On a Useful Taxonomy of Phases, Modes, and States in Systems Engineering

Mike Ryan
UNSW Canberra

On a Useful Taxonomy of Phases, Modes, and States in Systems Engineering

Anthony M. Olver & Michael J. Ryan
School of Engineering & Information Technology,
University of New South Wales at the Australian Defence Force Academy
Northcott Drive, CANBERRA, ACT, 2600
anthony.olver@me.com & m.ryan@adfa.edu.au

'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. If the system operates without states or modes, this paragraph shall so state, without the need to create artificial distinctions.'

MIL-STD-498B Data Item Description DI-IPSC-81431A (2000)

Abstract. Many system development methodologies require that the system is described in its various states and modes. While all would agree on the utility of using states and modes to describe a system, there is little guidance on the literature as to what constitutes a state or a mode, which is made worse by poor usage of the terms such that it is not commonly agreed as to whether states include modes, or vice versa. The aim of this paper is to establish a basis for research by considering a number of aspects around the use of phases, modes and states including: their definition; the need for such descriptions; areas where confusion may exist; a brief examination of a number of existing models; common problems encountered when implementing such models; and the need for a useful taxonomy. The paper concludes with an outline of areas of further investigation that are required before a suitable methodology can be developed for the use of phases, modes and states in system development.

INTRODUCTION

Many systems engineering processes require the development an Operational Concept Document (OCD) and, based on that OCD, the development of a Functional and Performance Specification (FPS) for the capability (DMO FPS Development Guide, 2011). The FPS not only describes the system's behaviour when the system is being used normally, but also often describes how the system should behave when operating under degraded conditions, or when the system is used in different ways. The accepted method used to describe how the system should function or behave beyond the norm is through the use of the concept of states and modes.

The use of states and modes in formal systems engineering is required by a number of systems engineering standards. By example DI-IPSC-81431 System/Subsystem Specification Data Item Description contained within (MIL-STD-498, 1994) states that "If the system is required to operate in more than one state or mode having requirements distinct from other states or modes, this paragraph shall identify and define each state and mode." DI-IPSC-81431 then provides a number of examples of states and modes. The Defence Materiel Organisation (DMO) FPS Development Guide states that "the OCD should have identified all of the applicable external and internal states and modes for the solution-class(es); however, the OCD may not have completely identified either the mapping of the functions and associated performance against each of the identified states and modes, or state or mode transition requirements and constraints, including the transition sequence and stimulus that triggers the transition." (DMO FPS Development Guide, 2011)

A number of inconsistencies exist around the definition and use of states and modes within the systems engineering field. US Department of Defense military standards, for example, do not adequately define nor provide sufficient guidance in the use of States and Modes. MIL-STD-498B states that the distinction between states and modes is arbitrary (DI-IPSC-81431A, 2000). The DMO FPS Development Guide (2011) requires that states and modes be used, but does not clearly define their meanings and is lacking in the level of guidance given to define how they should be developed for use within the Defence FPS structure. The INCOSE Handbook (2010) does not provide a definition for states and modes (SMSC, 2005). IEEE 610.12-1990 confuses the issue even more by defining state as a condition or mode of existence that a system, component or simulation may be in (IEEE 610.12, 1990). These inconsistencies can lead to poor or incomplete specifications if states and modes are used incorrectly which in turn could result in project delays or cost increases as the missing requirements are later incorporated into the system design.

Although states and modes are described in various standards and in textbooks, there is little in the way of academic research to investigate the different ways in which states and modes are defined and used within the systems engineering discipline. (Edwards, 2006) and (Wasson, 2007) have previously identified some of the anomalies around the use of states and modes, and have both suggested frameworks that could be used practically. Harel's statecharts have been the focus of a number of authors, including (Glinz, 2002 and 2005), (Davis, 1990), and (Laplante, 1993) and could also provide a basis for further development.

By example, the current methodology provided in the DMO FPS Development Guide does not provide a sufficiently detailed and comprehensive description of how to effectively define functional and performance requirements within a state and mode based specification.

This paper aims to establish a basis for research by considering a number of aspects around the use of phases, modes and states including: their definition; the need for phases, modes and states; areas where confusion may exist; a brief examination of a number of existing models; consideration on the use of a phase, mode and state-based model during the acquisition phase; common problems encountered when using state and mode based models; and the need for a useful framework for the use of phases, modes and states in systems engineering.

WHY DO WE NEED PHASES, MODES AND STATES?

One of the key benefits of using a phase, mode and state methodology is realised when acquisition organisations are attempting to describe requirements that exist outside the normal operating environment. By example, consider a system or subsystem that has failed in some way. The user requires the system to be restored to normal within a certain time frame. The user may also require fault logs to be sent to a diagnostic location following the system recovery. Both of these examples could be requirements placed on a system when it transitions from a failed state to a normal operating state. The initial failure of the system could be constrained by derived availability requirements. For this example we have described two states, normal and failed, and a state transitions from one to the other and a state transition back again. A simple diagram is shown at Figure 1.

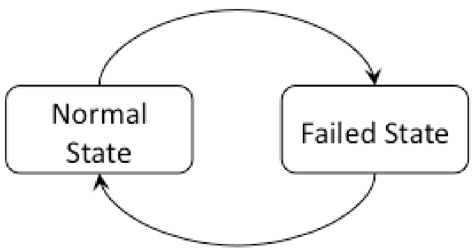

Figure 1. A Simple State Transition Diagram

The SMC Systems Engineering Primer and Handbook (2005) notes that the only reason for introducing states and modes is to provide a means to identify different sets of performance requirements for different sets of conditions that will be encountered by the system (SMSC, 2005).

