A Natural Approach to DoDAF: Systems Engineering and CORE®

Joseph Maley
Senior System Engineer
Vitech Corporation

James Long
President
Vitech Corporation

www.vitechcorp.com

In 1998, the United States Department of Defense mandated the Command, Control Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) Architecture Framework for all ongoing and future contracted architectures. The goal was to ensure that future military systems are interoperable and provide the warfighter with the support and effectiveness required for successful missions. In February 2004, the Department of Defense issued a new release, superceding C4ISR AF, for Architecture Frameworks, Department of Defense Architecture Framework (DoDAF), Version 1.0.

The DoD Architecture Framework specifies a set of "standard" views capturing various system perspectives. As with nearly all frameworks, the outline and contents are defined, but the methodology and support aids are left to the developmental organization's discretion. Many organizations implement processes that develop and manage the various DoDAF artifacts as independent deliverables leading to artifacts which are often inconsistent. Removing these inconsistencies occupies much of the time and resources at every stage of development. Failing to recognize inconsistencies leads to actual developmental, integration, and operational problems along with expensive retrofit efforts.

Often, Program Offices and other responsible organizations estimate the cost, schedule, and other resource requirements (for program justification) from the architecture documentation without significant use of systems engineering tools. Managing frameworks, not just the DoD Architecture Framework, and integrating the relevant information into a single integrated package by applying proven systems engineering processes, methodologies, and tools results in early system insight to support program justification and planning. In addition, this results in higher quality systems, manageable and effective processes, and is less expense than other approaches. All of these benefits lead to lower developmental risk.

Background on C4ISR and DoD Architecture Frameworks

To ensure that future military systems are interoperable, remain interoperable, and provide the warfighter with the support and effectiveness that is required, the Department of Defense (DoD) in 1998 mandated the use of the C4ISR Architecture Framework for use on all ongoing and future architectures. The current Department of Defense Architecture Framework (DoDAF) continues to provide guidance for describing architectures for both warfighting operations and business operations and processes. Systems structured according to the DoDAF should be comparable, thereby supporting management efforts to determine a solution that best promotes compatibility, interoperability, and utility to the user community. Therefore, a cohesive, collaborative, and integrated approach to analyzing, understanding, and specifying information exchange requirements is essential to the success of interoperable military systems. DoDAF is the means for determining interoperability. Furthermore, DoDAF products are used in developing a Capstone

Requirements Document (CRD)[1], Capability Development Document (CDD)[2] and Capability Production Document (CPD)[3]. However, developing and implementing DoDAF views independently — disjoint from themselves and subsequent requirement generation, procurement and development activities — leads to cost, schedule, and developmental inefficiencies. Using a systems engineering tool such as CORE® in concert with proven systems engineering processes significantly reduces these inefficiencies, establishes a stronger base for decision making, smoothes development, and allows management to control strategic and developmental risk more successfully.

The DoD Architecture Framework:

- Defines a set of architectural products and views in three perspectives: Operational, System, and Technical (Figure 1 shows the relationships between the DoDAF views);
- Provides direction on how to describe architectures; and
- Defines output products that are intended to provide a common basis for comparing and evaluating architectures. (Table 1 lists the prescribed DoDAF Products.)

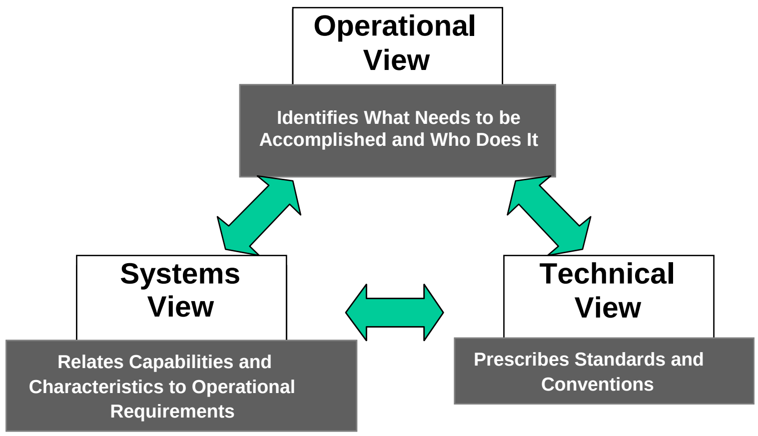


Figure 1 – Linkages Between the DoDAF Views

Table 1 - DoDAF Products

AV-1	Overview and Summary Information
AV-2	Integrated Dictionary
OV-1	High-Level Operational Concept Graphic
OV-2	Operational Node Connectivity Description
OV-3	Operational Information Exchange Matrix
OV-4	Organizational Relationships Chart
OV-5	Operational Activity Model
OV-6a	Operational Rules Model
OV-6b	Operational State Transition Description
OV-6c	Operational Event-Trace Description
OV-7	Logical Data Model
SV-1	Systems Interface Description
SV-2	Systems Communications Description

[1] CJCSI 6212.01C, Interoperability and Supportability of National Security Systems and Information Technology Systems, 20 November 2003, page D-3. See the six-step methodology recommended to develop CRD Interoperability Key Performance Parameters (I-KPPs).
[2] Ibid, page F-2. See the seven-step methodology recommended to develop CDD Net-Ready (NR) Key Performance Parameters (KPPs).
[3] Ibid, page G-2.

