2015 NDIA GROUND VEHICLE SYSTEMS ENGINEERING AND TECHNOLOGY SYMPOSIUM
SYSTEMS ENGINEERING (SE) TECHNICAL SESSION
AUGUST 4-6, 2015 – NOVI, MICHIGAN

The Linkage Between States & Modes and Performance Requirements

Kevin J. Dutcher
Maneuver Collaboration Center (mc2)
General Dynamics Land Systems
Sterling Height, MI 48090

While the identification of States and Modes is called out in some specification templates, it is missing from others. This paper contends that an item cannot be fully specified without identifying States and Modes. Furthermore, to easily document the States and Modes, the performance requirements are best defined within the following schema construct:

- *Functional Statement (verb – noun)*
  - *Performance Parameter 1*
    - *Condition 1*
    - *Condition 2*
  - *Performance Parameter 2*
    - *Condition 1*
    - *Condition 2*

INTRODUCTION

States and Modes seem to have many definitions, some claim that they are interchangeable terms, some refer to state machine definitions, while some definitions really just seem like circular logic and not really useful at all.

This paper will present definitions that were pieced together from different sources over the years, a requirements structure technique to support it, and a specification layout proposal to tie it all together.

STATES & MODES RELATED TO FUNCTIONS

My Mother had a 1969 Volkswagen Beetle which allowed the radio to play without having to have the key inserted, the radio was tied directly to the battery. Many mornings we would go out to start the car and find the battery was dead, caused by the radio having been left on. That inconvenience was balanced by the convenience of working in the garage and reaching over and switching on the radio without having to have the key (especially since I was only 11 at the time, and didn't have my own key, and liked listening to CKLW, The Big 8).

That was my first introduction to States and Modes. Some German engineer made a decision to allow the "Play Radio" function to be active during the "Key Out" state. Compare that to the recent General Motors air bag issue. With the key accidentally moved to off, but the vehicle still moving, the air bags were disabled. Clearly the key position creates state changes, but still having the vehicle moving is a State that more recently was discovered with hybrid vehicles, as the kinetic energy of the vehicle needs to be controlled. The "Key Off" equates to "Off State" is so engrained in the history of the automobiles, which has mostly been mechanical in nature, that it's not surprising that it was missed.

PRIOR AND CURRENT STANDARDS

A review of prior and current standards, around States & Modes brings up the following entries.

The first one we will look at is, DI-CMAN-80008 A, TYPE A, SYSTEM\SEGMENT SPEC1F1CATION

The template calls out the below structure:

3 System requirements
3.1 Definition
3.2 Characteristics
3.2.1 Performance Characteristics
3.2.1.X (State name)

3.2.1.X.Y (Mode name)
3.2.1.X.Y.Z (State capability name and project unique identifier)
3.2.2 System capability relationships

So this DID clearly got it. In fact, it put all the performance requirements under a State and Mode, and then required a description of the System Capability Relationships, essentially the States & Modes Table. Too bad it was superceded in 2002.

The description of State was:
. . shall identify and provide a brief description of a state in which the system can exist (e.g., weapon idle,weapon ready, weapon deployed).

The description of Mode was:
. . . shall identify and provide a brief description of a mode of operation (e.g.,surveillance, threat evaluation, weapon assignment, target designation

Those definitions of States and Modes aren't much help. You can see why there perhaps was confusion, leading to replacement.

It was replaced by DI-IPSC-81431A. It calls out the template below"

3 Requirements
3.1 Required states and modes
3.2 System capability requirements
3.2x System capability

This template delegates States and Modes to it's own section, separate from the requirements. But still states a relationship between them, within the DID:

"The correlation may be indicated by a table or other method in this paragraph"

Very interesting that they call out a table. I've never seen one used, however. The States and Modes definitions in this DID are not very good. They state:

"Examples of states and modes include: idle, ready, active, postuse analysis, training, degraded, emergency, backup, wartime, peacetime. The distinction between states and modes is arbitrary. A system may be described in terms of states only, modes only, states within modes, modes within states, or any other scheme that is useful."

Clearly those definitions are of no use.

The next standard to look at is MIL-STD-490A TYPE Bl, PRIME ITEM DEVELOPMENT SPECIFICATION. It's template structure is:

3.1 Prime item definition.
3.1.1 Prime item diagrams.
3.1.2 Interface definition.
3.1.3 Major component list.
3.1.4 Government furnished property list.
3.1.5 Government loaned property list.
3.2 Characteristics

It doesn't call out States and Modes directly at all. However, within the text it states:

"Paragraph 3.1.2, Interface definition. . . . where interfaces differ due to a change in operational mode, the requirements shall be specified in a manner which identifies specific functional interface requirements for each different mode."

