



INCOSE2001
INNOVATE INTEGRATE & INVIGORATE
MELBOURNE AUSTRALIA
1-5 JULY

The Architecture View Set in Action

Tara Morgan

Lockheed Martin Management & Data Systems

tara.morgan@lmco.com

610-354-7785

Purpose

.... to discuss the application of the new, modern System Engineering Architecture Based Design Process that is the current standard at Lockheed Martin Management & Data Systems in Valley Forge, PA, to a live, critical program....walk the talk.

Overview

- **AVS and ABD™ Background**
- **Program Examples**
- **Lessons and Observations**
- **Conclusion**
- **References**



AVS and ABD™ Background: Definitions



INCOS2001

INNOVATE INTEGRATE & INVIGORATE

MELBOURNE AUSTRALIA

1-5 JULY

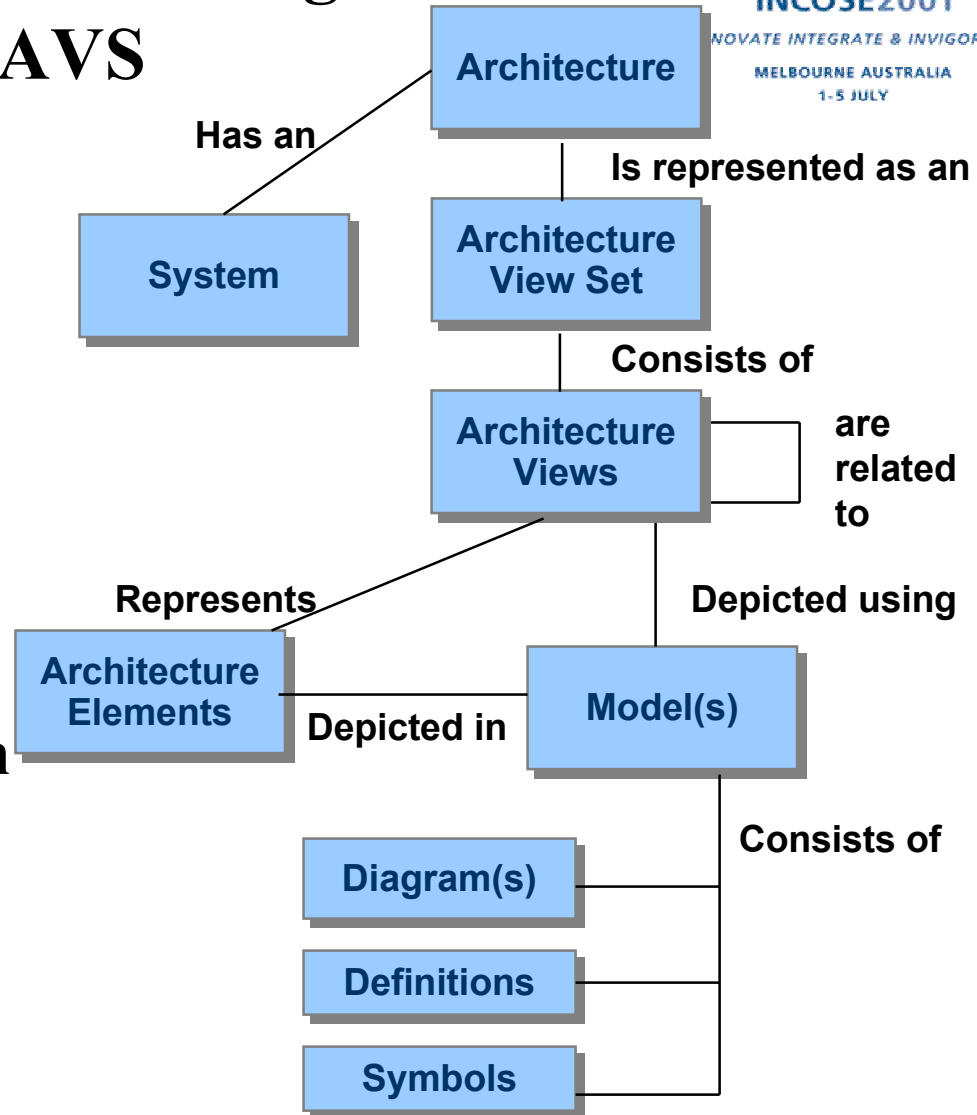
- **Architecture Based Design™ (ABD™)** : design process for defining all executing levels and aspects of a large, complex software intensive system through an Architecture View Set.
- **Architecture View Set (AVS)** : model-based, multi-dimensional, complete description of what a system is and how it works.
- **TTWCS (Tactical Tomahawk Weapon Control System)** : US Navy System responsible for managing and executing the launch and control of Tomahawks. (Significant reuse of ATWCS S/W and System Design)

