**Lab 10:** **Draw State Diagram (System S1)**

**Lab Learning Outcomes (LLO)**

By completion of the lab the students should be able to

1. Create state diagrams to represent the even-driven behaviors of the Traffic Violation Monitoring System in response to various events.

**Tools Required**

For example, [StarUML](https://staruml.io/), [Lucidchart](https://www.lucidchart.com/pages/), [penpot](https://penpot.app/) and [Figma](https://www.figma.com/) etc

**Task to Do**

**Introduction:**

1. Briefly explain the concept of event-driven modeling and how it is used to represent the behavior of systems in response to events.
2. Introduce the Traffic Violation Monitoring System problem and its suggested solution as previously done.
3. Emphasize the importance of identifying different states and events in the system.

**1: Identifying States and Events**

1. Identify a list of possible states and events for the Traffic Violation Monitoring System.
2. Analyze the states and events and select the most relevant ones for the state diagram.
3. **Creating State Diagrams**
4. Based on the selected states and events, guide students through the process of creating state diagrams for different components of the system.
5. Focus on a specific aspect of the system, such as "Vehicle State," "Notification State," or "Penalty State."
6. Identify the initial state, final state, and transitions between states based on events.
7. Consider the triggers (events) that cause transitions, the conditions for transitions, and the actions taken during transitions.
8. Provide guidance and clarification as needed.

For your convenience, you may consider the following state and events. Please note that these may not be sufficient or complete, and you might come across new ones.

1. **List of Possible States and Events:**

**Possible States:**

* Idle,
* Detecting Traffic Data
* Analyzing Data
* Verifying Violation
* Notifying Driver
* Processing Payment

**Possible Events:**

* Traffic Data Received
* Data Analysis Complete
* Violation Detected
* Verification Requested
* Violation Verified
* Notification Sent
* Payment Processed

**2) Analysis and Selection of Relevant States and Events:** For the state diagram, let's focus on the "Violation State" and "Notification State." These states are critical to the system's operation and reflect its main functionalities.

**3) Creating State Diagrams:** Let's create state diagrams for the "Violation State" and "Notification State."

**State Diagram for Violation State:** (You will draw it yourself. This is just guidance for your understanding)

* Initial State: Idle, Final State: Violation Verified
* Transitions:
  + Traffic Data Received: Idle → Detecting Traffic Data
  + Data Analysis Complete: Detecting Traffic Data → Analyzing Data
  + Violation Detected: Analyzing Data → Verifying Violation
  + Verification Requested: Verifying Violation → Violation Verified

**State Diagram for Notification State:**

* Initial State: Idle, Final State: Notification Sent
* Transitions:
  + Violation Verified: Idle → Notifying Driver
  + Notification Sent: Notifying Driver → Notification Sent

**4) Focusing on Specific Aspects and Other Changing States:** For the "Violation State," we can consider the changing states of the "Violation" object, including attributes like "Severity Level" and "Verification Status."

**5) Triggers, Conditions, and Actions:**

* Traffic Data Received:
  + Triggered by receiving traffic data from cameras or sensors.
  + Conditions: Data must be valid.
  + Actions: Begin data analysis.
* Data Analysis Complete:
  + Triggered when analysis of traffic data is complete.
  + Conditions: None or it may depend on the violation type.
  + Actions: Detect potential violations.
* Violation Detected:
  + Triggered when a vehicle do not follow the rule and is caught by the monitoring system.
  + Conditions: None or it may depend on the violation type. For instance, violation of Red-Light crossing will be detected when the vehicle crosses a stop line at red light ‘ON’.
  + Action: Request verification.
* Verification Requested:
  + Triggered when verification is requested by traffic police.
  + Conditions: None or it may depend on the violation type.
  + Actions: Verify the violation details.
* Violation Verified:
  + Triggered when violation is verified.
  + Conditions: None it may depend on the violation type
  + Actions: Notify the driver.

**QUESTIONS:**

1. You are required to draw the state diagram with the help of above analysis.
2. Also draw the state diagrams for the following scenarios.

