The Use of Finite State Machine Representation in Elevator Control Sequence Specifications

Jon B. Halpern
Millar Elevator Industries Inc.
620 Twelfth Avenue, New York, N.Y. 10036

Abstract

Finite State Machine representation has been used extensively in the design of Finite State Automatons (FSAs). Communication system protocols, microprocessor hardware and software systems as well as elevator systems are all well definable FSA's, and can be represented as such. However, when elevator systems are specified or described, less effective alternate methods are utilized. Consequently, the final result is a poorly defined set of operations, needing clarification and interpretation for proper utilization. What I would like to propose is the use of FSA representation in the specifications, and representation of elevator systems, as a standard when describing control sequences of operation.

I. Introduction

An elevator contractor or manufacturer of elevator control systems spends many hours interpreting elevator codes and specifications, provided from many sources. Code authorities, Consultants, Owners and Contractors themselves, provide specifications to control manufacturers, describing sequences of operation. These specifications/codes are usually written in such a fashion as a set of descriptive paragraphs or rules, that desire complex sequences of operation. From these rules, elevator contractor's and designer's must then read the rules and properly interpret them into control sequences. When they take on the task of interpreting these narrative rules, they find that there are many undefined sequences in the sets of rules, as well as undefined initial conditions. The method in which the initial conditions should be handled is left unspecified. The contractor must look for alternate sources to obtain these unusual or unspecified sequences. The process that usually is encountered is that the contractor goes to the source of the specification and requests what is known as an interpretation. From these published interpretations, the contractor can then continue his design process and engineering specification. Just the fact that interpretations exist, is an indication that the initial specification was incomplete.

Incomplete specifications is a major cost burden to the elevator industry and its customers. Numerous systems have been installed and then during the final inspection process, the system fails to meet the intended performance specification and or code requirements. Alternately, the Contractor and the Inspector, Consultant or Owner disagree on what the specifications intended. We can all agree that this can be an expensive as well as an embarrassing situation. The quality process is a process that has received a great deal of attention over the past several years. Companies are striving to provide higher quality products at more competitive costs. One of the root causes for non-conformance is incomplete specifications from any and all sources. A major missing component from the performance specification is the addition of a test specification. A test specification along with a test procedure included in the specification would provide tremendous insight. Think about the last time the code book or a consultants specification discussed the method for test compliance. It's almost non-existent.

One other area that stands to gain from core issues discussed in this article is standards in

remote monitoring. With state representation techniques and standard nomenclature for events and processes, implementation independent system state description can be accomplished. I would like to provide some insight on how this might work using FSA's.

I would propose that the industry should adopt some of these methods as a standard to represent control sequence specifications and that if we do this that the industry as a whole will improve its communication skills.

II. Structure Techniques

The introduction of structure techniques in the computer world was a major step forward, referred to by many people as the "structural revolution". These structures include methods of producing specifications and structure diagrams in a clear and obvious fashion. In the data processing environment, diagrams used for structure design were to act as a basis for software generation. Good diagramming techniques can:

Aid clearer thinking.
Be a precise communication between members of the development and the user team.
Provide standard interfaces.
Provide system documentation.
Force good structuring of specification.
Aid debugging of systems.
Aid changing of systems.
Aid acceptance testing.
Enable end users to review the design/specification.
Encourage end users to articulate their needs clearly.[1]

Similarly, in the elevator business, we all recognize that most, but not all, of the systems out there in the market also end up being coded into software. Many of these control sequences of operation end up being an essential part of a microprocessor controller. Therefore, if we were to incorporate structured techniques into the initial job and code specifications it could be useful to the elevator industry, especially to the engineers that generate the software that controls these sequences.

