

The *C4ISR* Architecture Framework: History, Status, and Plans for Evolution

P. Kathie Sowell ¹
The MITRE Corporation
McLean, Virginia

Abstract

An architecture is “the structure of components, their relationships, and the principles and guidelines governing their design and evolution over time.”

– C4ISR Architecture Framework Version 2.0

The *Command, Control, Communications, Intelligence, Surveillance, and Reconnaissance (C4ISR) Architecture Framework, Version 2.0*, developed by the U.S. Department of Defense (DoD) C4ISR Architecture Working Group, provides guidance for describing architectures. It is intended to ensure that architectures developed by the Commands, Services, and defense Agencies are interrelatable between and among the organizations’ operational, systems, and technical architecture views, and are comparable and integratable across Joint and multi-national organizational boundaries. The Framework is intended to ensure that a clear audit trail exists from mission operations and effectiveness measures to the characteristics of current and postulated C4ISR systems and their contributions (performance and interoperability metrics) to mission operations.

This paper describes the four main components of the Framework, i.e., Architecture Views (Operational, Systems, and Technical) and Linkages, Common Product Templates and Common Data, Universal Guidance, and Common Building Block References.

Relationships to other popular and emerging architecture frameworks are described, along with current plans for further evolution of the Framework.

1.0 History of the Framework

The initial impetus for the Framework came from the Defense Science Board, who determined in the early 1990s that one of the key means for ensuring interoperable and cost effective military systems is to establish comprehensive architectural guidance for all of DoD. Consequently, the C4ISR Integration Task Force developed version 1.0 of the *C4ISR Architecture Framework* in June of 1996, and the C4ISR Architecture Working Group completed version 2.0 in December of 1997, under the auspices of the Architecture Coordination Council (ACC) [C4ISR Architecture working Group, 1997].

¹ The author wishes to acknowledge the support of the agencies that sponsor the work related to this paper: the Office of the Assistant Secretary of Defense (C3I), Architectures and Interoperability Directorate; the Federal CIO Council; and the Department of the Treasury.

In a 23 February 1998 memorandum, the Under Secretary of Defense (Acquisition & Technology), the Acting Assistant Secretary of Defense (C3I), and the Joint Staff Director of C4 Systems (J6) mandated the *C4ISR Architecture Framework, Version 2.0* for use in all C4ISR or related architectures. In addition, they directed that the *Framework* be examined as the basis for a single architecture framework for all functional areas or domains within the DoD [Architecture Coordination Council, 1998].

2.0 Framework Overview

2.1 Why Do We Need an Architecture Framework?

Development of C4ISR architectures is a distributed process. Because there has been no uniform guidance governing architecture development, DoD components have described their respective architectures using a variety of disparate perspectives, formats and terminology. It has been virtually impossible to interrelate or compare one architecture with another, an integration process we must perform in order to identify interoperability issues and to find opportunities for technology leveraging and sharing.

Using the Framework over time, architectures can be dovetailed and opportunities for enhanced interoperability, integration, and cost-effectiveness will be easier to identify and act upon.

The Information Technology Management Reform Act and the Government Performance and Results Act require Federal Government organizations to measure the performance of existing and planned information systems and to report performance measures annually. The Framework can help DoD organizations to satisfy these reporting requirements by providing uniform methods for describing information systems and their performance in context with mission and functional effectiveness.

2.2 Framework Components

The Framework has four main parts: definitions of three standard views of any given architecture; common products (descriptive models) and data; common building block references; and high-level guidance in how to use the Framework to describe an architecture.

2.2.1 Definitions of Architecture Views

The Framework defines three views of any given architecture. Figure 1 illustrates these three views and their relationships.

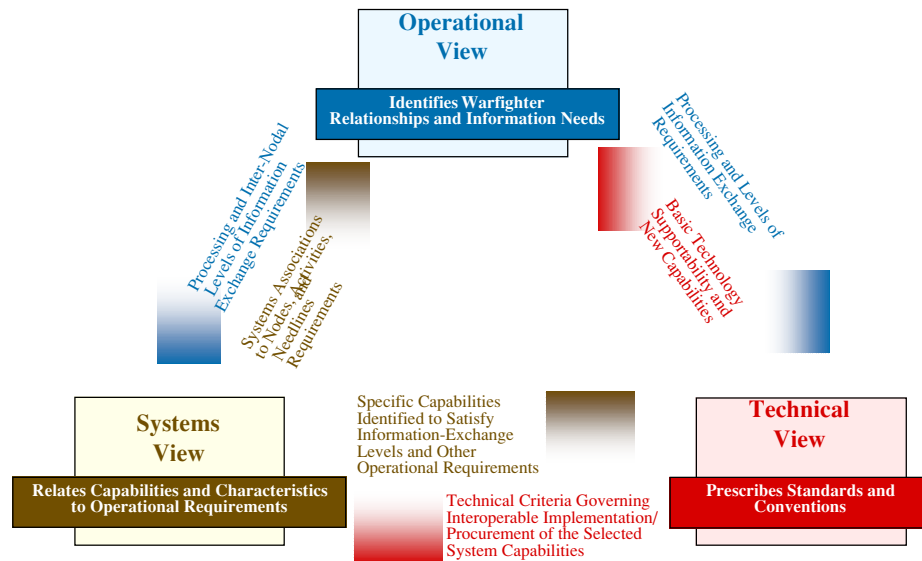


Figure 1. One Architecture...Three Views

The operational view describes the tasks and activities, the operational nodes, and the information flows between nodes that are required to accomplish or support an operation. The operational view describes the nature of information exchanges in detail sufficient to determine what specific degree of information-exchange interoperability is required.

The systems view translates the required degree of interoperability into a set of system capabilities needed, identifies current systems that are used in support of the operational requirements (or postulated systems that could be used), and facilitates the comparison of current/postulated system implementations with the needed capabilities.

The technical view articulates the criteria that govern the implementation of required system capabilities.

To be consistent and integrated, an architecture description must provide explicit linkages among its various views. The Framework product set, described briefly in the next paragraphs, provides a number of such linkages among the views.

2.2.2 Common Products and Data

Products are the representation formats, and the required data, that all of the DoD components will use to describe their C4ISR architectures. The **essential products** are those that every architecture description must include, provided that the subject view is included in the architecture. The **supporting products** are those that will be needed for some architecture descriptions, depending on the purpose of the architecture. The Framework describes seven essential products and 19 supporting products.