SV-3	**Systems-Systems Matrix**
SV-4	**Systems Functionality Description**
SV-5	**Operational Activity to Systems Function Traceability Matrix**
SV-6	**Systems Data Exchange Matrix**
SV-7	**Systems Performance Parameters Matrix**
SV-8	**Systems Evolution Description**
SV-9	**Systems Technology Forecast**
SV-10a	**Systems Rules Model**
SV-10b	**Systems State Transition Description**
SV-10c	**Systems Event-Trace Description**
SV-11	**Physical Schema**
TV-1	**Technical Standards Profile**
TV-2	**Technical Standards Forecast**

Note: An integrated architecture as referenced in DoDI 5000.2, DoDI 4630.8, CJCSI 3170.01, and CJCSI 6212.01 consists of AV-1, AV-2, OV-2, OV-3, OV-5, SV-1, and TV-1, at a minimum. Guidance for program-specific tailoring can be found in DoDAF Volumes 1, 2 and the DoDAF Deskbook.

Problems

1. The guidance provided by the DoD Architecture Framework suite of publications (two volumes and a deskbook) together is neither specific nor complete on how to design architectures. Without a common process or procedure applied to an engineering problem, it is still difficult to compare the products developed by various teams.
2. Additional design detail beyond that required for generating the identified DoDAF views is needed. The DoDAF documents permit “engineering by viewgraph” which is well known as an inadequate approach for the design of complex systems.
3. The process requirement for sufficient data to generate and maintain multiple, consistent, related views necessitates the use of a database management system.
4. Specialized tools on the market today provide the capability to generate DoDAF views without regard to how the resulting requirements are to be passed to other activities such as requirements document development, specification development, and systems development. Even though the intent is to bring interoperability, consistency, and cohesiveness to the development of new capabilities, the products from these tools impose an additional data and semantic interface that the requirements engineering and the systems engineering teams must resolve.
5. Once development begins, these DoDAF specialized views may be outside the mainstream of established system requirements, design processes, and frameworks. They must be managed to maintain synchronization with the mainstream as system development evolves because development affects the architecture as refinement occurs.
6. Any specific DoDAF architecture will most likely be needed for developing other or future DoDAF architectures. Maintaining architectures in a design repository is simpler to use and update than managing independent documents.

Solution

Applying the principles, processes, and comprehensive tools of systems engineering to DoDAF is one method to solve this problem set. Capturing the systems, operations, and design elements in a common design repository, unified by a strong systems engineering process, greatly reduces the program’s exposure to unrecognized risk during architectural comparison, selection, and development. Figure 2 presents an overview of CORE's implementation of the systems engineering process.

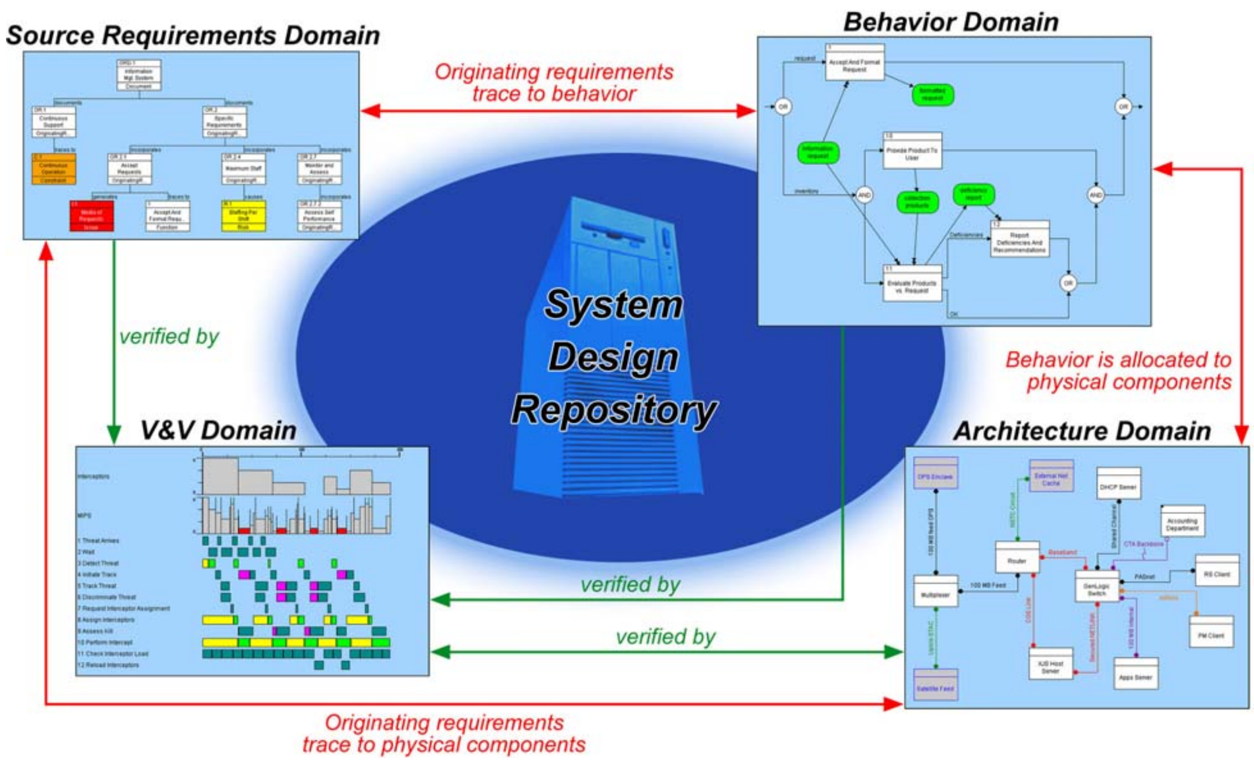


Figure 2 - Overview of the Systems Engineering Process

Vitech Corporation has extended their systems engineering environment to provide one that integrates DoDAF and systems engineering representations. This extension to the underlying systems engineering design language supports the development and specification of the specialized DoDAF architecture-specific requirements and views in harmony with the systems engineering methodology and their resulting views. The DoDAF language extension and its integration and interface with the systems engineering specification language are shown in Figure 3. This integrated environment, along with tailored view and document generators for specific DoDAF products, resolves the data and semantic interface problem associated with using weaker methodologies. This integrated environment is now an integral part of CORE, thereby enhancing collaboration, reducing inconsistencies, and maintaining the integrity of the systems engineering process as one transitions from one system life-cycle phase to another.

