**Modeling of Various Applications Using Finite State Machines**

Kavita Pandey

Dhruv Singh, Sumit Bansal, Vrishti Gahlaut

Jaypee Institute of Information Technology

**ABSTRACT**

Finite State Machines (FSMs) are mathematical abstractions which have a rich use in computer applications. If we build a FSM for software, then it becomes much easier and simpler to understand, debug and modify. The insights developed in FSMs have had great influence on various domains. This paper presents the application of FSMs over various such domains. These domains include *spoken web technology* that enables a user to access a massive network of voice sites through speech. Information transfer protocol for *Vehicular Computing* is another domain, where users can get on road support services using vehicular sensors and global position system. Adding on to the domain list, we have modeled FSMs for *Ankle Monitor* and *Sensor Dust* which are applications of mobile ad-hoc networks. Ankle monitor is primarily used to track movements of an individual through wireless communication and *Sensor Dusts* are tiny sensors which monitor the environment in which they are deployed. We have also modeled *Palm Operating System*, which is a mobile OS that runs on Linux kernel. This paper presents the FSM for the booting sequence of Palm OS. Furthermore, a FSM is modeled for *disaster management* of tsunamis and to conclude with we have taken up the domain of satellite simulation and have presented a FSM for the steps involved in *Satellite launching*.

**Keywords**: Finite State Machines; Spoken Web; Vehicular Computing; Ankle Monitor; Sensor Dust; Palm OS; Disaster Management.

1. **INTRODUCTION**

The theory Finite State Machines is one of the longest established areas in Computer Science and has a non-exhaustive list of applications. The application of FSM can be observed in many domains of the technical world which perform a sequence of actions depending on a sequence of inputs with which they are controlled. This paper describes some of these typical applications that are modeled specifically with FSMs with an aim to provide a single source of scattered application of this theoretical concept.

First of the applications that we are going to model is *spoken web* that was founded by Eyal Shalom in 2006. It is a massive web portal in which users can upload vocal sites and access them using unique phone numbers analogous to URL in World Wide Web. It is a concept primarily for underdeveloped areas lacking net connectivity and visually impaired people. In this paper we have presented a FSM for Spoken Web which can be used for tourism management where all inputs will be voice signals.

Vehicular Information Transfer Protocol is another application which is modeled using FSM. It provides assistance to the drivers based upon the information collected by vehicular sensors and GPS. Suppose a driver wants to find out information about a particular route, then it will send a request for traffic information along with its own location coordinates. The driver receives suggestions based upon the processing within the internal data structure of roads and vehicles embedded in the system. There are a number of applications where this protocol is needed, but we have specifically modeled some of events of accident reporting and GPS navigation.

Next domain which can borrow from the concept of FSM is networking. Sensor Dust is an application of mobile ad-hoc networks where sensors are deployed in an environment which is to be analyzed and controlled. Ankle monitor is also a mobile ad-hoc application which is extensively used by law enforcement to track individuals or is used to monitor pets. The monitor sends radio frequency which contains location and other information and this data is received after specified intervals of time. This paper presents the working of sensor dust using FSM and the application of Ankle Monitor for a parolee.

Another application where FSM can be used is operating systems. Of the several operating systems, we have taken up Palm OS. Palm webOS is Palm's proprietary mobile operating system running on the Linux kernel. WebOS’s graphical user interface is designed for use on devices with touchscreens. It includes a suite of applications for personal information management and makes use of a number of web technologies such as HTML5, JavaScript, and CSS. Under this vast domain we have specifically modeled FSMs for the booting sequence of Palm OS.

Further extending the list of applications for FSM is Disaster Management. Disaster Management of Tsunamis incorporates prevention, preparedness, response and recovery in case of a Tsunami. The implementation of these steps is depicted using a FSM.

The last domain we have touched upon is Satellite simulation. Satellite launching is an integral part of satellite simulation which describes the various steps involved in launching a satellite and these have been depicted using a FSM in this paper.

