

State-Transition Testing

Anuruddha Jayasekara Pathirana

02/27/2015

STATE-TRANSITION TESTING2

State-Transition Testing

State-Transition diagrams are excellent tools to capture certain types of system requirements and to document internal system design. State Transition testing, a black box testing technique, in which outputs are triggered by changes to the input conditions or changes to 'state' of the system. This is based on finite state machine(Chow, 1978). In general, a state machine is any device that stores the status of something at a given time and can operate on input to change the status and/or cause an action or output to take place for any given change. A computer is basically a state machine and each machine instruction is input that changes one or more states and may cause other actions to take place. In other words, tests are designed to execute valid and invalid state transitions(tutorialspoint.com, n.d.).

Example of State-Transition Testing

We have a Bank Account login screen, consider user need to enter correct user name, password and security question in the login screen to access the Account application at first attempt. Figure 01 provides full details about state transition.

STATE-TRANSITION TESTING3

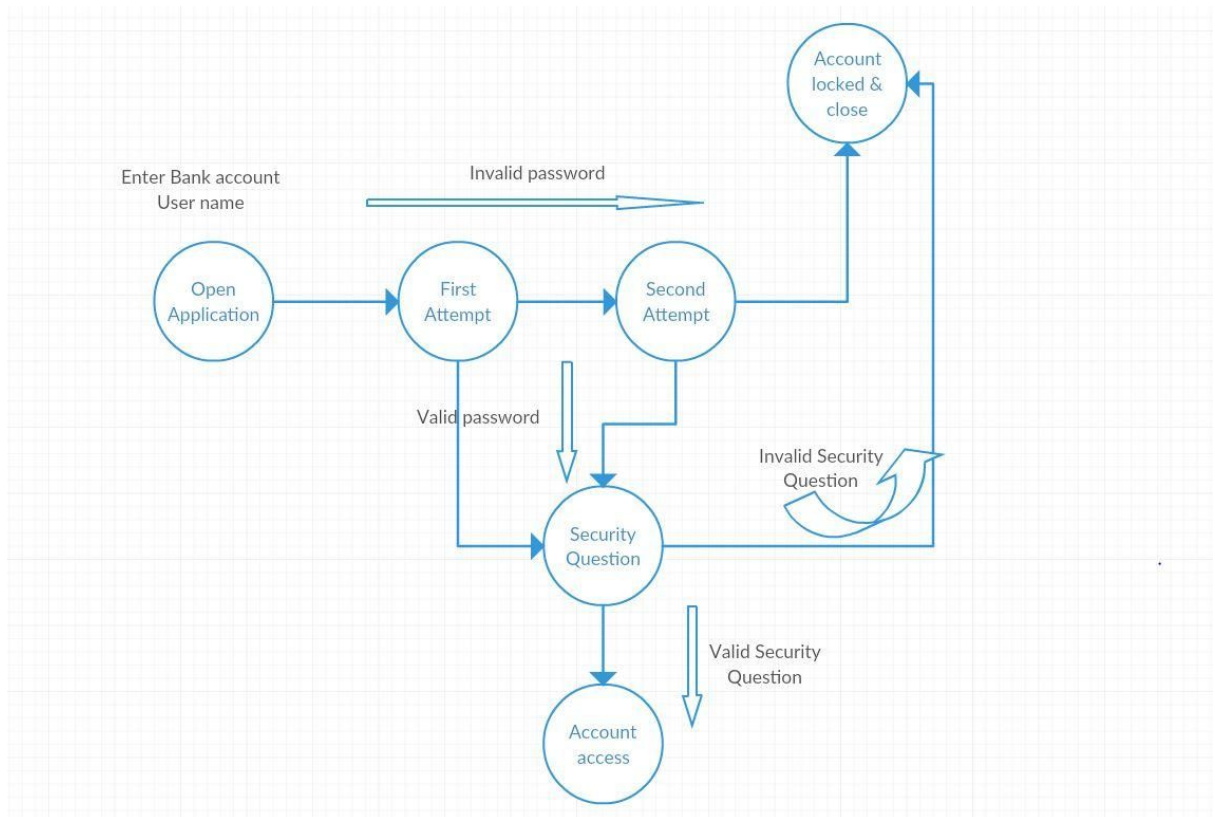


Figure 01: State Transition Diagram of Bank Login system

The system gives user the access to the application with correct password and login name, but what if user enters the wrong password.

How system decides how many login attempts should be allowed to users for the login screen with the wrong password before it finally closes the application.

In this login application, 2 attempts are allowed, and if users enter the wrong password at 4th attempt or wrong security question the system will be locked and closed the application automatically.

So, testing for scenario 1 with the correct password, the 2nd scenario with an incorrect password, 3rd correct security question and 4th incorrect security question is compulsory. For that

STATE-TRANSITION TESTING4

we use state transition testing. The tests are derived from the above state and transition and below are the possible scenarios that need to be tested.

		Valid Input	Invalid Input
State 01	Open Application	State 04	State 02
State 02	First Attempt	State 04	State 03
State 03	Second Attempt	State 04	State 06
State 04	Security Question	State 05	State 06
State 05	Account access		
State 06	Account locked & close		

Table 01. State Transition Table

We can use State Table to determine invalid system transitions. In a state Table, all the valid states are listed on the left side of the table, and the events that cause them on the top. Each cell represents the state system will move to when the corresponding event occurs. For example, while in state 01 user enter a correct password and the user is taken to state 04. Then if user entered correct answer for security question, user can access his/her account. Suppose if user has entered the wrong password at first attempt the user will be taken to state 03. Likewise, user can determine all other states.

When State-Transition Testing should be used

- When we have sequence of events that occur and associated conditions that apply to those events

STATE-TRANSITION TESTING5

- When the proper handling of a particular event depends on the events and conditions that have occurred in the past(“What is State transition testing in software testing?,” n.d.)

Types of software testing problems that State-Transition testing addresses

The State transition testing is a form of **Dynamic Testing Technique** that comes in use when the system explained as a finite number of states and the evolutions between the states is ruled by the rules of the system. Another use of this technique is when features of a system are characterized as states that converts to other state, this transition is explained by the method of the software

Key assumptions of State-Transition testing

- State-Transition testing is used for real time systems with various states and transitions involved.

Limitations of State-Transition testing

- 1) State transition diagrams are not applicable when the system has no state or does not need to respond to real-time events from outside of the system.
- 2) It needs to define all possible states of a system. While it is alright with small systems, it is not easy to perform in large systems as there is an exponential growth in the number of states. This state explosion problem leads to state transition diagrams becoming far too complex for much practical use.
- 3) Limited impact on business rules and test conditions.

STATE-TRANSITION TESTING6

- 4) Requires UML to be included in design phase(Ilgun, Kemmerer, & Porras, 1995).

References

- Chow, T. S. (1978). Testing software design modeled by finite-state machines. *IEEE Transactions on Software Engineering*, (3), 178–187.
- Ilgun, K., Kemmerer, R. A., & Porras, P. A. (1995). State transition analysis: A rule-based intrusion detection approach. *IEEE Transactions on Software Engineering*, 21(3), 181–199.
- tutorialspoint.com. (n.d.). State Transition Testing. Retrieved February 27, 2017, from https://www.tutorialspoint.com/software_testing_dictionary/state_transition.htm
- What is State transition testing in software testing? (n.d.). Retrieved February 27, 2017, from <http://istqbexamcertification.com/what-is-state-transition-testing-in-software-testing/>