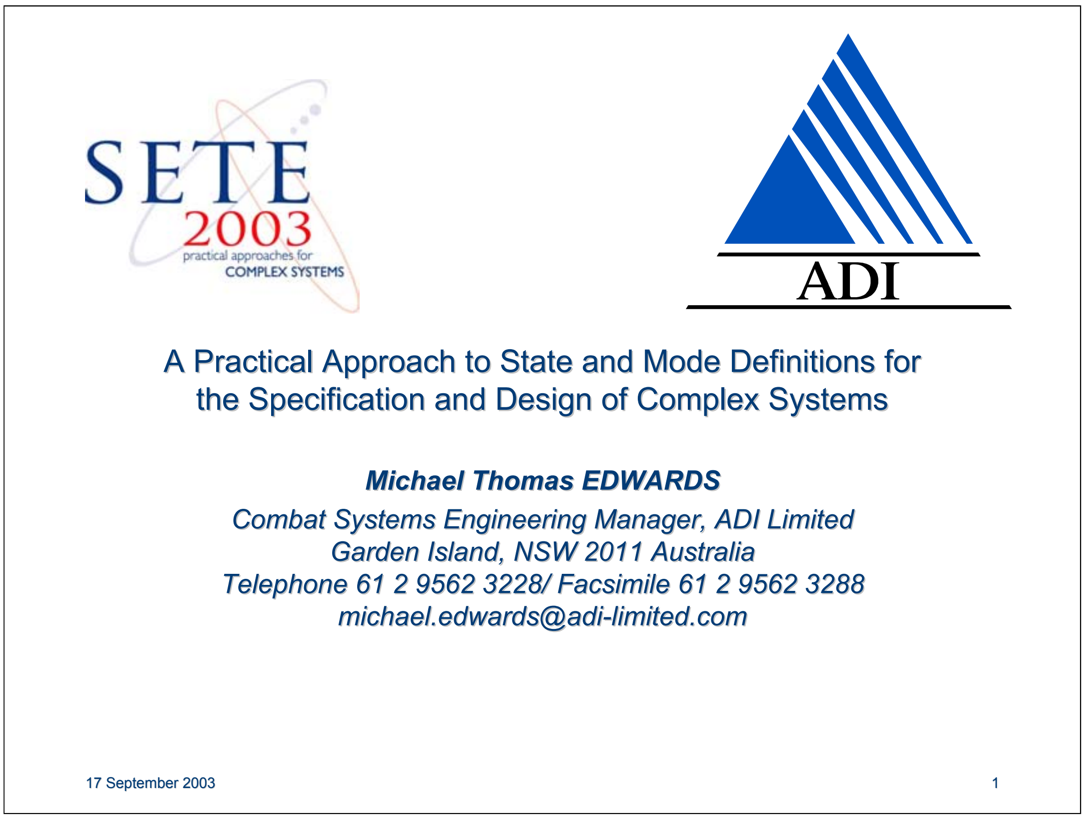


SETE 2003 practical approaches for COMPLEX SYSTEMS

ADI

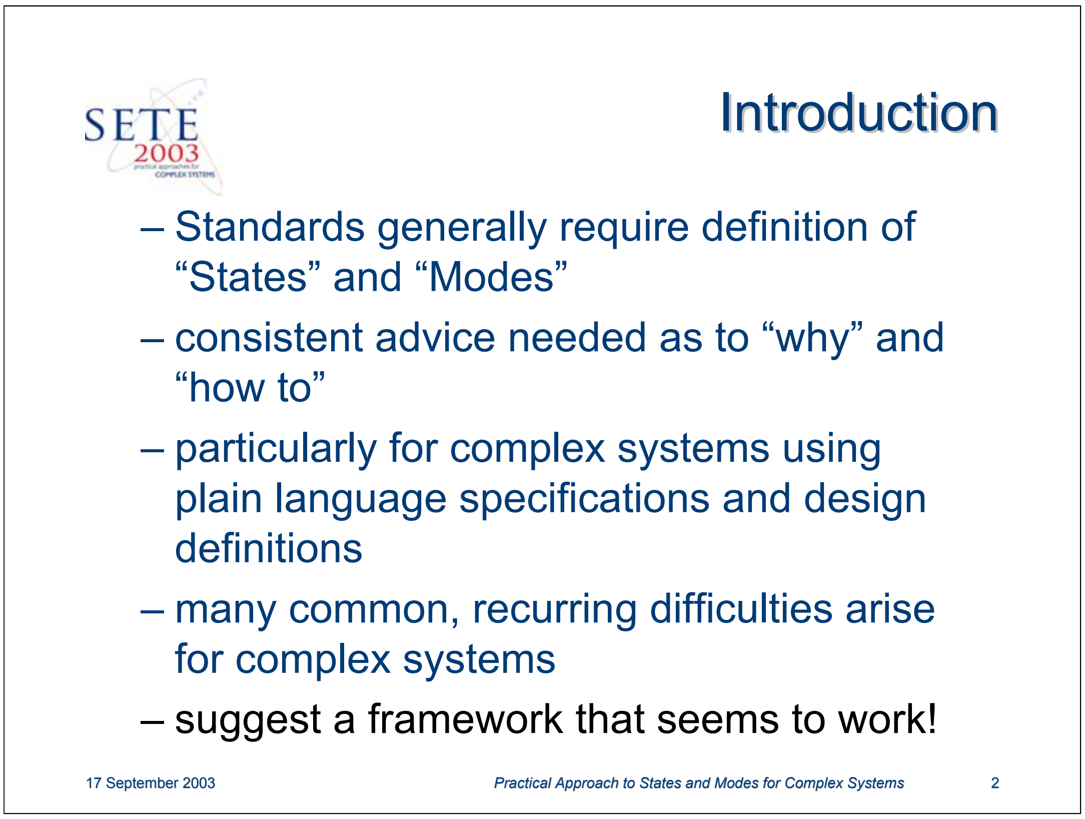
A Practical Approach to State and Mode Definitions for the Specification and Design of Complex Systems

Michael Thomas EDWARDS

Combat Systems Engineering Manager, ADI Limited
Garden Island, NSW 2011 Australia
Telephone 61 2 9562 3228/ Facsimile 61 2 9562 3288
michael.edwards@adi-limited.com

17 September 2003 1

Abstract: *Standards for system specifications and system design documents usually require the definition of the applicable states and modes of the system. However, little in the way of consistent advice of "why" or "how to" is available to the practitioner with the task of defining states and modes. This is particularly true in the domain of complex systems where formal specification methods are generally not in use and most system definitions are conducted using plain language. A practical approach to defining states and modes is presented which has been found to be useful in defining design and requirements for complex systems including where pre-existing subsystems with extant states and modes need to be incorporated into the functional definition of the overall system.*



Introduction

- Standards generally require definition of "States" and "Modes"
- consistent advice needed as to "why" and "how to"
- particularly for complex systems using plain language specifications and design definitions
- many common, recurring difficulties arise for complex systems
- suggest a framework that seems to work!

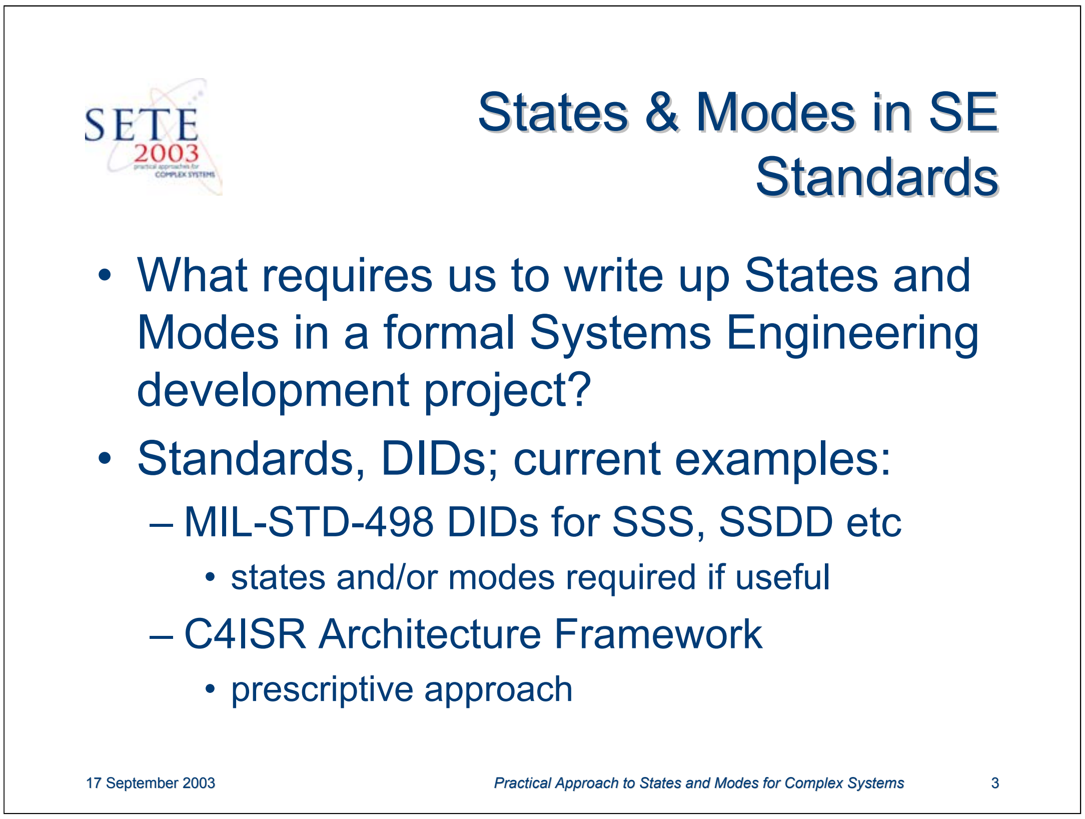
17 September 2003 *Practical Approach to States and Modes for Complex Systems* 2

Systems engineering standards and associated Data Item Descriptions (DIDs) for System Specifications and System Design Documents generally require definition of "States" and "Modes". This paper identifies and reviews a selection of these requirements.

In the author's experience, there seems to be little in the way of consistent advice or approach as to "why" and "how to" usefully define and use the concepts of states and modes. This is particularly true for complex systems where plain language specifications and design definitions remain the norm. These specifications and documents are the mechanisms through which the user, acquirer and developers reach consensus on the characteristics of complex systems under development, therefore a consistent and well defined approach is required to reduce misunderstandings between these stakeholders. A selection of formal and informal approaches are identified herein to demonstrate the lack of a common approach.

Many common and recurring difficulties arise in the specification and definition of complex systems. None of the problems are assisted by the lack of a common approach to states and modes. In this paper, a number of problems directly related to states and modes definition are identified.

Finally, a practical framework for states and modes is presented and applied to a representative system. The framework includes working definitions for states, modes, conditions, status and functions and recommends how to use these within System Specifications and System Design Documents.