AVS and ABD™ Background:



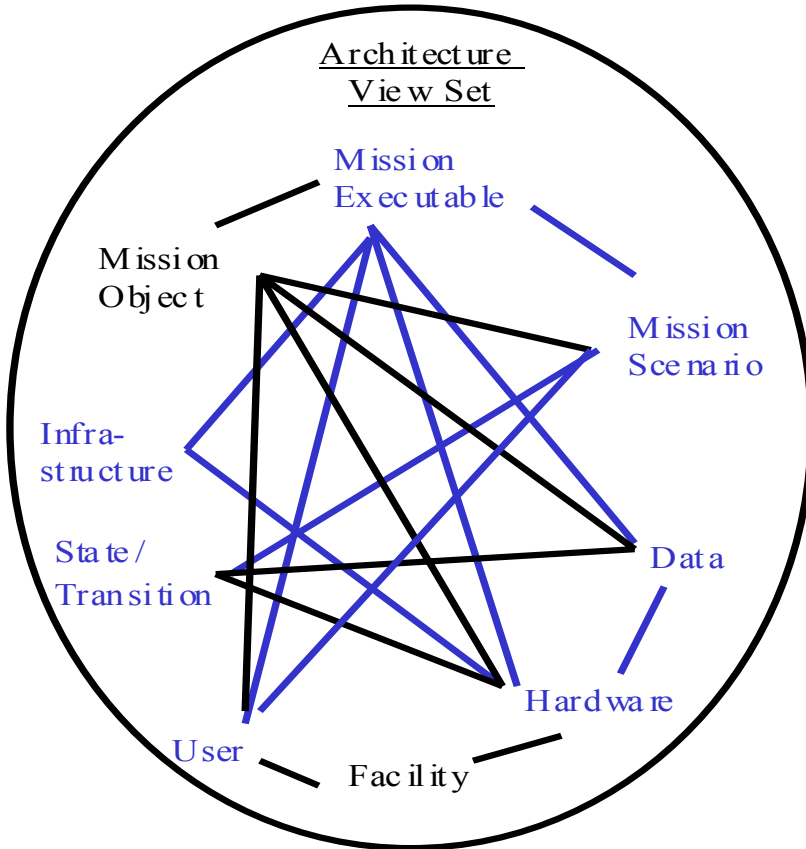
AVS

- **Architecture View Set** - a description of the system architecture consisting of a necessary and sufficient number of adequate architecture views and view relationships
- Each view focuses attention on a cohesive aspect of the system



AVS and ABD™ Background:

AVS



- **9 Core Views**
- **18 Core View Relationships**
- **TTWCS had no need for two core views**
 - Mission Object View - 50% reuse of system and software design
 - Facility View - Reused facility on ship and submarine
 - This removed the need for 9 core view relationships



AVS and ABD™ Background: Relationship to Other Models



INCOSE2001

INNOVATE INTEGRATE & INVIGORATE

MELBOURNE AUSTRALIA

1-5 JULY

- **Many Architecture Frameworks (AF) are currently available (C4ISR (a.k.a. DOD) AF, Treasury Enterprise AF, Zachman, others)**
- **Since the AVS represents the actual, physical, executing description of the system, it maps closely with**
 - the System View of the C4ISR AF
 - the middle layers of the Zachman framework
- **AVS focuses on relationships between models**
- **AVS focuses on depiction of emergent system properties**