III. Diagramming

"Good clear diagrams play an essential part in designing complex systems and developing programs." I believe the other quote, "a picture is worth a thousand words", is also appropriate. Diagramming is essential for clear thinking and for effective communication. An enterprise needs standards for control sequence definition, just as it has standards for data processing diagrams and engineering drawings.[1] The free form language and rules that are presently in our code books and specifications do not always present a clear picture to the design engineer as to exactly how he should approach a certain problem and how he should meet the intent of the specification. Although most specifications received are performance specifications and not necessarily design specifications, it is still essential to have a clear picture of exactly how a system should operate. Especially in control operations such as fireman's recall, emergency power, code blue hospital emergency, etc. We need a method for communicating those ideas clearly.

There are many types of diagrams that are used in structure techniques. Among those are decomposition diagrams, data flow diagrams, action diagrams, data structure diagrams, entity relationship diagrams, decision trees and tables, state transition diagrams. This list, although not complete, gives you an idea of different types of diagrams that are available. I will refer you to reference items J. Martin [1] and Ward and Mellor [2] for more detailed examples of the diagrams and structured techniques.

IV. Finite State Automata Formal Definition

The definition provides for a fixed set of well defined inputs, a fixed set of well defined set of states, and a completely defined deterministic set of transitions. It also describes a set of outputs that can be dependent on either the transition or the state. It provides for mathematical closure.

Definition : Deterministic Finite Automaton (DFA) - Mathematical model of a machine that accepts a particular set of words over some alphabet. [3]

A is the input Alphabet (Set of input combinations)
S is the finite set of non-empty states
s_0 is the starting or initial state in S
D is the state transition function D: S x A → S
F is the final states or empty set
W is the output function

symbol a ε A and a state s ε S can be viewed as a directed graph with vertices V and edges E.

$$< A , S , s_0 , D , F > \rightarrow G = < V , E >$$

as $V = S$ $\quad E = \{ <s,t,a> \mid S, t \varepsilon S , a \varepsilon A , D(s,a) = t \}$

each element E is an ordered triple < s , t , a >.

This directed graph model conforms very nicely to a descriptive method of displaying the automaton in a device called a state transition diagram.

The output function of the automaton can be a function of the state transition,a Mealy sequential machine

