

DoD Experience with the C4ISR Architecture Framework

William G. Wood
Sholom Cohen

September 2003

Architecture Tradeoff Analysis Initiative

Unlimited distribution subject to the copyright.

Technical Note
CMU/SEI-2003-TN-027

The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

Copyright 2003 by Carnegie Mellon University.

NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this report is not intended in any way to infringe on the rights of the trademark holder.

Internal use. Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

External use. Requests for permission to reproduce this document or prepare derivative works of this document for external and commercial use should be addressed to the SEI Licensing Agent.

This work was created in the performance of Federal Government Contract Number F19628-00-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (<http://www.sei.cmu.edu/publications/pubweb.html>).

Contents

Abstract	vii
1 Introduction	1
2 Context	2
2.1 Acquisition Strategy	2
2.2 Business Drivers.....	2
2.3 C4ISR/AF Views and Products.....	3
3 Observations	5
3.1 Requirements	5
3.1.1 Initial Requirements for Source Selection.....	5
3.1.2 Requirements for Incremental Builds.....	5
3.2 Organizational Issues	6
3.2.1 Government	6
3.2.2 Contractors	7
3.2.3 Government/Contractor Team	8
3.3 Acquisition Impacts.....	8
3.4 Reviewing the Products	9
4 Strengths	11
5 Challenges and Suggestions	12
5.1 Process.....	12
5.2 Evolutionary Development and Deployment.....	13
5.3 Technical View	14
5.4 Software Issues	14
5.5 Crosscutting Quality Attributes.....	15
5.6 Training and Testing.....	16
5.7 Tool Support.....	16
6 Conclusions	19
Appendix A: Summary of the Architectural Products	20
Appendix B: Acronym List	23

References..... 25

List of Figures

Figure 1: Architecture Framework Views [DoDAF 03].....	4
--	---

List of Tables

Table 1: Architectural Products [C4ISRAF 97]	20
--	----

Abstract

The Department of Defense (DoD) is mandating the use of the Command, Control, Communications, Computer, Intelligence, Surveillance, and Reconnaissance Architecture Framework (C4ISRAF) for large-scale software-intensive systems. The authors conducted eight interviews with personnel who have used the C4ISRAF in acquisition projects. The intent of the interviews was to find the strengths and weaknesses of the C4ISRAF, so that this information could be communicated to future users of the framework. This technical note discusses the context for using the C4ISRAF, the observations made during the interviews about its use, and the strengths and challenges of using it. Suggestions for overcoming these challenges also are included.

1 Introduction

The Command, Control, Communications, Computer, Intelligence, Surveillance, and Reconnaissance Architecture Framework (C4ISRAF), which is mandated by the Department of Defense (DoD) for large-scale software-intensive systems, describes how the architecture for a system or system of systems (SoS) should be documented [C4ISRAF 97]. It breaks the documentation into three views: the operational view (OV) consists of 9 products; the system view (SV) consists of 13 products; and the technical view (TV) consists of 2 products. An additional all view (AV) consists of two products, one of which is a graphic showing the weapons platforms involved, and the other is a data dictionary containing the data items defined in the OV, SV, and TV products. The C4ISRAF views and products are discussed in more detail in Section 2.3.

We conducted eight interviews with architects, designers, architecture documentation reviewers, and program managers associated with a number of acquisition category I (ACAT I)¹ programs to develop large-scale software-intensive SoSs. All of the program teams are using the C4ISRAF to build an architecture, consisting of many OV, SV, and TV products, that serves as the basis for acquisition planning, development, and deployment of the SoSs. One of the program teams is still developing a request for proposal (RFP), and others have completed source selection and are in the first phase of the development cycle. Some of the programs will evolve from legacy systems already in operation, while one is a new weapons platform with no legacy components (though it did have legacy interfaces). We conducted some of the interviews one-on-one, while others involved a number of people. Our intent was to determine the experiences associated with using the C4ISRAF as reported by practitioners on large-scale software-intensive programs.

In this report, we establish the context for using the C4ISRAF on ACAT I programs, discussing such issues as the business drivers for these systems and the uses for the architecture after it has been documented. In Section 3 we describe some observations about the use of the C4ISRAF that are associated indirectly with the framework, and in Section 4 we cover the strengths of the C4ISRAF. Section 5 includes challenges to using the framework, along with suggestions for overcoming them.