Program Examples: Models

- **Models supporting the views were generated using Meta-CASE Tool - System Architect by Popkin Software**
- **Customized Models made efficient and effective architecture depiction possible**
 - The best notation for the view
 - Facilitating managed and/or automatic relationships between models (and, therefore between views)

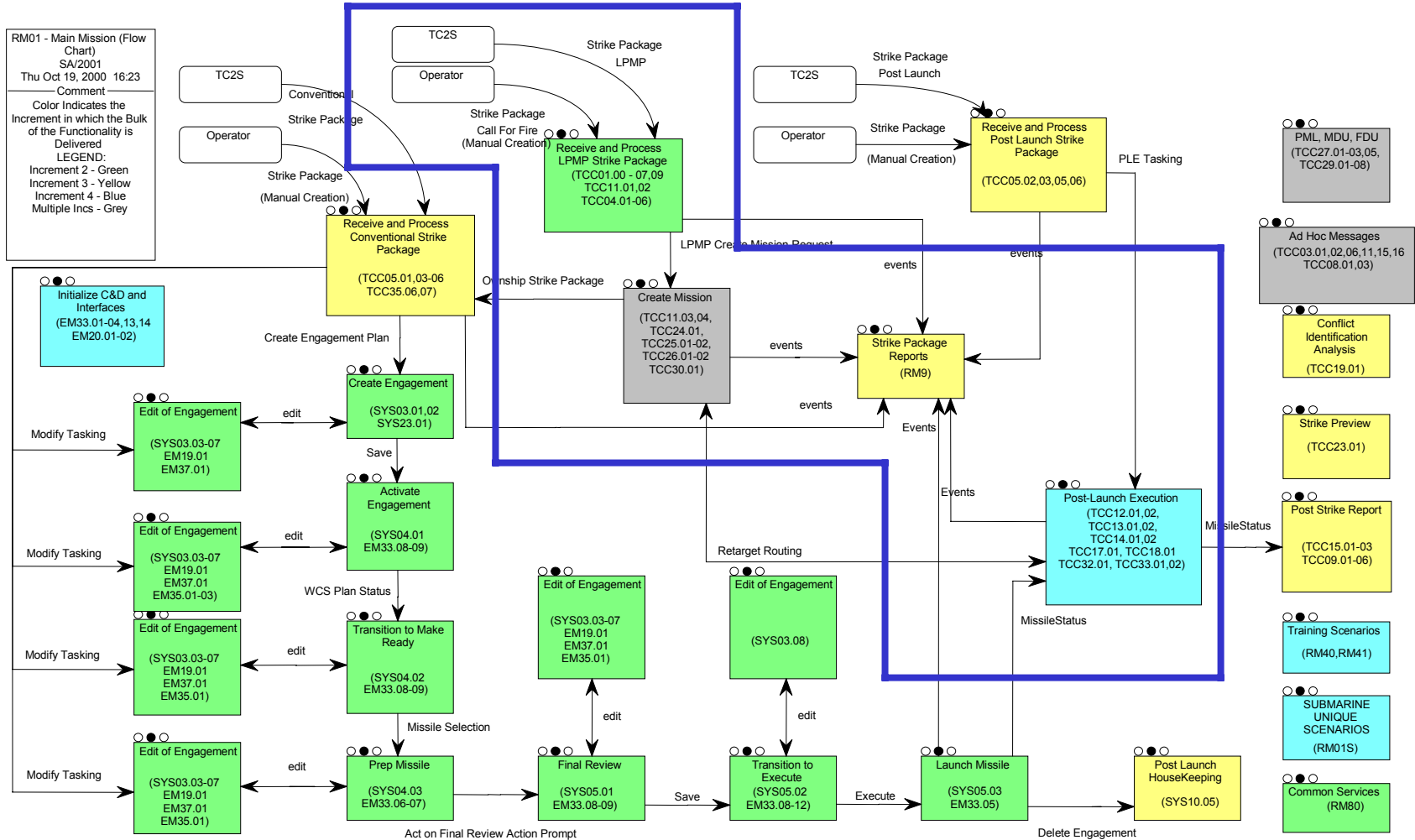
Program Examples: Model Documentation Method