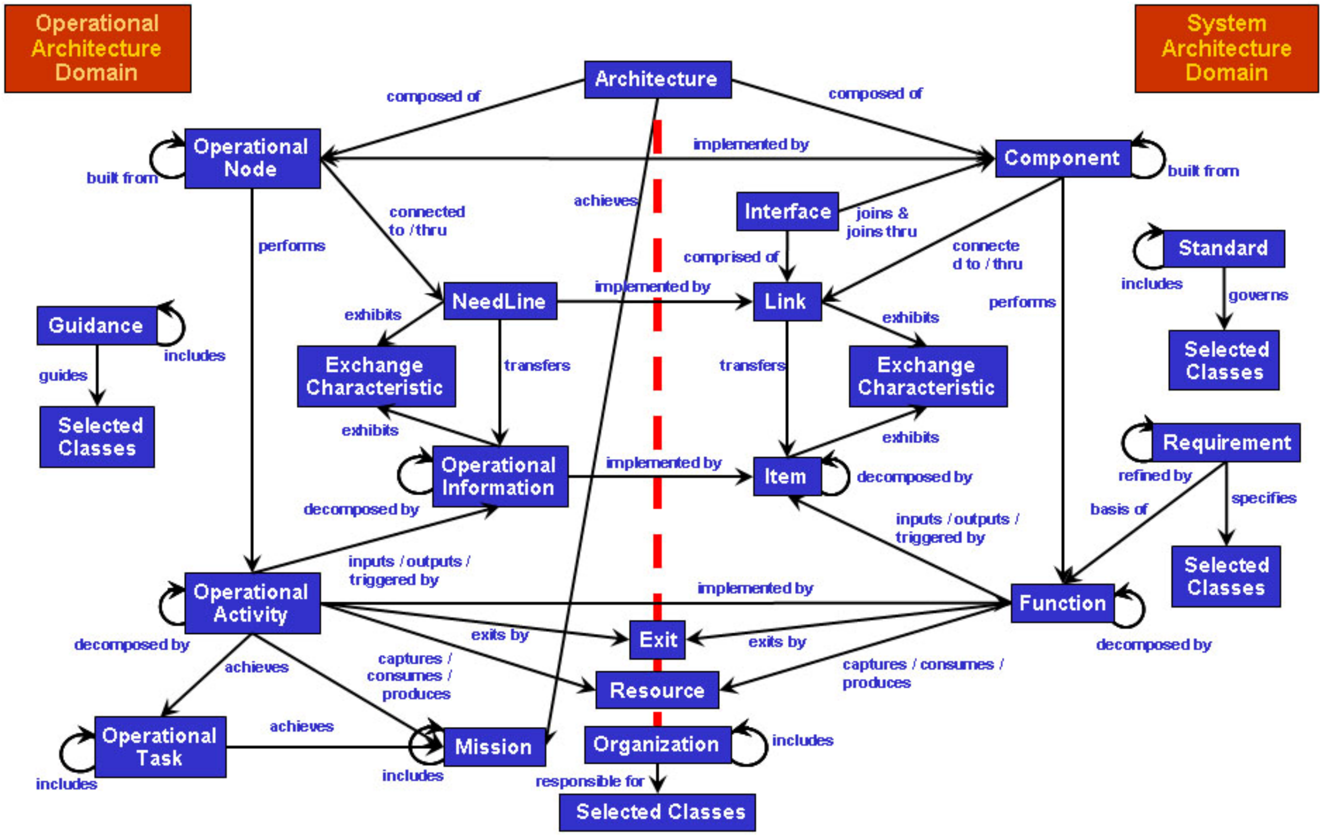


Figure 3 – DoDAF Semantics are Seamlessly Incorporated into the SE Process and Views

Using the same implementation approach as the CORE systems engineering environment, the products and views specified in the Architecture Framework are generated via specialized view generators and/or queries to the CORE design repository. This ensures that the DoDAF views are consistent with each other as well as with the current system design. Figure 4 indicates how the data in the design repository, structured per the semantic-based design language, serves as the basis of the specific views.