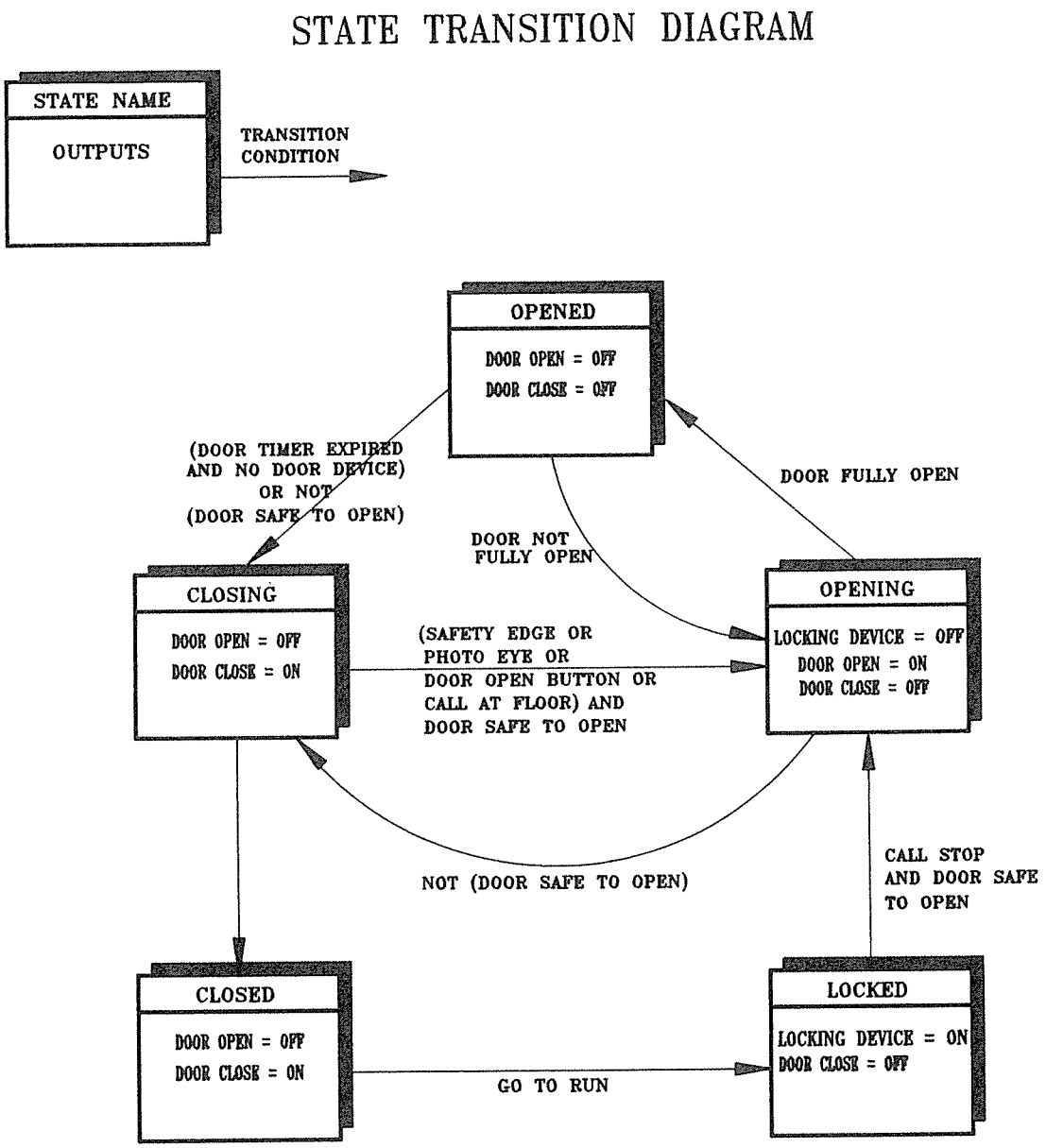
$$W : S \times A \rightarrow T$$

or a function of the state as in a Moore sequential machine.

$$W : S \rightarrow T$$

V. State Transition Diagrams

The state transition diagram is the device we will utilize to describe the FSA. The vertices will be represented as a box with the state name at the top and the outputs within the box (Moore



A SIMPLE DOOR STATE MACHINE FOR AN AUTOMATIC ELEVATOR

INPUTS	STATES	OUTPUTS
DOOR TIMER	OPENED	DOOR OPEN
DOOR FULLY OPEN	CLOSING	DOOR CLOSED
DOOR FULLY CLOSED	CLOSED	LOCKING DEVICE
GO TO RUN (FROM DIRECTION STATE MACHINE)	LOCKED	
CALL STOP	OPENING	
DOOR SAFE TO OPEN		
SAFETY EDGE		
PHOTO EYE		
DOOR OPEN BUTTON		
CALL AT FLOOR		

fig 1

machine). The edges are represented by lines with directional arrows indicating the direction of the transition. The line is labeled with the input combination (alphabet), which will cause the transition. See figure 1 for a detailed diagram.

VI. Finite State Machines for Elevators

Elevator systems can be modeled as a finite state machines. They typically have a finite number of inputs and outputs. Therefore, they must have a fixed number of states, if they are deterministic [Ignoring, for the moment, that there are some dispatching algorithms, and possibly some other aspects of an elevator's operation that might violate this statement]. Although the number of states is fixed, that number could be quite large, if we were to define every state for every combination of inputs. Since many inputs take precedent over other inputs, and some inputs run parallel, we can then functionally decompose the elevator system into a hierarchy of automatons. This decomposition eliminates a large number of invalid states and allows us to model the system with a manageable number of states. Although these automatons are related and need information from each other to run, we will treat that information as additional inputs and outputs.