States & Modes in SE Standards

- What requires us to write up States and Modes in a formal Systems Engineering development project?
- Standards, DIDs; current examples:
  - MIL-STD-498 DIDs for SSS, SSDD etc
    - states and/or modes required if useful
  - C4ISR Architecture Framework
    - prescriptive approach

17 September 2003 *Practical Approach to States and Modes for Complex Systems* 3

What requires us to write up States and Modes in a formal Systems Engineering development project? Generally most system/software engineering standards require the definition of states and modes. For example:

MIL-STD-498 DID for SYSTEM/SUBSYSTEM SPECIFICATION (SSS) DI-IPSC-81431:

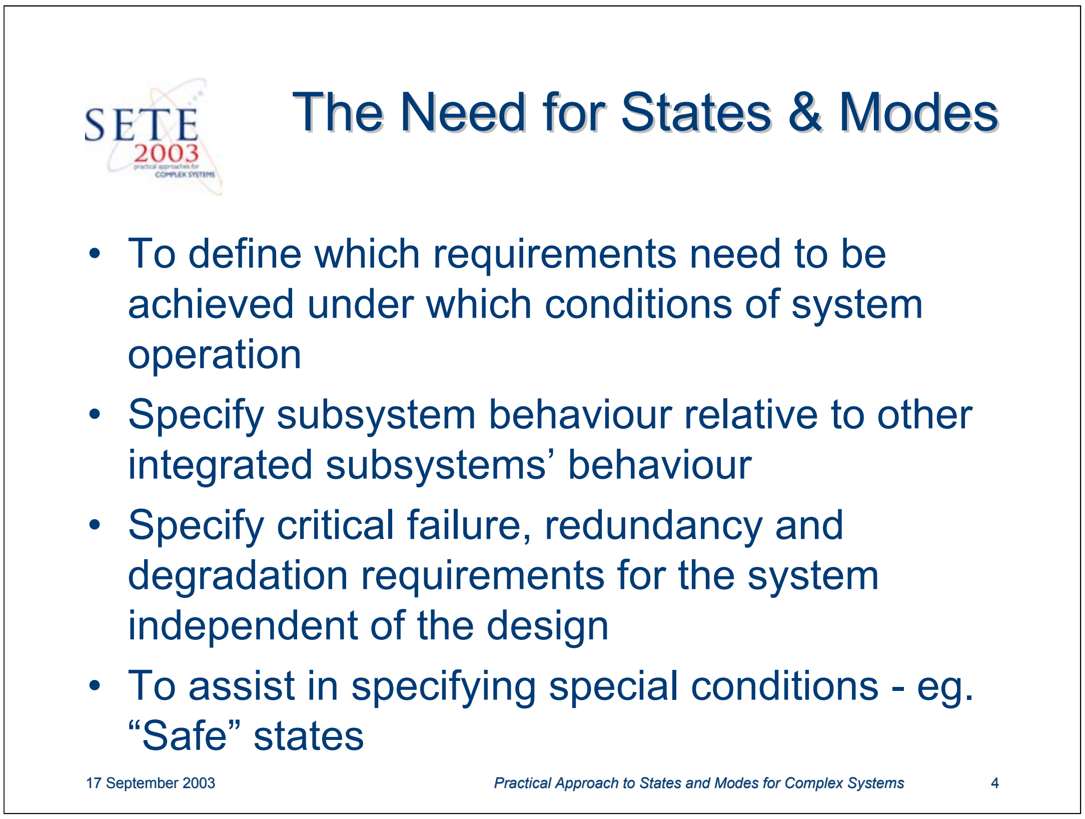
"3.1 Required states and modes. If the system is required to operate in more than one state or mode having requirements distinct from other states or modes, this paragraph shall identify and define each state and mode. Examples of states and modes include: idle, ready, active, post-use analysis, training, 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. If no states or modes are required, this paragraph shall so state, without the need to create artificial distinctions. If states and/or modes are required, each requirement or group of requirements in this specification shall be correlated to the states and modes. The correlation may be indicated by a table or other method in this paragraph, in an appendix referenced from this paragraph, or by annotation of the requirements in the paragraphs where they appear."

MIL-STD-498 DID for SYSTEM/SUBSYSTEM DESIGN DESCRIPTION (SSDD) DI-IPSC-81432

"4.2 Concept of execution. This paragraph shall describe the concept of execution among the system components. It shall include diagrams and descriptions showing the dynamic relationship of the components, that is, how they will interact during system operation, including, as applicable, flow of execution control, data flow, dynamically controlled sequencing, state transition diagrams, timing diagrams, priorities among components, handling of interrupts, timing/sequencing relationships, exception handling, concurrent execution, dynamic allocation/deallocation, dynamic creation/deletion of objects, processes, tasks, and other aspects of dynamic behaviour."

C4ISR Architecture Framework Version 2.0, Section 4.2.2.3.2 Operational Activity Sequence and Timing Descriptions - Operational State Transition Description.

"A state specifies the response of a system or business process to events. The response may vary depending on the current state and the rule set or conditions. The Operational State Transition Description relates events and states. When an event occurs, the next state depends on the current state as well as the event. A change of state is called a transition. Actions may be associated with a given state or with the transition between states."



The Need for States & Modes

- To define which requirements need to be achieved under which conditions of system operation
- Specify subsystem behaviour relative to other integrated subsystems' behaviour
- Specify critical failure, redundancy and degradation requirements for the system independent of the design
- To assist in specifying special conditions - eg. "Safe" states

17 September 2003 *Practical Approach to States and Modes for Complex Systems* 4

Why do we need to define States and Modes? States and modes need to be defined in a system specification/design document to allow:

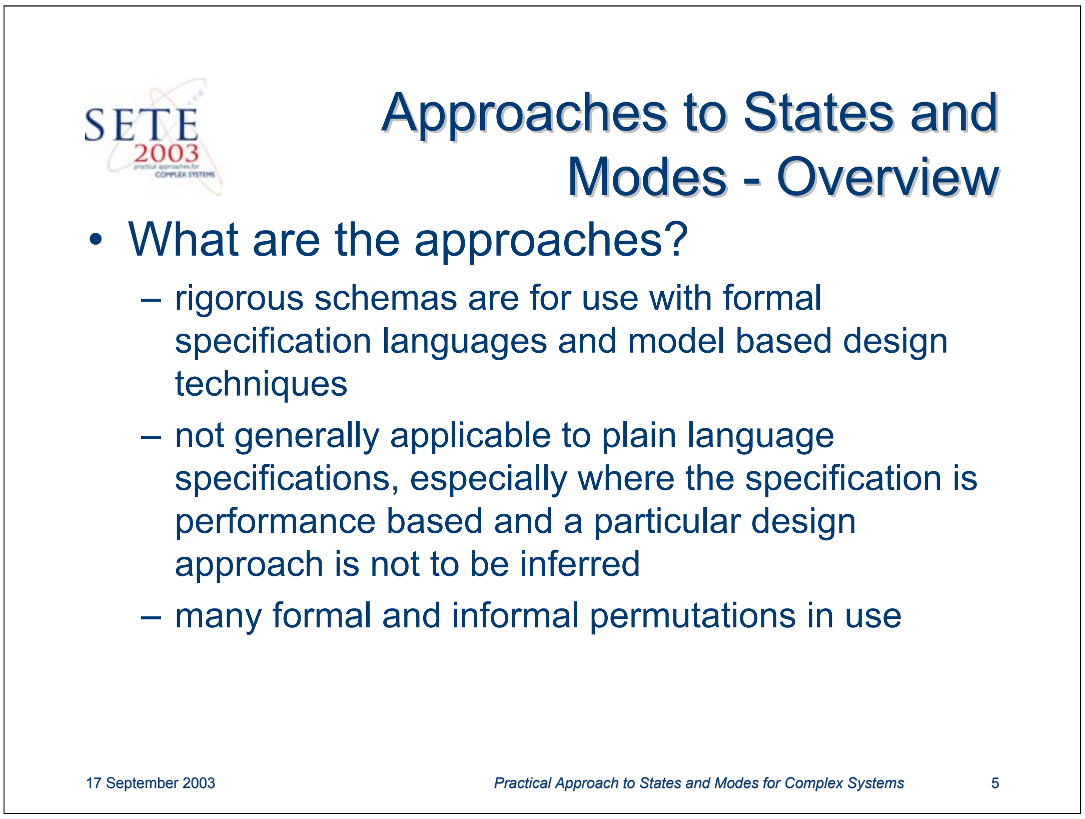
•Functional and performance requirements that do not need to be achieved under all conditions of system operation need to be specified relative to the defined states and modes. (e.g. the detection performance of a sensor may be specified for each search mode; Built In Test functions may be specified as being required during an "operational" state or just in a "maintenance" state).

•Requirements to be specified for subsystems that will define their behaviour relative to systems with which they are integrated. This includes the specifications of conditions under which systems may transition between the operating modes or states. The definition of status data exchanged between subsystems and which components are responsible for overall state determination (eg. A tracking system may need to pass both its own status and the status of integrated sensor systems on to a display system which determines and displays overall system status).

•As a means of defining required redundancy/allowed degradation/availability that does not pre-suppose system design characteristics (eg. "The operational state shall be maintained when at least n of m sensor subsystems are on-line.")

•Critical failures can be defined relative to which states and modes can be attained. (eg. We can define a MTBCF as being with respect to a failure which does not allow the "operational" state to occur; or does not allow the performance requirements of a "self defence mode" to be achieved.

•Special cases of system behaviour may be best defined using state and mode concepts. (eg. A "safe" state may require all mechanical movements of system equipment (eg antenna rotation) to be halted until error conditions are cleared.)



Approaches to States and Modes - Overview

- What are the approaches?
  - rigorous schemas are for use with formal specification languages and model based design techniques
  - not generally applicable to plain language specifications, especially where the specification is performance based and a particular design approach is not to be inferred
  - many formal and informal permutations in use

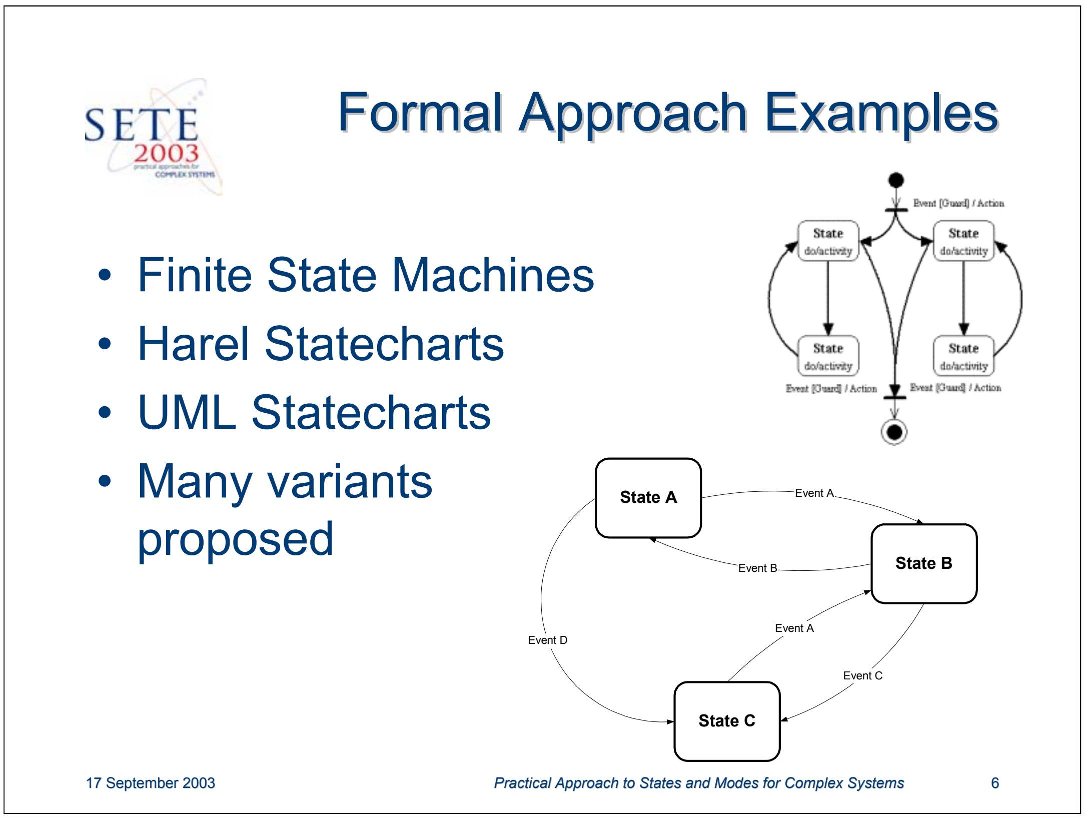
17 September 2003 *Practical Approach to States and Modes for Complex Systems* 5

What are the approaches to specifying/defining States and Modes?

Most rigorously developed examples and schemas are for use with formal specification languages and model based design techniques.