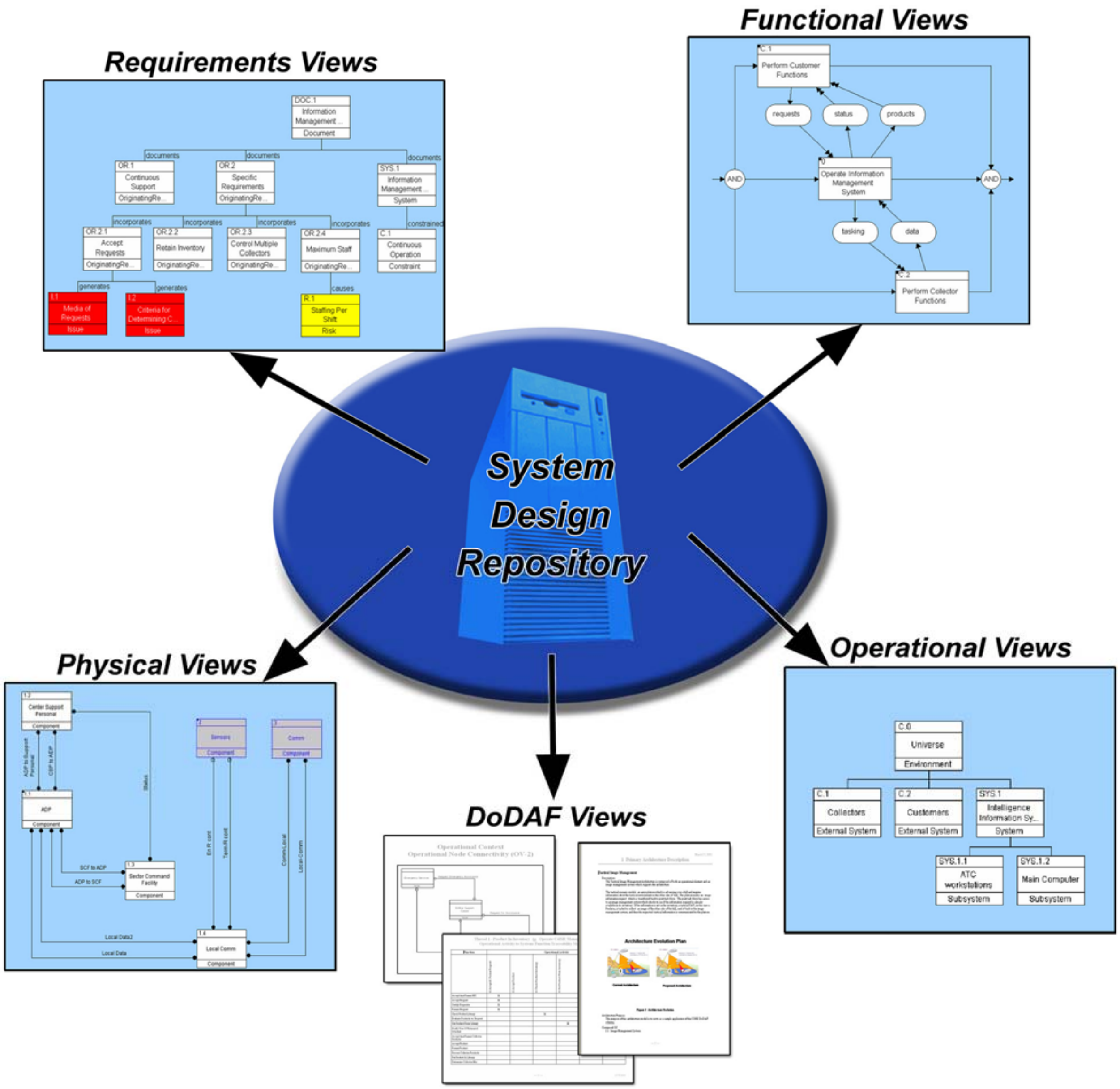
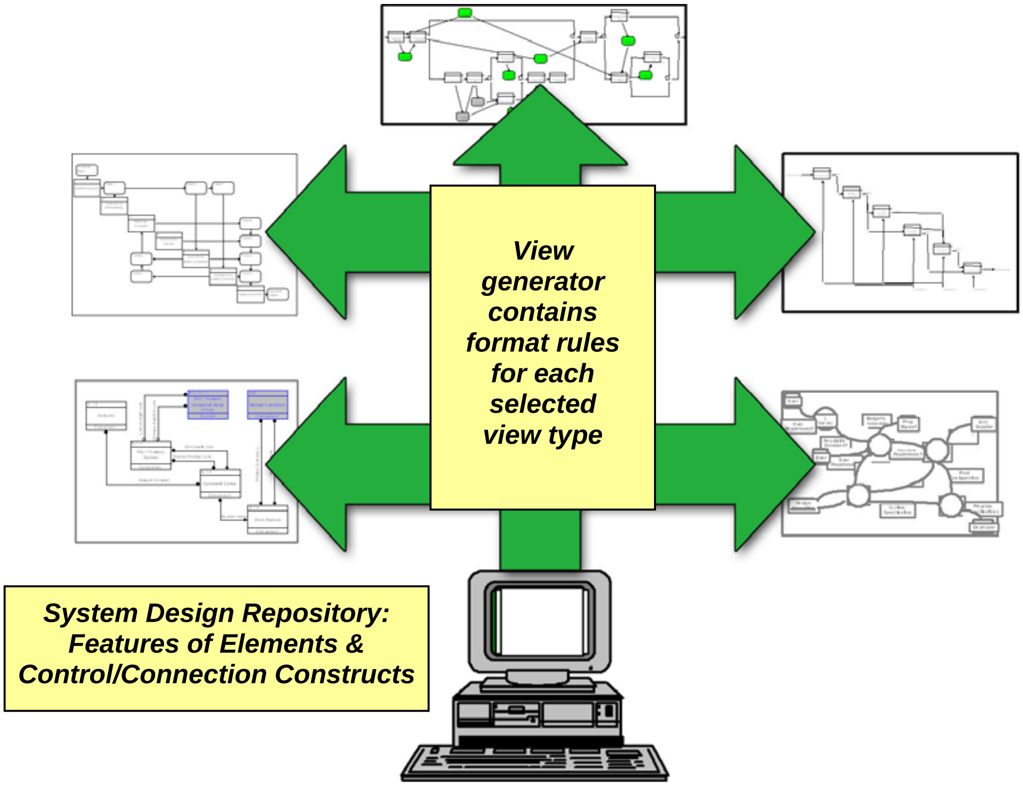


Figure 4 - Integrating the Repository and View Generators Provides Consistency

Figure 5 indicates how the design information in the design repository is combined with the format rules in the view generators to provide the desired set of graphical views, which are consistent by construct. Other views, which are of tabular and text format are likewise generated by use of scripts. This ensures that the DoDAF views are consistent with each other as well as the current system design.