The door state machine depicted in figure 1 is a simple door machine that is valid when the service state machine of the elevator is "group automatic". The state machine has been simplified for purposes of illustration ignoring some of the states and inputs that are found in a real implementation. States dealing with issues such as door protection, nudging and multiple door dwell times have been ignored for simplicity.

Sets of state machines that would describe an individual car on automatic group operation are shown in figure 2. The dependencies are structured in a hierarchy where the master state machine calls the proper service state machine, and based on what state the service state machine is in; the proper door direction, motor generator and motion state machines will run. Again this figure is but a microcosm of a complete elevator system.

The data definition defining each of the inputs and outputs is shown in figure 3.

From the narrative set of rules featured in figure 4 describing a simplified firemans recall sequence, I have generated state transition diagrams for the door and direction machines in figure 5.

What we end up with is a well defined set of operations that can supplement narrative test and event lists. It is by no means a cure for all bad specifications. It is possible to omit states and improperly define transitions. However, it provides a method that makes the discovery of errors and omissions easier and more visible. Similarly, we could provide a test specification in the same format to test the specified state machine. As I mentioned in the introduction, incomplete specifications allow too much latitude for the interpretation of whether the system meets the specification.

MASTER STATE MACHINE	ON	OFF

SERVICE STATE MACHINE (ON)	AUTOMATIC	ATTENDANT
	INDEPENDENT	TEST/INSPECTION
	SIMPLEX	ACCESS TOP
		ACCESS BOTTOM

DIRECTION STATE MACHINE (AUTOMATIC)	NO DIRECTION	UP DIRECTION
	DOWN DIRECTION	UP RUN
	DOWN RUN	STOPPING

DOOR STATE MACHINE (AUTOMATIC)	OPEN	OPENING
	CLOSED	CLOSING
		LOCKED

MOTOR GENERATOR STATE MACHINE (AUTOMATIC)	RUNNING	OFF
	STARTING	OVERSPEED

MOTION STATE MACHINE (AUTOMATIC)	STOP	ACCELERATE
	DECELERATE	CONTRACT SPEED
	LEVELING	RELEVELING
	REDUCED SPEED	INSPECTION SPEED
		FAULT

Figure 2

DATA DEFINITIONS

INPUTS/OUTPUTS	DEFINITION
Door Timer	Timer from the time manager, Door Dwell Timer
Door Fully Open	Input from the system signaling the door is open
Go To Run	Input from the system signaling the door is closed
Call Stop	Input signal from the direction state machine lock the doors so the car can run.
Door Safe To Open	Input from the system indicating that it is safe to open the doors
Safety Edge	Input from the door device called safety edge
Photo Eye	Input form the door device called photo eye
Call At Floor	Input from the direction sate machine indicating there is a call at the floor that we are presently at
Door Open	Output signal to open the door
Locking Device	Output signal for those systems requiring such a locking device on the doors
Door Close	Output signal to close the door

Figure 3

FIREMAN'S RECALL EXAMPLE

Excerpts from the ANSI A17.1 Firemans Recall Rule are provided below for the example.

Phase I Emergency Recall Operation. A three-position key-operated switch shall be provided only at the designated level for each single elevator or for each group of elevators.

When a switch is in the "ON" position:

All cars controlled by this switch which are on automatic service shall return nonstop to the designated level and power-operated doors shall open and remain open.

A car traveling away from the designated level shall reverse at or before the next available landing without opening its doors.

A car stopped at a landing shall have the in-car emergency stop switch or in-car stop switch rendered inoperative as soon as the car moves away from the landing. A moving car shall have the in-car emergency stop switch or in-car stop switch rendered inoperative without delay. Once the in-car emergency stop switch or in-car has been rendered inoperative, it shall remain inoperative while the car is on Phase I operation. All other stop switches shall remain operative.

A car standing at a landing other than the designated level, with the doors open and the in-car emergency stop switch or in-car stop switch in the run position, shall close the doors without delay and proceed to the designated level.