At least it gives Modes and mention and recognizes there may be performance differences across Modes.

Looking at another type within that standard, TYPE B2, CRITICAL ITEM DEVELOPMENT SPECIFICATION

3 Requirements.
3.1, Critical item definition.
3.2, Characteristics.

No references to States and Modes at all. Not good.

MIL-STD-490 was replaced with MIL-STD-961. It's structure template is:

A.2.1 General. Top-level performance requirements that may be included in program-unique specifications are described in A.2.2 through A.2.5. Typically, system specifications would begin with these requirements.
A.2.2 Missions.
A.2.3 Threat.
A.2.4 Required states and modes.
A.2.5 Entity capability requirements
A.2.5.1 Entity capability itemized requirements.

Like DI-IPSC-81431A, it calls out States and Modes separately from requirements. Within the description it states:

"A.2.4 Required states and modes. If the entity is required to operate in more than one state or mode having requirements distinct from other states or

modes, this paragraph should identify each state and mode. Examples of states and modes include idle, ready, active, post-use analysis, training, degraded, emergency, backup, wartime, and peacetime. If states or modes are required, each requirement or group of requirements in this specification should be correlated to the states and modes. A table or other method may be used to depict this correlation."

It effectively calls out to relate the States and Modes to the requirements, but the definitions of States and Modes are once again lacking, and are just a string of examples.

The last standard to look at is an FAA example specification that I found on the web a long time ago, and is no longer available. It had many good features. It's structure template is:

3.0 Requirements
3.1 System (Item) Definition
3.1.1 Functional Layouts
3.1.2 Interfaces
3.1.3 Major Components
3.1.4 FAA/Government Furnished Equipment:
3.1.5 States and Modes
3.2 Performance

Within the instructions, it states:

"3.1.5 States and Modes: The only states or modes which must be listed are those that impact/change the requirements expected from the product to be delivered. Each state and mode must be defined as to when that state or mode stops and starts. All modes must be identified as to which state it occurs in unless there is only one state. This section defines how XXXXXX will recognize a state or mod, such as power applied to a given pin. If there are other procurement specifications on a single program they must use a common definition for a state or a mode. This should define the relation to any higher-level state or mode. States are those conditions that the system in which XXXXXX resides may create. As an example a site in which XXXXXX resides may be in a dormant state and this may change the requirements expected from XXXXXX. The change in requirements must be documented in this section or in the following sections. A mode is a selected condition for XXXXXX, which changes the requirements on the product. As an example XXXXXX may be placed in the "low Power" mode which may change the performance delivered. States and modes must also contain the duty cycle and timelines for system or equipment operation. These must include maintenance, training and operational testing activities. It will also include anything like a mission profile. This section will include details about the element life (life time in modes and states, shelf life, operating life). This section includes the number of changes of modes and states as well as the time between changes."

And . . .

."If states and modes are used then a performance matrix should be included here that state; "The requirements herein shall apply to states and modes as listed in Table 3.2. Those requirements not listed apply during all states and modes.""

Wow. That was most certainly written by someone who has a firm opinion and who cares. They added transitions and timing, and included the table. However the States and Modes definitions are still lacking.

So across the last 40 years or so, the standards in our industry are very weak in reference to States & Modes and requirements. Let's fix that.

DEFINITION OF STATES AND MODES

Referring back to the two example from the introduction, the Volkswagen radio and the GM air bags, it's easy to see that States & Modes can be useful in controlling functions. When should they be available and when they should not be available. So what is a useful distinction between a State and a Mode?

I don't really get into arguments about States and Modes definitions because I found something that works for Systems Engineering, for me. Both States and Modes are containers that define control of the functions of the system. The difference between the two is that a change in State happens from outside the System and a change in Mode happens from within the system. For the automotive example, putting in a key and turning it, is an external interface. The system has no control over it. That assumes that the car driver with the key is not considered as part of the system. This is valid, as the automobile engineer has no control over them from a skills and training perspective. If we were talking about the Space Station, where the engineer has control over the selection and training of the astronaut, then it would be more likely that the astronaut flipping a switch would be considered a Mode change and not a State change.