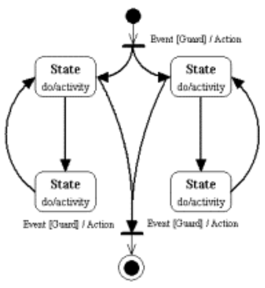
These approaches are not generally applicable to plain language specifications, especially where the specification is performance based and a particular design approach is not to be directed by the specifier nor inferred by the designer.

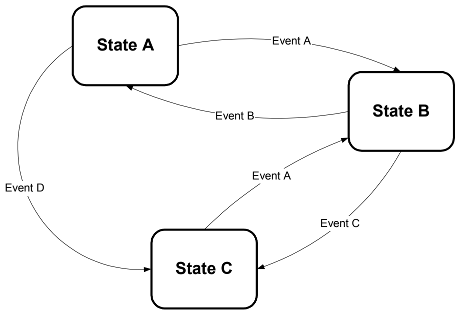
Many permutations have been put forward and are in use. It is not clear that any of these are more correct than any of the others. There is a potential to re-invent the wheel every time.



Formal Approach Examples

- Finite State Machines
- Harel Statecharts
- UML Statecharts
- Many variants proposed



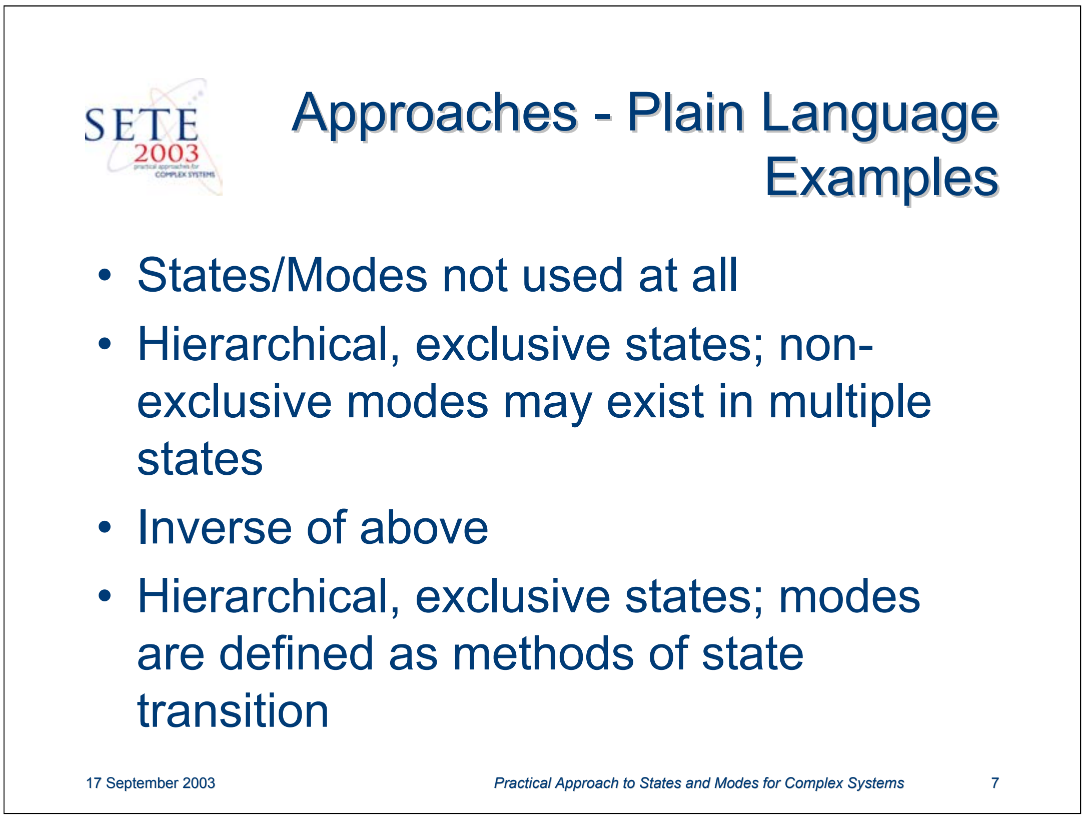


It has been well recognised since the early-mid 80s that the definition of states was important for the successful design of complex systems.

Landmark publications of that era by Harel [1] and its later extensions, led to definitions of Statecharts for formalising complex system behaviour. UML has adopted statecharts and refined their semantics.

Efforts continue to adopt these methods in a form suitable for requirements specification - the work of Glinz [2] is recommended reading.

Whilst these formal approaches are extremely useful for the detailed design of complex system components, their utility is limited during the overall concept definition stage of a system.



Approaches - Plain Language Examples

- States/Modes not used at all
- Hierarchical, exclusive states; non-exclusive modes may exist in multiple states
- Inverse of above
- Hierarchical, exclusive states; modes are defined as methods of state transition

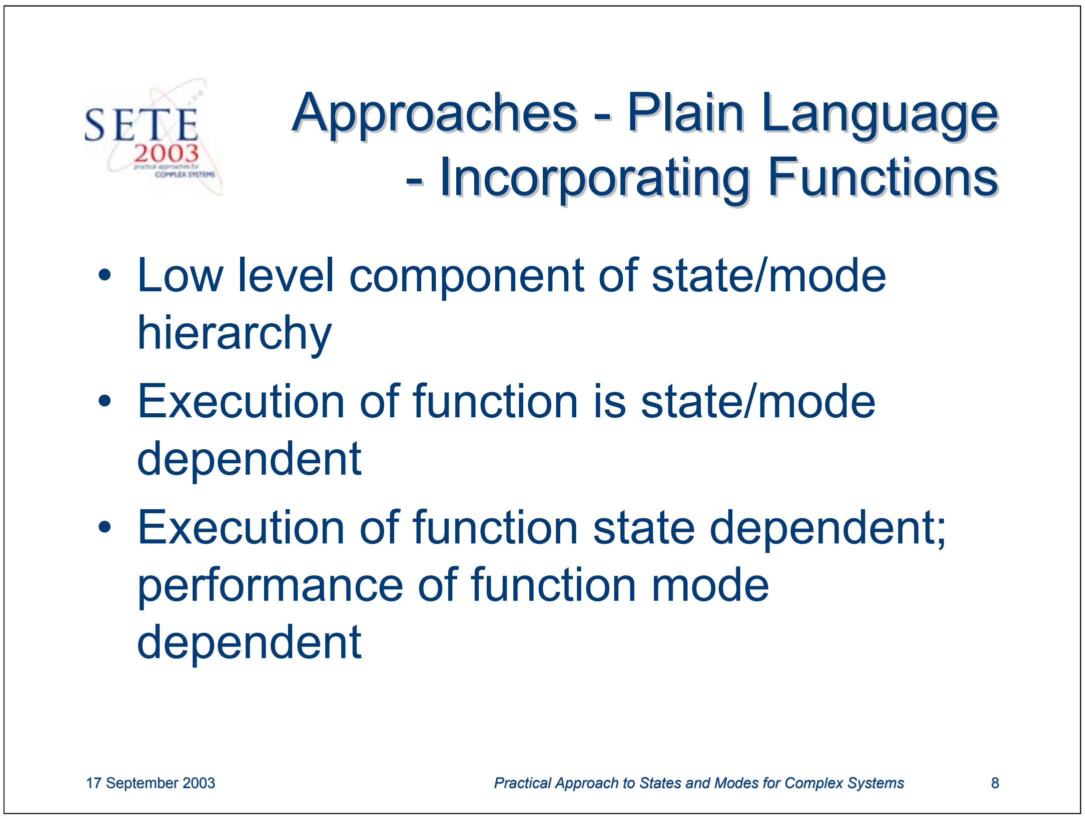
A non-exhaustive list of Plain Language Approach examples follows:

• States and modes are not defined or used at all

• States are mutually exclusive; substates may be related hierarchically within states. State transition diagrams useful for visualisation and defining transition behaviour. Modes exist within states; they are not mutually exclusive of other modes and may exist in more than one state.

• An inversion of states and modes in the aforementioned framework.

• States are mutually exclusive; modes are ways of achieving state transitions. That is, modes are the arrows on state transitions diagrams.



Approaches - Plain Language - Incorporating Functions

- Low level component of state/mode hierarchy
- Execution of function is state/mode dependent
- Execution of function state dependent; performance of function mode dependent

17 September 2003 *Practical Approach to States and Modes for Complex Systems* 8

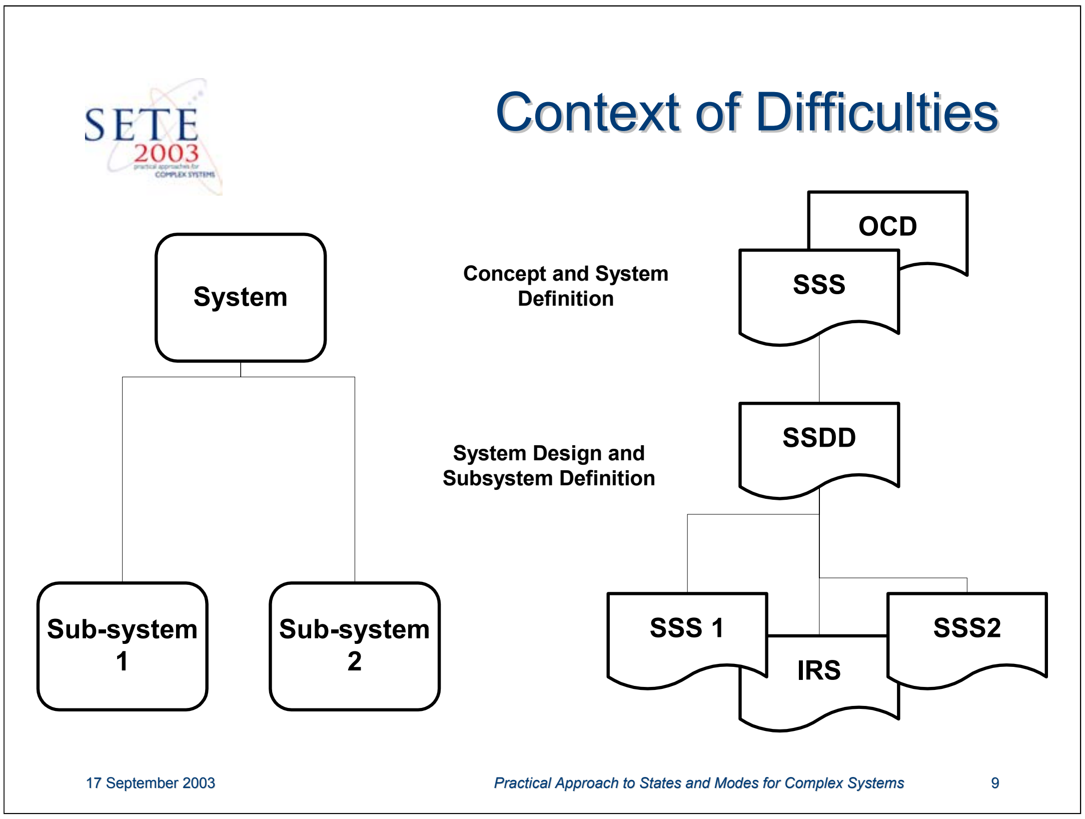
There are various ways of integrating the concept of "functions" with States and Modes. This becomes more important with the move towards function/performance based specifications for complex systems. Some examples follow. Note that a mixture of these approaches are sometimes used:

• Defining a function as another type of element in a state and mode hierarchy (usually a lower level component)

• Defining functions separately from the state and mode hierarchy but for each function defining explicitly which function/mode the function executes in.

• A refinement of the above where the execution of a function in a particular state(s) is explicitly required and the characteristics of the function may be defined differently for different modes (eg. A navigation function may always execute in an "operational" state but may be required to achieve different accuracies in "manual fix-dead reckon" mode and "Differential GPS" mode).