Another question that is not often broached is whether customer acquisition organisations should even be using a state and mode based specification, or whether states and modes should remain the domain of the system developer. (Wasson, 2007) wrote that "when a system acquirer develops modes and states-centric specifications, they have crossed the line form specifying what is to be accomplished to specifying how to design the system" which can potentially preclude suppliers from tendering. He does point out that exceptions will exist. We recognise that a complex system can exist in one of a number of possible states, and that within each of these states functional and performance requirements may differ.

The existence of different states implies that transitions between those states must be possible. As an acquisition organisation we may wish to define what functional and performance requirements exist in each state and define similar requirements for how the user may wish to trigger a transition. The acquisition organisation may also require certain performance requirements during transition such as "The system shall, when in the OFF STATE, and the ON/OF button is depressed, transition to the ON STATE in not less than 3 seconds."

If the acquisition organisation does not define required states and modes, and the transitions between them, the resulting specification may be incomplete. For many systems this can potentially have safety critical implications. By example, consider a networked system such as an Air Traffic Control system wherein one site has lost communications with the remainder of the network. Without a state based specification one may not be able to adequately define the requirements of the system as it recovers back to a networked state. This could potentially result in undesirable delays in synchronising critical aircraft flight data that may result in an incorrect air picture being displayed to Air Traffic Controllers.

CONFUSION ON THE USE OF STATES AND MODES

The various methodologies for states and modes do not provide a consistent message or framework for what constitutes a state or a mode. Some literature tells us that a state is the condition of a system and a mode is the condition of a system in a certain state when specific functions are valid (SMSC, 2005) (Jenney, 2010). (Aslaksen and Belcher, 1992) consider that different states correspond to different operating modes. (Wray, 1993) describes a boundary between modes and states that is based on partitioning between activities planned and directly controlled by humans (a mode) and those activities directed, in response, by the machine (a state). (Hatley and Pirbhai, 1988) astutely pointed out in 1998 that "theoreticians enjoy long philosophical discourses on the exact meaning of state."

EIA-632 Process for Engineering a System requires the establishment of states and modes as appropriate to the system products, but does not provide guidance on, or definition of, their usage within the model (ANSI/EIA 632, 1999).

Some literature adopts a 'modes-in-states' taxonomy (SMSC, 2005) (Bellagamba, 2012) where other literature adopts 'states in modes' (DMO FPS Development Guide, 2011). This inconsistency adds to the lack of clarity. (Wasson, 2006 and 2007) introduces the concept of phases to his methodology that relate to mission-based operations. Harel developed Statecharts to describe reactive systems (Harel, 1987, 1997, and 2009).

The DMO FPS Development Guide (2011) provides an example of a State Transition Diagram that shows four different modes with a number of transitions between them. The figure itself does not depict any states at all. This inconsistency is not an isolated incident within the Development Guide.

Appendix A to this paper provides a comparison of the definitions of states and modes given within various pieces of literature, including an indication where states and/or modes are not defined. The wide variation of definitions demonstrates that no consistent structure exists.

EXISTING METHODS OF PHASES, MODES AND STATES PRESENTATION

A number of different models have been developed which either directly incorporate states and modes in their implementation or use similar methodologies such as statecharts to help describe the behaviour of the system. This section of the paper briefly describes some of the key characteristics of a number of such models.

Statecharts

Harel's Statecharts have been presented in various pieces of literature since the model was first published. Of particular relevance in this instance is his 1987 paper that applies statecharts to a Citizen Quartz Multi-Alarm III digital watch. This model does not define modes as part of its diagrammatical solution, but instead takes modes described in the user guide and integrates them into the overall state-based model. Unlike conventional state transition diagrams, statecharts use a formal methodology. Statecharts are described by Harel as an extension to conventional state transition diagrams through the inclusion of hierarchy, concurrence and communication. One differencing factor with statecharts is that the states may be exclusive, orthogonal or ancestral (Harel, 1987) unlike many other models that define states to always be mutually exclusive.

Phases, Modes, and States

Wasson presented a paper to INCOSE in 2010 that introduced the concept of phases alongside the more traditional model of states and modes. 'State' was also broken into system state, operational state, dynamic state and physical state. All human-made systems have at least three phases according to Wasson's model (2006 and 2007). These phases are defined as pre-mission, mission and post-mission. Incidentally, Wasson prefers a 'modes in states' approach (Wasson, 2007). Further sub-phases also exist under this model.

Operational Phases

Although not specific to states and modes, Chestnut's 'discrete operation' model has relevance where it utilises operational phases as demonstrated through an example showing a Re-entry Vehicle Mission that comprises three discrete system phases (launch, space flight and return) each broken into sub phases (Chestnut, 1965). Chestnut also describes a 'continuous process' system in which the system operates over a long period of time performing the same set of functions (Chestnut, 1965).

States and Modes

The DMO FPS Development Guide adopts an approach whereby a system will exist at any given time in one of a number of system states. Examples of a state may include operational state, training state or maintenance state. Within each state there exists one or more modes. Examples given include normal, degraded, emergency or diagnostic modes. This model requires not only that states and modes be defined for the operational system, but also for the support system. The model uses state/mode tables (see Figure 2) to show coexistence or otherwise and state transition diagrams (see Figure 3) to show transition sequences (DMO FPS Development Guide, 2011). This model is reasonably consistent with the definitions provided in MIL-STD-498B.

