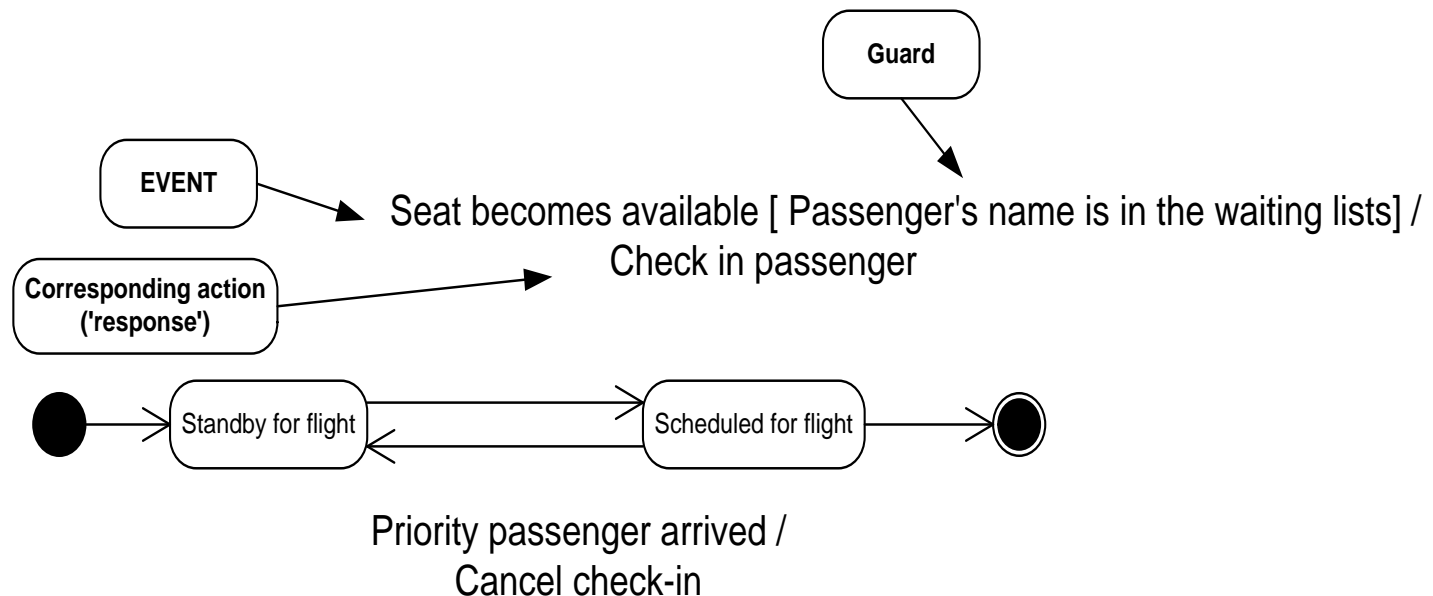
Response to events

- State change as before
- Here, there are corresponding actions
- Observe the “/” is used to separate events from actions



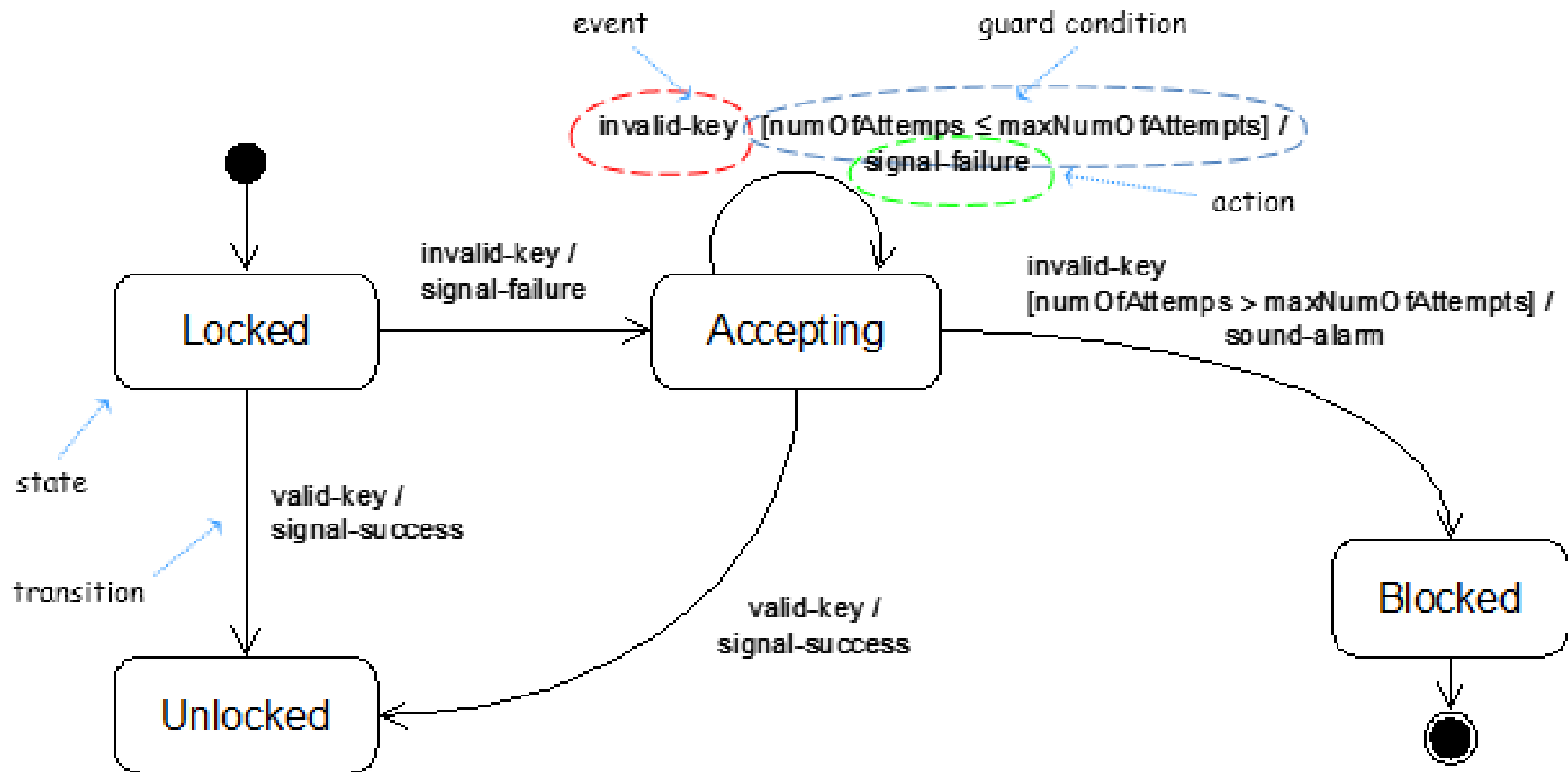
Guards

- Guard is a boolean condition that must be true for the transition to happen



State-Based Testing

State-based testing defines a set of abstract states that a software unit can take and tests the unit's behavior by comparing its actual states to the expected states



Consistency Check vs Transition

- Every state is reachable
- Every state must have one incoming and one outgoing state
- No equivalent state
- All define event and action appear in at least one transition
- Missing / incorrect transitions
- Missing/ incorrect events
- Corrupt state
- Sneak path (an event is accepted when it should not be)
- Transition with undefined events