**Possible Scenarios for State Diagrams:**

* Scenario 1: Detection and Verification of Over Speeding Violation
* Scenario 2: Notification and Verification of Red-Light Violation
* Scenario 3: Payment Processing for Verified Violation
* Scenario 4: Notification for Non-Verified Violation

These scenarios cover different aspects of the system's behavior and allow for the creation of relevant state diagrams.

Remember, these are just examples to guide you through the process. You can modify and adapt them based on your needs as guided by the lab instructor and the requirements of the Traffic Violation Monitoring System.

**Examples**

You may also get help from the following samples taken from the book in a different context i.e. Microwave oven. Y**ou are required to complete them and identify any mistakes/errors or space for improvements or changes, if required.**

In Figure 1, representing a state diagram, you can see that the system starts in a waiting state and responds initially to either the full-power or the half-power button. Users can change their mind after selecting one of these and press the other button. The time is set and, if the door is closed, the Start button is enabled. Pushing this button starts the oven operation and cooking takes place for the specified time. This is the end of the cooking cycle and the system returns to the waiting state.

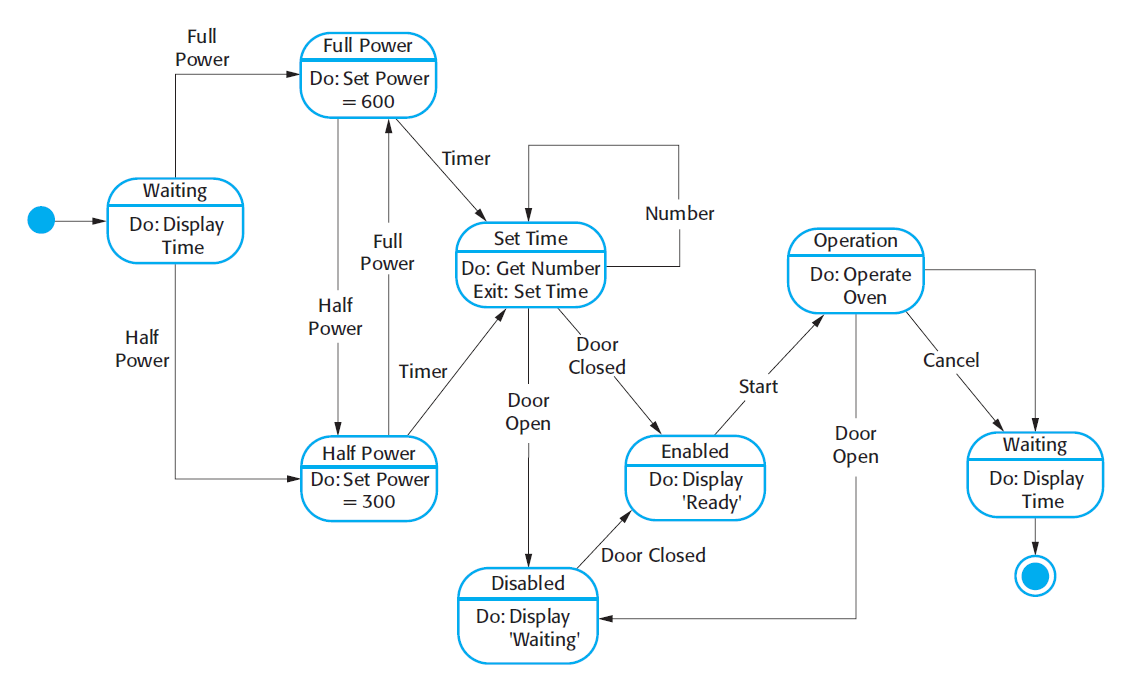


Figure 1 State Diagram of Microwave oven

The UML notation lets you indicate the activity that takes place in a state. In a detailed system specification, you have to provide more detail about both the stimuli and the system states. It is illustrate in Figure 2, which shows a tabular description of each state and how the stimuli that force state transitions are generated.