- A graphical view is defined by features and a format
- The features are in the repository
- The format for each view type is defined in the view generator

Figure 5 - View Generators Guarantee Consistent Graphics

Table 2 shows how the specific DoDAF views are generated from the integrated CORE environment, using a combination of graphic view generators, custom report scripts, and links to external files (text and graphics) available from other sources (e.g., PowerPoint, Visio, etc.).

Table 2- CORE Generates DoDAF Views

DoDAF Architecture Products	Product	CORE Representation
Overview and Summary Information	AV-1	Description, purpose, scope, time frame and mission; optional external graphic
Integrated Dictionary	AV-2	Architecture Description Document (ADD) Report
High-Level Operational Concept Graphic	OV-1	Identified in CORE database as an External Graphic. Viewable in CORE and incorporated in CORE reports.
Operational Node Connectivity Description	OV-2	Physical Block Diagram (PBD) for Operational Node displaying the OperationalNodes decomposition with Needlines connecting internal and external nodes.
Operational Information Exchange Matrix	OV-3	OV-3 Report script (Outputs a summary or full information exchange matrix listing Needlines, Operational Information exchanged and the ExchangeCharacteristics.)
Command Relationships Chart	OV-4	Hierarchy Diagram showing command and coordination relationships of the Organization
Activity Model	OV-5	EFFBD, FFBD or IDEF0 representing behavior models including control, input/output, sequencing and decomposition of OperationalActivities
Operational Activity Sequence and Timing Descriptions	OV-6a, 6b, 6c	Complete, executable operational activity model as EFFBDs. Optional output includes rules, a captured COREsim timeline file or an external event trace file.
Logical Data Model	OV-7	OperationalInformation characterization table
System Interface Description	SV-1	Physical Block Diagram for Component or an external file that augments the Component
Systems Communications Description	SV-2	Physical Block Diagram for Component or an external file that augments the Component
Systems-Systems Matrix	SV-3	Matrix representing the interfaces between Components
Systems Functionality Description	SV-4	EFFBD, FFBD, N2 or IDEF0 representing behavior models including control, input/output, sequencing and decomposition of Functions
Operational Activity to Systems Function Traceability Matrix	SV-5	Matrix tracing between the OperationalActivities and Functions with an option to show the Component performing each Function.
System Information Exchange Matrix	SV-6	Summary or full data exchange matrix listing Links, Item exchanged, and ExchangeCharacteristics.
System Performance Parameters Matrix	SV-7	Table containing quantitative performance characteristics (current, threshold, objective for near-term, mid-term, and far-term time frames) for Components and their associated Interfaces, Links and Functions.
System Evolution Description	SV-8	An ExternalFile (text or graphic).
System Technology Forecast	SV-9	Table containing technology forecast information for Components and their associated Interfaces, Links, Functions and Items.
System Activity Sequence and Timing Descriptions	SV-10a, 10b, 10c	Complete, executable functional model as EFFBDs. Optional output includes rules, a captured COREsim timeline file or an external event trace file.
Physical Data Model	SV-11	Item characterization table and/or an ExternalFile.
Technical Standards Profile	TV-1	Table listing Standards governing the Components and their associated Interfaces, Links, Functions, and Items.)
Technical Standards Forecast	TV-2	Table listing future Standards governing the Components and their associated Interfaces, Links, Functions, and Items.)

Table 3 indicates the inefficiencies (leading to extra cost and schedule) of using an independent, ad hoc DoDAF approach instead of an integrated DoDAF / systems engineering approach.

Table 3 – Benefits of an Integrated Approach

Independent	Integrated
Independent drawings	Consistent views
Static diagrams	Executable behavior[4]
Data storage	Linked repository
Stored views	Dynamic view generation
Ad hoc process (inconsistent results)	Repeatable process (consistent results)
Manual change propagation across all affected products (by the systems engineer)	Automatic change propagation across all current and future products (by the engineering environment)

The DoDAF/Systems Engineering Process – Analytical Domains

The following sections are a brief description of the analytical domains in the integrated DoDAF/systems engineering process.