It is critical to understand that a given product type is designated as “essential” because it is one that higher-level integrators and decision makers will need to see in

order to make comparisons and budget decisions across multiple architectures. Other products may be equally necessary, or even more necessary, for a specific architecture effort, but if those products do not need to be seen by the higher level decision-makers, those products are designated “supporting.” Simply stated, “essential” means “essential for cross-architecture analysis.”

The product set was designed with relationships and connections among the products in mind. These connections and relationships not only facilitate a more complete representation of a given architecture, they also provide a means for relating technology to mission requirements.

There are three essential products in the operational view:

The **High-level Operational Concept Graphic** (figure 2) is the most general of the architecture-description products. Its main utility is as a facilitator of human communication, and it is intended for presentation to high-level decision makers. This kind of diagram can also be used as a means of orienting and focusing detailed discussions. The graphic should be accompanied by explanatory text.

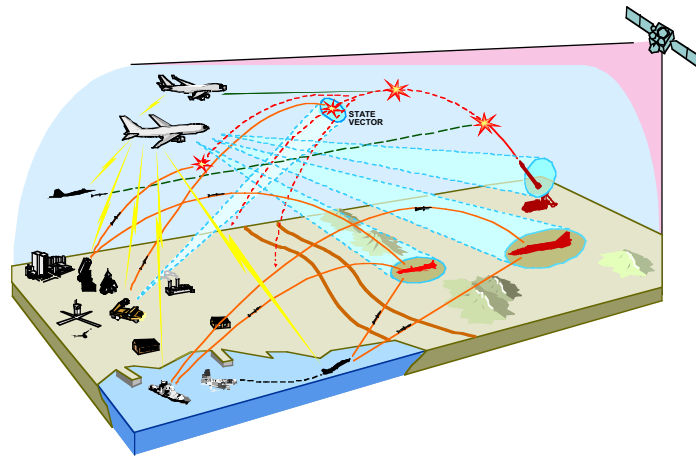


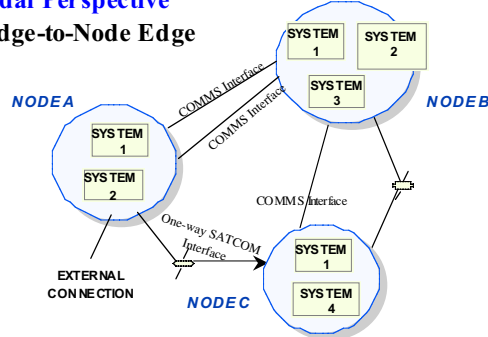
Figure 2. High-Level Operational Concept Graphic

The **Operational Node Connectivity Description** (figure 3) depicts the operational nodes, the needlines between them, and the characteristics of the information exchanged. A needline is simply an indication that two nodes need to exchange information; it does not state how that exchange is accomplished, i.e., with what systems or networks.

Essential Product: System Interface Description

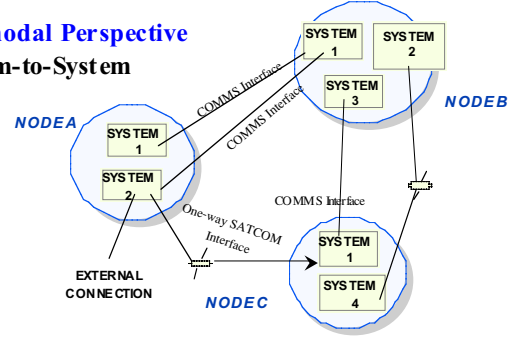
Internodal Perspective

Node Edge-to-Node Edge

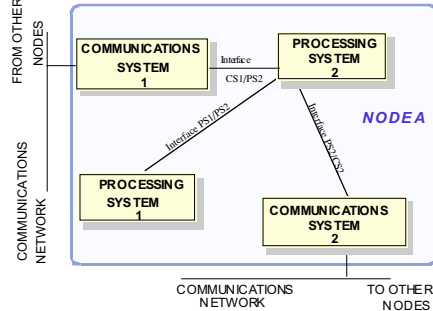


Internodal Perspective

System-to-System



Intranodal Perspective



Intrasystem Perspective

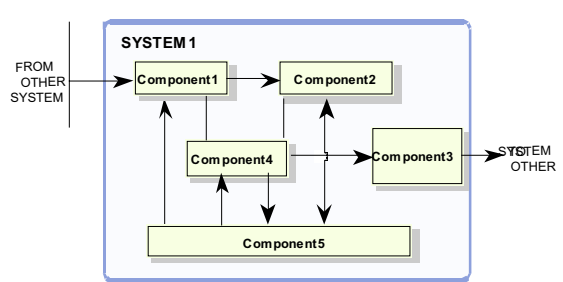


Figure 5. System Interface Description, in Four Levels of Detail

The System Interface Description links together the operational and systems architecture views by depicting the assignments of systems and their interfaces to the nodes and needlines described in the Operational Node Connectivity Description.

The System Interface Description identifies the interfaces between nodes, between systems, and between the components of a system, depending on the needs of a particular architecture.

There is also just one essential product in the technical view, the **Technical Architecture Profile** (figure 6). The Technical Architecture Profile references the technical standards that apply to the architecture and how they need to be, or have been, implemented.

Service Area	Service	Standard
Operating System	Kernel	FIPS Pub 151-1 (POSIX.1)
	Shell and Utilities	IEEE P1003.2
Software Engineering Services	Programming Languages	FIPS Pub 119 (Ada)
User Interface	Client Server Operations	FIPS Pub 158 (X-Window System)
	Object Definition and Management	DoD Human Computer Interface Style Guide
	Window Management	FIPS Pub 158 (X-Window System)
	Dialogue Support	Project Standard
Data Management	Data Management	FIPS Pub 127-2 (SQL)
Data Interchange	Data Interchange	FIPS Pub 152 (SGML)
	Electronic Data Interchange	FIPS Pub 161 (EDI)
Graphics	Graphics	FIPS Pub 153 (PHIGS)
• • •		

Figure 6. Example Technical Architecture Profile

In addition to the view-specific essential products described here, there are two more essential products that apply to all three views. These are the Overview and Summary and the Integrated Dictionary.

For each product, appendix A of the Framework contains a table presenting details of the product attributes or characteristics. Each product attribute represents a piece of information about a given architecture that should be captured in the product and stored in the Integrated Dictionary.

As the Framework is used and lessons-learned are compiled, the set of information elements needed to describe architectures will be refined.

