LOCKHEED MARTIN

Capturing the Army Battle Command System (ABCS) Architecture Using the C4ISR Architecture Framework

Abe Meilich, Ph.D., C.C.P.
Lockheed Martin Mission Systems
Defense Information Systems
abraham.w.meilich@lmco.com

®CMM and Capability Maturity Model are registered in the U.S. Patent and Trademark Office

Background

- ABCS System of Systems Engineering and Integration (SSE&I) Contract :
  - Improved integration of business processes, data, and systems amongst the core Tactical Army Battle Command Control Systems
    - from GCCS-A , through MCS, and down to FBCB2
  - Technically manage the enterprise architecture that results from this integration
    - Provide analysis framework required to better facilitate system of systems integration
- Challenge: Framework for collecting, categorizing, and utilizing the architecture and engineering information

Background

- Multiple contractors, geographically dispersed, and chosen from amongst systems developers
- Minimize Traditional DOD development deliverables (e.g., SSDD, SSS, etc.)
- New major DOD Enterprise systems are requiring compliance with C4ISR Architecture Framework (C4ISR/AF)
- Review of alternative documentation and analysis approaches evaluated
  - Result: C4ISR/AF fit the bill
  - model-based approach reduced documentation requirements

LOCKHEED MARTIN

Results

- **A comprehensive enterprise architecture analysis and design framework**
- **Capture of architecture analysis artifacts of each system (Program) key architectural components**
  - **Accomplished through forward and reverse engineering to capture legacy and evolving modernized components of each system in ABCS**
  - **Using this information capture to conduct architecture and system engineering studies**
  - **Utilized to capture common data model data elements across systems with the result of an integrated data model**
  - **A greater understanding of the key integration challenges being experienced with existing field integration of the existing systems**

LOCKHEED MARTIN

Lessons Learned

- The C4ISR AF Products are robust in terms of capturing the architecture, but are problematic when they need to be rigorously modeled on the path to producing a system that is compliant with that architecture.
- The recommended documentation of the artifacts from this architecture is not consistent across all the products from a system or software "methodology representation" standpoint.

LOCKHEED MARTIN

Lessons Learned

- The domain of Systems Engineering (SE) and the essential the elements of an Enterprise Architecture Framework (EAF) are tightly coupled
  - SE artifacts (information) is basically leveraged to provide the basis for a "system architecture”
  - Good SE and producing the C4ISR/AF are synonymous
- In addition, SE has become synonymous with building systems that are cost effective and most likely will meet users' real (as opposed to perceived) needs.

LOCKHEED MARTIN

Lessons Learned

- The C4ISR Architecture Framework is the first instance of the Federal Architecture Framework to be extensively adopted and implemented throughout the DOD for all new systems.
- The C4ISR/AF, in its present form (Version 2.0), can be found at www.c3i.osd.mil
  - Represented as 27 products that capture information (or views) about the architecture (see Table below).

The C4ISR/AF Standard Products

LOCKHEED MARTIN

Product	C4ISR Name	System Architect Solution
AV-1	Overview and Summary Information	AV-1 Overview and Summary Information Definition*
AV-2	Integrated Dictionary	The System Architect Repository
AV-3	Capability Maturity Profile	Future
OV-1	High-Level Operational Concept Description	OV-1 Operational Concept Diagram*
OV-2	Operational Node Connectivity Description	OV-2 Operational Node Connectivity Diagram*
OV-3	Operational Information Exchange Matrix	OV-3 Operational Information Exchange Report*
OV-4	Organizational Relationships Chart	OV-4 Organization Chart Diagram*
OV-5	Activity Model	OV-5 Activity Model Diagram and Node Tree Diagram
OV-6a	Operational Rules Model	OV-6a Operational Rules Model Diagram
OV-6b	Operational State Transition Description	OV-6b Operational State Transition Diagram*
OV-6c	Operational Event/Trace Description	OV-6c Operational Event/Trace Diagram*
OV-7	Logical Data Model	OV-7 Logical Data Model Diagram
SV-1	System Interface Description	SV-1 System Interface Diagram*
SV-2	Systems Communications Description	SV-2 Systems Communication Diagram*
SV-3	Systems(2) Matrix	SV-3 System to System Matrix and Report*
SV-4	Systems Functionality Description	SV-4 Data Flow Diagram and Decomposition Diagram
SV-5	Operational Activity to System Function Traceability Matrix	SV-5 System Function to Operational Activity Matrix and Report*
SV-6	System Data Exchange Matrix	SV-6 System Data Exchange Report*
SV-7	System Performance Parameters Matrix	Performance Relate Definitions and Report*
SV-8	System Evolution Description	Derived from use of SA repository information
SV-9	System Technology Forecast	Technology & Technology Area Definitions + Report*
SV-10a	System Rules Model	Property on System Function using BNF Syntax
SV-10b	Systems State Transition Description	SV-10b Systems State Transition Diagram*
SV-10c	Systems Event/Trace Description	SV-10c Systems Event/Trace Diagram*
SV-11	Physical Data Model	SV-11 Physical Data Model Diagram
TV-1	Technical Architecture Profile	Technical Architecture Profile Definitions + Report*
TV-2	Standards Technology Forecast	Standards Technology Forecast Definitions + Report*
*Capability added with C4ISR extension to System Architect		