Door reopening devices for power-operated doors which are sensitive to smoke or flame shall be rendered inoperative without delay. Door reopening devices not sensitive to smoke or flame are permitted to remain operative.

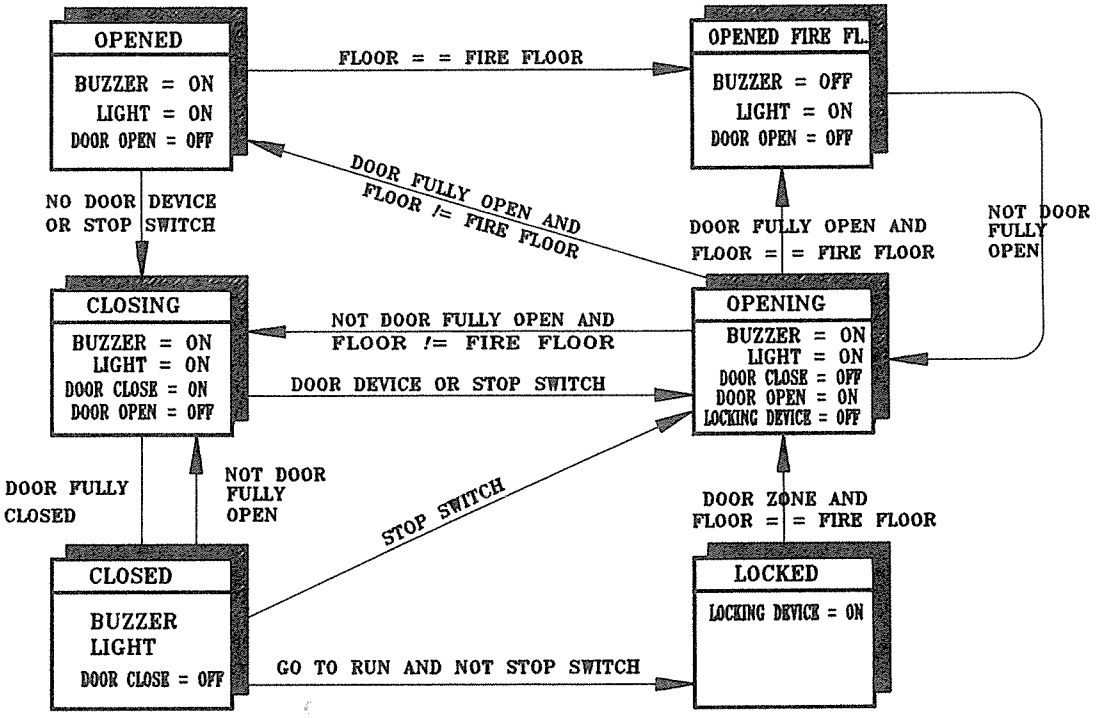
All cars shall be provided with a visual and audible signal system which shall be activated to alert the passengers that the car is returning nonstop to the designated level. The signal shall remain activated until the car has returned to the designated level.

A car stopped at a landing shall have the in-car door open button rendered inoperative as soon as the car moves away from the landing. A moving car shall have the in-car door open button rendered inoperative without delay. Once the in-car door open button has been rendered inoperative, it shall remain inoperative until the car has returned to the designated level. [4]

Figure 4.