The C4ISRAF is currently being upgraded and renamed as the DoD Architecture Framework (DoDAF) [DoDAF 03]. The DoDAF (and its views and associated products) does not differ substantially from the C4ISRAF.

¹ This acquisition terminology is defined in the DoD Regulation 5000.2-R, which may be found at <http://www.aitso.com/simval/Documents/5000.2-R/DoD5000.2-R_part1.htm>.

2 Context

2.1 Acquisition Strategy

Most of the systems under consideration were built using a rolling-down-select acquisition strategy. In this approach the government releases an initial RFP, and many offerors respond to it. A few (typically two or three) of the offerors receive a contract and work independently for a time period (the fly-off) of a few months to a few years, developing a proposed architecture for the system as a number of contract deliverables (CDRLs). Each CDRL consists of a number of C4ISRAF products and the associated explanatory documentation. During the fly-off, a government analysis team (GVAT) works with each contracting team to answer its technical questions and perform technical reviews on the products developed. At the end of the fly-off, the government issues a call for improvement (CFI), and the contractors enter final proposal bids, including C4ISRAF products. These bids are evaluated by a government-source-selection technical analysis team (GVSSAT) as part of the source selection process used to select a single contractor.

2.2 Business Drivers

Large-scale programs implementing stand-alone stovepiped weapons systems have been somewhat successful over the last 10 years,² albeit often with budget overruns, schedule slips, and reduced functionality. Nonetheless, significant problems have occurred with integrating these stovepiped systems into network-centric operations when system users within one stovepipe need access to information that is within his or her authority and responsibility but resides in another stovepipe. While the business drivers listed below could apply to many classes of systems, they are specifically applicable to network-centric ACAT I programs:

- Achieve interoperability between stovepiped programs, providing the necessary knowledge to decision makers at all echelons.
- Reduce life-cycle costs in a number of ways, including the following:
 - replacing obsolete systems with systems based on industry standards and commercial off-the-shelf (COTS) and government off-the-shelf (GOTS) products, thus reducing maintenance costs
 - reducing warfighter, logistics, and support staffing

² Crosstalk's annual survey of top quality software projects documents case studies of organizations using well-defined and proven processes and practices that successfully deliver software products to the U.S. government (<<http://www.stsc.hill.af.mil/top5projects/index.html>>).

- Introduce operational concepts that are aligned with the new system capabilities and new technology. Add the capability that “everyone can know everything” within the bounds of the person’s authorization.

2.3 C4ISRAF Views and Products

As noted in the introduction, the C4ISRAF offers a collection of products to describe the architecture as three views.³ The products are listed in Appendix A.

1. The OV describes the activities, operational elements, and information flows required for a number of missions supported by the SoS. This information is described in terms of the mission’s nodes, the operational activities conducted within these nodes, and the needlines connecting the internodal activities. A node often means a mobile weapons platform (e.g., a ship, tank, or aircraft) or a fixed ground platform (e.g., a command and control center, or a communication hub). An operational activity is usually associated with a warfighter and consists of a combination of manual and automated actions taken by the warfighter, such as tactical planning and engagement of an enemy platform. A needline is a description of the information that moves from one activity to another, such as a tactical plan, by some automated or manual (“sneaker net”) route.
2. The SV describes the system-level structures that support the OVs. The SV includes the systems required at each node, the communication media connecting the nodes, and the functions contained within each system. In addition this view contains tables describing how each needline listed in the OV products is implemented by the communication media listed in the SV products and tables. These tables show how each activity described in the OV products maps onto the functions described in the SV products.
3. The TV describes the minimal set of rules governing the arrangements, interactions, and dependencies of system components. This view includes standards used in the SoS, and the major COTS and GOTS components used.

A fourth set of products, the AV, includes two products that provide an overview perspective of the entire system. Figure 1 illustrates the relationships among the three primary views of the architecture framework.

³ In the literature on documenting software architecture (SA), the word *view* is used differently from how it is used in the C4ISRAF. In the SA context, view is usually a single documentation format; for example, a logical view or a process view.