LOCKHEED MARTIN

Lessons Learned

- What was not addressed in the framework is how each organization would capture and utilize these products in a “modeling environment”.
  - What “process” and “methods” should be used to capture these artifacts of the architecture?
  - What modeling “methodologies” should be used to capture these artifacts of the architecture?
- Each contractor and government agency must decide how they will utilize the framework to mange the Enterprise

LOCKHEED MARTIN

Lessons Learned

- Issue: focus by users of the framework on the "cells" of the framework which delineate these products.
- The "cross-cell" linkages need to be more explicit for architects and engineers to use these views effectively:
  - to ensure consistency between views
  - Accuracy and completeness of those views (or "cells" within the framework)
- Problem: Moving too quickly in "modeling" the architecture before maturity and stabilization of architecture

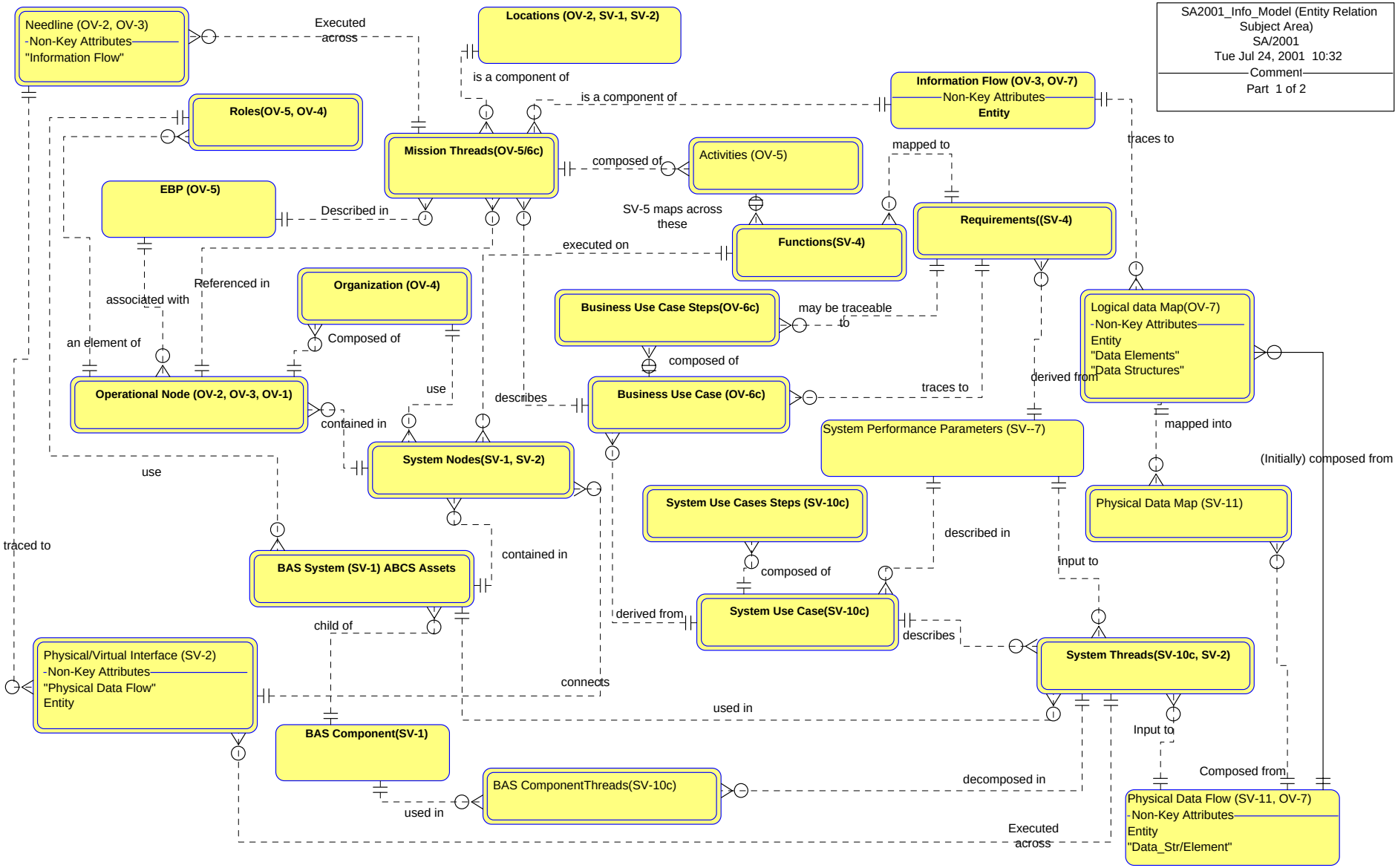
LOCKHEED MARTIN

Lessons Learned

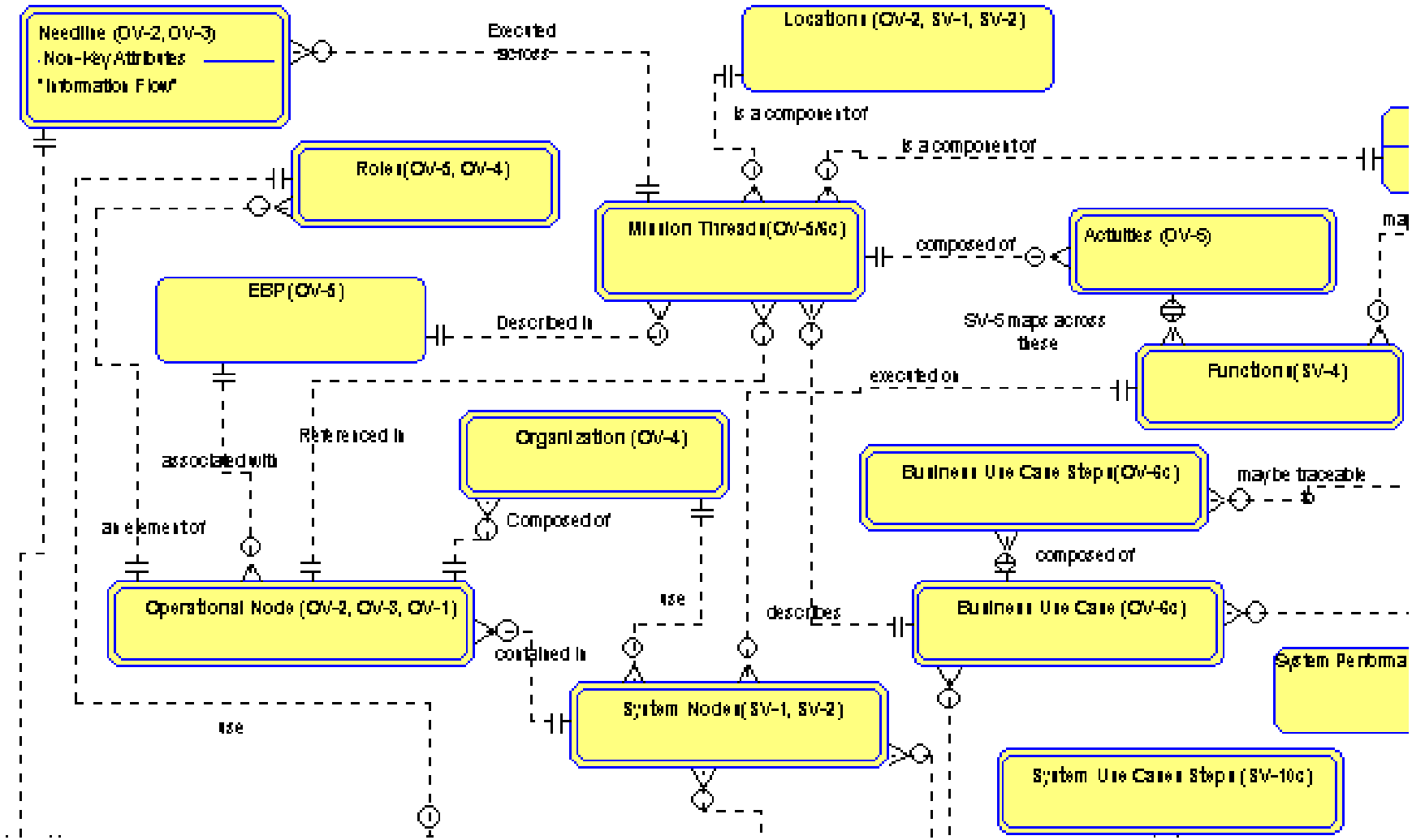
- Important: Before attempting to capture information in the tool, develop the schema
  - This plan for capturing “information”about, or applicable to, architecture
- Don't capture data in the tool from day one
  - Maturity of key information drives the architecture
  - Capture outside the tool quickly then import into components of C4ISR/AF Products