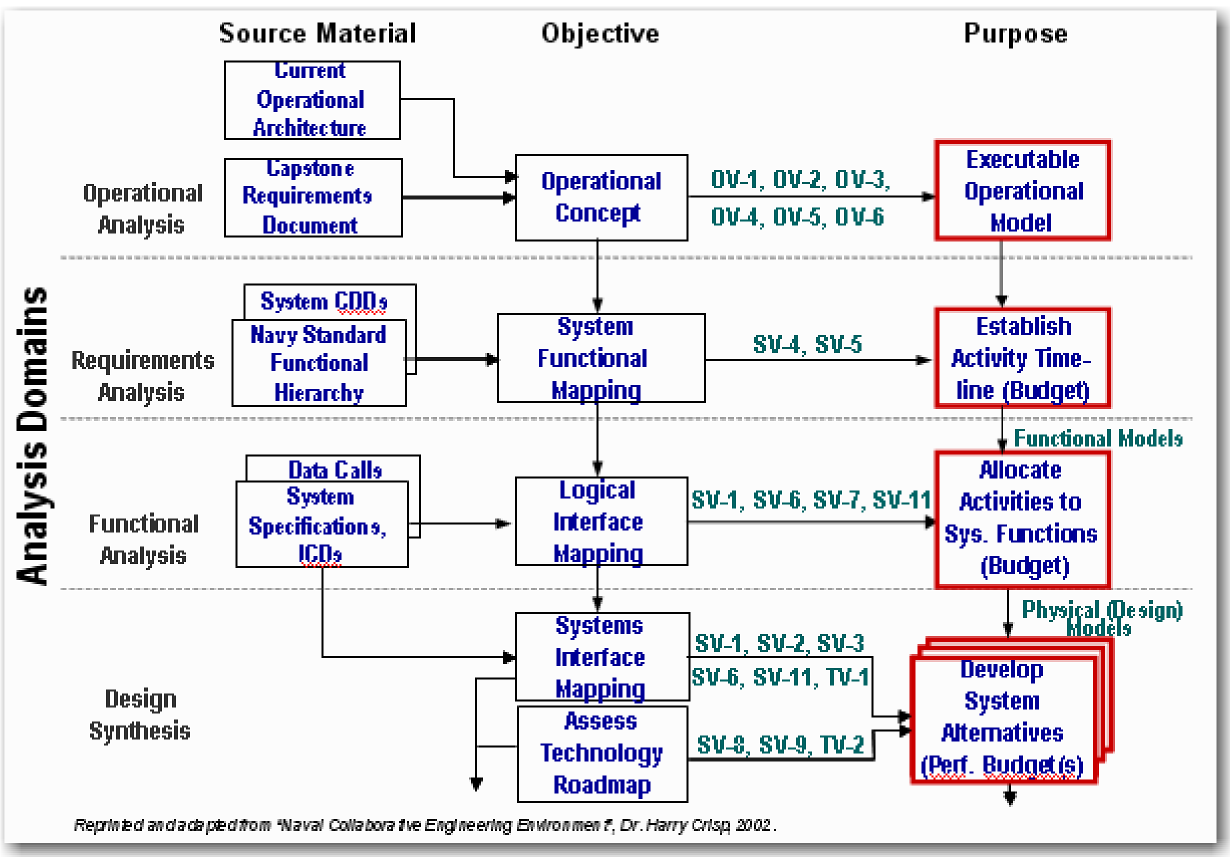
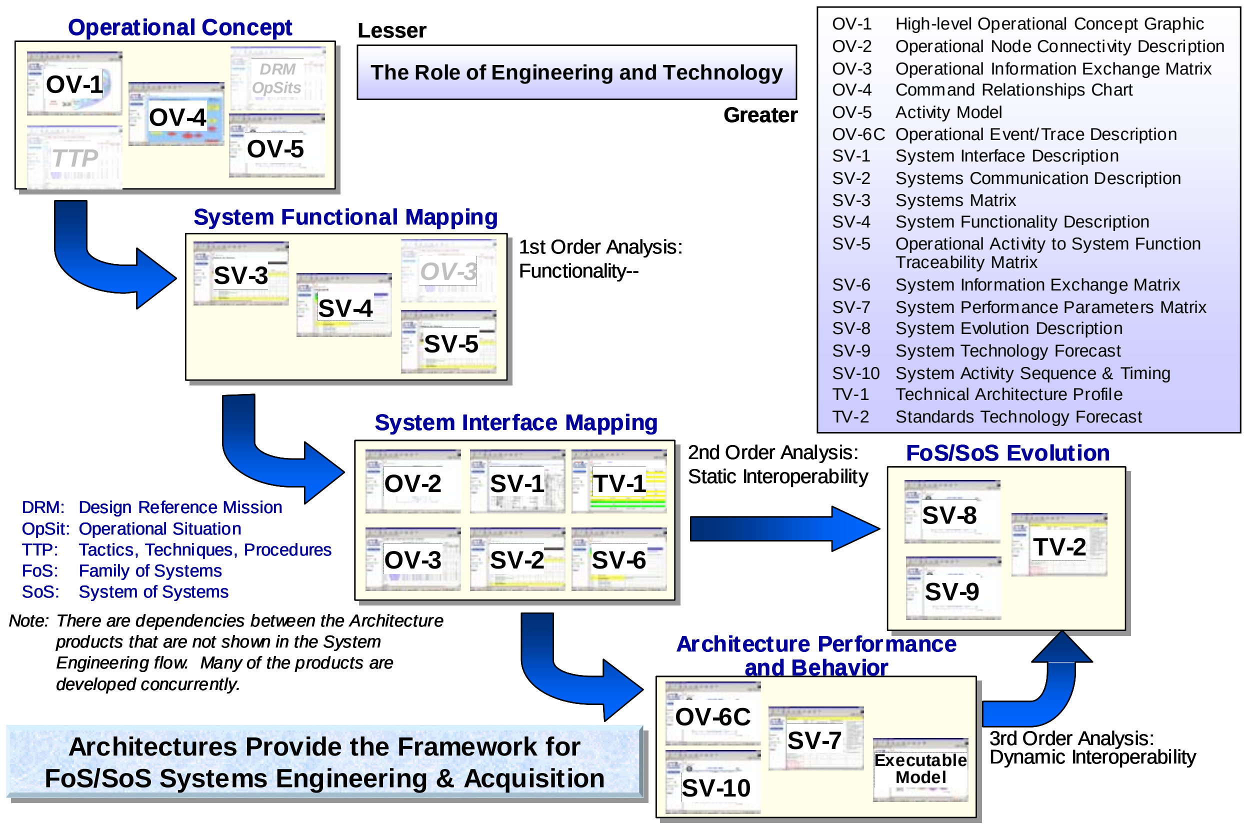


Figure 6 – DoDAF Products as a Function of Analysis Activities

[4] Executable behavior eliminates structural or dynamic inconsistencies from the requirements.

