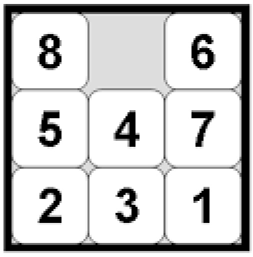
FSM

Stare

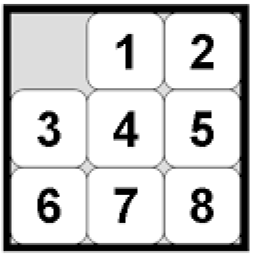
Starea unui program este data de valorile variablilelor la un moment dat

Starea unui obiect este data de valorile atributelor (variabilelor de instanta)

Exemple:



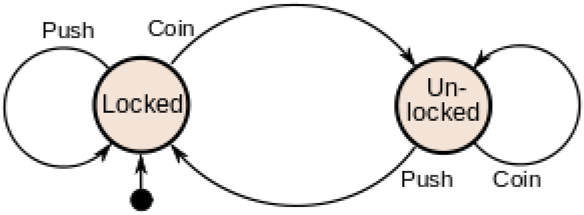
8		6
5	4	7
2	3	1



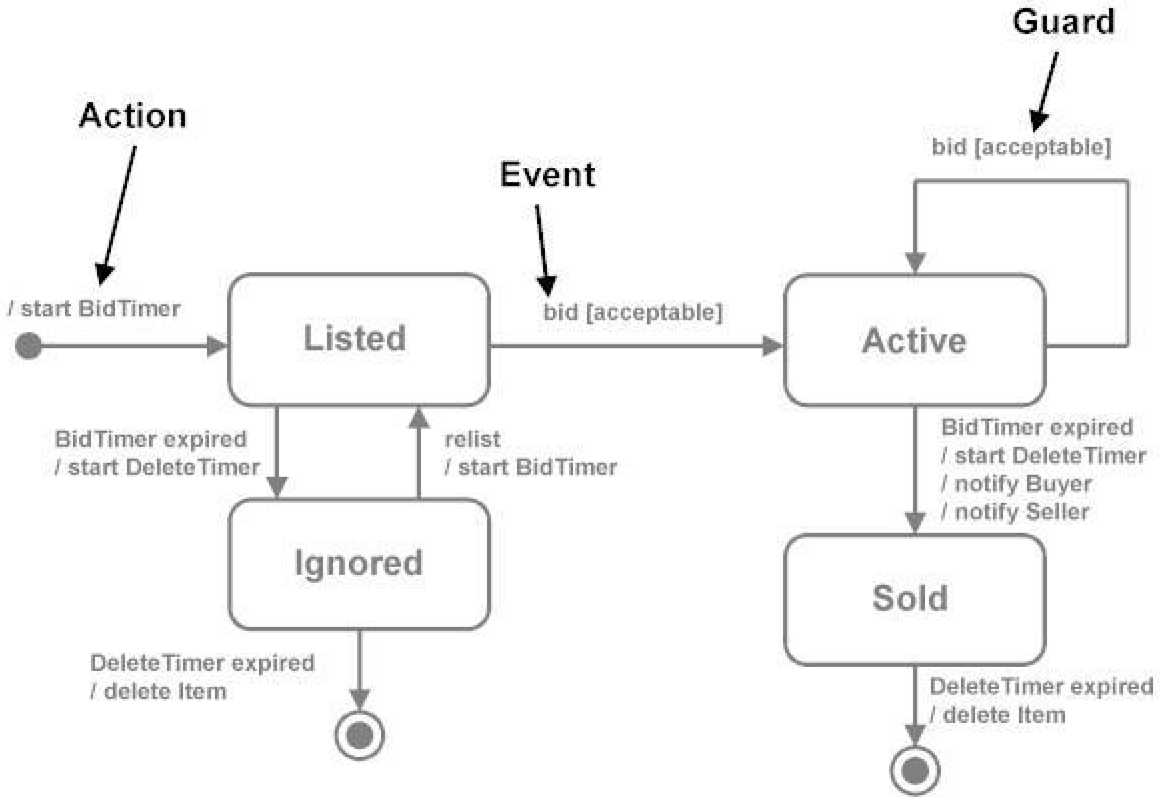
	1	2
3	4	5
6	7	8

Wikipedia:

A **finite-state machine** (**FSM**) or **finite-state automaton** (**FSA**, plural: *automata*), **finite automaton**, or simply a **state machine**, is a mathematical model of computation. It is an abstract machine that can be in exactly one of a finite number of *states* at any given time. The FSM can change from one state to another in response to some inputs; the change from one state to another is called a *transition*



The UML notation for state-transition diagrams is shown below:



Notation

For those not familiar with the notation used for state-transition diagrams, some explanation is in order.