State/Mode	Operational State	Training State	Maintenance State	Normal Mode	Degraded Mode	Emergency Mode	Diagnostic Mode
Operational State		O	O	X	X	X	X
Training State	O		O	X	O	X	X
Maintenance State	O	O		X	O	O	X
Normal Mode	X	X	X		O	O	X
Degraded Mode	X	O	O	O		X	X
Emergency Mode	X	X	O	O	X		X
Diagnostic Mode	X	X	X	X	X	X	

Figure 2. An Example State/Mode Table
[Source: DMO FPS Development Guide, 2011]

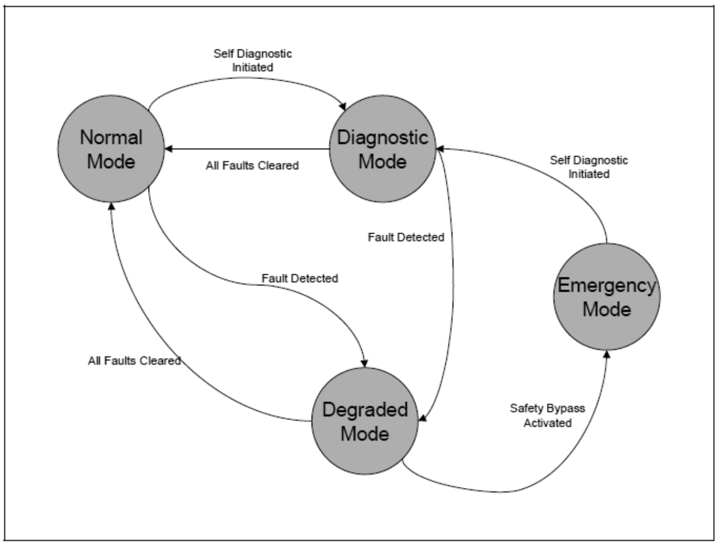

Figure 3. An Example State-Transition Diagram
[Source: DMO FPS Development Guide, 2011]

Flow Graphs

(Van Court Hare, 1967) introduces flow graphs to model state-descriptive systems. The flow graphs are similar in nature to later developed state transition diagrams, although in this instance the diagrams are also accompanied by a 'to-state' versus 'from-state' transition matrix that creates a model used to represent a probabilistic state-descriptive system. The matrix elements contain percentage likelihoods of that event occurring unlike the DMO FPS Development Guide state/mode matrix that shows mutual coexistence or otherwise in each element. An example flow graph and transition matrix for a simple deterministic state-descriptive system is shown at Figure 4. The DMO model for states and modes does not depict transition probabilities (DMO FPS Development Guide, 2011) and although the transition matrix and graph are useful, the probabilistic aspect does not have relevance in a plai language specification and so will not be considered further.

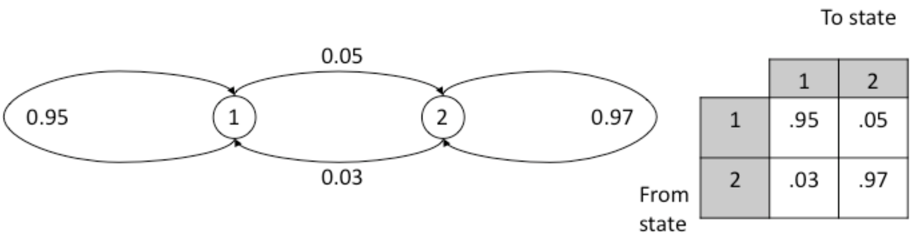

Figure 4. An Example Flow Graph and Transition Matrix
[Source: Van Court Hare, 1967]

UML State Machines

UML uses two different types of state machines. Behavioural state machines are used to specify the behaviour of various model elements and Protocol state machines are used to express usage protocols (Object Management Group, 2009). The Behavioural state machine is an object-based variant of statecharts (Object Management Group, 2009). The basic elements of a UML state machine diagram are states which describe what is happening within a system at a given point in time, transitions which depict how to transition between states, and events which describe what message is passed on the transition (Holt, 2004). Each state models a period of time during the object's life wherein it satisfies certain conditions. Under this model state machine diagrams can be used to describe user interfaces, device controllers or other reactive systems. The state machine can also be used to describe passive objects as they transition through a number of qualitatively distinct phases over their lifetime wherein each phase has distinct behaviour (Rumbaugh, et. al., 2005). A sample state machine diagram is shown at Figure 5.

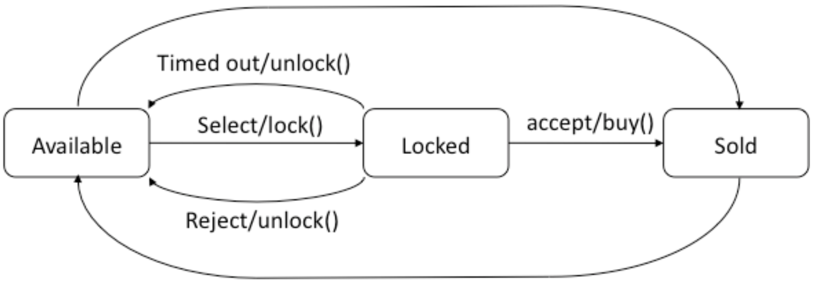

Figure 5. UML State Machine Diagram
[Source: Rambaugh et. al., 2005]

Model-based Systems Engineering

(Wymore, 1993) describes "modes of behaviour" as part of his model-based systems engineering model. Wymore asserts that components of a computer system such as printer, screen, keyboard, disk drive, hardware logic and memory all exist and function as part of the computer system at all times. He also identifies that transient modes will exist including "off", "on", "waking up", "waiting", "reading a diskette", "writing to a diskette", "computing", "printing" and "down."