AVS View	Diagram	Details
Mission Executable View	Ward Mellor Data Flow Diagram	Symbol Defintions
Mission Scenario View	Gane & Sarson Data Flow Diagram	Symbol Defintions
Data View	N/A	Data Defintions & Adjuct Documents (1)
Hardware View	Sketch in SA	H/W Schematics
User View	Workflows	Microsoft Office (2)
State/Transition View	State Transition Diagram	Symbol Defintions
Infrastructure View	Sketch in SA	Referenced (3)
Composite Views	N/A	Referenced Documentation and Incorporated into Other Views
<p>(1) External Messages reference ICDs (Interface Control Documents. IDD (Internal Design Documents) messages are defined in SA and external Spreadsheet. Intra-Subsystem messages are defined in SDDs.</p>		
<p>(2) HCI Presentations are in PPT: HCI Specification is a Word Document.</p>		
<p>(3) Operating Environments goverened by Requirements and Navy Standards documented in Vendor Documents.</p>		

Program Examples: Model Relationships

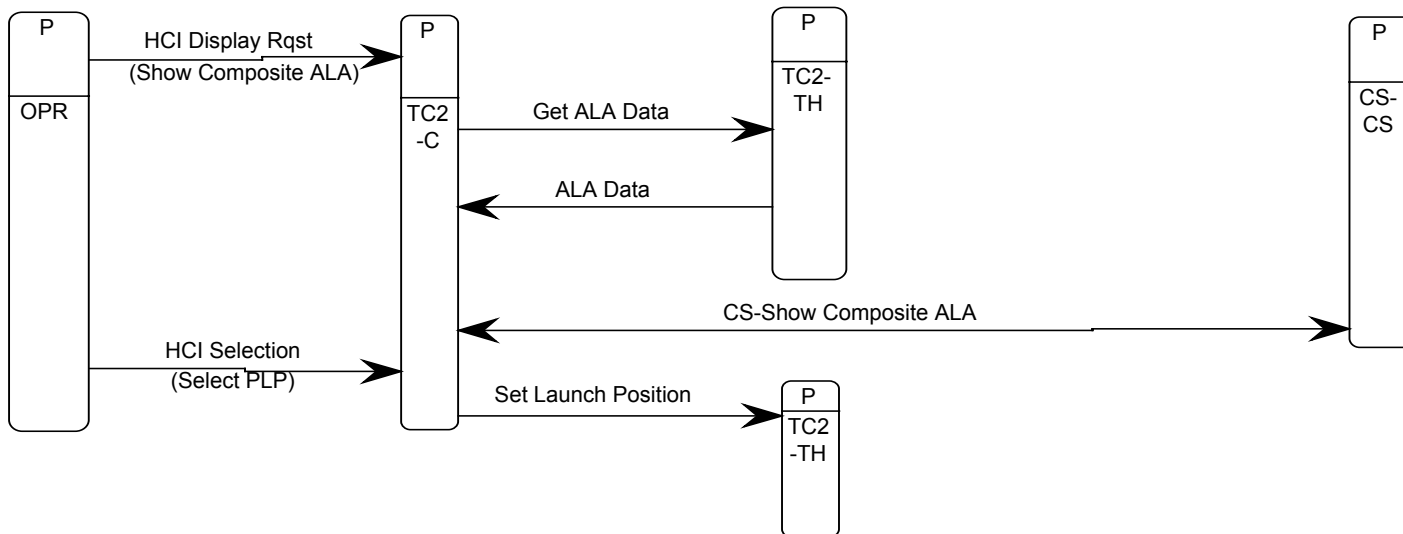
Relationships between Views in the TTWCS AVS		
Views		Relationship Mechanism
Mission Executable	Hardware	Executables Mapped to H/W component definition
Mission Scenario	Mission Executable	Scenario depicts Executables/Messages and data stores in Action
User	Mission Scenario	Detailed HMI Specs correspond to operator interaction in Scenario Diagrams
State Transition	Mission Scenario	Behavior of Executables during State Transitions is captured in Scenario Diagrams
Mission Executable	Data	Message and Data Store symbols are defined
Mission Executable	User	User symbol is defined
Mission Executable	Infrastructure	Services used are listed in Executable Symbol Definition
Data	Hardware	Persistent Data Stores Mapped to H/W component definition
Infrastructure	Hardware	Infrastructure Executables Mapped to H/W Component definitions