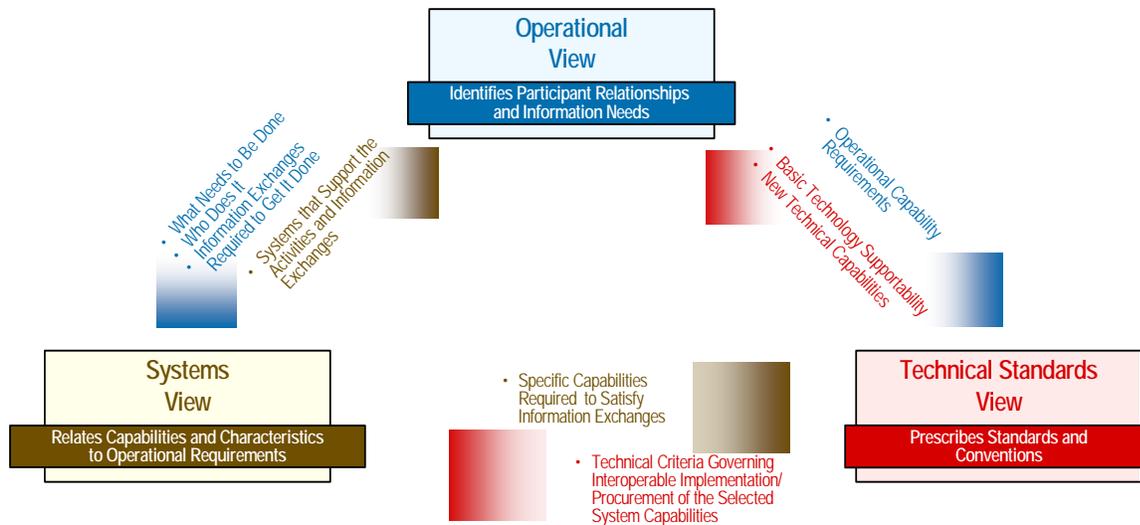


Figure 1: Architecture Framework Views [DoDAF 03]⁴

Within each of these views is a set of products that provide increasing levels of detail concerning the operations and system structures. Most program teams tailor the collection of products they will produce at each major acquisition stage. They also select tools to produce and maintain the products.

Program teams use the products to support various aspects of development. For example, most of them use OV products as a source for requirements. They use the SV products as a blueprint for planning, budgeting, acquiring, and building an SoS. And they use the TV products to either suggest or enforce standards and COTS or GOTS products used in the system implementation. The products include sufficient detail to accommodate the aspects of development.

The C4ISR/AF products serve as a communication vehicle among stakeholders and provide sufficient detail for individual classes of stakeholders. The C4ISR/AF products help the program teams identify and manage the significant risks, analyze mission effectiveness, build prototypes, and integrate work efforts of different organizations.

⁴ The term *Technical Standards View* is the new DoDAF standard for the existing *Technical View* in the C4ISR/AF, on which this report is based.

3 Observations

During the interviews, we made many important observations about using the C4ISR/AF that do not relate directly to the products. These observations are gathered into this section.

3.1 Requirements

In a rolling-down-select acquisition, an initial RFP is issued that includes the end-vision requirements for the program over its complete life cycle. If the development is to be deployed in a sequence of incremental deliveries, a more detailed requirements definition will also be needed for each increment.

3.1.1 Initial Requirements for Source Selection

The program teams interviewed had performed the source selection in different ways:

- In one case, the program team issued an RFP without developing any OV, SV, or TV products. The RFP stated the requirements generally and defined which C4ISR/AF products should be developed as part of the fly-off effort. These views were then included as part of the final rolling-down-select proposal.
- In other cases, the program office developed some exploratory OV, SV, and TV products to investigate the problem space before developing an RFP, and then used the results of that investigation to develop the technical requirements in the RFP. In these cases, the exploratory products were made available to the offerors during the initial proposal development, though the offerors were not required to use these products.