PHASE I FIREMANS

DOOR STATE MACHINE AUTOMATIC PHASE I



DOOR DEVICE = ANY DOOR DEVICE UNAFFECTED BY SMOKE

DIRECTION STATE MACHINE AUTOMATIC PHASE I

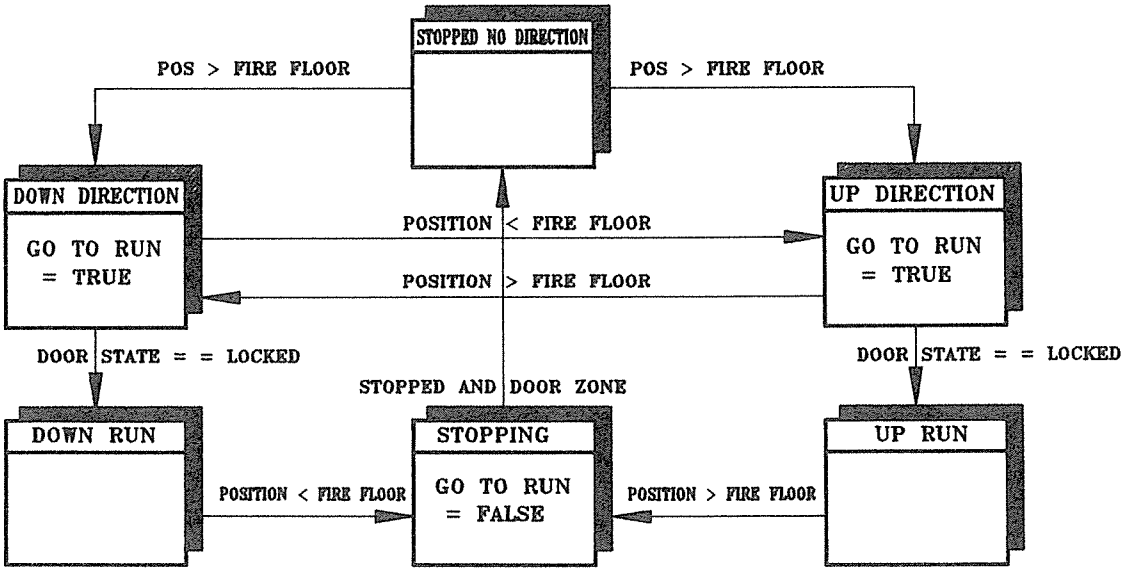


fig 5

VII. Remote Monitoring

All prior approaches to specifying a remote monitoring standard dealt with how one could connect a monitoring device to a an existing elevator. It specified connections, connectors and set of standard messages.

The proper approach to remote monitoring should be implementation independent. A standard set of monitoring functions should be defined and manufacturers of control equipment would then conform to the monitoring functions by translating their systems into the standard. Look at the door state machine in figure 1 and for the sake of this discussion, agree that this is the standard for automatic door operation. If I were to send you the current state of the door state machine (opening,closing,opened,closed,...) think about how much information you already have about the system without any other additional information. If you would collect the prior door states and the inputs that created the transition, you could reconstruct all that has happened, remotely.

VIII. Conclusions

The final result of utilizing these state diagrams is that the final document is graphic, rigorous, maintainable, logical, precise, concise, and highly readable. [2]

This is all not to suggest that state transition diagrams are the only way to describe these sequences and rules. But, what one can conclude is that they would be a wonderful supplement to the present day narrative specifications that we receive. It would add completeness and provide a better means of communications between generators of specifications and users of those specifications.

Bibliography

[1] J. Martin, RECOMMENDED DIAGRAM STANDARDS FOR ANALYSTS AND PROGRAMMERS: A Basis For Automation. Prentice Hall, Englewood Cliffs, NJ, 1987.

[2] P. Ward & J. Mellor, STRUCTURED DEVELOPMENT for REAL TIME SYSTEMS, Vol 1-3. Yourdon Press, Englewood Cliffs, New Jersey, 1986.

[3] J. Carroll & Darrell Long, THEORY of FINITE AUTOMATA, Prentice Hall, Englewood Cliffs , New Jersey, 1989.

[4] SAFETY CODE for ELEVATORS and ESCALATORS A17.1 PP 83-84, The American Society of Mechanical Engineers, 345 East 47th Street, N.Y. , N.Y. , 1989

Biography

Jon Halpern is the Executive Vice President of Millar Elevator Industries Inc. in New York City. Mr. Halpern has a Masters of Science and Professional Degree in Electrical Engineering. He has spent his entire 13 year career with Millar and is presently responsible for Modernization and Maintenance Sales, Research and Development, and Manufacturing.