Architecture Product Linkages – the Audit Trail That Relates Technology to Mission Operations

The three architecture views provide a basis for analyzing proposed investments in terms of their contributions to mission effectiveness. Because the Framework products are closely interrelated, this kind of trace-back can be readily accomplished.

In figure 7, each of the three architecture views (operational, systems, and technical) is represented by examples of the appropriate performance measures for that view, the information that must be captured to evaluate whether those measures can be or are being met, and the main Framework product or products used to capture that information.

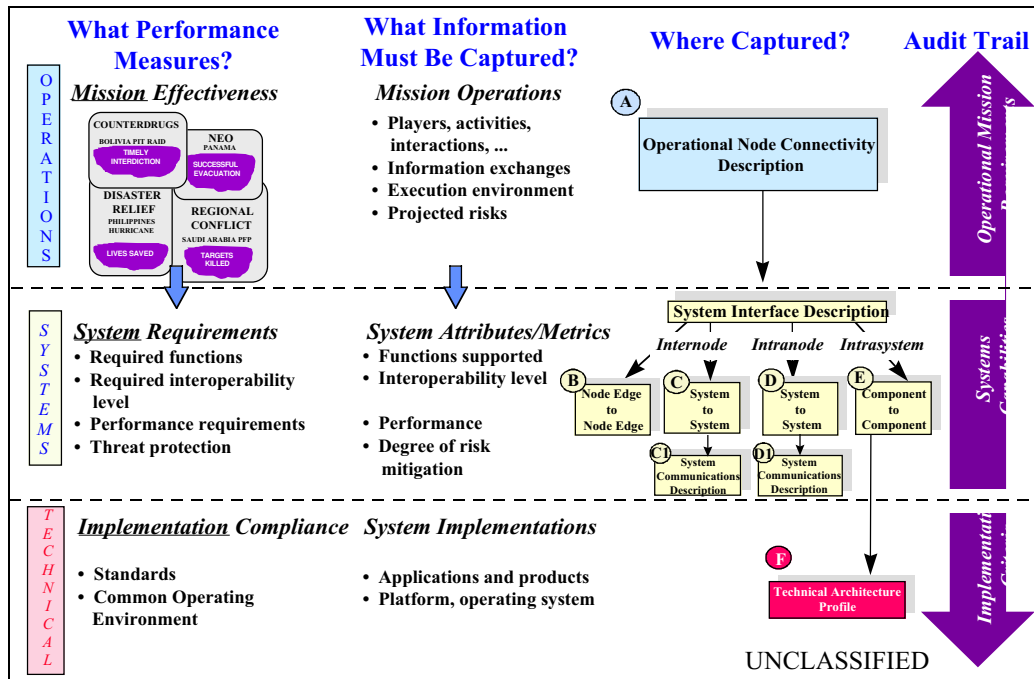


Figure 7. Audit Trail Linking Technology to Mission Operations

As the architecture description moves from the operational view to the systems view, information about the systems that satisfy the operational needs is overlaid on the operational information, and the mission effectiveness measures are translated into system performance measures. The prevailing technical architecture standards provide implementation criteria for the systems that satisfy the operational requirements.

The sequence of products, indicated by the circled letters above, provides a mechanism for relating the systems solutions (investments) back to their operational requirements (mission effectiveness). The Framework does not explicitly dictate the sequence in which products should be built, but by taking advantage of the inherent relationships among the products, one can tailor an appropriate sequence to suit the analysis at hand.

2.2.3 Common Building Block References

There are many efforts ongoing and evolving in DoD that focus on the common goals of interoperability, integration, and cost-effective investments. A number of reference models and information standards exist that serve as sources for guidelines and attributes that must be consulted while building architecture products. Each of these resources is defined and described in its own document. Table 1 lists some of these resource building blocks.

Applicable Architecture Views	Universal Reference Resource	General Nature
All Views	<i>C4ISR Core Architecture Data Model (CADM)</i>	Logical data model of information used to describe and build architectures
All Views	<i>Defense Data Dictionary System (DDDS)</i>	Repository of standard data definitions, formats, usage, and structures
All Views	<i>Levels of Information Systems Interoperability (LISI)</i>	Reference Model of interoperability levels and operational, systems, and technical architecture associations
Operational	<i>Universal Joint Task List (UJTL)</i>	Hierarchical listing of the tasks that can be performed by a Joint military force
Operational	<i>Joint Operational Architecture (JOA)</i>	(In development) -- High-level, evolving architecture depicting Joint and multi-national operational relationships
System Technical	<i>Technical Reference Model (TRM)</i>	Common conceptual framework and vocabulary encompassing a representation of the information system domain
System Technical	<i>DII Common Operating Environment (COE)</i>	Framework for systems development encompassing systems architecture standards, software reuse, sharable data, interoperability and automated integration
Technical	<i>Shared Data Environment (SHADE)</i>	Strategy and mechanism for data-sharing in the context of DII COE-compliant systems
Technical	<i>Joint Technical Architecture (JTA)</i>	IT standards and guidelines

UNCLASSIFIED

Table 1. Universal Reference Resources

2.2.4 Universal Guidance

Guidance in the Framework concerning the *process* of describing an architecture, i.e., what steps to perform and in what order, has intentionally been kept to a very high level to allow organizations to design their own processes tailored to their own needs.

The most critical aspect of the guidance is that the purpose for building the architecture description should be clearly understood and articulated up front. This purpose will influence the choice of what information to gather, what products to build, and what kinds of analysis to apply. Figure 8 illustrates this.