This model is based on an existing system component having the desired functionality (or system mode) only when it receives a correct external input. This is represented by a component that has the system mode and a second component that generates inputs to stimulate the first component to exhibit the desired system mode. That is to say that the concept of system mode is a relationship between two systems.

Figure 6 shows Wymore's system mode as a system within a system. (Wymore, 1993)

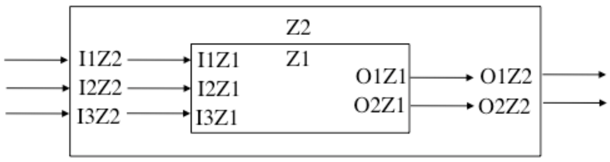

Figure 6. Wymore's System Mode as a System Within a System
[Source: Wymore, 1993]

Temporal Logic Modelling

(Hoffer, et. al., 1996) present a discussion on the modelling of temporal logic using state-transition diagrams and tables stating that for on-line and real-time applications analysts will often use state-transition tables and state-transition diagrams as a supplement to other modelling techniques. Object-oriented systems analysis and design, as well as process-oriented system development, are presented as possible applications of state-transition tables and diagrams. This model (from an object-oriented viewpoint) considers that a state consists of all the static properties of an object and the dynamic values of those properties. Transitions are triggered by events such as user mouse input or the system clocking over to a new day.

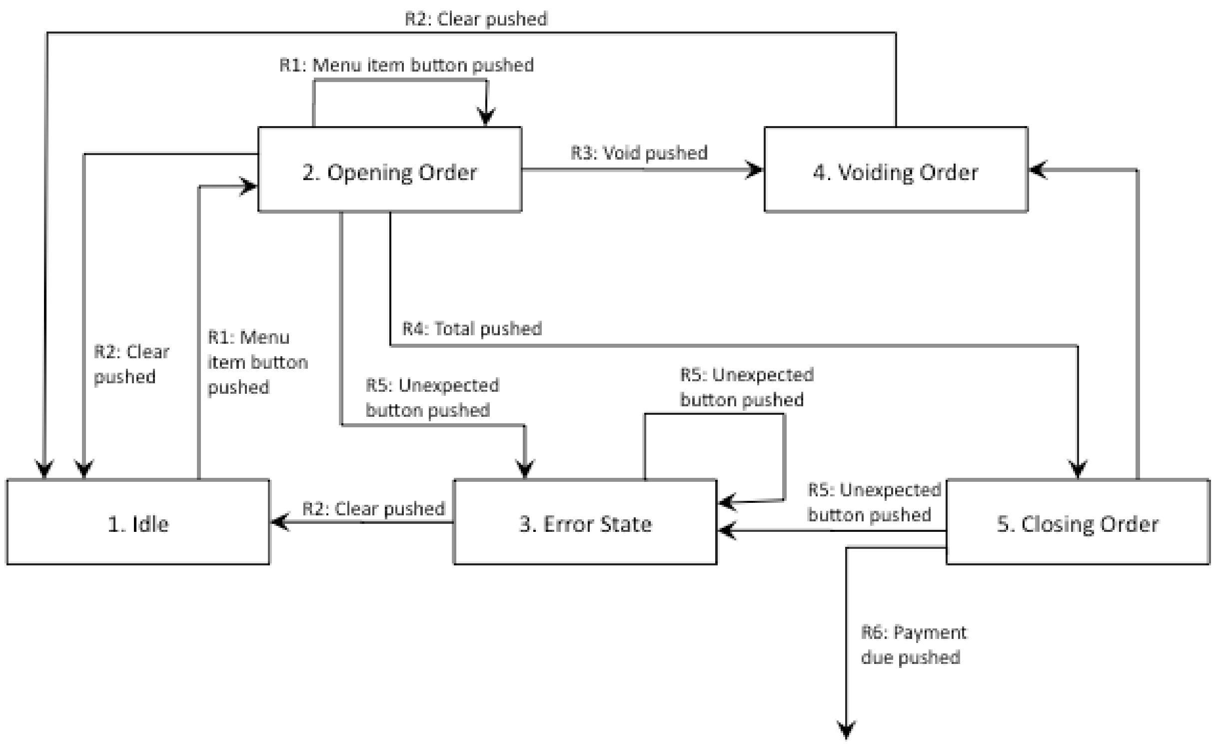

Figure 7. Example State-Transition Diagram Showing Hoosier Burger's Food-ordering System
[Source: Hoffer, et al, 1996]

Markovian Systems and the Probability of Transition

Although many models consider transitions between states or between modes, the probability of such a transition occurring is not usually considered. A number of authors have used Markov chain theory to develop models that account for such a probability when describing those aspects of a continuous system.

(Aslaksen and Belcher, 1992) consider the transitional aspect between different states by describing a Markov process that, although similar to Markov chains, also includes time as a variable to account for those systems that exist within a continuous time frame.

(Machol, 1965) presents a model where an operational system can exist in a number of possible states that will change over time as the system reacts to external and internal forces as they act upon it. Machol applies the Markov chain theory to continuous systems by setting an infinite number of discrete time periods *dt* in length.

Finite State Machines

(Hatley and Pirbhai, 1988) present finite state machine models as being suitable for representing the behaviour of a complex system due to the effect of both internal and external, and past and present events, on the system's behaviour. State machines are defined as either combinational or sequential.