Program Examples: Mission Scenario Roadmap

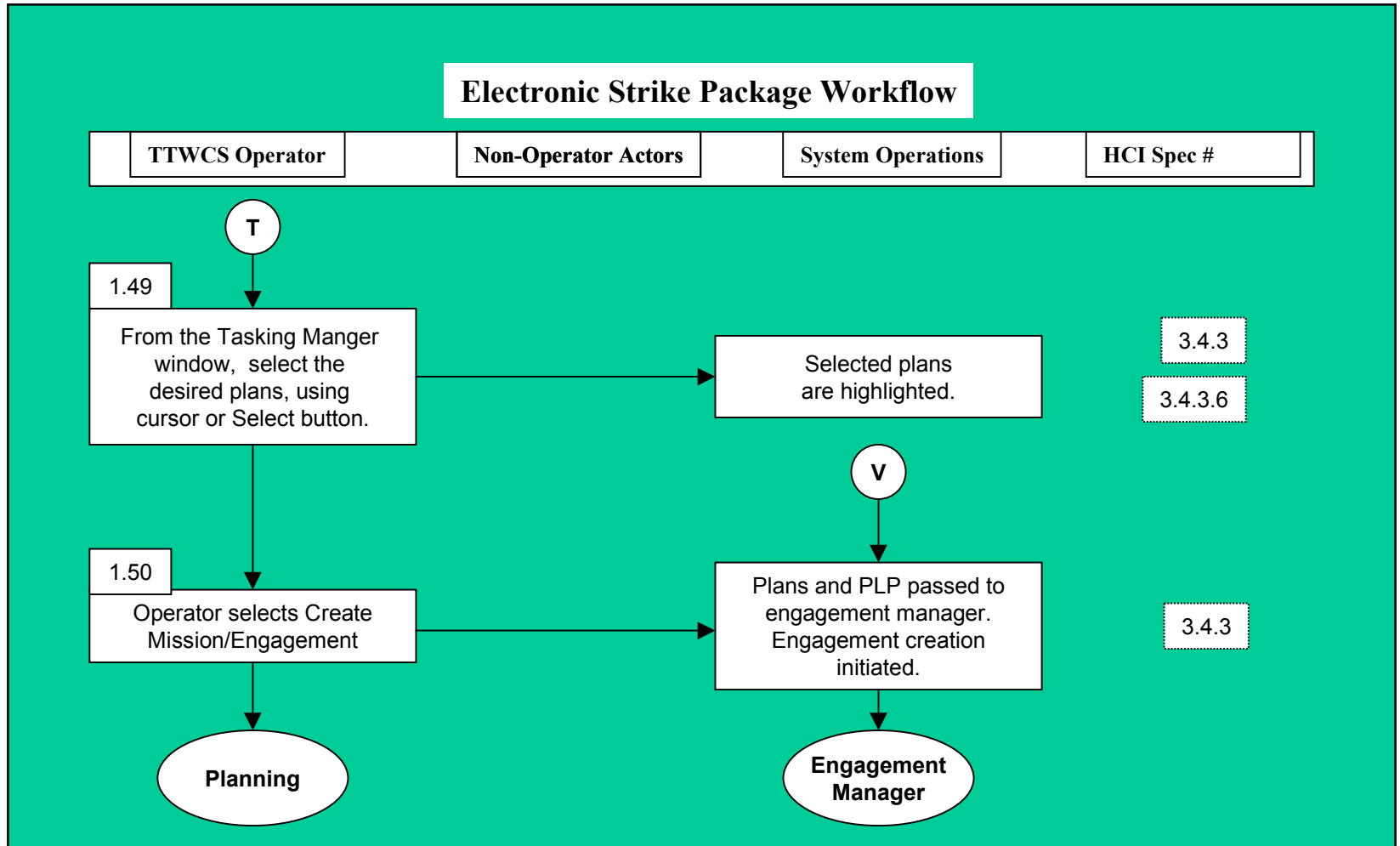


Program Examples: Mission Scenario Diagram

TCC05.04 - Display CALA (Data Flow Gane & Sarson) SA/2001 Sat Apr 22, 2000 15:47 — Comment —




Program Examples: User View Flow Chart



Program Examples: User View HMI Example




UNCLASS
232133:55Z MAR 00

System Chart Views Comms Misc
Help

Job Bar
Window Help

Tasking
Planning
Launch
Post-Launch
PML Mgt
Prompts

Prompt Filter: Tasking Launch
 Planning Post Launch

Tasking Manager

Reports
Help

Strike Packages

Message Type	Use State	Message ID	Strike Execution Time	Time Received
Tactical 12 100000Z Oct 2000	Execute	001000001	12 114000Z Oct 2000	

Display TSN Data
Conflict Identification Analysis Selection...
Strike Preview...
Edit Strike Package...
Create Strike Package...
Print Report
Delete

Prospective Plans

Supportable Plans
Unsupportable Plans

Status	ASN	MSN	P/RS/B	Type	Mission ID	BrID	Time /Type	RS Delta Time	Missile Type	Cell/Tube
Waiting Blk III C	124	538		Primary	Pre-Planned	048-001-01000		121400	/TGT	
Waiting 00:00	124	539		RSpare	Pre-Planned	048-001-01000		121401	/TGT	
Waiting Blk IV	125	540		Primary	Pre-Planned	048-001-01001	6	121400	/TGT	
Waiting Blk IV	127	542		Primary	Pre-Planned	048-001-01002	2	121422	/TGT	
Waiting Blk IV	132	561		Backup	Pre-Planned	048-001-01007		121450	/TGT	

Select...

Mark Plan Unsupportable
Send Exception Message(s)...
Show Allowable Launch Areas
Create Mission/Engagement
Call For Fire (CFF)...
Close

Lessons and Observations: Communicating Design



- **HTML Version of Architecture on Program Website**
- **Effective Communication Internally & Externally**
- **Customer Presentations done in Logical Sequence**
 - HMI (User View: Flow Chart and HMI Examples)
 - Architecture Design (Mission Scenario Roadmap and Mission Scenario Views)
 - Transition to Software Design Details
- **Resulted in Positive Customer Involvement**
 - Introduced Tool Early
 - Confidence in System Design
 - Useful Meetings at all levels, Valuable Interaction