In the author's opinion, these approaches seem to be an ad-hoc response to the requirements in MIL-STD-498 DIDs and their predecessors. Whilst there does not seem to be an obvious right and wrong, a consistent approach (where states and modes are needed) would be useful. At least within a project!



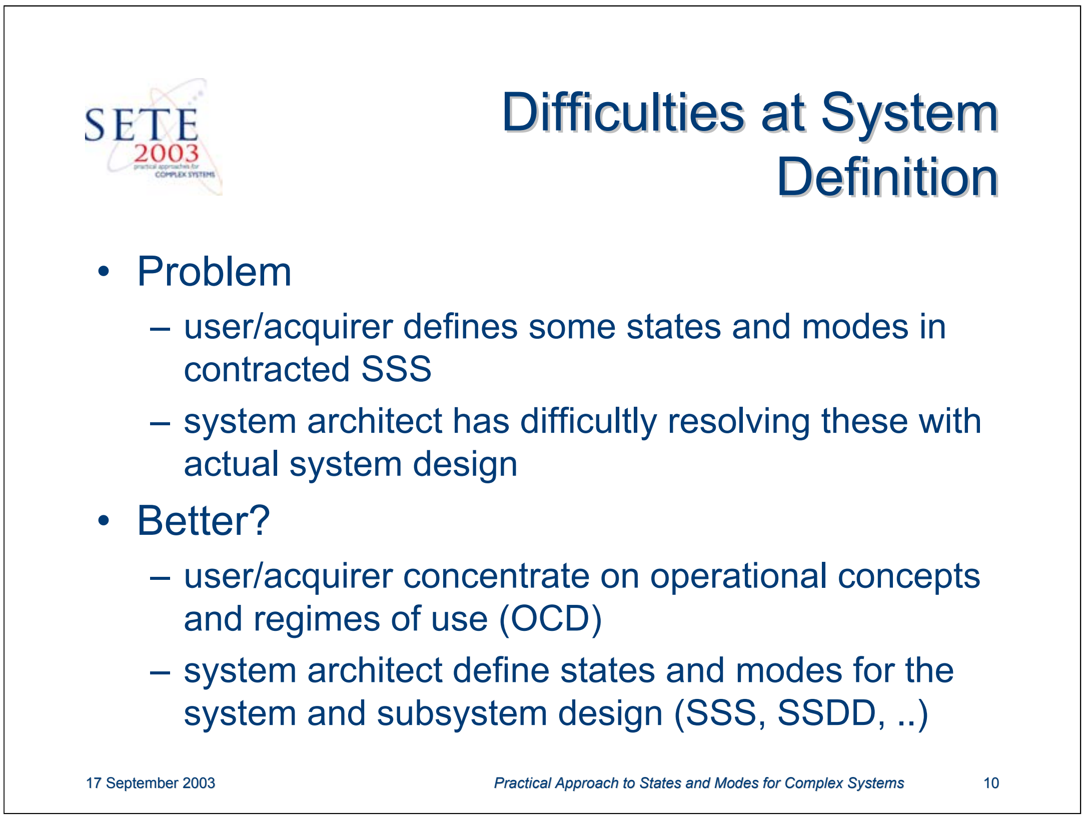
Context of Difficulties

SETE 2003
practical approaches for COMPLEX SYSTEMS
System
Sub-system 1
Sub-system 2
Concept and System Definition
System Design and Subsystem Definition
OCD
SSS
SSDD
SSS 1
IRS
SSS2

The “System” is usually contracted with a SSS or equivalent forming part of the delivery contract. Increasingly, the system concept and definition is augmented by an OCD from which the users intended use is communicated through the acquirer to the designer. These documents are generally all written in plain language.

The designer’s job is to take the required characteristics of the SSS and derive a compliant system design from these. This design derivation includes the specification of subsystems and their interfaces (SSS, IRS). This stage of the system design usually sees the bulk of the customers review and comment as they seek to ensure that the specified subsystems will indeed fulfil the overall system requirements.

These processes provide the context within which the difficulties with states and modes are identified and an approach is suggested.



Difficulties at System Definition

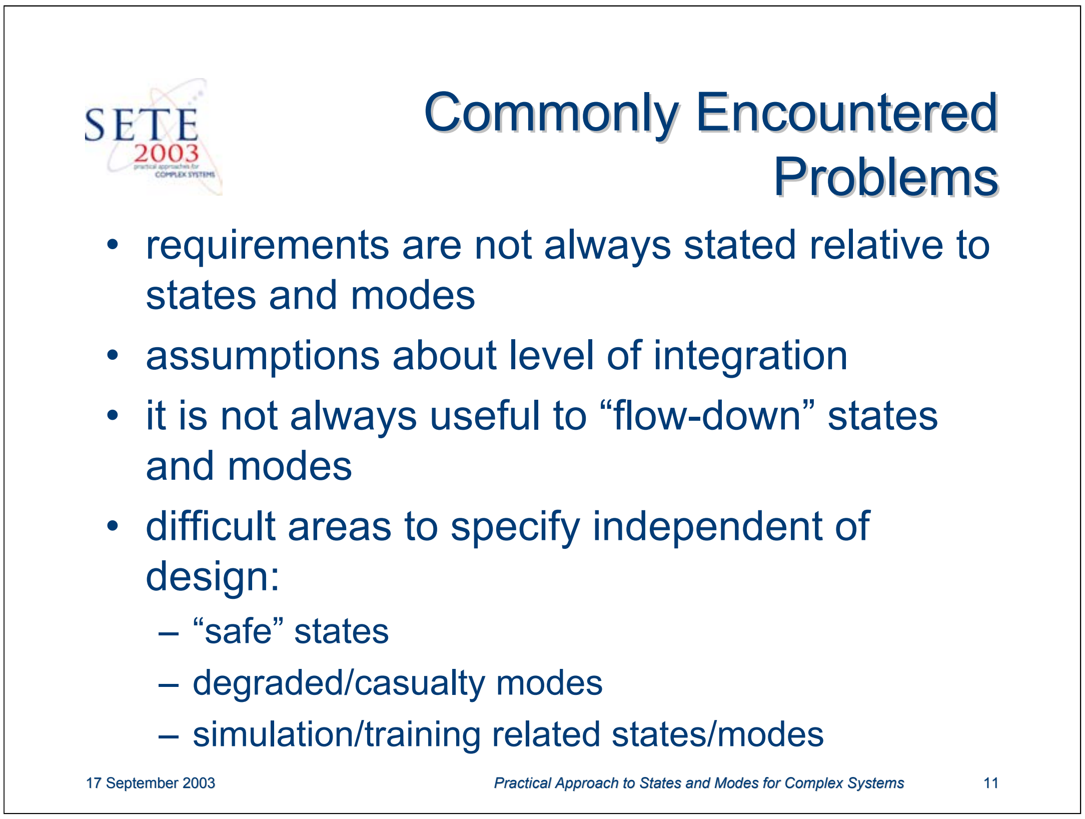
- Problem
  - user/acquirer defines some states and modes in contracted SSS
  - system architect has difficultly resolving these with actual system design
- Better?
  - user/acquirer concentrate on operational concepts and regimes of use (OCD)
  - system architect define states and modes for the system and subsystem design (SSS, SSDD, ..)

17 September 2003 *Practical Approach to States and Modes for Complex Systems* 10

Practical problems and difficulties are often encountered when defining and using states and modes in plain language specifications and design documents. This is particularly true in the domain of complex systems where formal specification methods are generally not in use and most system definitions are conducted using plain language.

Where the initial specifications are drafted by the end users and acquirers of the system (people who generally have operational and/or broad technical experience in the domain, but who do not have significant systems architecture/engineering experience) the risk of initiating ill-defined states and modes which then permeate the system design is high. Where this is the case, the defined states/modes should be treated as non-binding on the designer.

These early efforts would be more appropriately applied to defining regimes of use, to be captured in OCDs. The system architect could then define appropriate states and modes that support these regimes of use consistent with other program goals that may include the retention of particular subsystems or the nominated NDI subsystems.



Commonly Encountered Problems

- requirements are not always stated relative to states and modes
- assumptions about level of integration
- it is not always useful to “flow-down” states and modes
- difficult areas to specify independent of design:
  - “safe” states
  - degraded/casualty modes
  - simulation/training related states/modes

17 September 2003 *Practical Approach to States and Modes for Complex Systems* 11

In the author's experience, the following difficulties related to states and modes definitions are often encountered:

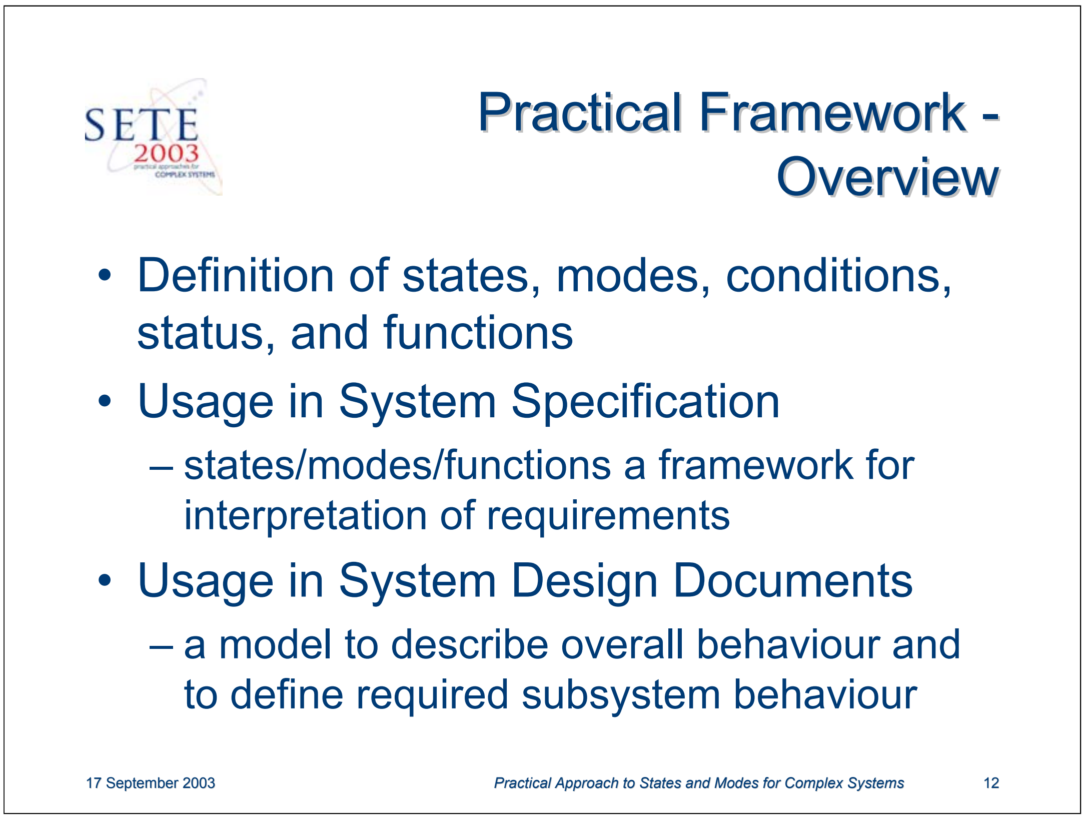
• Requirements are not always stated relative to states and modes when they are defined. This leaves the interpretation of when the requirements are achieved up to the designers interpretation and may result in disagreement with the acquirer. This is particularly the case when the user/acquirer has an expectation that some requirements will be met by the system concurrently. (eg. Detection performance and range resolution of a sensor subsystem may need to both be achieved in the same mode of operation rather than in separate "detect" and "localise" modes). Similar differences in interpretation between the designer and the test and evaluation team may lead to unnecessary difficulties.

• Inappropriate assumptions about the level of integration of subsystems. Is an overall state an appropriate and useful concept at the system level, in particular when the system is not tightly integrated? (eg. the status of a stand-alone component of the system may have little if any bearing on the status of the integrated core of the system).