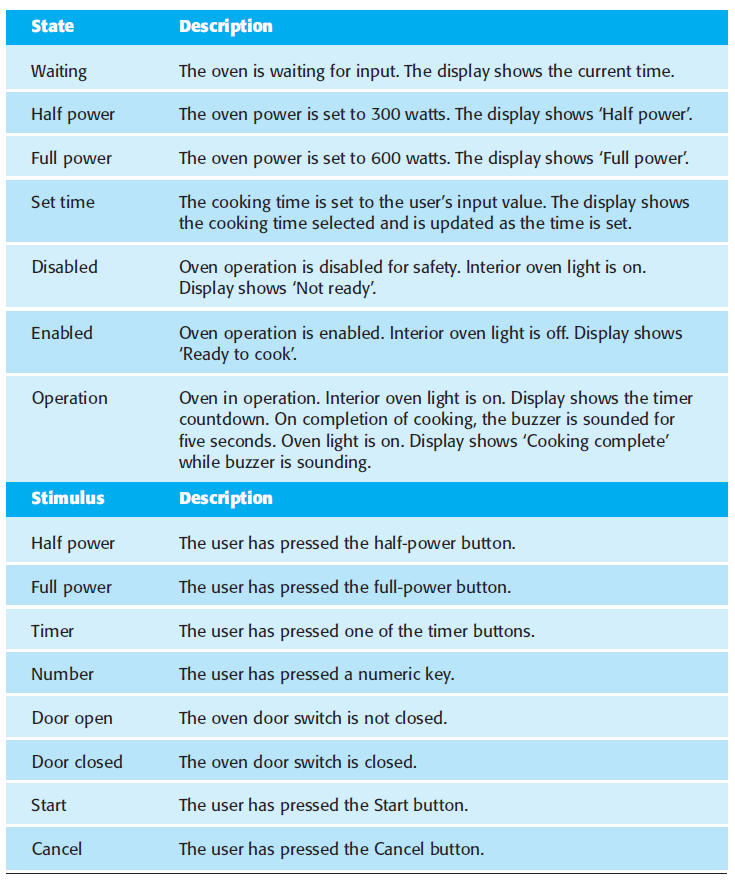


Figure 2 State and Stimuli for the microwave oven

The problem with state-based modeling is that the number of possible states increases rapidly. For large system models, therefore, you need to hide detail in the models. One way to do this is by using the notion of a superstate that encapsulates a number of separate states. This superstate looks like a single state on a high-level model but is then expanded to show more detail on a separate diagram. To illustrate this concept, consider the ‘Operation’ state in Figure 1. This is a superstate that can be expanded, as illustrated in Figure 3.

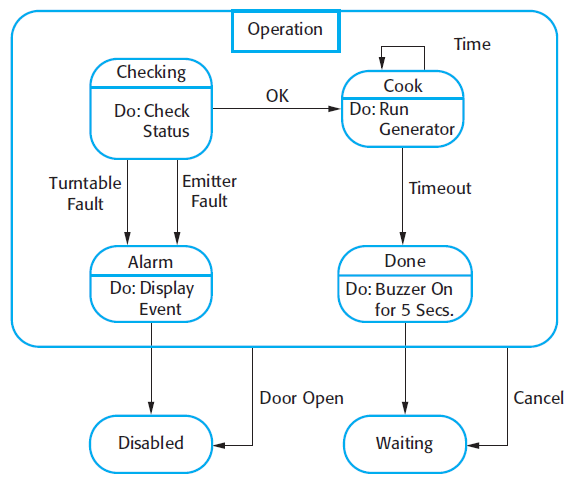


Figure 3 Microwave's Operation State as Superstate

The Operation state includes a number of sub-states. It shows that operation starts with a status check and that if any problems are discovered an alarm is indicated and operation is disabled. Cooking involves running the microwave generator for the specified time; on completion, a buzzer is sounded. If the door is opened during operation, the system moves to the disabled state, as shown in Figure 1.