Lessons and Observations: Composite View



INCOS2001

INNOVATE INTEGRATE & INVIGORATE

MELBOURNE AUSTRALIA

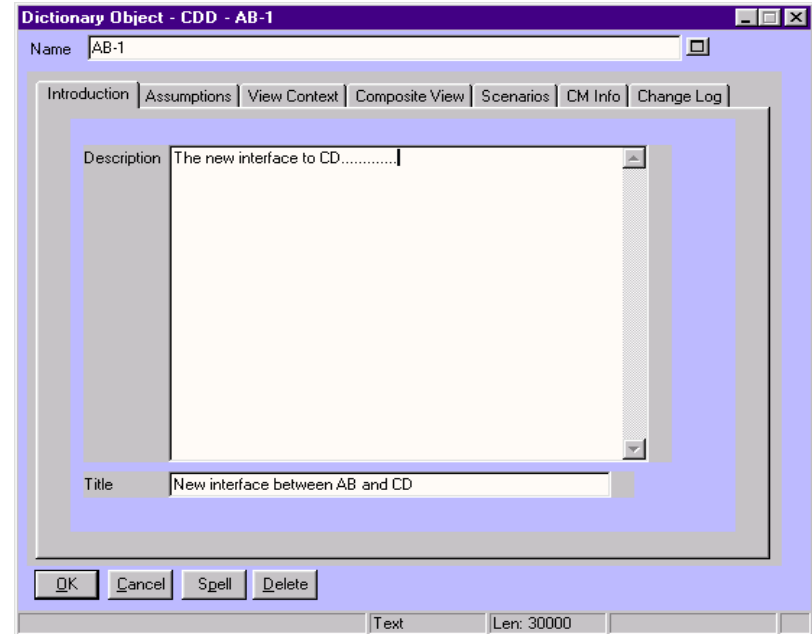
1-5 JULY

- **Composite Views: describe and design emergent properties in the system**
 - Emergent properties are visible across many architecture views
 - Impacts and conflicts with the overall architecture must be resolved
 - TTWCS Composite Views: Safety, Security, Process Management, Training, Submarine, System Performance
- **A dedicated engineer responsible for each composite view**
 - Selected appropriate documentation approach
 - Reviewed all Capability Design Documents
- **Composite Views are a powerful and cohesive approach to solving system-level problems**

Lessons and Observations: Capability Design Document (CDD)

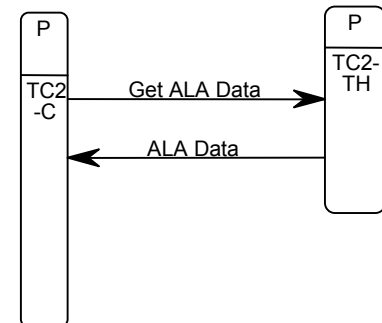


- Each Increment consists of the delivery of a set of Capabilities to the test team
- **Capability: A Deliverable Piece of Functionality, generally specific to Increment and SubSystem**
- **CDDs Document Operational Concept Definition and Applicable System Functionality through References to Applicable Models, including Composite Views, Series of Scenario Diagrams, etc.**
- **CDD managed by System Architect in the Architecture View Set**
- **CDDs serve as a Contract between SE and S/W Development**