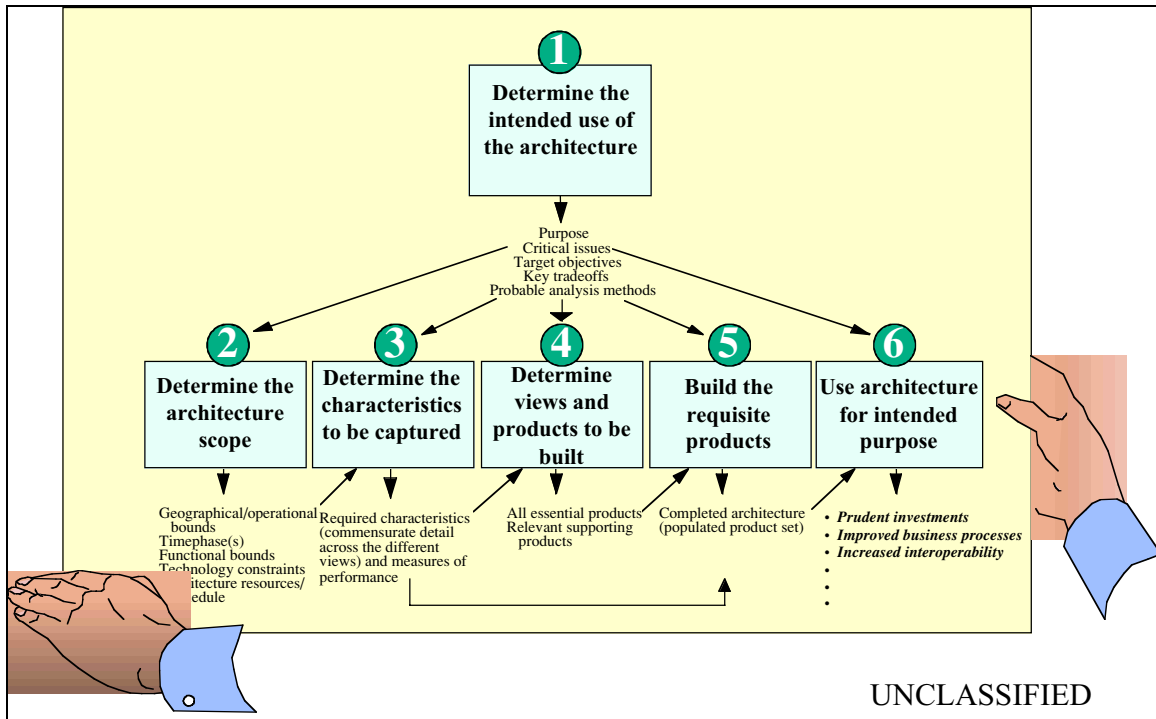


Figure 8. Framework Process Guidance

3.0 Adaptations Beyond DoD and Relationships to Other Frameworks

Although it was developed for C4ISR, the Framework has been used successfully in other DoD domains, such as electronic commerce, logistics, and health services, as well as in the Intelligence Community. Other Agencies and Departments of the Federal Government are also adopting the descriptive product types developed for the DoD Framework. Governments outside the United States have expressed interest in the DoD Framework, most notably Australia, Sweden, and Israel.

A far-reaching goal of architecture development is to have a common means for expressing architectures so that architectures can be understood and compared at many levels -- for example, within Agencies and Departments (e.g., Service-to-Service within DoD, or bureau-to-bureau within the Treasury Department) and across Agencies or Departments (e.g., between DoD and the Treasury Department). Some can even envision a world in which industry architectures can be readily compared with those of the Federal Government (e.g., to compare the way DoD performs payroll operations with the way IBM performs payroll operations). To this end, a number of organizations are developing architecture frameworks that tell their architects how to describe their architectures using common techniques and templates.

In addition to Government, industry is beginning to show interest in the C4ISR Architecture Framework as well. The Open Group is a multi-national organization, one of whose purposes is to develop an industry-wide common practice for architecture

description. The group has expressed a desire to incorporate some of the principles and techniques of the *C4ISR Architecture Framework* into their document, which is entitled *The Open Group Architecture Framework (TOGAF)* [Author's Correspondence, 2000].

The *C4ISR Architecture Framework* does not dictate a specific methodology to be used when describing architectures. An organization can devise its own method for developing the products (that is, which supporting products should be built, in what order, and to what level of detail), or it can use the *Framework* in conjunction with another, existing methodology or framework. This flexibility has made it easier to adapt DoD's Framework to other domains. The sections below describe the relationships between DoD's Framework and some other frameworks, and how other communities are using the architecture products prescribed in DoD's Framework.

3.1 Relationship to the Zachman Framework

The Zachman Framework is a way of thinking about an enterprise in an organized way so that it can be described and analyzed. The columns represent various aspects of the enterprise that can be addressed, and the rows represent various viewpoints from which those aspects can be described [Zachman, 1987]. Thus, each cell, formed by the intersection of a column and a row, represents an aspect of the enterprise modeled from a particular point of view. The architect selects and models the cells that are appropriate to the purpose of the analysis.

As shown by the color coding in figure 9, the views and individual products of the *C4ISR Architecture Framework, Version 2.0* map to the cells of the Zachman Framework [Sowell, 1999]. (The figure maps only the most frequently-used DoD products, not all of them.)

Blue cells indicate that the *C4ISR Architecture Framework* contains operational view products that map to the cells; orange cells indicate that the *C4ISR Framework* contains systems products that map to the cells; and blue/orange cells indicate that the *C4ISR Framework* contains both operational and systems products that map to the cells. (Note that there are no red cells; this reflects the fact that no technical view products map to the Zachman Framework. This is because the Zachman Framework does not call for the explicit modeling of the applicable rules and standards themselves, but assumes standards to apply within multiple cells.)

Ovals have been overlaid onto the color-coded cells. These ovals represent individual products from the *C4ISR Architecture Framework* that correspond to the Zachman cell or cells onto which the oval is overlaid. Operational products are represented by blue ovals, and systems products by yellow or orange ovals.