C4ISR/AF Information Entities Schema

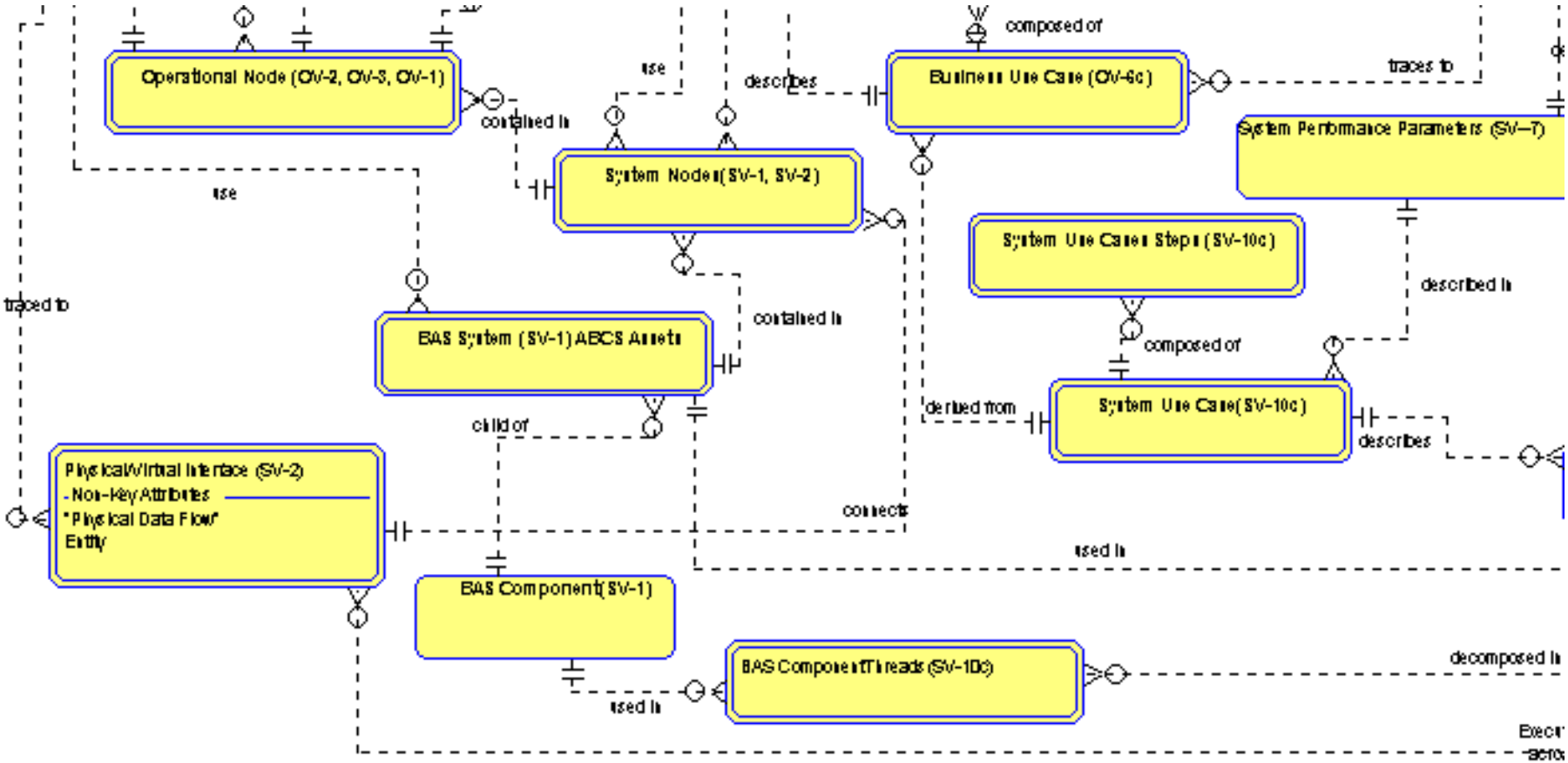
LOCKHEED MARTIN

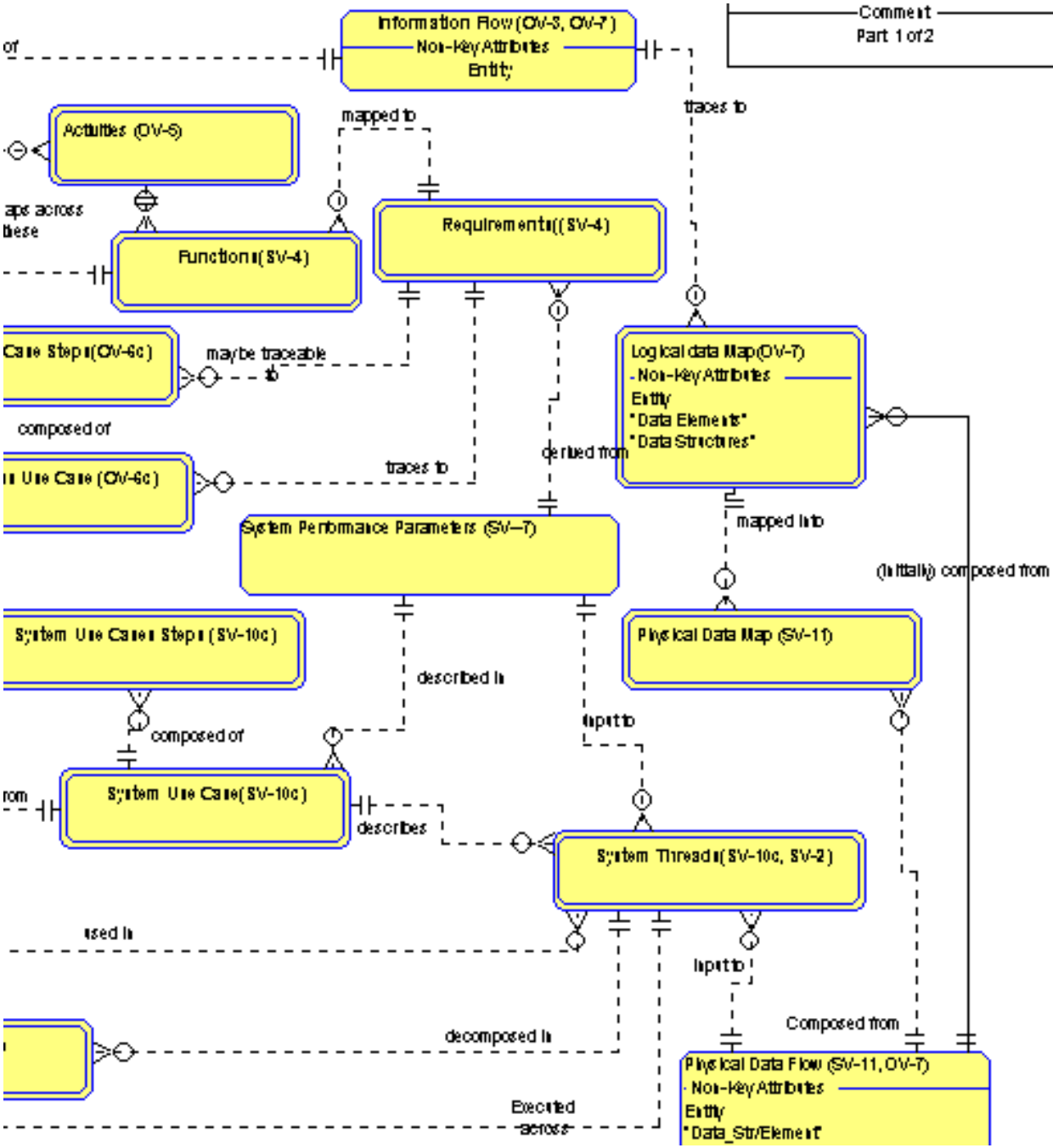