States and modes are then used to control the coincidentiality (perhaps a word I just made up) of functions. Which one can operate with others and which one cannot.

The simplest way to look at the relationship between States & Modes and functions, is as a table matrix, show in Figure 1, below. A couple of the standards recommended using such a table.

	State 1				State 2			
	Mode 1			Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
	Mode 7	Mode 8	Mode 9					
Function 1	On	On	On	On	On	On	On	
Function 2	On	On	On	On	On		On	
Function 3	On	On	On	On	On			
Function 4	On	On	On		On			
Function 5	On	On	On					
Function 6	On		On		On			
Function 7	On				On			On

Figure 1. States & Mode / Function Matrix

This approach allows you to see which functions are available under each State and Mode, and maybe more importantly, which functions are not available.

In reality, the table is much more complicated. The entries within the body of the table need to include:

- Which functions have to be on in that State or Mode
- Which functions have to be off in that State or Mode
- Which functions are enabled in that State or Mode (allowed to turn on/off at will)
- Which functions must be on simultaneously
- Which functions cannot be on or off simultaneously
- Which functions default to On or Off during a transition into or out of that State or Mode

And then there is even a more complicated aspect to the table related to the functions – partial performance. Modes where the function may be enabled, but either full performance in not allowed or not required. Essentially requiring the functional performance to be fully restated in those States or Modes. I have yet to see any literature delve into this, at the System level. Most of that detail is beyond the scope of this paper, the goal of which is to introduce a construct for documenting performance requirements that fits into the State & Modes / Function matrix concept in Figure 1.

REQUIREMENTS TEMPLATE

This brings us to the question, is there a way to specify requirements in a manner which maps easily into the States & Mode table? Historically, requirement statements have been written around a "shall" statement and a verb-noun combination. The vehicle (noun) *shall* accelerate (verb) from 0 – 60 mph within 5 seconds. Almost all the requirements training states begin with the verb – noun combination and then to add text to make it readable, and add the shall, so it's binding. In my opinion, that is going in the wrong direction. We need it less wordy, not more.

A revelation came to me while in the automotive industry. There were requirements for 0-30 mph, 0-60 mph, 50-70 mph, and 0-100 mph. Each one had a well written legible requirement statement with conditions and verification requirements. But aren't they different flavors of the same thing? Are they all performance parameters of an Accelerate Vehicle function? Can't they all be measured in one test?

Referring back to Figure 1, would it be better to include the four separate acceleration requirements in different rows, or just the one function, Accelerate Vehicle. As each of the entries in the body of the table would be the same (on / off, etc), it clearly is better to just include the function one time.

That leads to the requirements template of Figure 2, which was partially taught to me by Jozef Bedocs. The construct rolls up the related Performance Parameters under a single Functional Statement, which can easily be put into the States & Modes table, simplifying it tremendously.

3.2.x Functional Statement
3.2.x.1 Performance Parameter 1
3.2.x.1.1 Conditions
3.2.x.2 Performance Parameter 2
3.2.x.2.1 Conditions
3.2.x.3 Performance Parameter 3
3.2.x.3.1 Conditions
3.2.x.4 Performance Parameter 4
3.2.x.4.1 Conditions

Figure 2. Requirements Template

Each functional statement (i.e. Accelerate Vehicle) can have multiple performance parameters underneath them, with perhaps separate conditions for each of them. For Accelerate Vehicle there may be a wet, dry, soft soil, fully loaded, or other performance parameters that have their own unique conditions. However, for a States & Modes standpoint, they are likely only going to be active during a Power On State and within a Self-Powered Mobility Mode. An effective, and well-communicated, top-level States & Modes increases the effectiveness of the engineering organization and allows effective decisions across the organizations. For example, If you are not in the Self-Propelled Mobility Mode, do you need to have the engine and transmission control units powered? Having them turned off can save electrical energy.

Looking back at older military vehicle specifications you see headings of Shoot, Move, Communicate, and Survive. These were treated merely as headings, but make pretty good top level functions, under which to put a performance parameters. A few efforts have show you may only have to go one or two levels from there to have a very effective, top level set of functions.

SUMMARY

This paper has introduced:

A working States & Modes definition
A States & Modes Matrix with Functions identified
A template for requirement statements

These are three related items that allow specification at all levels to be done more quickly, more concisely, and consistently. You are encouraged to learn more about them and try it on your next program.

Approved for public release, LogNo. 2015-33, Distribution Unlimited, 06/07/2015.