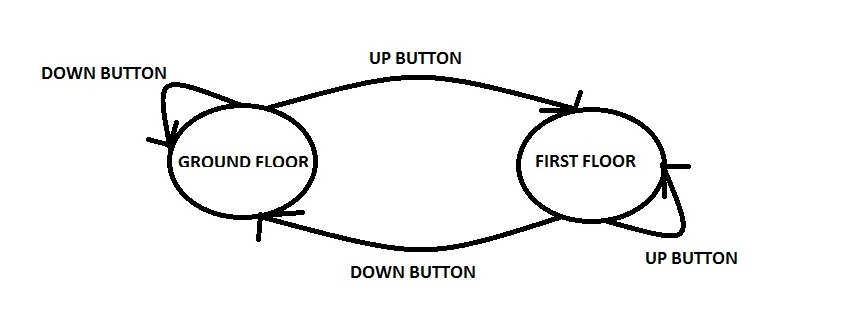
Basically, the aim of choosing such a diverse list of applications is twofold. First, is to show the applicability of FSM concept over various domains and the next is to provide a single source document where users can find and understand various applications using the concept of FSM. Now, let’s briefly introduce the finite state machine concept with a small example.

A particular FSM is defined by the following elements

* set of states which include a start state, one or more finish states and other intermediate states.
* set of Inputs.
* a transition set stating the transitions from one state to another based on an input.
* set of action performed on each transition/ on each state.

Finite state machines can be represented either as a state diagram or a state transition table.

A state diagram is a very basic representation where states are represented by circles, arrows denote transition from the base state to the state pointed by the arrowhead along with input required for that transition. *Figure 1* shows a simple example for state diagram of a two floor elevator. If you are present at the ground floor then the down button keeps you at the ground floor and the up button takes you to the first floor. Similarly at the first floor, the up button keeps you at the first floor and the down button takes you to ground floor.



*Figure 1- State Diagram of two floor elevator*

This can also be represented in the form of state transition table where rows represent the various states and the columns are for current state, input and next state. The *table 1* shows the same example of two floor elevator using state transition table.

|  |  |  |
| --- | --- | --- |
| **STATE** | **INPUT** | |
| **UP** | **DOWN** |
| Ground Floor | First Floor | Ground Floor |
| First Floor | First Floor | Ground Floor |

*Table 1- State Transition Table*

The remaining paper is organized as follows: Section 2 describes the related work, Section 3 to 8 present the FSM for various applications like spoken web, information transfer protocol, sensor dust & ankle monitor, palm OS, disaster management of tsunamis and satellite launching. At the last conclusion is given in section 9.

1. **RELATED WORK**

Various researches have been carried out in order to demonstrate the applications of Finite State Machines over multiple domains. Some of these researches are: M. Ali Qureshi et al. [1] proposed A Verilog Model of adaptable traffic control system using mealy state machine in which the design traffic control system is carried out for a chowk consisting of five roads and each road is divided into main road (for straight movement) and cross road (for crossing) with the aim to prevent traffic jams. Also, Noman Sohaib Qureshi et al. [2] summarized the design of an arcade game using Automata theory tool. They primarily used Deterministic finite state automaton and nondeterministic finite state automaton in various levels of designing the game. Another research that used the concept of FSM was conducted by Pengyu Hong et al. [3]. They proposed an approach for gesture learning and recognition in which they built a FSM recognizer for each gesture. The computational efficiency of these FSM recognizers allowed them to reach real time on-line performances. Further implying to the never ending applications of FSM, Ana Monga and Balwinder Singh [4] described the designing of multi select Vending machine with Auto-Billing Features that was modeled using mealy machine to process four states namely user Selection, Waiting for money insertion, product delivery and servicing.