Combinational state machines are those whose output is uniquely determined by its current input. Figure 8 shows a decision table for a heating system. Of particular note in this diagram is the use of the term 'Mode' in the title of the left-most column identifying the mode of the heating system as an input to the model (Hatley and Pirbhai, 1988).

INPUTS			OUTPUTS	
MODE	TEMP	PRESSURE	HEATER	PUMP
Idle	High	High	Off	Off
		Low	Off	Off
	Low	High	Off	Off
		Low	Off	Off
Auto1	High	High	Off	Off
		Low	Off	On
	Low	High	On	Off
		Low	On	On
Auto2	High	High	Off	Off
		Low	Off	On
	Low	High	On	Off
		Low	On	Off

Figure 8. Condensed decision Table for a Heating System
[Source: Hatley and Pirbhai, 1988]

In their discussion on finite state machines, (Hatley and Pirbhai, 1988) posit that most nontrivial systems are sequential machines citing the example of a flight management system. Sequential machines are those whose outputs are determined by both current and past inputs and are described as having 'memory' represented in the form of states. Similar to the conventional representation of states and modes, a state machine is always in one of its defined states. Figure 9 shows a state transition diagram for an example system with its associated state transition table shown at Figure 10. The associated state transition matrix is shown at Figure 11.

The state-transition diagrams presented by Hatley and Pirbhai comprise states, transitions between states, input events and output actions. The state-transition table and matrix contain the same information, although it is presented in a different format.

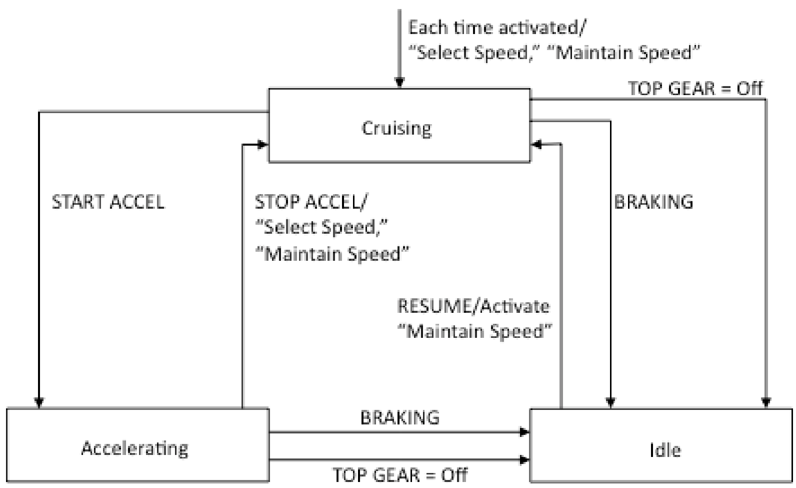

Figure 9. Example State-Transition Diagram for a Sequential Machine
[Source: Hatley and Pirbhai, 1988]

Current State	Event	Action	Next State
Start	Each time Activated	"Select Speed," "Maintain Speed"	Cruising
Cruising	TOP GEAR = Off	"	Idle
	BRAKING	"	Idle
	START ACCEL	"	Accelerating
Idle	RESUME	"Maintain Speed"	Cruising
Accelerating	BRAKING	"	Idle
	TOP GEAR = Off	"	Idle
	STOP ACCEL	"Select Speed," "Maintain Speed"	Cruising

Figure 10. Example State-Transition Table for a Sequential Machine
[Source: Hatley and Pirbhai, 1988]

From State` / To State	Cruising	Idle		Accelerating
Start	Each time Activated			
	"Select Speed," "Maintain Speed"			
Cruising		TOP GEAR = Off	BRAKING	START ACCEL
Idle	RESUME			
	"Maintain Speed"			
Accelerating	STOP ACCEL	TOP GEAR = Off	BRAKING	
	"Select Speed," "Maintain Speed"			

Figure 11. Example State-Transition Matrix for a Sequential Machine
[Source: Hatley and Pirbhai, 1988]

COMMON PROBLEMS WITH THE USE OF STATES AND MODES

Number of States

Consideration needs to be given to the number of states necessary to sufficiently define the system. (Harel, 1987) identifies that if two components each had one thousand states, then the resultant number of states in the product would be one million. Harel further proposes orthogonality to split states according to physical subsystems. The application of such a concept to systems engineering should be considered.

States and Modes Based Specifications

Most literature suggests that once the acquisition organisation decides to use states and modes in their specification they must ensure that all of the requirements are included within this structure (SMSC, 2005). If this is not done then there is a risk of an incomplete or inconsistent specification that could have cost, schedule, quality or safety implications.

A Methodology to Develop State-based Specifications

Although as we have seen there are many different ways in which state/ mode-based specifications can be represented, there are few detailed methodologies which provide a sound development framework to assist the acquisition organisation to develop the state based specification. This research intends to address that problem as part of its proposed methodology to improve useability and repeatability of the methodology.

Static versus Dynamic

A static condition of the system could be defined as a state. A dynamic condition, such as 'shuttle re-entry' could be defined as a state, but may better be defined as an operational phase, for example the re-entry phase. One could reasonably consider that a shuttle could, during the re-entry phase, be in a degraded state. Such a differentiation could be established by examining the temporal effects on the system with no new external influences. Those already present would continue to act. By example if the shuttle was in the 're-entry phase' and no external changes were applied, the shuttle would likely crash. If the shuttle were in a degraded state, all things being equal, it would continue to remain in a degraded state until an external influence was applied to the system. A shuttle re-entry system would be considered to be a dynamic system. Systems such as a power station would be considered a static system.