Partial View - C4ISR AF Information model



Partial View - C4ISR AF Information Model



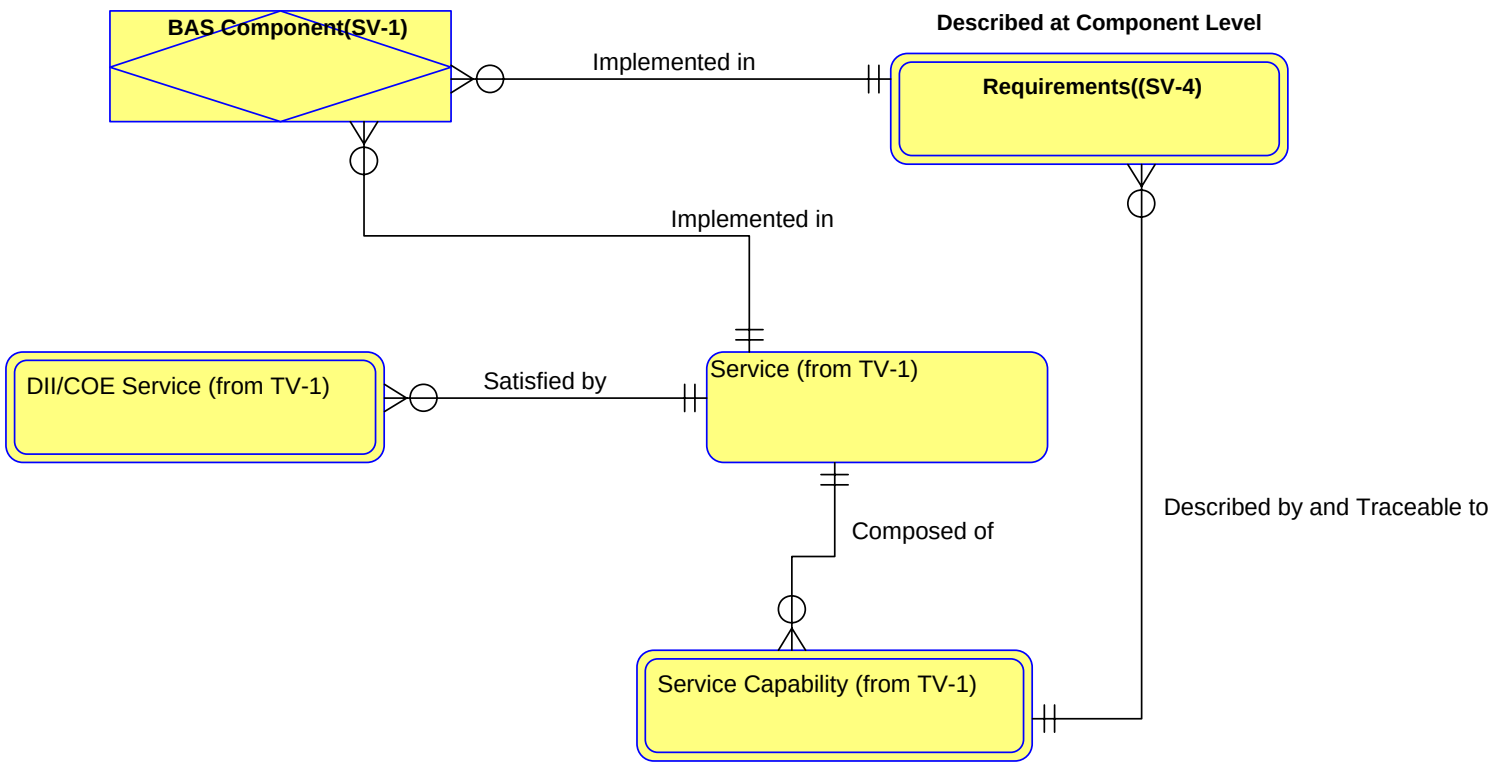


Partial View C4ISR AF Information Model

LOCKHEED MARTIN

SA2001_Info_Model_part2
(Entity Relation Subject Area)
SA/2001
Tue Jul 24, 2001 10:40
Comment
Part 2 of 2