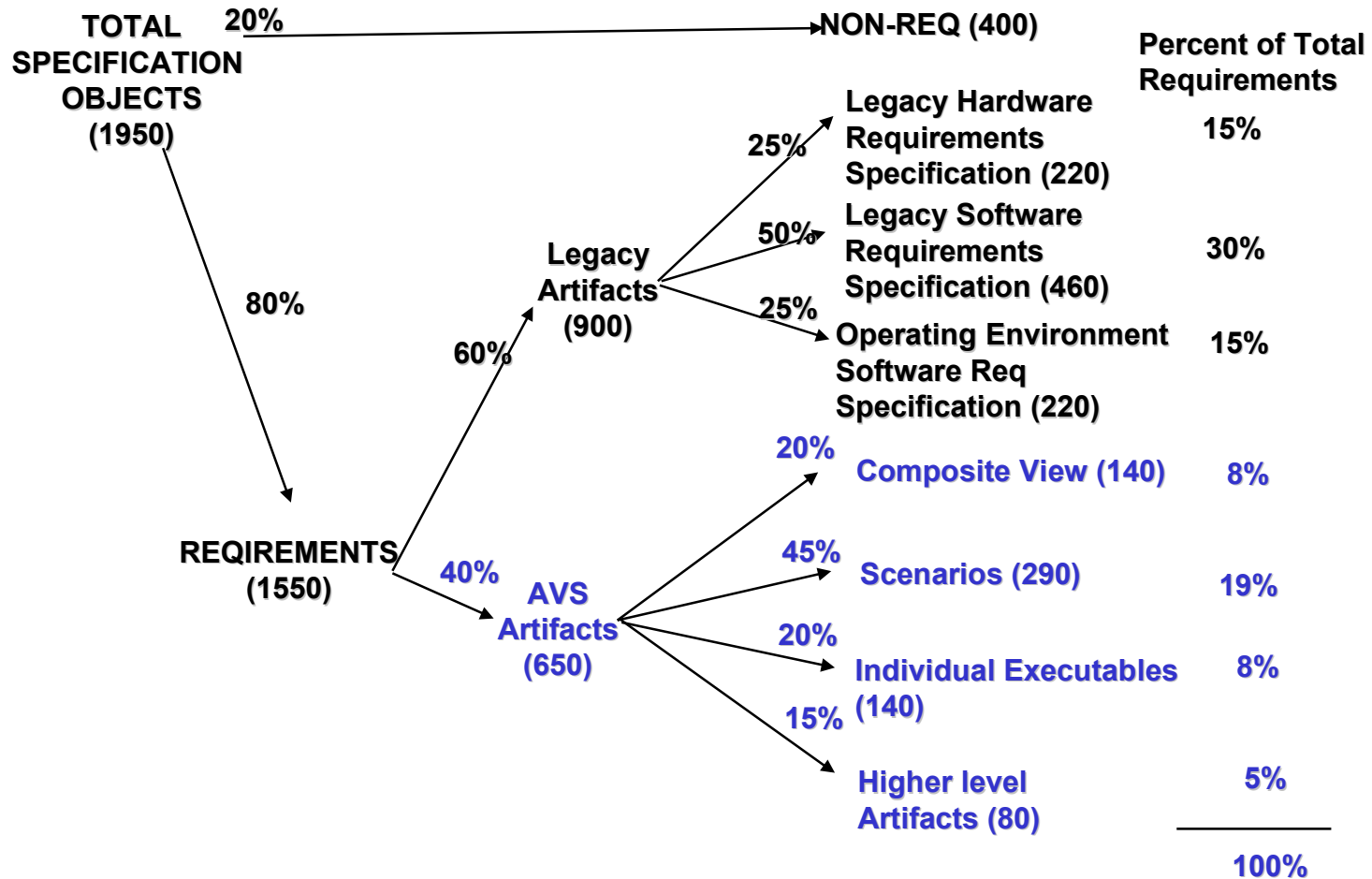


Lessons and Observations: Requirement Traceability

- **Allocation of System Level Requirements directly to Architecture Artifacts**
 - Capabilities, Scenario Diagrams, Executing Units, Composite Views
 - Maintained in RTM(Requirement Management Tool) through Architecture Element Attribute
- **Lower Level Requirements (Executable Spec) captured directly in the Architecture View Set**
 - Capability Design Documents
 - Mini-Spec associated with functionality internal to executable



Lessons and Observations: Requirement Traceability



Lessons and Observations: Additional Points

- **Design Confidence**
 - Multiple Related Views of System Architecture
- **Communicating and Documenting Design**
 - Modern Method - Models vs. Requirements
 - Facilitated coordination between SE and S/W throughout the development of the AVS Artifacts
 - Prevalence of Design Artifacts throughout Work Environment
- **Separation of Executing and Construction Aspects**
 - Design stopped at definition of Executables
 - Smooth transition to Software Design using UML
- **Extended design Approach to Scenario Based Test**

Conclusion

- **Architecture Based Design™ (ABD™) and the Architecture View Set is a Valuable and Successful Method for Architecting a System**
- **Experiences on TTWCS are noteworthy and helpful for improving the process**
 - Navy customer mandated continuance of method into the next generation of the system
 - Lessons Learned are being incorporated back into the evolving ABD™ process.

References

- **Rodger Frank: Co-Author and pioneer of ABD™ Methodology (rff@hrb.com)**
- **“System Architecture - A View Perspective”. Rodger Frank. INCOSE 1998**
- **“A Solution to the Architecture/Software Design Bind - The Mission Object View”. Rodger Frank. INCOSE 1999.**