A POSSIBLE FRAME WORK FOR THE USE OF PHASES, MODES AND STATES IN SYSTEMS ENGINEERING

The states and modes framework outlined in the DMO FPS Development Guide lacks detail and contains conflicting information regarding the nature of states and modes. There is no 'step-by-step' guide provided that details how to develop a phase/mode/state-based specification. The inconsistent usage of the terms 'state' and 'mode' do not aid the specification developer in understanding the approach. This research aims to develop a viable framework for use in the development of phase/mode/state-based functional and performance specifications within systems engineering.

The models identified in this paper form a core sample set from which key elements can be considered for inclusion into the new methodology. The new methodology will be focused on usability with a particular focus on its use by an acquisition organisation in the development of a phase/mode/state-based functional and performance specification.

A number of characteristics will be considered as part of the development of the proposed methodology including:

- The definition of phases, states and modes as applicable to the proposed methodology;
- The temporal aspect of phases, states and modes (e.g. Hoffer et. al., 1996);
- The level of complexity in the proposed methodology versus the benefit gained;
- The practical limitations on the number of phases, states and modes within a system definition;
- Static systems such as power stations or air traffic control systems versus dynamic systems such as aircraft or land vehicles;
- Continuous systems versus discrete systems;
- A sound guide for the practical use of the proposed methodology; and
- Graphical representations and/or models used as part of the proposed methodology.

A number of definitions of phases, states and modes are listed at Table 1 where they have been further grouped into the following three categories:

- States within Modes;
- Modes within States; and
- Other.

The 'Other' grouping contains those definitions that: do not defined states and modes; define only a state or a mode or; provide a definition that does not clearly distinguish between, or clearly describe the relationship between, states and modes.

There is no clear preferred method of relating states and modes amongst those reviewed in this paper. The majority of sources do not provide any direct correlation between states and modes and some even consider them to be the same thing. In order to develop a useful taxonomy it is important to make a distinction between the various levels of abstraction in the model. One could choose to have level one, level two, level three, etc, however this does not provide a plain-English taxonomy that would assist systems engineers in their discussions with the end-users. To that end we have elected to define a deliberate relationship between phases, modes and states such that phases contain modes, and modes contain states.

Phases broadly relate to the higher-level user-centric language to describe the operational uses of the system or subsystem and should typically relate to the system in the context of user developed operational scenarios. In the case of an aircraft for example it could be considered to have phases such as ‘taxiing’, ‘take off’, ‘cruising’, and ‘landing’. Within these phases, the system or subsystem under consideration would exist in a number of modes. Each mode may exist in one or more phase, or may be deliberately excluded from operating in a particular mode in a given phase. For example the landing gear subsystem may have a ‘engaged’ mode where the landing gear is deployed for taxiing, take off or landing, and a ‘cruising’ mode where the landing gear is stowed during flight. An ‘up/down’ sensor switch may be in an ‘open’ or ‘closed’ state to indicate which mode the landing gear is operating in.

Through this example one can see that phases would relate to user defined operational levels for systems or subsystems. Modes relate to functional and performance requirements such as ‘the landing gear shall have an engaged mode’ and ‘the landing gear shall have a cruising mode’. States relate to the actual implementation of the design. In the aircraft example the number and type of states that the switch (or sensor) has would be determined by the system designer such that a higher level functional requirement such as ‘the aircraft shall indicate to the pilot when the landing gear is engaged’.

CONCLUSIONS

The use of states and modes in formal systems engineering processes has been shown to be wide and varied. They have been shown to exhibit inconsistent definitions and applications when applied to the definition of a system.

This paper has shown that there are wide ranging applications of (phase)/state/mode, or state/mode-like, models that can be studied in further detail to determine the potential applicability of various elements to the development of a suitable methodology for use in systems engineering. A number of models show potential in applicability or relevance to the development of a methodology for phases, modes and states in systems engineering. The key areas include:

- Statecharts provide that states may be exclusive (as per what one considers normal), orthogonal or ancestral, although the overall methodology may be too complex for a large-scale system definition.
- Phases provide a different viewpoint that would appear to complement a traditional state/mode based specification.
- The phase/mode/state methodology to be applied may be dependent on the discrete or continuous nature of the system.
- The concept that ‘state’ could relate to a system, operational, dynamic or physical state would allow for a more applicable assignment of ‘state’, although complexity may be increased.
- Complexity may be reduced through the use of orthogonality in the representation of more complex state diagrams.
- A probabilistic state-descriptive system could provide for percentage likelihoods of state transitions that may be useful in a state/mode-based methodology when considering the likelihood of entry to unwanted states.

The development of a functional and performance specification is important for communicating acquisition organisation requirements to suppliers and in establishing an initial baseline to support system specification development and subsequent system design. The ability to accurately, completely and effectively utilise phases, modes and states in the definition of functional and performance specifications could contribute to a more successful mutual understanding between customer and supplier of the required system functionality.

Afterword

Quote from the SMC Primer Handbook

The following extract from the SMC Primer Handbook provides an interesting viewpoint against the use of states and modes in systems engineering.

"The use of states and modes in system level requirements documents probably came into widespread use as a result of Data Item CMAN 80008A. This was the document that specified the format, content, and structure for A-Specs (system and segment level specs). However, trying to apply states and modes to an entire system may not have been a great idea. Often, while states and modes may make sense for a subsystem or element of a system, they would be difficult to apply (or meaningless) to the entire system. Although no longer mandated, some engineers still use states/modes within their requirements documents." (SMSC, 2005)

REFERENCES

ANSI/EIA 632 'Process for Engineering a System', EIA Engineering Department, Arlington, VA, 1999.