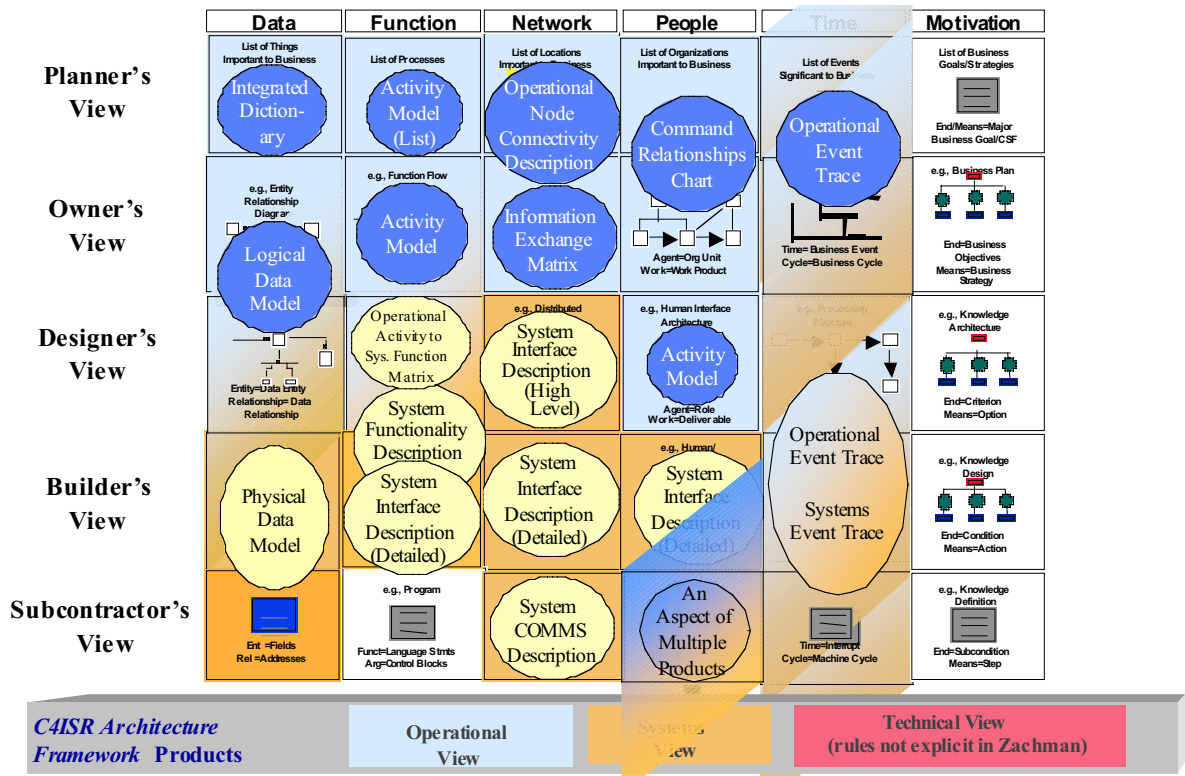


Figure 9: Selected DoD Framework Products Mapped to Zachman Framework Cells

Note that in some instances a cell is blue and orange, indicating that the *C4ISR Architecture Framework* contains both operational and systems products that correspond to the cell, but only a blue oval is shown in the cell. This is because not all the *C4ISR Architecture Framework* products are represented, only some of those that have been most frequently used in DoD architectures to date. The Function/Designer cell is blue and orange because the Operational Activity to Systems Function Matrix, while shown in the *C4ISR Architecture Framework* as a systems view product, is actually a pivot point between the operational and systems views.

Through this product-to-cell mapping, the *C4ISR Architecture Framework* can provide templates and guidelines for modeling the enterprise features that correspond to the Zachman cells.

3.2 Using the DoD Framework Product Types with the Federal Enterprise Architecture Framework

The Federal CIO Council has developed the *Federal Enterprise Architecture Framework (FEAF) Version 1.1*, which provides guidance in how to describe architectures for multi-organizational functional segments of the Federal Government [Federal CIO Council, 1999]. As shown in figure 10, the FEAF partitions a given architecture into a Business architecture and a Design architecture, with the Design architecture further partitioned into Data, Applications, and Technology aspects.

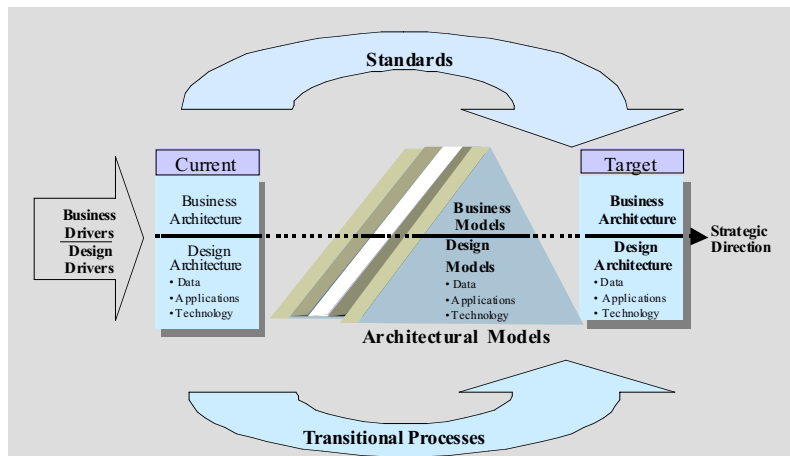


Figure 10: Structure of the Federal Enterprise Architecture Framework Components

The FEAF guidance is built on the foundation of a modified Zachman Framework, with the Spewak Enterprise Architecture Planning overlaid onto the first two rows. The first two rows are considered essential for all architectures built in accordance with the FEAF. Very high-level text descriptions are provided of the models that should be built to fulfill the cells of the modified-Zachman matrix.

3.2.1 Comparison of the Federal Framework and the DoD Framework

Figure 11 illustrates the following correspondence between the FEAF components and the DoD Framework Views: the Business architecture roughly corresponds to the DoD's Operational View, the Design architecture roughly corresponds to the DoD's Systems View, and the FEAF's Standards roughly correspond to DoD's Technical View. (Data is distributed across the Operational and Systems Views in the DoD Framework.)

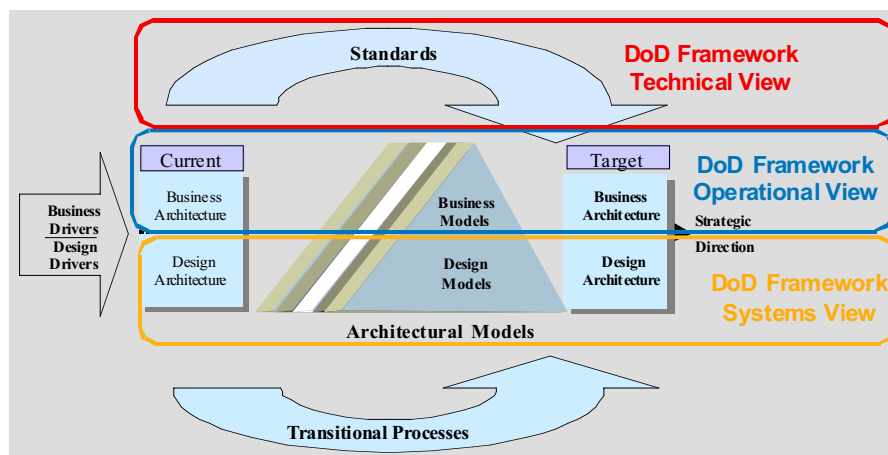


Figure 11: DoD Framework Views Mapped to the Federal Framework Components

As stated above, the FEF guidance is built on the foundation of the Zachman Framework, with the Spewak Enterprise Architecture Planning overlaid onto the first two rows. Because, as shown earlier, the DoD Framework products can be used to fill out the cells of the Zachman Framework, the DoD products can also be used to fill out the cells of the FEF. Figure 12 illustrates the mapping of selected DoD Framework products to the corresponding cells of the FEF. Note that the FEF has made some modifications and annotations to the Zachman Framework rows and column names.