• When states and modes are defined as requirements at the system level, it is not always useful to "flow" these down to subsystems/ components (it is better to impose required states and modes of subsystems based on required integrated system behaviour). This can lead to some difficulties in requirements traceability and verification.

• When an operational system has requirements for stimulation and simulation in support of training, training related states and modes need careful definition. For example the degraded, off-line or maintenance state of some subsystems probably does not need to preclude the assumption of a system training state unless some safety related constraints exist.

• "Safe" states and degraded/casualty modes are difficult to define independently of system design. For example a "safe" state may require the cessation of all mechanical movements (eg antenna rotation) or the cessation of intentional emissions (eg. Radar transmission) but depending on the operational circumstance this could result in mission failure with more dire results than the potential safety hazard mitigated (eg. If a Fire Control System is detected as overheating while controlling a missile engagement, it may be better to risk equipment damage than to risk halting an engagement). For a complex system, with many disparate subsystems, degradation may not be best considered as a state or mode (eg the failure of a sonar subsystem has little relevance to the Anti Air Warfare (AAW) capabilities of a Naval Combat System).



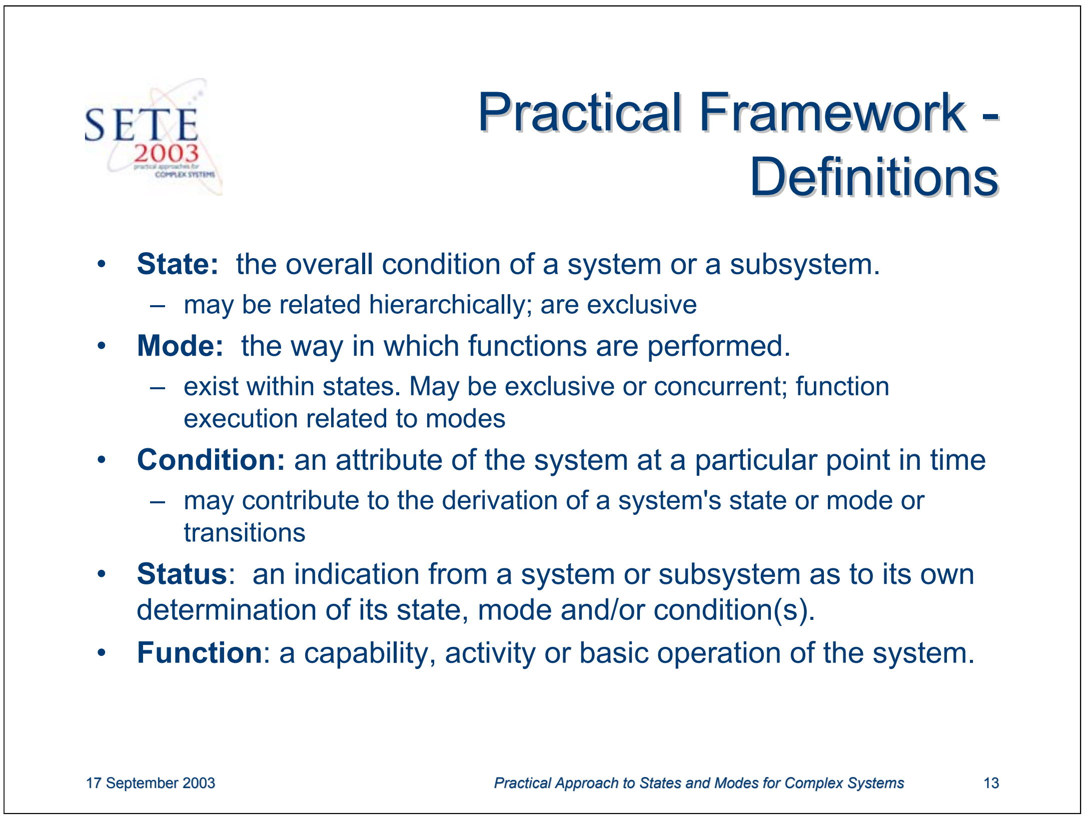
Practical Framework - Overview

- Definition of states, modes, conditions, status, and functions
- Usage in System Specification
  - states/modes/functions a framework for interpretation of requirements
- Usage in System Design Documents
  - a model to describe overall behaviour and to define required subsystem behaviour

A practical framework firstly requires an agreed definition of states, modes and other terms that will be used unambiguously by all stakeholders in the system life cycle.

Within the system specification(s), a minimum requirement is to define all states and modes that are useful in supporting the definition of the required characteristics of the system. Stated another way, the states and modes are an essential framework for ensuring the system requirements will be correctly interpreted.

In the system design documentation the complete set of states and modes are documented. The system design document(s) resolve how these system states and modes relate to the required behaviour of new subsystems and the existing behaviour of retained or NDI subsystems.



Practical Framework - Definitions

- **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
- **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
- **Status**: an indication from a system or subsystem as to its own determination of its state, mode and/or condition(s).
- **Function**: a capability, activity or basic operation of the system.

17 September 2003 *Practical Approach to States and Modes for Complex Systems* 13

The following definitions for states, modes and related terms are recommended:

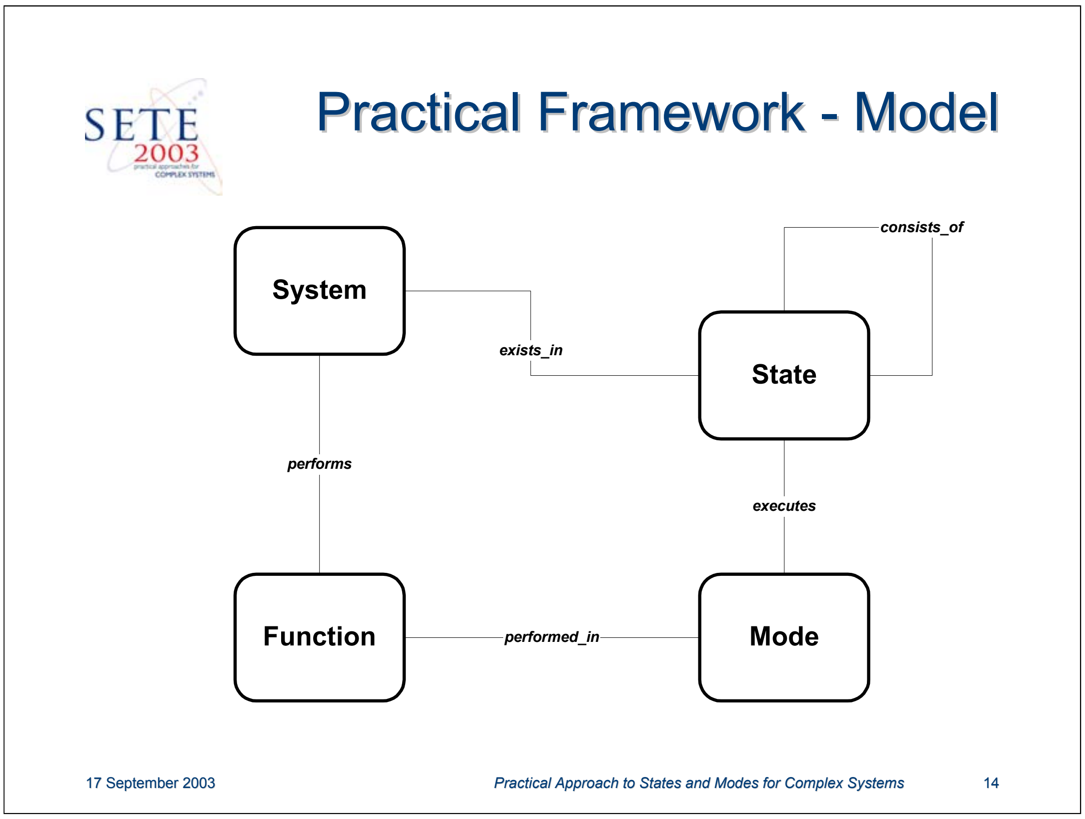
a. State: the overall condition of a system or a subsystem. Typically states may be like "on", "off", "standby", "operational". States may be related hierarchically. States are exclusive - only one state (at a given level of the hierarchy) can exist at any one time. Requirements may be stated relative to the State, but generally are assumed to be specified relative to an "operational" state unless otherwise indicated.

b. Mode: the way in which functions are performed. Modes exist within states. Some functions may execute differently, or execute or not execute based on the current mode of the system. Modes may be related hierarchically. Modes may be defined to exist concurrently with other modes or exclusive of other modes. Requirements may be stated relative to a mode where applicable. (As examples, within a sensor system “Search” Mode and “Localise” Mode may both exist only at different times within the “Operational” State. Different detection range and range/bearing resolution requirements may be specified with respect to each mode. A “Diagnostic” mode may also be defined within the “operational” state and it could exist concurrently with either of the other modes.)

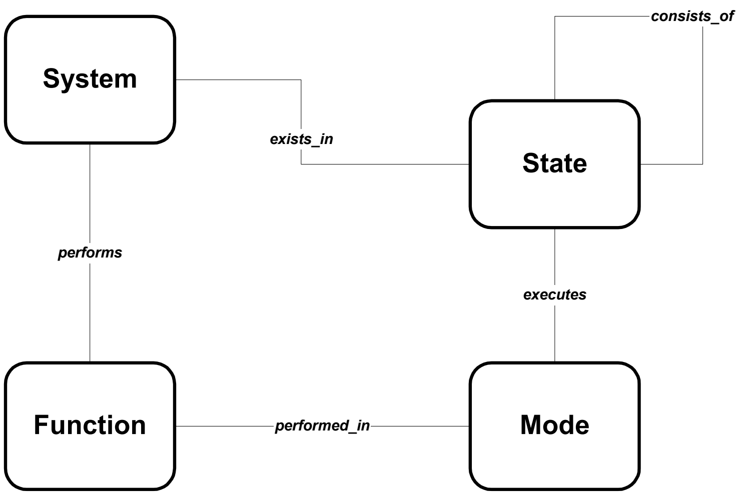
c. Condition: an attribute of the system at a particular point in time that may contribute to the derivation of a system's state or mode or the transition between states and modes. A condition may be used to describe failure, degradation or availability of components that may exist for a system or subsystem and affect the state or state transitions. (eg. If a fail condition exists for a sub-system's power supply it must be in an "Off" state and cannot transition from this state). Other conditions may include the presence or absence of particular entities within a system at a particular time (e.g. the existence of real engagement orders in a Combat System may preclude transition to a training state).

d. Status: an indication from a system or subsystem as to its own determination of its state, mode and/or condition. Status include displays and indicators provided to operators and maintainers and signals and data passed to other systems. It is usually implicit in requirements that reported status is correct.

e. Function: a capability, activity or basic operation of the system. (eg. Following on from the example in b. above, the function “Detect Entities” may be performed in both the “Search” and “Localise” modes of the “Operational” state, but it may have longer detection range performance requirements in the “Search” mode.