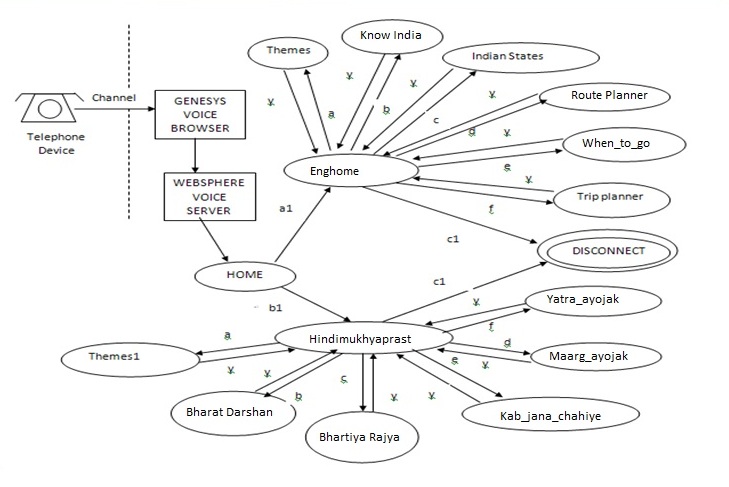
All the above researches were focused on one application and have concentrated their efforts in gaining appreciation for FSM in one domain. However in this paper, Finite State Machines lend themselves for various applications over scattered domains. Here, we attempt to provide appreciation for this concept in more than one application.

**3. SPOKEN WEB**

Finite State Machines can be effectively used to describe a Spoken Web Voice Site where the user can access a network of voice sites through speech. We have taken the example of tourism Voice Site which is modeled using the FSM shown in *figure 2.* This Voice Site has many features where user can get information on various themes according to which they can plan the trip in India, users can avail the services of route planner and trip planner provided by the site in both Hindi and English languages.

Input alphabets:

**a1**- select language English so that you listen in English, **b1**- Select language Hindi so that you listen in hindi, **a**- To get info about themes (when the current state is enghome), **b**- To go to state know India (when current state is enghome), **c**- To get info about Indian states (when current state is enghome), **d**- To go to state Route Planner (when current state is enghome), **e** To go to state When\_to\_go (when current state is enghome), **f**- To go to state Trip Planner Planner(when current state is enghome), **b**- Select language Hindi so that you listen in hindi, **a-** To get info about themes(when the current state is Hindimukhyaprasht), **b**- To go to state Bharat darshan (when current state is Hindimukhyaprasht), **c**- To get info about Bhartiya Rajya(when current state is Hindimukhyaprasht), **d**- To go to state Maarga\_ayojak(when current state is Hindimukhyaprasht), **e**- To go to state Kab jaana chahiye (when current state is Hindimukhyaprasht), **f**- To go to state Yatra\_ayojak(when current state is Hindimukhyaprasht)



*Figure 2 –FSM for Voice Site*

Web Technology and networks go hand in hand, therefore further we have discussed about the application of FSM in modern vehicular ad-hoc networks in the next section.

**4. INFORMATION TRANSFER PROTOCOL**

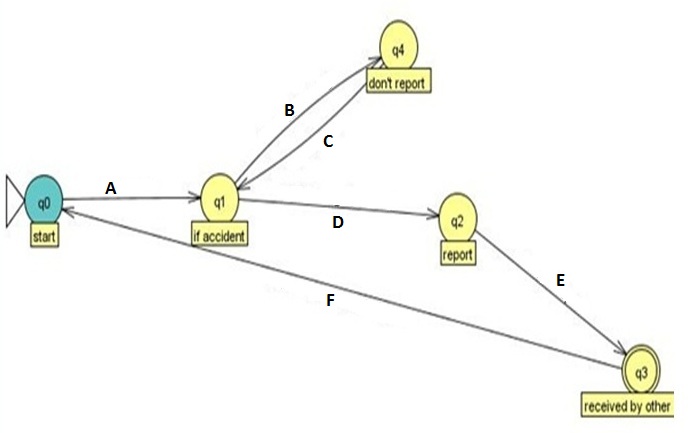
Information transfer protocol is an application of vehicular ad-hoc networks (VANETs) whose purpose is to assist the drivers based upon the information collected by the various sensors. This domain constitutes various sub domains, but in this paper we have modeled FSMs for two sub domains namely accident reporting and GPS navigation.