Aslaksan, E. and Belcher, R., Systems Engineering, Prentice Hall, 1992.

Bellagamba, L., Systems Engineering and Architecting, CRC Press, 2012.

Buede, D.M., The Engineering Design of Systems, John Wiley & Sons, Inc, 2009

Chestnut, H., Systems Engineering Tools, John Wiley and Sons, 1965.

Davis, A.M., Software Requirements Analysis and Specification, Prentice Hall, New Jersey, 1990.

Defence Materiel Organisation, DMH (ENG) 12-3-005 Function and Performance (FPS) Development Guide, 2011.

Department of Defense, Unmanned Systems Safety Guide for DoD Acquisition, First Edition v.96, 2007

DI-IPSC-81431A, MIL-STD-498B (Cancelled) Data Item Description, System/Subsystem Specification, 2000

Edwards, M.T., "A Practical Approach to State and Mode Definitions for the Specification and Design of Complex Systems", SETE Conference 2003.

Glinz, M., "An Integrated Formal Model of Scenarios Based on Statecharts", 1995.

Glinz, M., "Statecharts for Requirements Specification – As Simple As Possible, As Rich As Needed", Proceedings of the ICSE 2002 International Workshop on Scenarios and State machines: Models, Algorithms and Tools. Orlando, May 2002.

Harel, D., "Statecharts: A Visual Formalism for Complex Systems", in Science of Computer Programming, Vol. 8, pp. 231-274, North Holland, 1987.

Harel, D., "Some Thoughts on Statecharts, 13 Years Later", 1997.

Harel, D., "Statecharts in the Making: A Personal Account", Communications of the ACM, March 2009, Vol. 52, No. 3, 2009.

Hatley & Pirbhai, Strategies for Real-Time System Specifications, Dorset House Publishing, New York, 1988.

Hoffer, J.A., George, J.F., and Valacich, J.S., Modern Systems and Design Analysis, Benjamin/Cummings, 1996.

Holt, J., UML for Systems Engineering – watching the wheels, IEE, [ADD], 2004.

IEEE 610.12-1990 IEEE Standard Glossary of Software Engineering Terminology, 1990

INCOSE-TP-2003-002-03.2 Systems Engineering Handbook v3.2 Appendix D, 2010

Jenney, J., Modern Methods of Systems Engineering: With an Introduction to Pattern and Model Based Methods, CreateSpace Independent Publishing Platform, 2010.

Knowledge Based Systems, Inc., Information Integration For Concurrent Engineering (IICE) IDEF4 Object-Oriented Design Method Report, Air Force Systems Command, Ohio, 1995.

Laplante, P., Real-Time Systems Design and Analysis - An Engineers Handbook, IEEE Press, New Jersey, 1993.

Machol, R.E., Queues and Markov Processes – The Response of Operating Systems to Fluctuating Demand and Supply, System Engineering Handbook, chapter 28, McGraw-Hill Book Company, Military Standard, 'MIL-STD-498 Software Development and Documentation', Department of Defense (USA), 1994.

Object Management Group Unified Modelling Language OMG UML Superstructure Specification v2.2, 2009

Rumbaugh, J., Jacobson, I., and Booch, G., The Unified Modeling Language Reference Manual, Addison-Wesley, 2005.

SMSC (Space and Missile Systems Center), SMC Systems Engineering Primer and Handbook, 3rd Edition, U.S. Air Force, 2005.

Wasson, C.S., System Analysis, Design, and Development: Concepts, Principles, and Practices, Wiley, 2006.

Wasson, "System Phases, Modes, and States – Solutions to Controversial Issues", Presented at INCOSE 2011, 2010.

Wray, R.B., An Analysis of the Modes and States for Generic Avionics, Lockheed Engineering and Sciences Company, Houston, Texas, 1993.

Van Court Hare, Systems Analysis A Diagnostic Approach, [ADD], 1967.

Wymore, A.W., Model-Based Systems Engineering, CRC Press, 1993.

BIOGRAPHIES

Anthony Olver is postgraduate doctoral student with the School of Information Technology and Electrical Engineering, University of New South Wales, Canberra, at the Australian Defence Force Academy. He holds a Bachelor degree in electrical engineering, and Masters degrees in systems engineering and in project management. He has over 20 years experience in systems engineering and project management within the Defence aerospace environment.

Dr Mike Ryan is a Senior Lecturer with the School of Information Technology and Electrical Engineering, University of New South Wales, Canberra, at the Australian Defence Force Academy. He holds Bachelor, Masters, and Doctor of Philosophy degrees in electrical engineering as well as a Graduate Diploma in Management Studies. He lectures and regularly consults in a range of subjects including communications and information systems, systems engineering, requirements engineering, and project management. He is the conference chair of two international conferences each year, he is the editor-in-chief of the Journal of Battlefield Technology, and is chair of the Requirements Working Group in the International Council on Systems Engineering (INCOSE). He is the author or co-author of ten books, three book chapters, and over one hundred technical papers and reports.