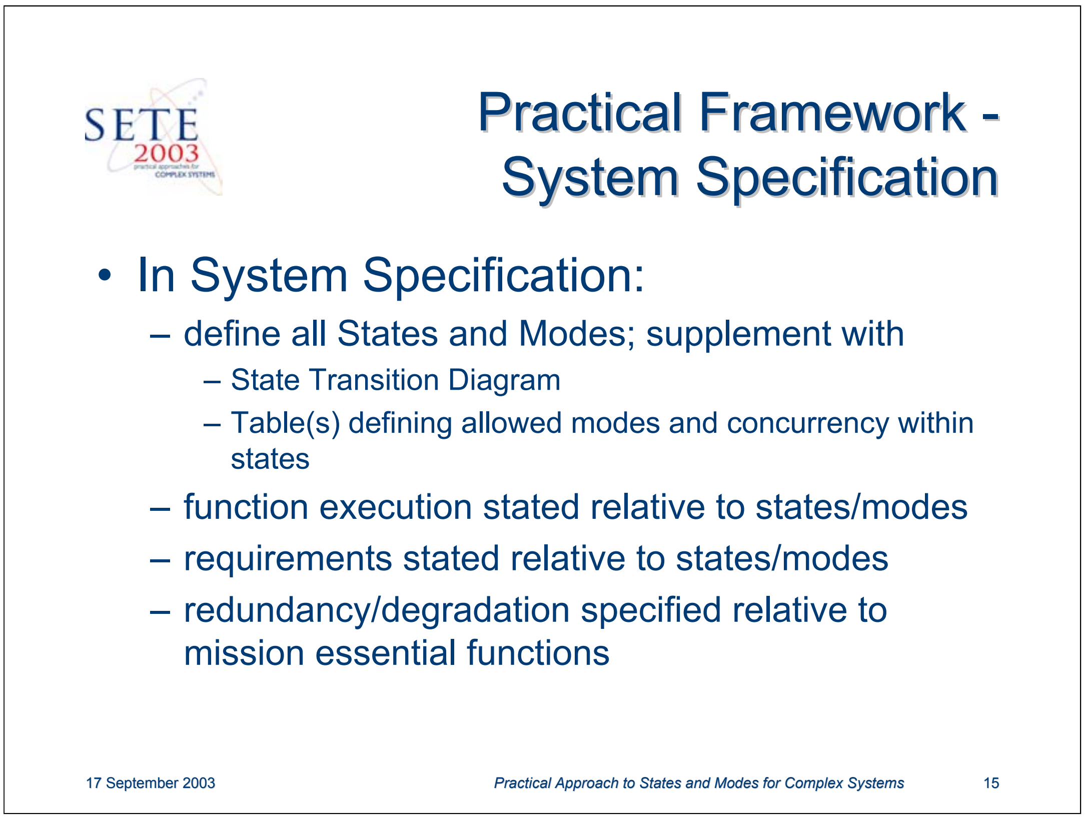
Practical Framework - Model



The proposed model is shown diagrammatically herein.

A system may be defined to exist in zero or more states. States may consist of zero or more substates. A mode may execute in one or more states. A function may be performed in one or more modes. The system must perform one or more functions. Functions may be decomposed further (not shown).

These constructs have been found to be sufficient in defining the required characteristics and behaviour for a complex system using plain language specifications and design documents.



Practical Framework - System Specification

- In System Specification:
  - define all States and Modes; supplement with
    - State Transition Diagram
    - Table(s) defining allowed modes and concurrency within states
  - function execution stated relative to states/modes
  - requirements stated relative to states/modes
  - redundancy/degradation specified relative to mission essential functions

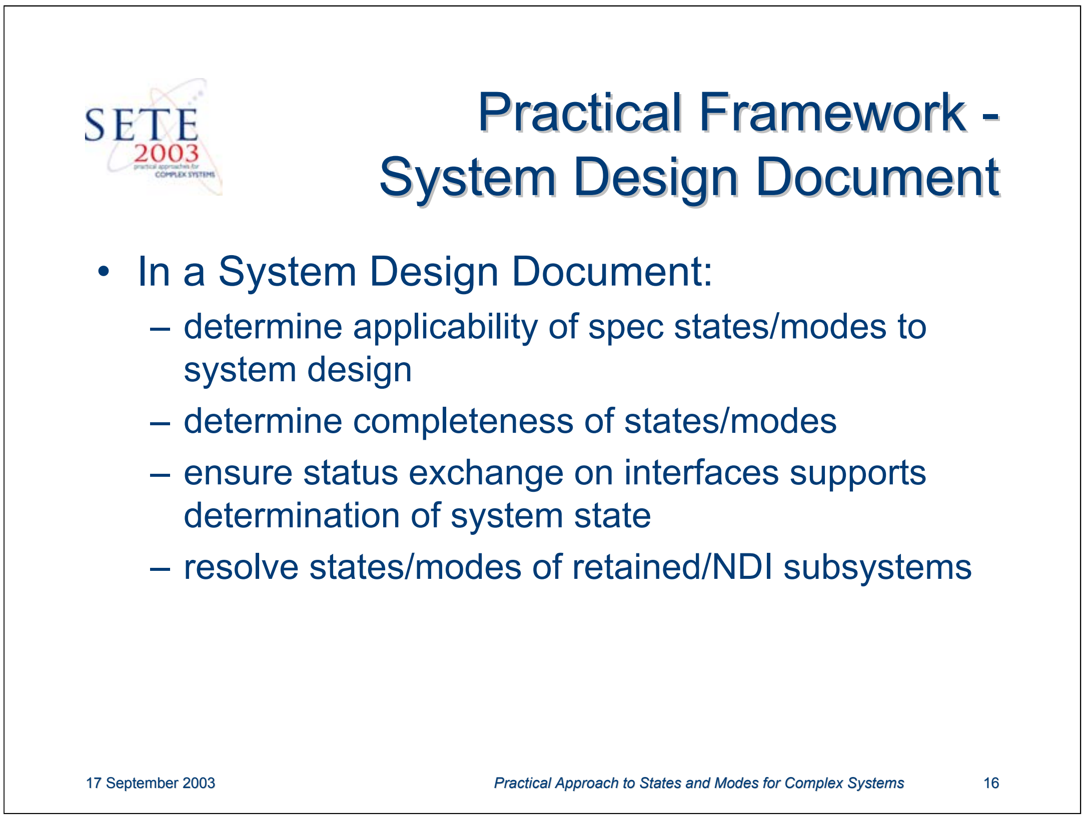
17 September 2003 *Practical Approach to States and Modes for Complex Systems* 15

Within the system specification(s), a minimum requirement is to define all states and modes that are useful in supporting the definition of the required characteristics of the system. Stated another way, the states and modes are a framework for ensuring the system requirements will be correctly interpreted. To support this definition simple State Transition Diagrams and tables defining modes that are allowed within states and any concurrency of modes within states should be used.

The number of states and modes defined should be as small as possible (to reduce complexity) but as many as needed to provide a framework for the system requirements. Note that it is not considered necessary that this state and mode definition be exhaustive (ie the designed system may exhibit many more states and modes) or that the state and mode definitions necessarily take the form of requirements (leave flexibility for system design where possible).

Once the framework of states and modes within the system specification has been established, the functions (or capabilities) of the system need to be defined relative to these states and modes. Functional/performance requirements associated with the function may then be implicitly assumed to be tied to the same state/mode framework as their associated function unless specified elsewhere. Arguably, it is better to define each requirement explicitly with respect to the state/mode framework (this may be particularly true where the requirements are extracted to a database and not always read in the context of the associated function).

Where redundancy/degradation requirements need to be established in the system specification, it is recommended that the relevant functions of the system be tagged “mission critical” and then the tolerance of these functions to failure conditions within the system may be defined.



Practical Framework - System Design Document

- In a System Design Document:
  - determine applicability of spec states/modes to system design
  - determine completeness of states/modes
  - ensure status exchange on interfaces supports determination of system state
  - resolve states/modes of retained/NDI subsystems

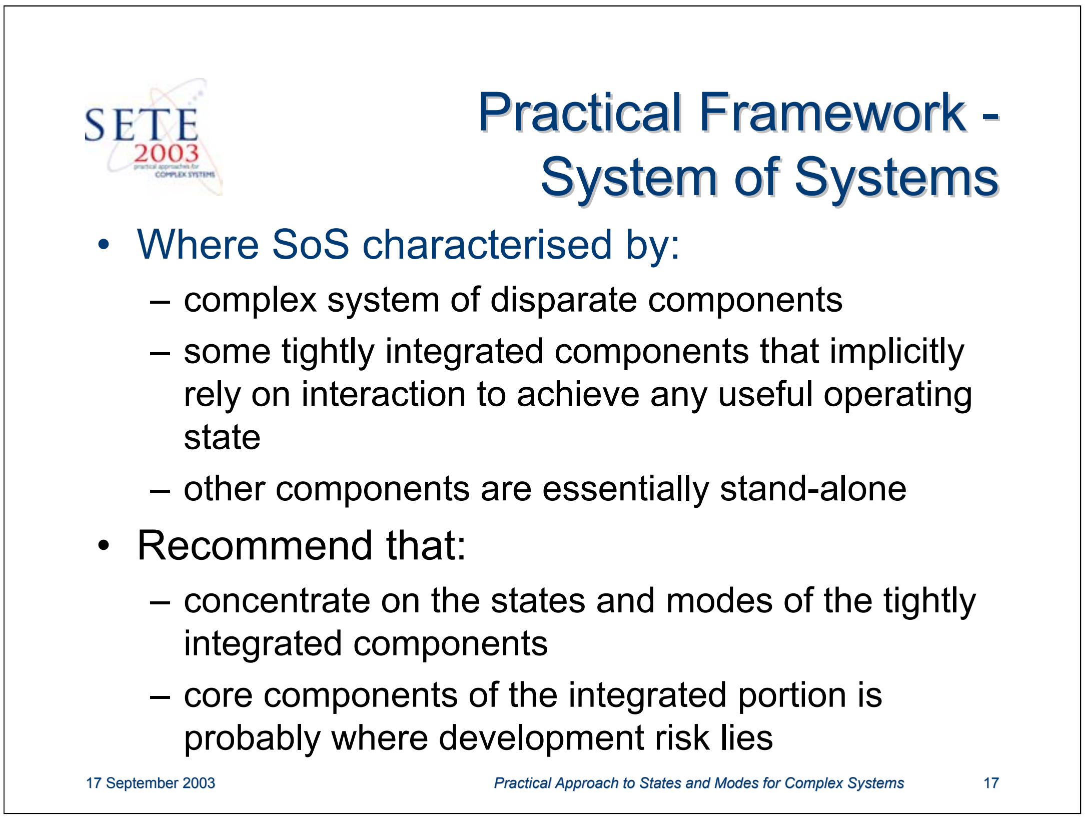
In the system design documentation the complete set of designed states and modes are documented. The system design document(s) resolve how these system states and modes relate to the required behaviour of new subsystems and the existing behaviour of retained or NDI subsystems.

In particular, the applicability of the defined states/modes in the System Specification need to be established. They may well be appropriate as the basis for the designed states and modes but other approaches that simplify system behaviour or better allow for retained or NDI subsystem behaviour should not be precluded.

It is likely that the specified states/modes are sufficient for system definition but not for realisation. The complete set of states/modes for desired system and subsystem behaviour need to be established and documented in the system design document.

If the overall state of the operational system is to be determined (usually determined by a centralised element of a system, but increasingly the state may be determined in a distributed manner) then the exchange of status between subsystems needs to sufficient to support this. Thus the designed states/modes of the system and subsystems may well find themselves into the detail specification and design of interfaces.

Where retained or NDI subsystems are to be incorporated into the system their extant states and modes are also retained. The system design document needs to resolve how these extant states/modes map into the states and modes of the system as a whole.