State. A condition during the life of an object in which it satisfies some condition, performs some action, or waits for some event.

Event. An occurrence that may trigger a state transition. Event types include an explicit signal from outside the system, an invocation from inside the system, the passage of a designated period of time, or a designated condition becoming true.

Guard. A boolean expression which, if true, enables an event to cause a transition.

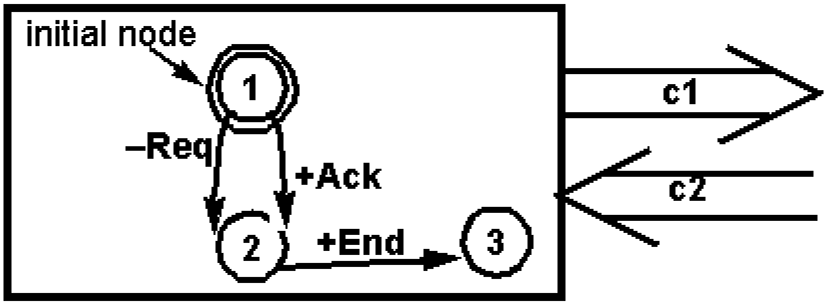
Transition. The change of state within an object.

Action. One or more actions taken by an object in response to a state change.

Communicating Finite State Machines (CFSM)

- Protocol is described as a set of Communicating Finite State Machines.
- Each CFSM represents a component (or process) of the network (in OSI term, a protocol entity, e.g. sender, receiver).
- Each CFSM is represented by a directed labelled graph where

--Nodes represent states (conditions) of the process;
--Edges represent transitions (events) of the process.



- Transitions include actions taken the process (e.g. the sending a message) or external stimuli (e.g. the reception of a message).

- The <u>sending message</u> transition is labelled as

-Msg where Msg is the type of messages being sent.

- The receiving message transition is labelled as +Msg where Msg is the head message on the incoming FIFO queue of the CFSM.

CFSM operating semantic

- The channels that connect CFSM's are assumed to be FIFO queues.

An error-prone channel is modelled as a CFSM.

- Initial node--starting state of a CFSM.

Final node--no transition.
Receiving node--all (outgoing) transitions are receiving transitions. If no message or incorrect msg in the channel, the node will be blocked.
Sending node--all transitions are sending transitions. They are not blocked.
Mix node--has both receiving and sending transition.

- Starting at the initial node, a CFSM traverses the nodes and transitions.

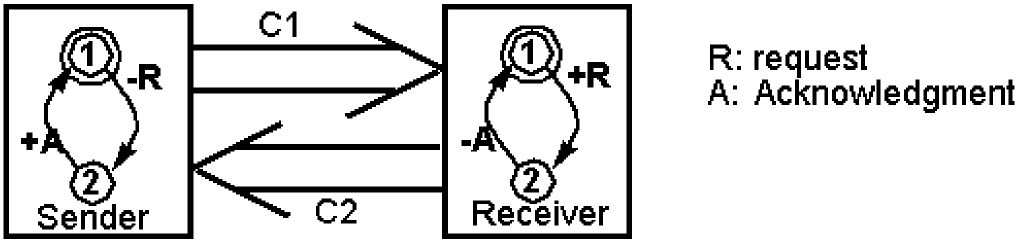
The node currently being visited is called the current node.

- When a machine traverses a sending transition, it sends/appends a message with the same label to its outgoing channel.
- A machine at a node cannot traverse its receiving transition unless there is a message matched with the same label on the head of its incoming channel.

- When a machine traverses a receiving transition, it removes the matched head message of its incoming channel.
- Among several possible transitions, a machine traverses one non-deterministically

Networks of CFSMs

- Example 1: Simple request-response protocol.


C1
1
-R
+A
2
Sender
1
+R
-A
2
Receiver
C2
R: request
A: Acknowledgment