**Accident Reporting**

Accident reporting is a vital component of VANET whose functioning is depicted using FSM shown in *figure 3.* When a car is on the move and an accident occurs, if there is no other car in the range of 100 meters then the system will not report the accident and will continue moving ahead. In the case of presence of a car within 100 meters, the system will transmit a warning report. If such a report of an accident is received by any system then the driver of that car is warned.

Input alphabets:

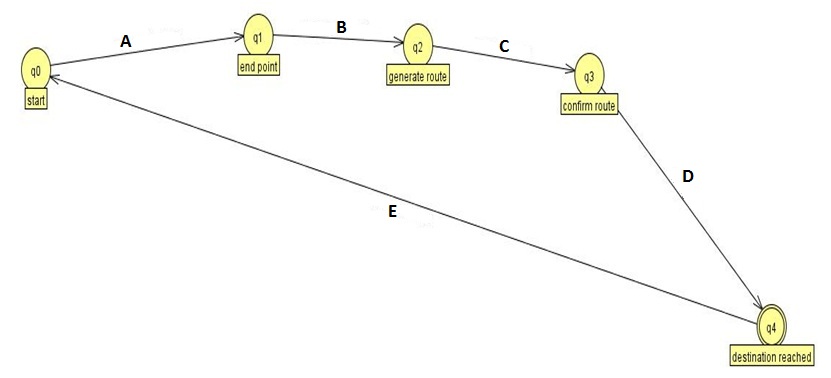
1. Car moves
2. Node not in range
3. Moves Ahead
4. Node in range(0-100m)
5. Transmit Report
6. Warn or report accident ahead



*Figure 3 –FSM for Accident Reporting*

**GPS Navigation**

Another most widely used application of VANET is GPS. When the driver of a car inputs the destination location, the system accesses the satellite connection which suggests a number of routes from which the car chooses an appropriate route. Then the car navigates through that route and reaches the destination. If the car wants to navigate to a new destination then the same process will again commence. The FSM for this whole procedure is shown in *figure 4.*

**

*Figure 4 –FSM for GPS Navigation*

Input alphabets:

1. Enter destination
2. Satellite connection
3. Choose route
4. Navigate
5. New navigation

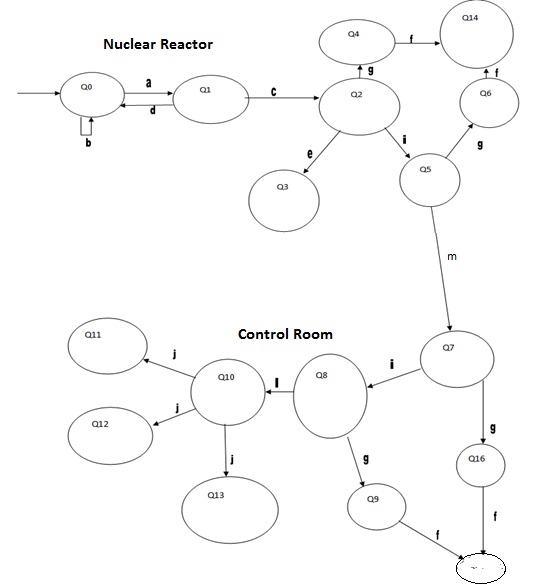
Continuing with the application of FSM in networking, further we have presented the use of FSMs in mobile ad-hoc networks like Sensor Dust and Ankle Monitor in the next section.

**5. MOBILE AD-HOC NETWORKS**

Of the various mobile ad-hoc networks, in this paper we have used FSM to model Sensor dust and ankle monitor both of which use wireless communication techniques to fulfill their purpose.

**Sensor Dust**

To be able to analyze - for example - a possible atomic reactor leak, it may be a pleasant solution to place tiny sensors throughout a huge area, probably in regions where no information infrastructure is present or destroyed, instead of risking human lives by sending them to hazardous areas. Those sensors would be very limited in their energy supply. Figure5 depicts the FSM for the functioning of these Sensors.