Practical Framework - System of Systems

- Where SoS characterised by:
  - complex system of disparate components
  - some tightly integrated components that implicitly rely on interaction to achieve any useful operating state
  - other components are essentially stand-alone
- Recommend that:
  - concentrate on the states and modes of the tightly integrated components
  - core components of the integrated portion is probably where development risk lies

17 September 2003 *Practical Approach to States and Modes for Complex Systems* 17

When a "system of systems" is a large and complex system of disparate components and where:

- some of these components are tightly integrated and implicitly rely on the interaction of components to achieve any useful operating state (e.g. sensor systems may not incorporate their own HMI but rely on that provided by an integrating system),
- other components are essentially stand-alone in that they can achieve a useful operating state independent of other elements of the system; they have their own set of functions and behaviours independent of the system as a whole (e.g. a GPS receiver usually has its own position readout display that can be used for manual navigation irrespective of how many other system components receive position data feeds from the GPS), and
- the definition of states like "Off", "Standby" and "Operational" for the system as a whole may provide little assistance in the description and specification of system behaviour,

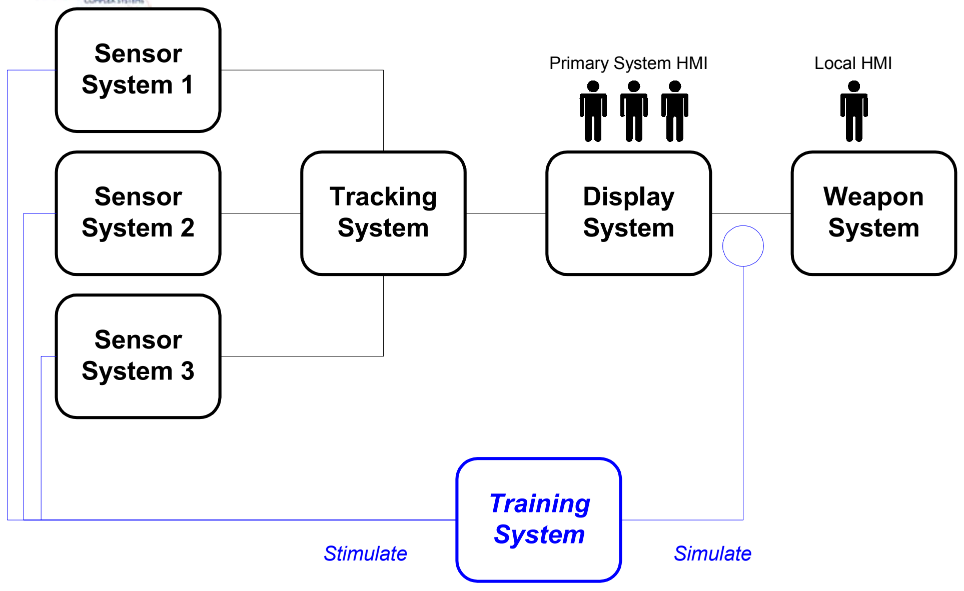
then the definition of states and modes therefore must concentrate on the behaviour of the tightly integrated components of the system of systems. This will assist in the definition of system and interface initialisation requirements and the correct and safe transition between different modes of this part of the system.

Moreover, the core of this integrated portion of the system is most likely to be:

- the most complex (processing, functions and number of interfaces),
- the most developmental (probably will be customised in order to integrate existing and NDI subsystems), and
- have the most HMI (to collect the status presented by own and external subsystems to the operators and to exercise control of the system)

and is therefore most worthy of the focus of the system architect and design engineers.

Example: Reference System



17 September 2003 *Practical Approach to States and Modes for Complex Systems* 18

A reference system is illustrated to convey the concepts of the recommended approach. A description of the reference system is as follows:

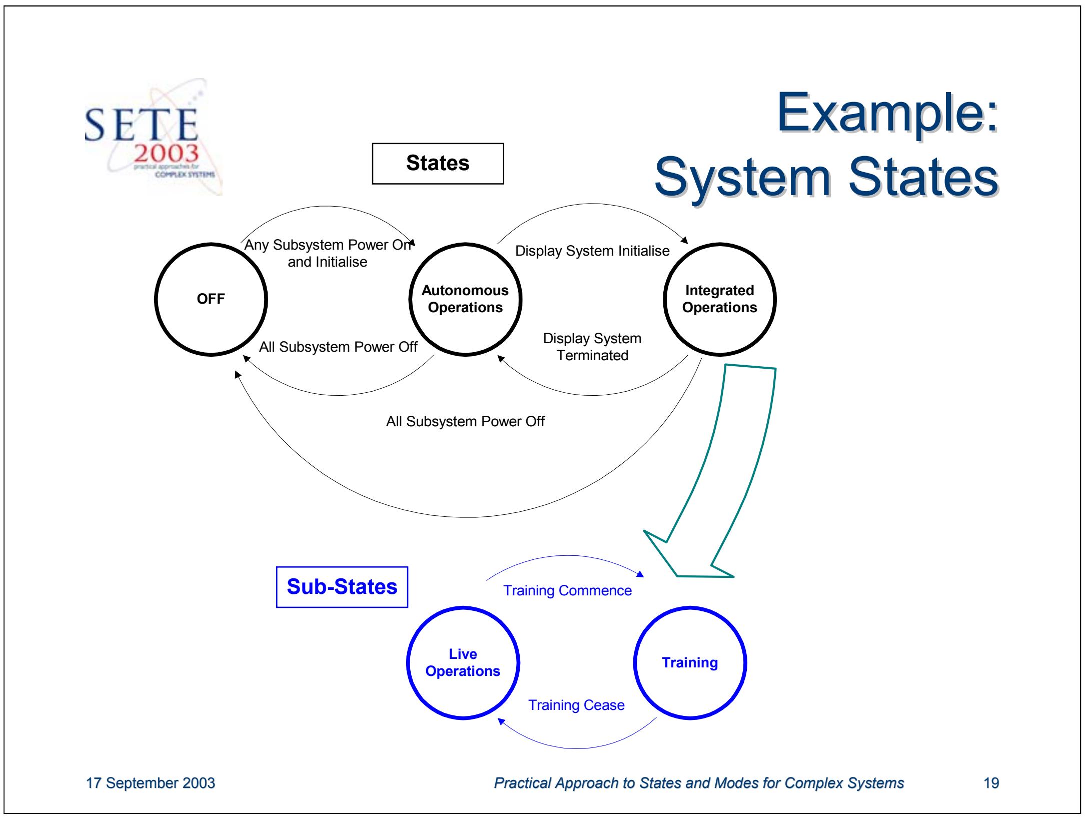
•Sensor systems, independently feed detection data into the tracking system and receive sensor setting controls from the tracking system.

•Tracking system forms tracks from raw detections and provides these to the Display System. It controls sensor settings either automatically or based on operator settings from the Display System.

•The Display System is the primary HMI component of the system. It receives and displays tracks for operator attention and action. It accepts operator entries for sensor settings, and for weapon system employment orders and passes these to the relevant systems. It gathers status from all other subsystems and determines the overall system state and status for display to the operators.

•The Weapon System has a local HMI capability and may be manually employed independent of the other systems. Usually it is cued from the Display System.

•The Training System's purpose is to provide realistic training for operators. It achieves this by stimulation of sensors (injection of simulated data) and by simulating (substituting for) the Weapon System.



Example: System States

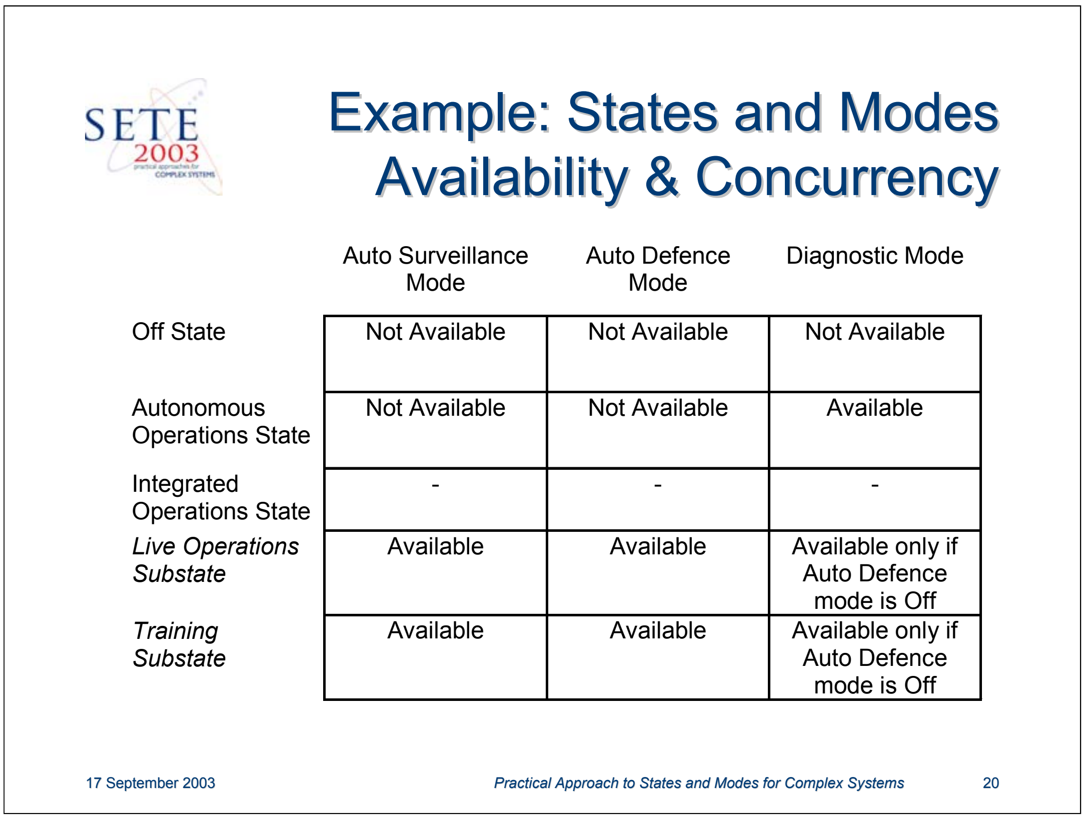
17 September 2003 *Practical Approach to States and Modes for Complex Systems* 19

The "Off" state exists when no system component has power applied. In a complex system, depending on its employment, it may be expected that this state will not occur often (e.g. an inertial navigation sensor in a submarine would be infrequently powered down completely). Not many system requirements are likely to be specified relative to this state.

The "Autonomous Operations" state exists when one or more subsystems have power applied and have initialised. Subsystems may execute operations to the extent they are able (eg. The Weapon System may be fired manually under local operator control). Subsystems may establish interfaces with other subsystems that have initialised and some integrated operations may occur (e.g. the Tracking System may be forming tracks from sensors). The Display System is not initialised during this system state, and no integrated operations involving the Display System are possible in this state. No overall System status may be displayed to the operators in this state. No training states/modes exist in this state.

The "Integrated Operations" state exists when the Display System is initialised. This state exists for a range of conditions from the Display System only up to all other Systems available and interfaces operational. Overall System status (as reported to the Display System) may be displayed to operators in this state.

The Integrated Operations State consists in turn of a Live Operations State and a Training State (substates). The default state on entering the Integrated Operations State is the Live Operations State. The Integrated Operations State supports various modes of operation, most of which may exist in either the Live Operations or Training States.



Example: States and Modes Availability & Concurrency

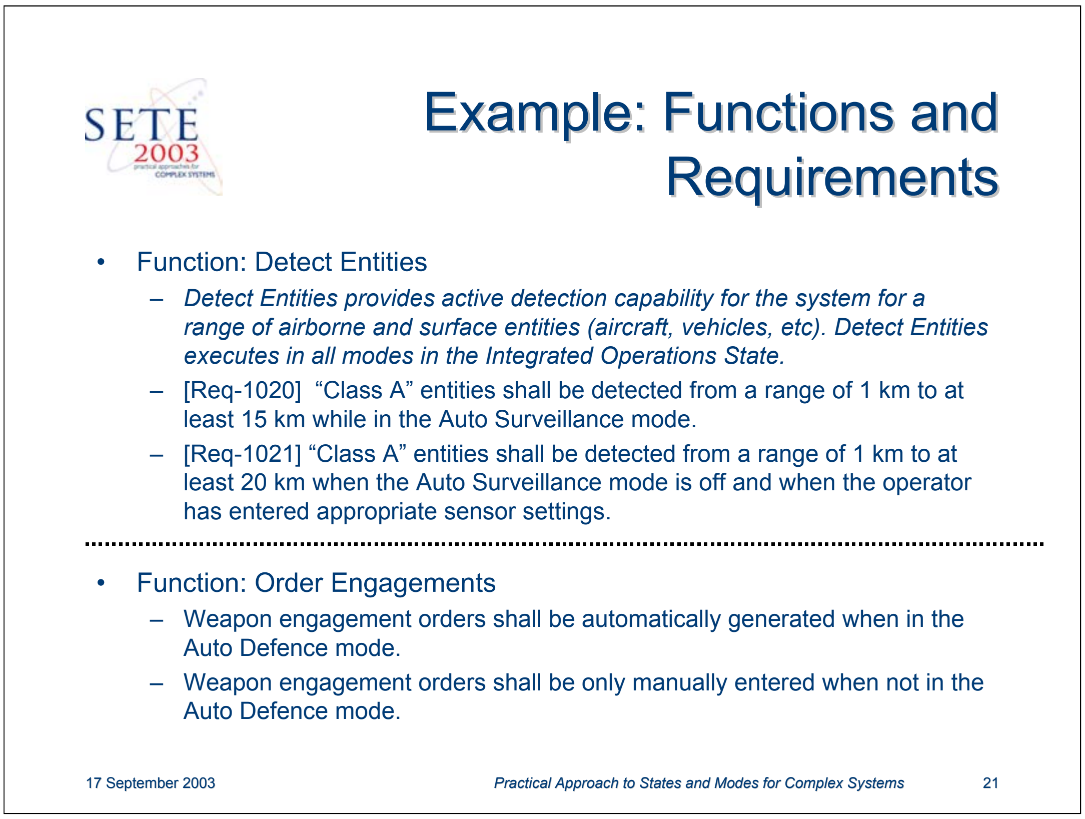
	Auto Surveillance Mode	Auto Defence Mode	Diagnostic Mode
Off State	Not Available	Not Available	Not Available
Autonomous Operations State	Not Available	Not Available	Available
Integrated Operations State	-	-	-
Live Operations Substate	Available	Available	Available only if Auto Defence mode is Off
Training Substate	Available	Available	Available only if Auto Defence mode is Off

17 September 2003 *Practical Approach to States and Modes for Complex Systems* 20

An example of a table that establishes the relationships between the system states and modes is illustrated. It allows the concurrency of modes within states to be defined.