TABLE A-1. A COMPARISON OF THE DEFINITIONS OF PHASES, MODES AND STATES

Modes Within States				
Source	**State**	**Mode**	**Phase**	**Comment**
Bellagamba, 2012	Not defined. Mentions that states have subsets called modes.	Not defined. Mentions that modes are subsets of states.	Not used.	Modes within States.
Edwards, 2006	State: the overall condition of a system or a subsystem. - may be related hierarchically; are exclusive.	Mode: the way in which functions are performed. - exist within states. May be exclusive or concurrent; function execution related to modes	Not used.	Defines Modes within States. Also defines Condition: an attribute of the system at a particular point in time. May contribute to the derivation of a system's state or mode or transitions.
USAF SMC, 2005	A state is the condition of a system or subsystem when specific modes or capabilities (or functions) are valid.	A mode is the condition of a system in a certain state when specific capabilities (or functions) are valid.	Not used.	Defines Modes within States.
States Within Modes				
Source	**State**	**Mode**	**Phase**	**Comment**
Buede, 2009	A static snapshot of the set of metrics or variables needed to describe fully the system's capabilities to perform the system's function.	A distinct operating capability of the system during which some or all of the system's functions may be performed to a full or limited degree.	Not used.	Defines States within Modes.
Unmanned Systems Safety Guide (Department of Defense, 2007)	States identify conditions in which a system or subsystem can exist. A system or subsystem may be in only one state at a time. States are unique and may be binary. A state is a subset of a mode.	Modes identify operational segments within the system lifecycle generally defined in the Concept of Operations. Modes consist of one or more sub-modes. A system may be in only one mode, but may be in more than one sub-mode, at any given time.	Not used.	Defines States within Modes.
Wasson, 2006	"State of operation": The operational or operating condition of a system of interest required to safely conduct or continue its mission.	An abstract label applied to a collection of system operational capabilities and activities focused on satisfying a specific phase objective. (Note: Wasson also includes system phases in his model)	Phase of Operation A. high-level, objective-based abstraction representing a collection of SYSTEM OF INTEREST (SOI) operations required to support accomplishment of a system's mission. For example, a system has pre-mission, mission, and post mission phases.	Defines States within Modes.
Wasson, 2007	A state is an attribute used to characterize the current logistical status or performance-based condition of a	A mode is an abstract label applied to a user (UML Actor) selectable option that enables a set of use case-based	A phase is a designation applied to segments of the System / Product Life Cycle or Mission	Defines States within Modes.

	system, product, or service or system components at the element, subsystem, assembly, subassembly, etc. levels of abstraction.	system capabilities to be employed in conjunction with organizational processes and procedures to command and control a system, product, or service to achieve a specified set of mission objectives, outcomes, and levels of performance.	Lifecycle of a system, product.	
States, States in States, or Modes in Modes				
Source	**State**	**Mode**	**Phase**	**Comment**
Holt, 2004	States describe what is happening with the system at any point in time.	Not defined.	Not used.	Defines States only.
IDEF4 (Knowledge Based Systems Inc., 1995)	States described the value of object characteristics. Object states represent situations or conditions of an object instance that are meaningful in the design.	Not defined.	Not used.	Defines States only.
Machol, 1965	The operational system can exist in a number of possible states… As the system reacts to the external and internal forces acting on it, its state will change with time.	Not defined.	Not used.	Defines States only.
Rumbaugh, et. al., 2005	State is a condition or situation during the life of an object during which it satisfies some condition, performs some do activity, or waits for some event.	Not used.	Not used.	Defines States only.
Van Court Hare, 1967	The term state refers to a particular and discrete system condition.	Not defined.	Not used.	Defines States only.
Wray, 1993	States are the activities that a system or subsystem adopts in response to a mode input.	Modes are those activities planned in advanced and accordingly implemented along a timeline established by humans, or are activities directly under human control.	Not used.	Uses states within states and modes within modes.
States and Modes Not Defined or No Distinction Given				
Source	**State**	**Mode**	**Phase**	**Comment**
Aslaksan and Belcher, 1992	The state refers to the system configuration or to the condition of its components. Different states correspond to different modes.	Not defined.	Not used.	No clear distinction.
Chestnut,1965	Not used.	Not used.	Not defined, but used to describe the steps that a discrete operation, such as a space re-entry vehicle,	Phases are used in relation to identifying functional needs of a system.

			goes through.	
DI-IPSC-81431A, 2000	The distinction between states and modes is arbitrary.	The distinction between states and modes is arbitrary.	Not used.	States and modes are required under this model, but are not defined.
DMO FPS Development Guide, 2011	Not defined.	Not defined.	Not used.	States and modes are required under this model, but are not defined.
ANSI/EIA 632, 1999	Mentions states and modes but does not define.	Mentions states and modes but does not define.	Not used.	States and modes are not defined.
Glinz, 2002 and 2005	Not defined.	Not defined.	Not used.	States and modes are not defined.
Hatley and Pirbhai, 1988	State is synonymous with the mode or condition of the machine.	Not defined.	Not used.	No clear distinction.
Hoffer, et. al., 1996	A mode or condition of existence for a process or other system component, as determined by current circumstances.	Not defined.	Not used.	No clear distinction.
Harel, 1987, 1997 and 2009	Not defined.	Not defined.	Not used.	States and modes are not defined.
IEEE 610.12-1990	State is a condition or mode of existence that a system, component, or simulation may be in.	Mode is an operating condition of a function or sub-function or physical element of the system.	Not used.	No clear distinction.
INCOSE Handbook (INCOSE-TP-2003-002-03.2, 2010)	Not defined.	Not defined.	Not used.	States and modes are not defined.
Jenney, 2010	States define an exact operating condition of a system.	Modes define the set of capabilities or functions, which are valid for the current operating condition.	Not used.	Defines Modes and States as unrelated entities.
Laplante, 1993	Not defined.	Not used.	Not used.	Not defined.
MIL-STD-498, 1994	Not defined.	Not defined.	Not used.	States and modes are required under this model, but are not defined.