****

*Figure 5.1 – FSM for Sensor Dust*

State description:

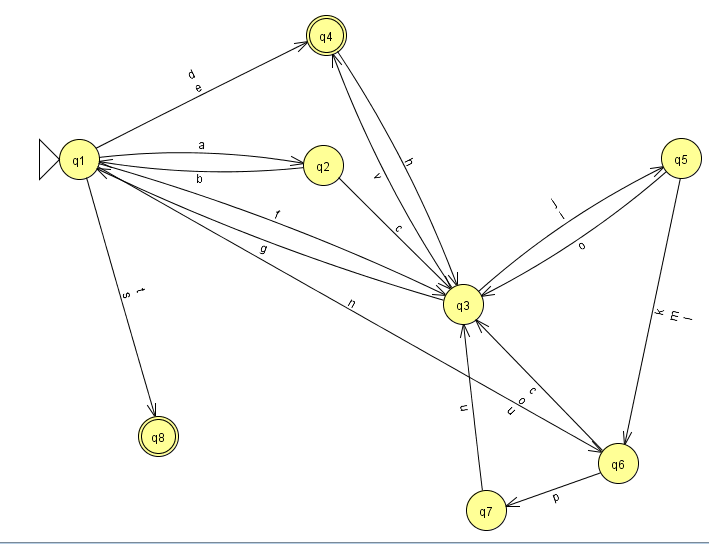
**Q0** - nuclear reactor, **Q1** - check particle level, **Q2** - sensor signal trigger, **Q3** - Automatic shower, **Q4** - Sensor Failure, **Q5** - Alarm rings, **Q6** - Alarm failure, **Q7** - Control Room Sensor, **Q8** - Control Room alarm, **Q9** - Control Room alarm failure, **Q10** - Mobile Signal Transmitted, **Q11** – Team 1(fire brigade), **Q12** - Team 2 (security team), **Q13** - Team 3 ( technician), **Q14** – Backup 1 (nuclear), **Q15** – Backup 2 (control room), **Q16** - Control Room sensor failure

Input alphabets:

**a** - Gas leakage, **b** **-** Gas not leaked, **c** - particle level high, **d** - particle level low, **e** - shower starts, **f** - call for backup, **g** - fail, **h** - send signal to control room sensor, **i** - sensor starts, **j** - transmit signal to team, **k** - control room signal detected, **L** - send signal to mobile

**Ankle Monitor**

This setup involves parole officer, parolee, server room and local police. A radius is set for the parolee to be able to move in. Any movement outside the radius or he tries to break the ankle monitor circuit it sets off an alarm in the server room. Server room first informs the parolee to back out and also alerts his parole officer. In case of non-cooperation of the parolee local police is called in by the server room and the parolee is arrested. Also, if the prisoner is found anywhere near a crime scene or taking part in some illegal actions, etc. he gets arrested. Then the court decides based on the severity of his nuisance whether to leave him on parole or to get him back to prison. In the end, keeping parolee’s good conduct (if any) or at the time when his supposed parole time is coming to halt the court assesses the parolee. The court can either release him or can make him continue the parole time. The FSM for this implementation is shown in *Figure 6*



*Figure 6 – FSM for Ankle Monitor*

State description:

**q1**-court, **q2**- committee, **q3**- prisoner, **q4**- prison, **q5**- server room, **q6**- parole officer, **q7**- local police, **q8**- discharge, **q9**- ankle monitor, **q10**- satellite, **q11**- transmitter

Input alphabets:

**a**-sets, **b**- submit report, **c**- assess, **d**- order of continuation of imprisonment, **e**- order release on parole, **f**-conducts hearing, **g**-gives statement in, **h**-releases(on parole), **i**-track via. G.P.S., **j**- track via. LAN cable or phone line, **k**- send location whenever asked, **l**-send location whenever needed, **m**- send location after certain interval of time, **n**-appoints, **o**-alerts and asks to cooperate, **p**-in case of non-cooperation, **r**- arrests and taken to, **s**- positive assessment or petty crimes (before parole), **t**-completion of parole time(without nuisance), **u**-arrest, **v**-taken to, **w**-send signal.