In this example the system has three distinct modes that may exist relative to the aforementioned states and substates. The Automatic Surveillance Mode provides for the system itself to configure the sensor settings to optimal values for current conditions. The Auto Defence Mode allows for the system to evaluate tracks for threat and to automate weapon employment orders. The diagnostic mode allows for background system Built In Test to execute.

In this example the Automatic Surveillance Mode and the Auto Defence Mode are available in the Live Operations and Training substates of the System. The Diagnostic Mode is available in the Autonomous Operations State (ie allows systems to conduct their own diagnostics even if not centrally reported). It is allowed in the Live Operations Substate but only when the Auto Defence Mode is Off (this is a constraint placed on the system such that diagnostics do not interfere with nor utilise resources that are more properly assigned to defensive functions). Similarly it has been constrained to be limited in the Training substate so that Training replicates Live Operations to the maximum extent possible.



Example: Functions and Requirements

- Function: Detect Entities
  - *Detect Entities provides active detection capability for the system for a range of airborne and surface entities (aircraft, vehicles, etc). Detect Entities executes in all modes in the Integrated Operations State.*
  - [Req-1020] "Class A" entities shall be detected from a range of 1 km to at least 15 km while in the Auto Surveillance mode.
  - [Req-1021] "Class A" entities shall be detected from a range of 1 km to at least 20 km when the Auto Surveillance mode is off and when the operator has entered appropriate sensor settings.

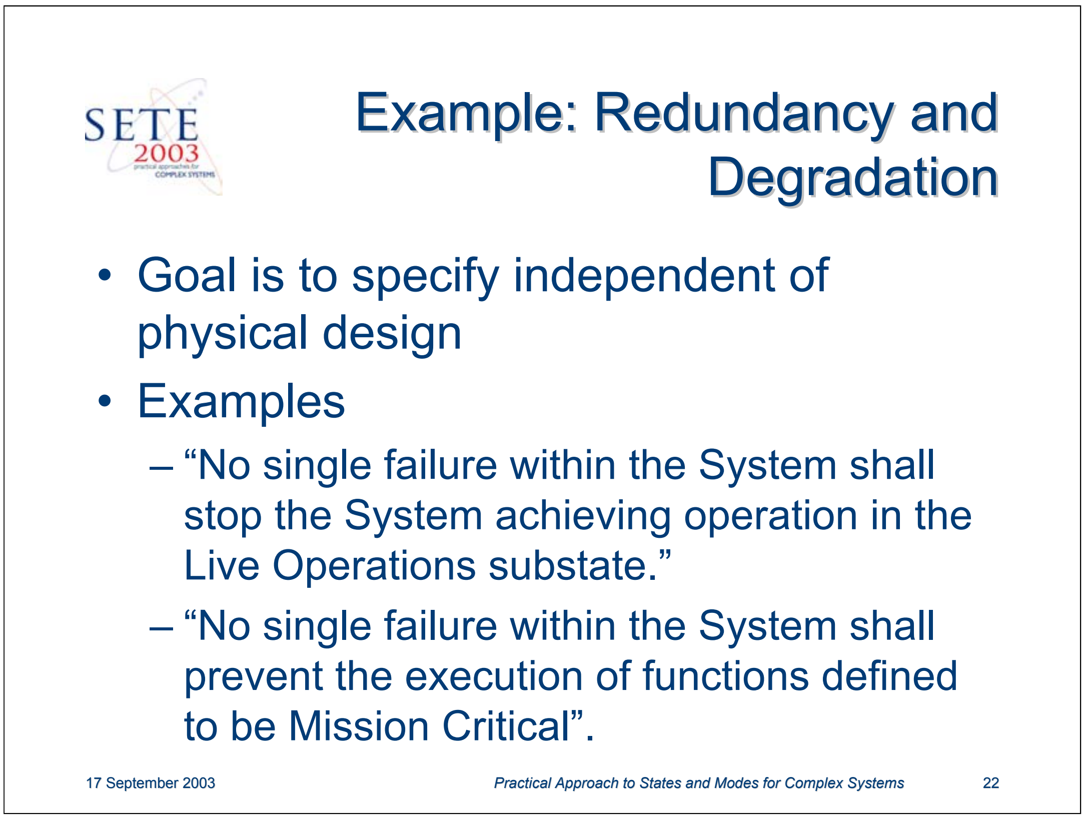
- Function: Order Engagements
  - Weapon engagement orders shall be automatically generated when in the Auto Defence mode.
  - Weapon engagement orders shall be only manually entered when not in the Auto Defence mode.

17 September 2003 *Practical Approach to States and Modes for Complex Systems* 21

Two sets of examples are given for the reference system.

Firstly, a function "Detect Entities" is defined relative to the states and modes framework already established. Then two performance requirements associated with that function are defined. In this example the requirements are explicitly related to the mode of system operation. The example contemplates that detection will be better when the operator is able to optimise sensor settings compared to automatically generated sensor settings. In verification then, this function will need to be tested in at least these two operational modes.

Secondly, two functional requirements are defined with respect to a function "Order Engagements". This example illustrates a method of specifying different methods of a function executing dependent on the current system mode.



Example: Redundancy and Degradation

- Goal is to specify independent of physical design
- Examples
  - "No single failure within the System shall stop the System achieving operation in the Live Operations substate."
  - "No single failure within the System shall prevent the execution of functions defined to be Mission Critical".

17 September 2003 *Practical Approach to States and Modes for Complex Systems* 22

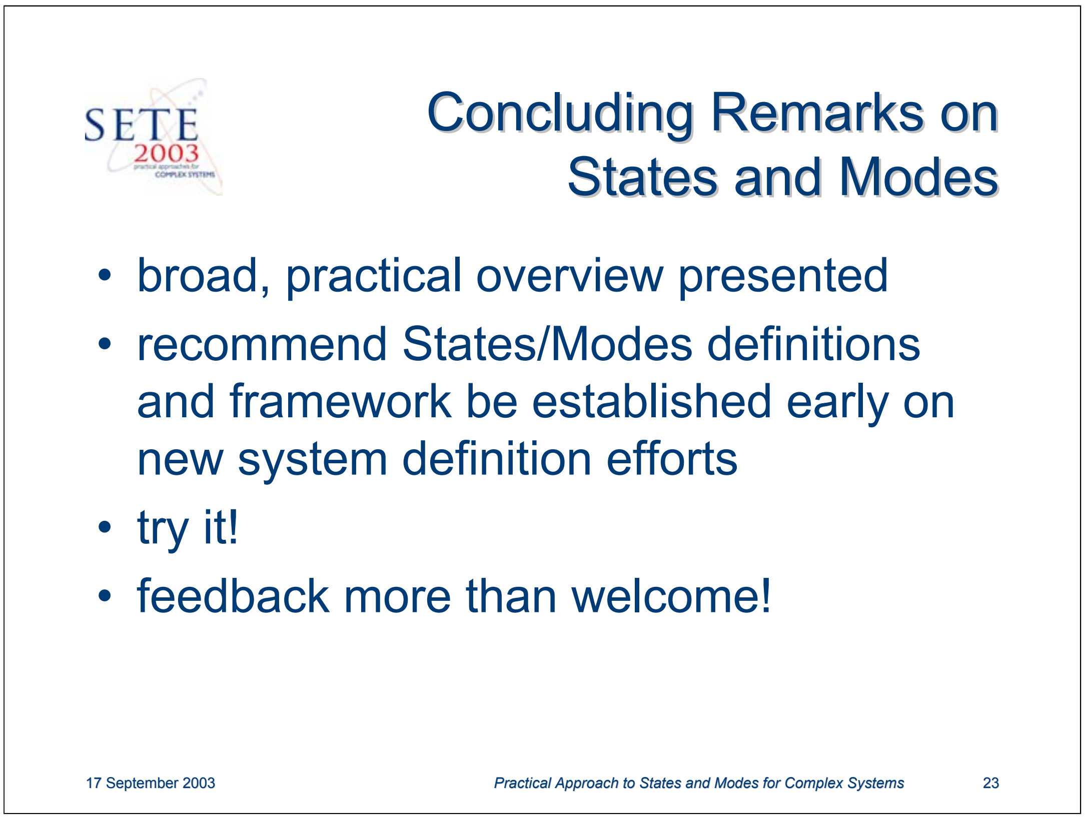
Degradation is best considered a condition of the system rather than a specific state or mode. A degraded condition exists when one or more functions of the system are unavailable to the operators.

Redundancy requirements are often specified in terms of interfaces (eg. "all interfaces shall utilise redundant physical connections"). This is not really related to the real requirements of the system, ie the achievement of available functions.

Two examples are suggested herein to fulfil the goal of specifying degradation/redundancy requirements independent of the physical realisation of the system.

The first example, "No single failure within the System shall stop the System achieving operation in the Live Operations substate", utilises the framework of states already established. It says to the architect of the system that enough redundancy must be designed in to allow the Live Operations substate to be attained even if there is one failure in the system.

The second example is similar: "No single failure within the System shall prevent the execution of functions defined to be Mission Critical", but utilises the concept of Mission Critical functions. Redundancy therefore will need to be designed into the system components that support Mission Critical functions but can be kept simple for non-mission critical functions.



Concluding Remarks on States and Modes

- broad, practical overview presented
- recommend States/Modes definitions and framework be established early on new system definition efforts
- try it!
- feedback more than welcome!

17 September 2003 *Practical Approach to States and Modes for Complex Systems* 23

The author welcomes discussion and questions about the contents of this paper and welcomes any opportunity to gather useful information and insights from the wider systems engineering community on this and related topics.

References:

[1] Harel, D. (1987). Statecharts: A Visual Formalism for Complex Systems. *Science of Computer Programming* 8 (1987) 231-274.

[2] Glinz, Martin (2002). 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, Algorithm and Tools.* Orlando, May 2002.

Author's Biography

Mick Edwards holds a B.E. (Hons) and a M.Eng.Sci majoring in communications from the University of NSW and a MBA (Technology Management) from APESMA/Deakin University.

Mick is currently Combat Systems Engineering Manager at ADI Limited where his primary activity is the Combat System Design Authority for the upgrade of the RAN's FFGs. His experience and continued interests lie in the application of systems engineering methods and domain knowledge to the design and integration of functionally complex systems, largely in the naval and maritime domains.