Technical Reference Model



Lessons Learned

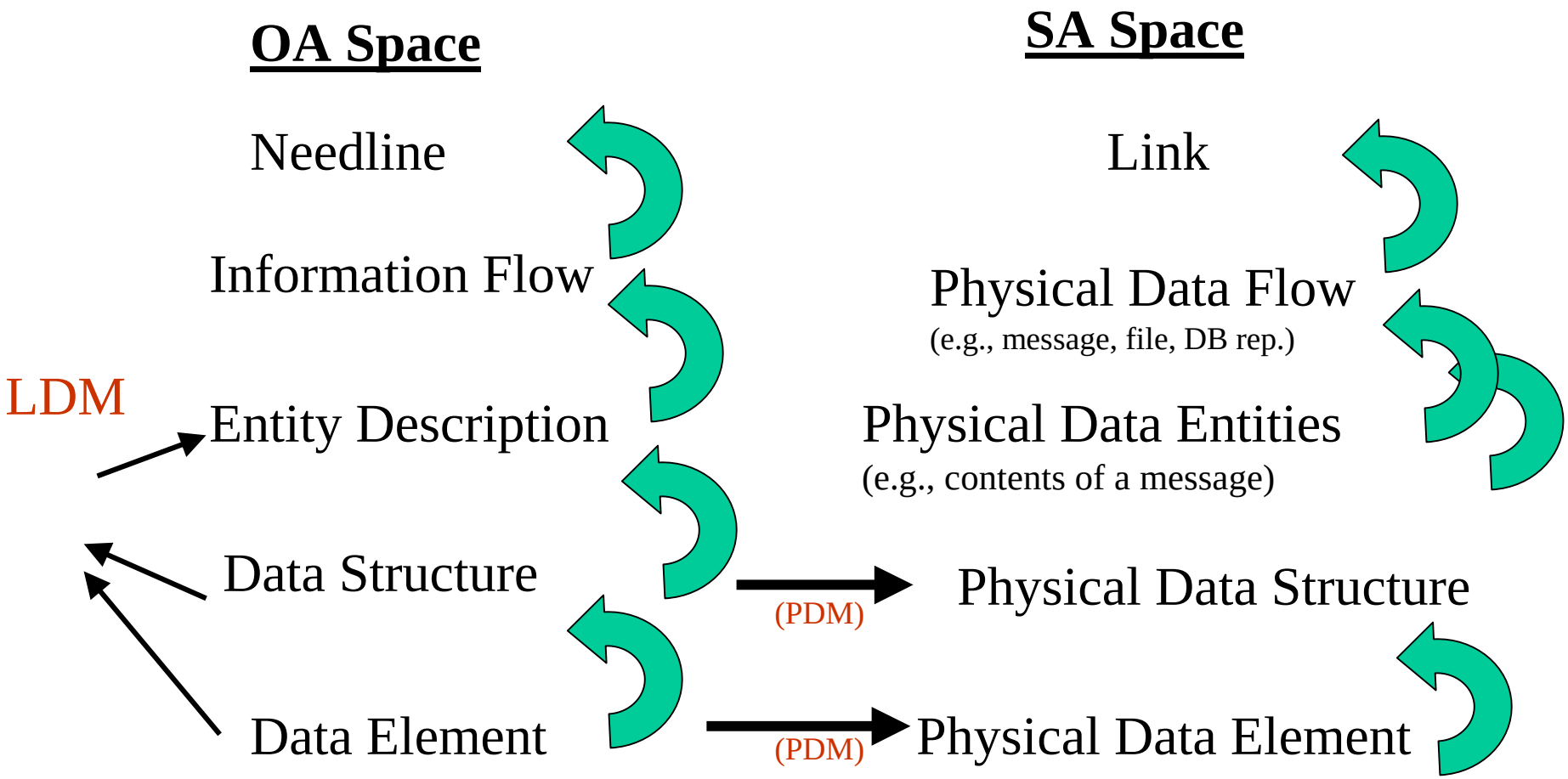
- There are problems with taking these views and relating them to SE products, methodologies, or processes.
  - Example: (OV-5 product - Activity Model) and linking time ordered relationships of the activities (OV- 6c product - Sequence diagram)
  - No clean way to relate the two (i.e., driving out information flows across processes and into Node interactions).
  - Bottom Line: The principles of the Framework indicate views are “recommended” representations of cells
    - i.e., IDEF0 may not fulfill your system developer’s needs, but may be fine for the business modeler

LOCKHEED MARTIN

Lesson Learned

- Follow the data!
- Developers often interpret the logical and physical data models in C4ISR/AF as the database design artifacts; in fact they are not
- You must account for all data flowing throughout the system
- Next chart highlights the implied data hierarchy