Figure 6 illustrates the relationship among the four analytic domains of systems engineering and the DoDAF products developed within each domain. In each analytic domain, the source, objective, and purpose for the analysis are shown and their dependencies. Figure 7 presents the DoDAF views aggregated in terms of their context and engineering usage.

For this discussion, the DoDAF process using CORE is assumed to begin with the availability of certain information or documents such as an Initial capabilities Document (ICD), Universal Joint Task List (UJTL), etc. Note, however, that the systems engineering approach and CORE are very useful during the earlier concept definition stages in the exploration of mission need and benefit as well as development of the mission concept and operations. This preliminary work is used as reference material for starting the DoDAF process as indicated in Figures 3 and 4. From the operational viewpoint, the analytical space may be partitioned into operational analysis including validation and verification, requirements analysis (including information exchange and assurance requirements[5]), functional (activity) analysis, and design synthesis. These areas of operations engineering are performed concurrently because the decisions in one domain affect one or more other areas. Alternatives may be developed so that various benefits and deficiencies may be compared and weighted.



Reprinted from "Naval Collaborative Engineering Environment", Dr. Harry Crisp, 2002.

Figure 7 – DoDAF Views in Context

Operational Analysis (Validation and Verification)

This analytical domain includes the analysis, validation, and verification activities that support the development of the operational needs and architecture. When DoDAF products support capabilities development, the analysis effort provides the rationale for the information exchange requirements (IERs)

---

[5] Information assurance is defined as: "Information operations that protect and defend information and information systems by ensuring their availability, integrity, authentication, confidentiality, and nonrepudiation."

contained in the Initial Capabilities Document (ICD) and thus provides support for the formal validation of those capabilities as development progresses through acquisition milestones B and C, resulting in the Capability Development Document (CDD) and Capability Production Document (CPD), respectively.

The Operational Concept analysis specifies the organizations, roles, and the interaction of the proposed system in the operational context. This analysis draws on the other activities to support the proposed system definition. The system concepts developed here provide the rationale to the stakeholders for justifying the certain needs or capabilities determined to be necessary for mission support and interoperability.

DoDAF products produced

- OV-1 High-Level Operational Concept Graphic
- OV-2 Operational Node Connectivity Description
- OV-3 Operational Information Exchange Matrix
- OV-4 Command Relationships Chart
- OV-5 Activity Model
- OV-6 Operational Event and Sequence Timing Descriptions

Requirements Analysis

This activity domain identifies some of the system requirements necessary to satisfy the operational needs. Note that the DoDAF does not address all system requirements; the CDD will provide some, for example. However, at least the information exchange and assurance requirements are available. IERs are developed through the identification and refinement of information items crossing the system boundary at each level of refinement. This activity depends upon the threads used as part of the activity analysis domain as they are represented as behavioral models. Information assurance requirements are partially identified by the activity analysis, where the sensitivity of each information element can be determined. However, the threat and vulnerability aspects become more apparent when activities are allocated to the system elements defined during synthesis. Changing allocation affects the choice and ease of developing or employing proper countermeasures or safeguards.

DoDAF products produced

- SV-4 System Functionality Description
- SV-5 Operational Activity to System Function Traceability Matrix

Functional (Activity) Analysis

As part of operations, various organizations and systems implement or perform certain activities to implement the operational concept. CORE's *OperationalActivity* class supports the modeling of operational concepts through identification and definition of operational threads or scenarios. This better supports the identification and capturing of the information flowing across the system boundary. From an information assurance perspective, sensitivity of information is better correlated with possible threats. The degree of detail associated with this effort is directly proportional to the level of analysis required; i.e., the context level description of a system is considerably simpler than describing the more detailed aspects of the lower-level system activities.

This activity also transforms the higher-order abstract functions the system must perform into more refined and detailed descriptions, or lower-level functions. The system-level performance requirements are decomposed as well and allocated to the lower-level functions. By adding structure (behavior modeling), these functions may be sequenced or arranged such that a control viewpoint is established, input/output data dependencies established, and timing relationships determined.

DoDAF products produced

- SV-1 System Interface Description

- SV-7 System Performance Parameters Matrix

For tools such as CORE, where the behavioral models (EFFBDs) are executable using the COREsim discrete event simulator, these models may be simulated to determine if the proposed design will meet the operational needs. Requirements defining performance, resource utilization, data exchange, and processing capacity are just a few of the many such requirements that must be modeled to confirm that the system logic is dynamically consistent (in other words executable) and, furthermore, that the model meets the system performance and timing constraints. Simulation supports analysis of resource utilization, data queuing, communication loading, component activation, loads and duty cycle, and changes in operational environment.