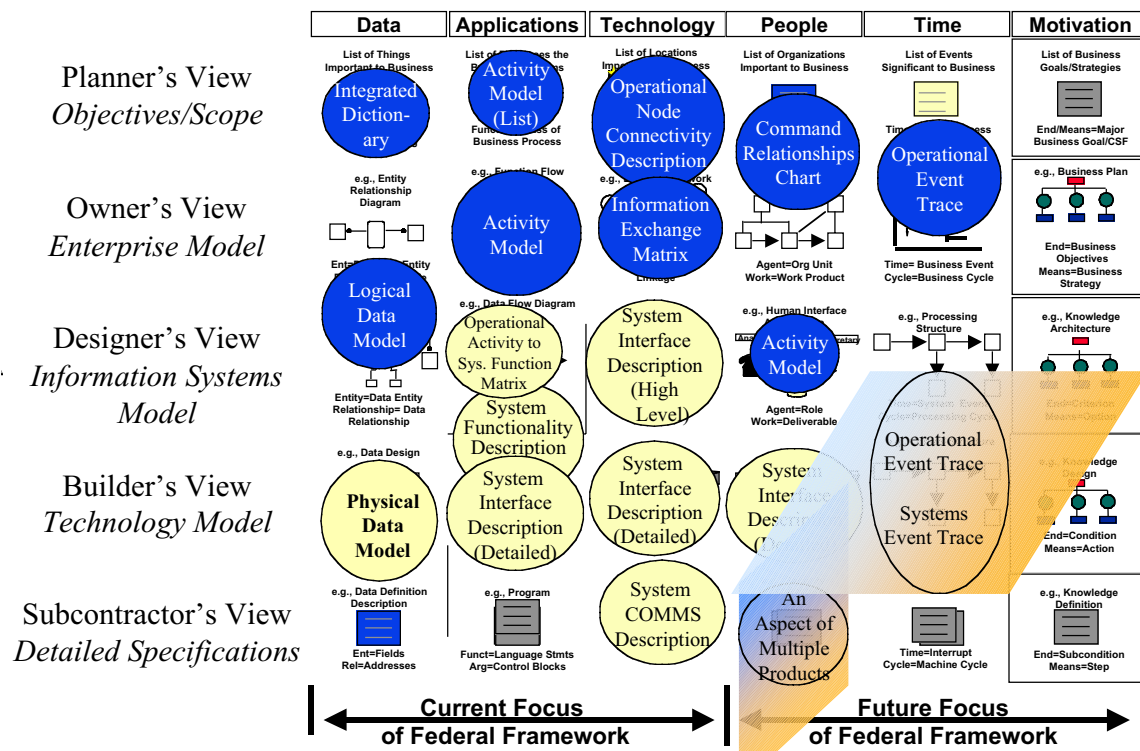


Figure 12: Selected DoD Product Types Mapped to the Federal Framework's Zachman-Based Cells

3.2.2 Using the DoD Products in the FEF: Federal Framework Pilot Architectures

The Federal CIO Council seeks to develop, maintain, and facilitate the implementation of the top-level enterprise architecture for the Federal enterprise. This architecture will serve as a reference point to facilitate the efficient and effective coordination of common business processes, information flows, systems, and investments among Federal agencies.

The approach taken to develop this Federal enterprise architecture is to develop architectures for selected high-priority cross-agency business lines, or "Federal Segments," which will collectively constitute the enterprise architecture.

With technical leadership provided by MITRE, a pilot effort is being conducted in which architecture descriptions will be constructed for one of the Federal Segments, to test the utility of the FEF guidance. The candidate functional segment as of this writing is Grants.

There was concern within the Federal Agencies Information Architectures Working Group (FAIAWG) that the Zachman Framework did not provide enough detailed direction to enable a useful architecture analysis. At this point, the FAIAWG turned to the *C4ISR Architecture Framework* products for this additional architecture information. Representatives of the FAIAWG worked with MITRE to examine the DoD products; they determined that the products were usable for the Federal Pilot with almost no modifications. Accordingly, four of the *C4ISR Architecture Framework*'s essential products and one supporting product will be used to populate the appropriate cells of the modified Zachman Framework.

The pilot effort will produce, in accordance with the *Federal Enterprise Architecture Framework* (as amended by the *DoD C4ISR Architecture Framework* products), a narrow-scope architecture pilot segment that can be used to gather lessons-learned for further development or improvement of the *Federal Enterprise Architecture Framework*. This effort will also support the activities of the Federal CIO Council's Emerging Information Technology and Interoperability Committee and contribute to the Committee's near term vision, which is *increased interoperability of Federal business processes to achieve a cost-effective, value-added contribution to the efficiency of the Federal enterprise*.

Figure 13 illustrates the products selected from DoD's Framework that will be used as templates for populating the Federal Framework cells selected for the Pilot [Sowell, 1999]. Although, as shown previously, many more DoD products map to the FEAF cells, only a few products were selected for a thin-thread example architecture for the Pilot.

		Data Architecture (entities = what)	Applications Architecture (activities = how)	Technology Architecture (locations = where)
<p>Blue Text: C4ISR Framework Products</p> <p>Shading = Products to be built in Pilot (No separate Activity Model; activities shown in Node Connectivity Description</p> <p>NOTE: C4ISR Architecture Framework's "All Views" products are needed by all cells!</p>	Perspectives	List of Business Objects	List of Business Processes	List of Business Locations
	Planner's View Objectives/ Scope		Activity Model (Hierarchy of activities)	Operational Node Connectivity - • Major nodes only • Needlines not annotated
	Owners View Enterprise Model	Semantic Model Operational Information Exchange Matrix also contributes)	Business Process Model	Business Logistics System Op'n'l Node Connectivity • All nodes • Needlines annotated Operational Information Exchange Matrix
	Designer's View Information Systems Model	Logical Data Model	Application Architecture	System Geographic Deployment Sys. Interface Description (Internodal, System-to System) Technical Architecture Profile
	Builders View Technology Model	Physical Data Model	System Design	Technology Architecture
	Subcontractors View Detailed Specifications	Data Definition "Library or Encyclopedia"	Programs "Supporting S/W Components"	Network Architecture

Figure 13. DoD Framework Products Mapped to the Federal Pilot Architecture Models

Note that the Technical Architecture Profile does not actually map to the FEAF cells, because “Standards” are not explicit in the FEAF’s modified Zachman Framework. It is included here for completeness of the Pilot.