Having discussed about application of FSM in the field of networking, in the next section we have presented the implementation and working of Palm operating system using FSMs.

**6. PALM OPERATING SYSTEM**

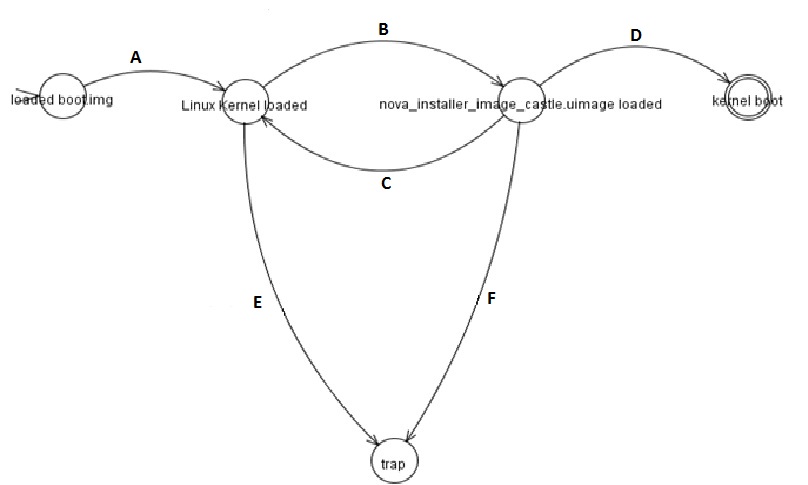
The Palm webOS is based on the Linux 2.6 kernel whose graphical user interface is designed for use on devices with touchscreens. The booting sequence and the application launch cycle are two crucial components of palm OS that are modeled using FSM in this paper.

**Booting sequence of Palm OS**

The whole sequence of steps involved in the booting of palm OS is depicted in using FSM in *figure 7*.

Input alphabets:

1. Loading Kernel
2. Loading nova\_installer\_image\_castle.uimage
3. Loading error
4. Loading initrd
5. Kernel not found
6. Initrd not found

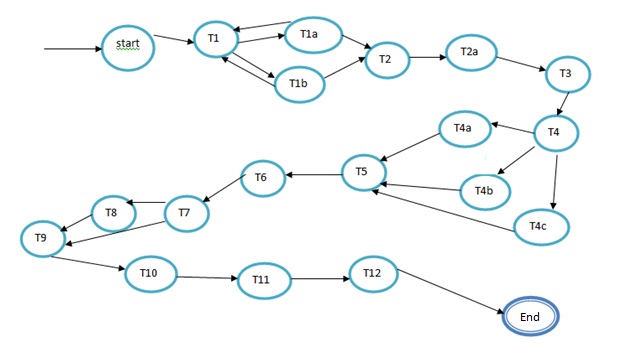


*Figure 7 – FSM for Booting Sequence of Palm OS.*

After applying FSMs in the domains of Operating Systems, networking and Web Design, in the next section we move on to a lifesaving application of FSM i.e. Disaster Management.

**7. DISASTER MANAGEMENT FOR TSUNAMI**

Disasters occur unexpectedly and cause huge amount of loss in terms of life and property. Therefore every concept should contribute towards relieving the world from such disasters. Here we have chosen tsunami as the disaster whose ill effects we want to minimize. *Figure 8* depicts the FSM for steps to be taken for prevention, preparedness, response and recovery in the event of a Tsunami.