<u>DoDAF products produced</u>

- SV-10 Operational Event and Sequence Timing Descriptions

Design Synthesis (Physical)

The system, the focus of the DoD Architecture Framework effort, will in itself be comprised of other systems, segments, subsystems, etc. Potentially, many architectural choices are available and the selection of one or more architectures to carry forward for further refinement or selection depends on selection criteria. Selection criteria vary and which ones to use are dependent upon the goals and needs of the various stakeholders. Along with any architectural benefits there are usually some tangible or intangible costs or deficiencies that need to be considered during selection. However, regardless of architectural choice, the activities identified through activity analysis must be supported by the selected architectural elements.

<u>DoDAF products produced</u>

- SV-1 System Interface Description
- SV-2 Systems Communication Description
- SV-3 Systems Matrix
- SV-6 System Information Exchange Matrix
- SV-9 System Technology Forecast
- SV-11 Physical Data Model
- TV-1 Technical Architecture Profile
- TV-2 Standards Technology Forecast

DoDAF and Subsequent Engineering and Deployment Support

Creating a system's DoD system's architecture is one of several necessary activities to advance from a mission concept to reality. Typically, a new program will develop other requirement documents along with a procurement package[6]. These are all developed and managed as independent documents requiring considerable effort to maintain technical, logical, and editorial consistency among these documents. Even during the development, fielding, operations and support phases, these documents should be maintained so that the next system upgrade or procurement of another interfacing system can be done without engaging in activities to update the documentation to reflect current operations. By using CORE, all the system information, external context, and external environment are contained in the CORE design repository (database). CORE's systems engineering support functions simplifies the upkeep and allows documents to be produced on an as-needed basis. Since the documents and views are produced from the CORE design repository, consistency is automatically maintained. When the design repository is changed; all the views and documents produced from the design repository reflect that change with no extra effort. Therefore, the system knowledge developed and captured in CORE is readily handed off to the next level team regardless of whether the program is in the concept, development, fielding, or operational stage. In addition, the various stakeholders do not have to be concerned whether they have

[6] Systems engineering practices and CORE can support these other activities as well.

(or have access to) the latest documentation – such documentation is accessible upon demand from the design repository.

CORE Technology

At the heart of the CORE systems engineering environment is an object data repository or design repository that maintains every aspect of a program's design. The object structure of the design repository allows easy query and representation of the data useful to the many stakeholders, whether they be customers, accreditors, managers, engineers, software designers, or testers, to name a few. As an option, CORE2net provides authorized users the ability to browse and comment on the proposed architectures through the use of a standard web browser.

The design repository uses a robust schema, or data structure, built to relate all the engineering artifacts that are critical to defining, specifying, or developing a system that satisfies mission needs. This structure, along with a powerful query and reporting engine, allows the various stakeholders and engineering disciplines to see and modify only their relevant information, without the clutter and distraction of the entire database.

The design repository stores all the architecture framework and systems engineering artifacts in an integrated and consistent database, rather than as a database of separate artifacts, which includes:

- Requirements
- Functional descriptions and graphical models
- Behavioral executable models
- Performance characteristics and constraints
- Operational architectures
- Physical architectures
- Interfaces, data flows and rates
- Responsible organizations
- Technical guidance

Summary

The stated goals of the DoD Architecture Frameworks require more than graphics and simple data capture to be satisfied. Comparability of interoperable systems requires that the system alternatives satisfy the same requirement set, operational context, and the like. Systems engineering with its five decades of successful application to complex technical problems and overlap with the problem space is the natural approach. All systems engineering domain activities (Requirements, Activities, Synthesis, Verification and Validation) need to be employed concurrently; otherwise, only surface or pseudocomparisons can be made. The low risk approach for defining, specifying, and developing complex systems requires the application of both a proven process and tools to capture and present information produced by those participating in the engineering effort.

CORE's implementation of the integrated DoDAF and systems engineering processes provide:

- A repeatable and proven systems engineering methodology with over 30 years of successful application
- Integrated models for technical, operational, and system architectures
- DoDAF products produced directly from the design repository
- A graphical notation to enhance the capture and representation necessary for communication and evaluation of candidate architectures
- Support for the product life cycle
- Executable models (simulation) for behavioral and performance analysis
- Rigorous and extensible database schema to allow for tailored and expanded views.

The CORE systems engineering environment is a powerful, yet flexible tool supporting the latest standards as well as providing organizations with a common design repository in which to store, analyze, and produce engineering information in a collaborative environment. The application of systems engineering and CORE to DoDAF is a straightforward decision for the defining, specifying, and developing of successful systems. If one wants to take the effort from concept to operations, then DoDAF via systems engineering and CORE is unquestionably the answer.

References

C4ISR Architecture Framework, version 2.0, December 1997.

DoD Architecture Framework Version 1.0, (Volume I: Definitions & Guidelines; Volume II: Product Descriptions; Deskbook), 09 February 2004.

INCOSE Systems Engineering Handbook version 2.0, July 2000.

Crisp, *Harry E., Naval Collaborative Engineering Environment*, Presentation to INCOSE WMA, May 2002.

Dam, Steven H. and James D. Willis, *Developing a Defensible C4ISR Architecture*, INCOSE Tutorial, February 2002.

CJCSI 3170.01D*, Joint Capabilities, Integration and Development System,* 12 March 2004.

CJCSI 6212.01C, *Interoperability and Supportability of Information Technology and National Security Systems*, 20 November 2003.

Note CORE and the CORE logo are registered trademarks of Vitech Corporation. This paper was originally published in August 2002; and substantially updated May 2005. Patrick MacDonald co-authored the original paper.