3.3 Using the DoD Framework Products with the Treasury Department’s Enterprise Architecture Framework

On 3 January 1997 the Treasury Department published its *Treasury Enterprise Architecture Framework (TISAF)* [Department of the Treasury, 1997]. At this writing, the TISAF document is evolving to become the Treasury Enterprise Architecture Framework (TEAF). Some of the goals for this evolution are to make the document more user-friendly, to make it more explicit in its direction (more prescriptive rather than just descriptive), and to make it consistent with the FEAF. To further all of these goals, the TEAF uses the same DoD Framework products that are being used in the FEAF Pilot, and makes those products mandatory.

The TEAF is based on an adaptation of the Zachman Framework, using essentially the same rows (perspectives) as the Zachman Framework, collapsing the bottom two rows into one, and modifying the columns into four “Views” as shown in figure 14.

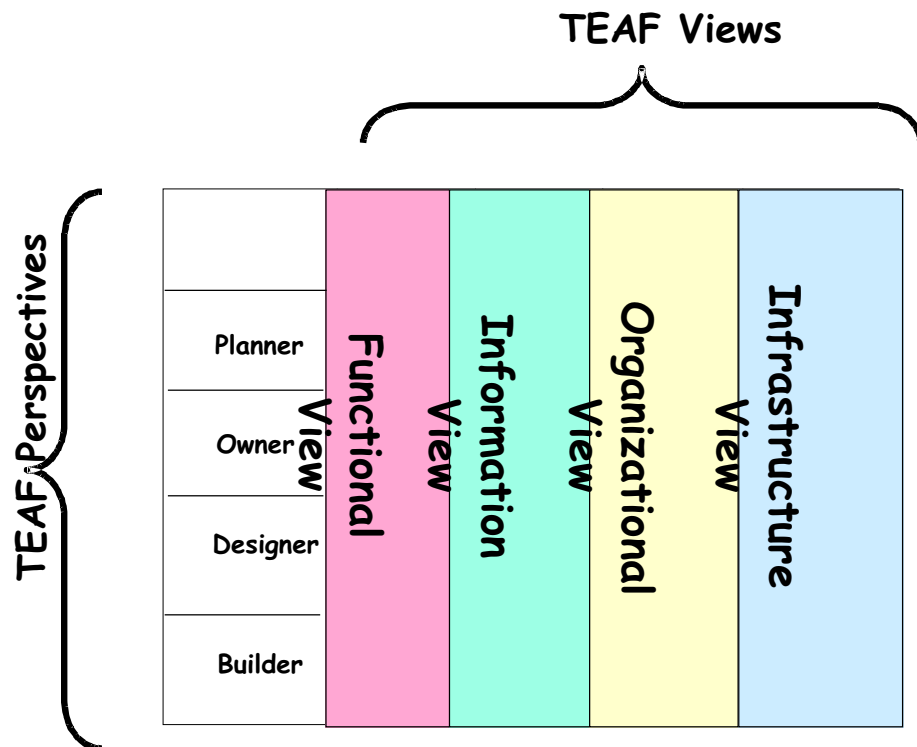


Figure 14. The Views and Perspectives of the Treasury Enterprise Architecture Framework

The current draft TEAF adopts the distinction between essential and supporting products that is used in the DoD Framework (the TEAF calls the products “work products”). The TEAF has adopted the DoD Essential product set as a starting point, to which Treasury-specific products may be added later. In addition, the TEAF lists many of the DoD’s supporting products, as well as other products derived from IRS work and other sources.

Figure 15 illustrates the selected DoD products mapped to the cells of the Treasury Framework [Department of the Treasury, 2000]. Note that this mapping is representative and for illustration only; the TEAF document was in draft as of this writing and the exact mapping may therefore be subject to change.

The DoD Framework allows for constructing products at multiple levels of detail, as needed; the TEAF has accounted for this by giving each level of detail a distinct product name. For example, the DoD’s Operational Node Connectivity Description is represented three times in the TEAF matrix: once as Node Connectivity Description (Conceptual), once as Node Connectivity Description (Logical), and once as Node Connectivity Description (Physical).

	Functional View	Information View	Organizational View	Infrastructure View
Planner Perspective	Mission & Vision Statements	Information Dictionary	Organization Chart	Technical Reference Model
				Standards Profile
Owner Perspective	Activity Model	Information Exchange Matrix (Conceptual)	Node Connectivity Description (Conceptual)	Information Assurance Risk Assessment
	Information Assurance Trust Model			System Interface Description Level 1
Designer Perspective	Business Process/ System Function Matrix	Information Exchange Matrix (Logical)	Node Connectivity Description (Logical)	System Interface Description Levels 2 & 3
	Event Trace Diagrams	Data CRUD Matrices		
	State Charts	Logical Data Models		
Builder Perspective	System Functionality Description	Information Exchange Matrix (Physical)	Node Connectivity Description (Physical)	System Interface Description Level 4
		Physical Data Models		System Performance Parameters Matrix
	Work Products from DoD	Other Work Products		

Figure 15. Representative Mapping of the DoD Framework’s Products to the Cells of the TEAF

3.4 U.S. Customs Service Use of the DoD Framework Products

The U.S. Customs Service is using several of the DoD Framework's product types in developing an architecture description of its Automated Commercial System. This effort is described in another CCRTS conference paper [Thomas *et al.*, 2000].

4.0 Current Plans for Evolution of the Framework

The Office of the Assistant Secretary of Defense (C3I) is undertaking an effort to evolve the *C4ISR Architecture Framework* beyond its current Version 2.0. The plan is to develop a Version 2.1, which makes some improvements and refinements to Version 2.0, and to develop the broad outlines for an evolution to a Version 3.0. The main objectives in developing Version 2.1 are to

1. Widen the scope of the document from C4ISR to more explicitly address DoD-wide application.
2. Improve on version 2.0 to leverage lessons learned and emerging tools and methodologies. These improvements will include simplification, clarification, and a discussion of the implications of object-oriented techniques and tools on the Framework.