****

*Figure 8 – FSM for Disaster Management of Tsunami*

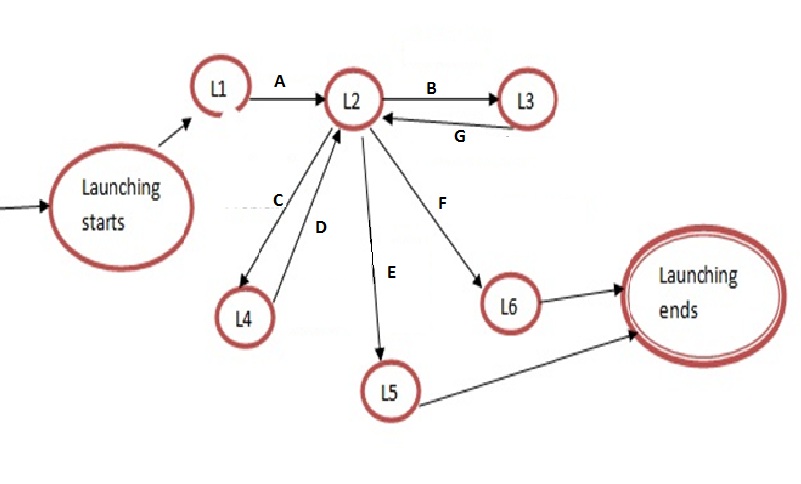
State description:

**T1**: check tsunami watches, **T1a**: radio, **T1b**: televisions, **T2**: Check warnings, **T2a**: use of radar, **T3**: establish a base camp to provide communication to everyone, **T4**: Structural engineer to designate shelter areas, **T4a**: construct some underground areas, **T4b**: if not go for rooms at lowest floor without windows, **T4c**: do not consider auditoriums, cafeterias, **T5**: evacuation plan, **T6**: Tsunami Drills, **T7**: message received, **T8**: perform drills, **T9**: evacuation process, **T10**: provide shelter, **T11**: food, **T12**: first aid.

The next section we have taken up the final application of FSM for out paper and that is Satellite Launching procedure which is a highly critical component of satellite simulation.

**8. SATELLITE LAUNCHING**

Satellite launching is the elementary procedure for any satellite simulation, thus it needs to be accurate and easy to test. Therefore we have used FSM to model the steps involved in satellite launching. *Figure 9* depicts the various stages, testing and calculations involved in satellite launching.



*Figure 9- FSM for satellite launching*

State description:

**L1**: fuel filled, **L2**: vehicle ready to take off, **L3**: vehicle waiting, **L4**: vehicle waiting, **L5**: satellite in Leo orbit, **L6**: satellite in Geo orbit.

Input alphabets:

**A**- Ignition, **B**- Bad weather, **C**- Technical problem, **D**- Problem solved, **E**- Speed >=11.3 km/s, **F**- Speed >=11.3 km/s, **G**- Good weather

**9. CONCLUSION**

In this paper, Finite state machines have been used to model various applications over multiple domains. FSM has the power to make any application easier to understand, implement, debug, verify and modify. We have effectively used FSM to model a tourism voice site where users can access a network via speech signals, also we have modeled FSMs for a robust information transfer to assist drivers with GPS navigation and accident reporting. In addition to this we have presented FSMs to depict the functionalities of sensor dust and ankle monitor. We effectively modeled a FSM for the booting sequence of palm webOS. Also, we displayed the steps to be taken in the event of a tsunami in the format of a FSM. Towards the end we used FSM to depict the steps involved in launching a satellite. In all, we have successfully complied a single source document pertaining with the application of finite state machines in scattered areas of science and management.

**REFERENCES**

[1] M. Ali Qureshi, Abdul Aziz, and S. Hammad Raza, “A Verilog Model of Adaptable Traffic Control System

Using Mealy State Machines”, International Journal of Computer and Electrical Engineering, Vol. 4, No. 3, June 2012.

[2] Noman Sohaib Qureshi, Hassan Mushtaq, Muhammad Shehzad Aslam, Muhammad Ahsan, Mohsin Ali and Muhammad Aqib Atta, “Computing Game Design with Automata Theory”, International Journal of Multidisciplinary Sciences and Engineering, Vol. 3, No. 5, May 2012.

[3] Pengyu Hong, Matthew Turk, Thomas S. Huang, “Gesture Modeling and Recognition using Finite State Machine”, Microsoft Research.

[4] Ana Monga, Balwinder Singh, “Finite State Machine based Vending Machine Controller with Auto-Billing Features”, International Journal of VLSI design & Communication Systems (VLSICS) Vol.3, No.2, April 2012.