Data Map Hierarchy

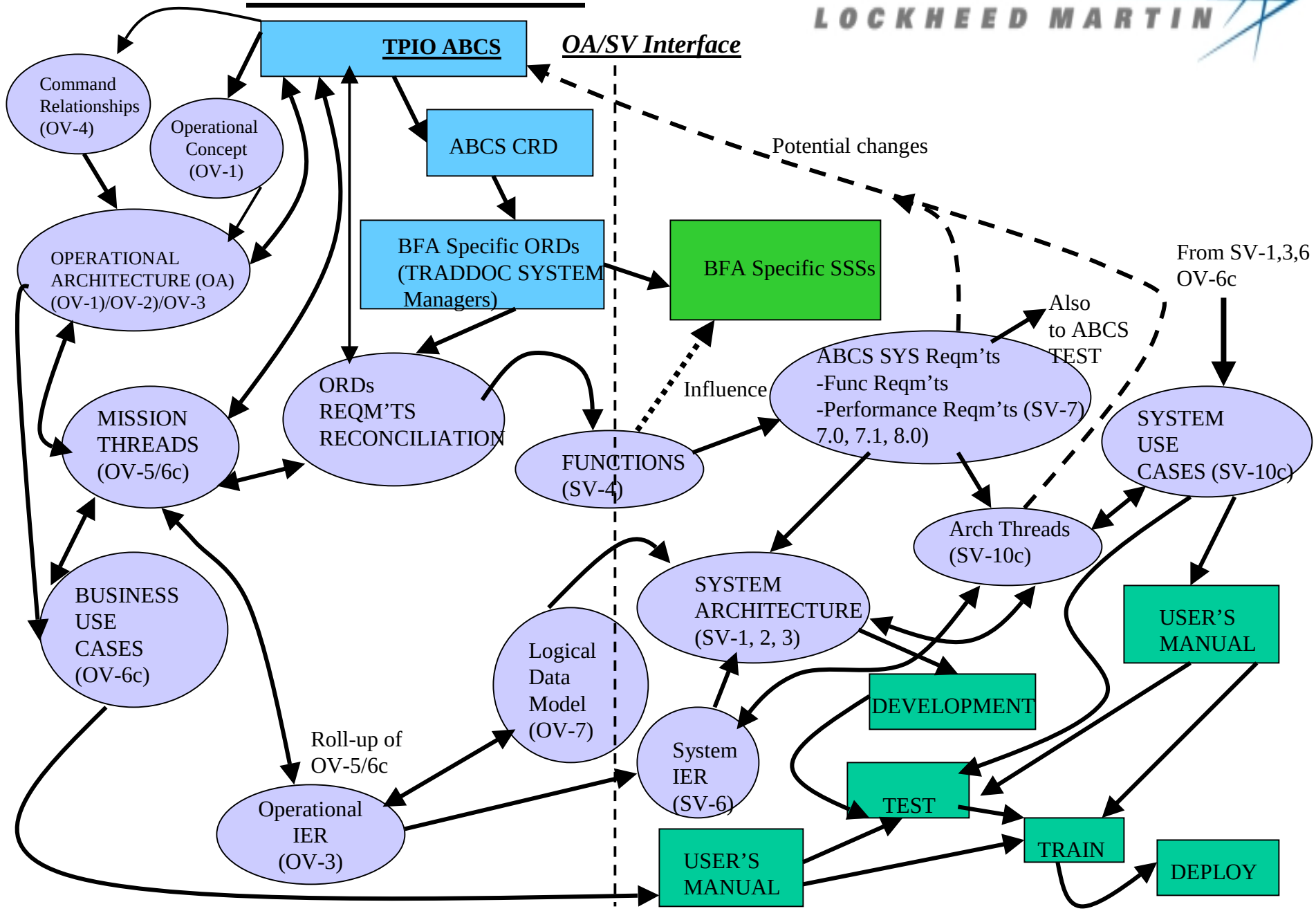


LOCKHEED MARTIN

Lessons Learned

- Many people focus on working to get the data that goes into a cell in the Framework.
  - What must be done is organizing around the "Information Entities" that make a cell in the framework
  - These Entities are artifacts of a system design and development process.

S of S SE Process



LOCKHEED MARTIN

Lessons Learned

- **To briefly summarize what was learned, the key here is for the systems engineers and mission analysts to:**
  1) **Understand all the information that needs to be captured about the mission, system, and technical views of the system**
  2) **Develop a strategy for gathering this information in a structured way (Levis, et.al, 2000)**
  3) **Combine the information Entities in ways that create stakeholder (architecture) views (e.g., the 27 C4ISR AF recommended products)**
  4) **Choose complementary methodologies in the CASE tool environment that allow relationships to be maintained between information Entities through complementary modeling methodologies for each architecture view.**

LOCKHEED MARTIN

Lessons Learned

- Utilize a CASE tool (e.g., SA 2001 by Popkin Software) for capturing the multiple methodologies needed to capture C4ISR AF products and link the Information entities from all the views
  - This adds tremendous value to the communication and synchronization of these views
  - Facilitates the linking of views and architecture artifacts without trying to bridge amongst different CASE tools

LOCKHEED MARTIN

Acronym List

- GCCS-A = Global Command and Control System - Army
- MCS = Maneuver Control System
- FBCB2 = Force XXI Battle Command Brigade and Below
- ABCS = Army Battle Command System
- SSEIC = System of Systems Engineering & Integration Contract
- BFA = Battlefield Functional Area
- ORD = Operational Requirements Document
- TPIO = Technical Program Integration Office
- OA = Operational Architecture
- SV = System Architecture
- S of S = System of Systems

LOCKHEED MARTIN

Reference

– Levis, Alexander H. and Wagenhals, Lee W., "C4ISR Architectures: I. Developing a Process for C4ISR Architecture Design." *Journal of the International Council on Systems Engineering*, vol. 3, no. 4, 2000, pp. 225-247.