It is the intent that any architectures that are compliant with version 2.0 will also be compliant with version 2.1 without modifications. Any major changes such as making the Activity Model a required product will be "grandfathered" to make exceptions for architecture efforts already underway.

Overall reaction to the *C4ISR Architecture Framework, Version 2.0* guidance was quite positive. Most organizations supported the requirement for such guidance, and the consensus was that, if executed properly, the guidance can provide a valuable vehicle for streamlining the architecture process as well as related processes.

Several suggestions were submitted with respect to Framework enhancements. Some of the more significant suggestions are described in table 1. As of this writing, these are some of the changes that are expected to appear in the draft of Version 2.1. The final product may differ, pending working group recommendations.

Table 1. Some Version 2.1 Changes Planned Resulting from Community Feedback

Community Feedback on Version 2.0	Resulting Changes Planned for Version 2.1
<ul style="list-style-type: none"> • Remove C4ISR from the title. 	<ul style="list-style-type: none"> • Plan to do
<ul style="list-style-type: none"> • Streamline, clarify definitions. 	<ul style="list-style-type: none"> • Plan to do
<ul style="list-style-type: none"> • Please make the Activity Model “Essential.” 	<ul style="list-style-type: none"> • Plan to do, and redesignate as “OV-2” • NOTE: Operational Node Connectivity Description and Operational Information Exchange Matrix must be renumbered because of the insertion of the Activity Model into the Essential product set.
<ul style="list-style-type: none"> • Provide more process guidance, i.e., how to use the Framework. 	<ul style="list-style-type: none"> • Some organization-specific tailorings of the Framework generic process will be referenced and excerpted.
<ul style="list-style-type: none"> • Provide some guidance for those of us using object-oriented techniques. 	<ul style="list-style-type: none"> • Mappings to object-oriented notations are planned. • Example architecture showing object-oriented versions of products is planned.
<ul style="list-style-type: none"> • Give some guidance on how to move from “AS-IS” to “TO-BE” architectures. 	<ul style="list-style-type: none"> • A Capability Maturity Profile product is planned (AV-3).
<ul style="list-style-type: none"> • Address the “value of architectures” question. 	<ul style="list-style-type: none"> • A new section is planned to begin addressing the value of architectures and architecture metrics.

Table 1. Some Version 2.1 Changes Planned Resulting from Community Feedback – concluded

Community Feedback on Version 2.0	Resulting Changes Planned for Version 2.1
<ul style="list-style-type: none"> • The examples are confusing and too numerous. 	<ul style="list-style-type: none"> • Real-world examples will be omitted in favor of generic templates.
<ul style="list-style-type: none"> • The distinction between “essential” and “supporting” products is confusing. Please eliminate the distinction or change the names. 	<ul style="list-style-type: none"> • Will use clarified terms to emphasize the intent – that some products are needed for Joint analysis, integration, and reuse and are therefore mandatory even if they are not directly needed for the individual architecture’s specific purpose.
<ul style="list-style-type: none"> • We need more structure in the Operational Information Exchange Matrix, less freedom. 	<ul style="list-style-type: none"> • The matrix will be further detailed.
<ul style="list-style-type: none"> • Clarify the connections between the operational view and the systems view. 	<ul style="list-style-type: none"> • The Operational Information Exchange Matrix and the Systems Data Exchange Matrix (formerly called the “Systems Information Exchange Matrix”) will be refined and more closely related to facilitate the transformation of operational requirements into system requirements. • Product Relationship Charts will illustrate other relationships among the views. • A “value of architectures” section is planned to address an audit trail through the views.
<ul style="list-style-type: none"> • The Framework is too big. 	<ul style="list-style-type: none"> • Document will be restructured, with key information in a relatively small body and supplementary information in appendices.
<ul style="list-style-type: none"> • Please provide information on automated tools (also some advice not to get into detailed tool discussions). 	<ul style="list-style-type: none"> • Some general information about tool will be provided, but no particular commercial tools will be endorsed.
<ul style="list-style-type: none"> • Please clarify that the Operational Node Connectivity Description (OV-2) can be used to portray “functional” rather than physical nodes. 	<ul style="list-style-type: none"> • Existing words to that effect will be supplemented with a template showing functional nodes.
<ul style="list-style-type: none"> • We need more guidance on the degree of freedom allowed in the graphical appearance of the product; as it is, the Framework seems to imply that only Structured Analysis representations can be used. 	<ul style="list-style-type: none"> • Activity Model write-up will be revised to be less IDEF0-specific. • A mapping of products to object-oriented notations is planned. • Example architecture is planned that illustrates object-oriented product alternatives.

References

[Architecture Coordination Council, 1998] Memorandum, 23 February 1998, Subject: Strategic Direction for a DoD Architecture Framework.

[Author's Correspondence, 2000] Communication with TOGAF committee members.

[C4ISR Architecture Working Group, 1997] *C4ISR Architecture Framework Version 2.0*. Office of the Assistant Secretary of Defense for Command, Control, Communications and Intelligence, Washington D.C., November 1997.

[Department of the Treasury, 1997] *Treasury Information System Architecture Framework, version 1.0*. Office of the Deputy Assistant Secretary for Information Systems and Chief Information Officer, Department of the Treasury, Washington, D.C., 3 January 1997.

[Department of the Treasury, 2000] *Treasury Enterprise Architecture Framework, Version 1*. Department of the Treasury Chief Information Officers' Council, Washington, D.C., July 2000 (in draft as of this writing).

[Federal CIO Council, 1999] *Federal Enterprise Architecture Framework, Version 1.1*. Department of Commerce, Technology Administration, National Technical Information Service, Springfield, VA., 1999.

[Sowell, 1999] P. Kathie Sowell. *The C4ISR Architecture Framework and the Federal Enterprise Architecture Framework*. The MITRE Corporation, McLean, Virginia. 1999.

[Sowell, 1999] P. Kathie Sowell. *Consolidated Mapping of C4ISR Framework Products to Federal Framework Models*. The MITRE Corporation, McLean, Virginia, 1999.

[Thomas et al., 2000] Rob C. Thomas II, Raymond A. Beamer, P. Kathie Sowell. *Civilian Application of the DoD C4ISR Architecture Framework: A Treasury Department Case Study*, 5th International Command and Control Research and Technology Symposium, 2000.

[Zachman, 1987] John A. Zachman. *A Framework for Information Systems Architecture*. IBM Systems Journal, 26(3): 276-291, 1987.