3.1.2 Requirements for Incremental Builds

Planning for many of these large, complex SoSs includes incremental deployment to the field, which is discussed in more detail in Section 5.2. For these types of systems, the C4ISR/AF products built initially document an end-vision architecture as much as 20 years in the future and do not contain sufficient detail to serve as a blueprint for building or upgrading the individual systems. The actual system is usually built in update blocks about 18 months apart. This process uses multiple extensions to a single contract, with each extension including the part of the system to be delivered as the next increment. In this case, the requirements for each block must be developed in sufficient detail to support evaluation and provide sufficient guidance for the subsequent blocks. The procedure outlined below describes the way the interviewed program teams are handling multiple block delivery:

1. Build a set of OV products representing the next system increment, based on the

- end-vision architecture
- baseline system from the previous increment (or the legacy at stage 1)
- changes to be made during this increment

If significant technical advances have been made while the last increment was in progress, update the TV also.

2. Derive the technical requirements document (TRD) for this increment based on the OV products. The TRD is much smaller than the OV products and documents system requirements better.
3. Have the contractor build the SV products based on the OV products. Then proceed to build this increment of the system.

The above process can be accomplished in two ways:

1. The government can build and control the OV and TV products at each stage, and the contractor then builds the SV products for that stage. In this case, the government must either have the organic skills to develop the OV and TV products, or contract this development separately from the SV products.
2. The contractor can develop, with government oversight, the OV, TV, and SV products.

3.2 Organizational Issues

Government mandates for the use of a framework such as C4ISR/AF often raise organizational issues within the government and with the contractors. Discussions about many such issues arose during the interviews, and we documented them in this section. The government and contractor personnel commented on government organization, contractor organization, and after-award team organization. *While many of these issues would have come up even if the C4ISR/AF had not been mandated, we documented them here anyway.*

3.2.1 Government

The government is often asking for new operational, technical, and system approaches to resolving large-scale problems that may require changes in force structure, the introduction of new weapons and sensor platforms, and updates to legacy platforms. However, this method can lead to problems such as

- The government reviewers are themselves part of the military organization stovepipes and are reluctant to adopt changes that might remove funding from their own stovepipe or diminish its role.
- If the new SoS uses legacy components and their associated force structure and concept of operations (CONOPS) as a baseline from which to evolve, a satisfactory inventory of the baseline must be developed. To do so, the government can build an “as-is” architecture before the contract is let. If no such architecture is built, it is difficult for all

contractors during a fly-off competition to capture a sufficiently robust understanding and baseline definition of the legacy components.

- The government reviewers are often reluctant to play “technology leapfrog.” If they are using a legacy system and think that current technology will provide a great improvement over it, they are reluctant to select an even more advanced technology that they might not understand.
- The C4ISRAF products can conveniently be used as mechanisms to justify the continuation of system acquisition at all stages. Such use can lead high-level government managers to believe that these products function mainly as a marketing tool and thus use them to convince even higher-level government managers that the program is being executed successfully. One must always remember that the real significance of the products is to serve as a blueprint for planning, scheduling, and building the system.

3.2.2 Contractors

The contractors must organize teams to build the products required by the government and take care to avoid the pitfalls listed below:

- The contractors should avoid organizing their teams around the C4ISRAF product structure because some products that seem quite different are actually closely related. The best example of this relationship includes the OV products, the CONOPS, the operational requirements document (ORD), and results of a simulation study of mission effectiveness. If teams organized around these products work in different parts of the organization without close supervision and coordination, they tend to document different operations, and a considerable effort must then be expended to bring them back together. Of course only the OV products are part of the C4ISRAF, but in most ACAT I systems, the other documents and studies are also needed. An organizational goal may be to use the OVs to capture all information, and to tag information as “management support” if it is needed for programmatic and budgeting efforts (for inclusion in the CONOPS or ORD) or as “technical support” if it’s used to support SVs.
- If the contractor is proposing some innovation in operations, system design, or operational concepts, or a combination of all three, the contractor must supply a strong rationale. The contractors should keep in mind that the GVAT they are working with during the fly-off phase will differ from the GVSSAT; hence they should include sufficient rationale in the proposal to be understood by the GVSSAT. This rationale is especially important if the members of the GVAT are “early adopters” who encourage use of new technology and concepts, but the higher echelons involved at source selection are reluctant to accept such changes.
- One of the strengths of the C4ISRAF is that the requirements can be written at the operational level. That enables the contracting team to make design choices based on its understanding of technology, COTS, and non-developmental item (NDI) products, and offer operational changes to the acquiring agency based on this knowledge. However, the contractors are often not accustomed to working this way and may prefer that the

government “tell them what they want” rather than making choices on their own that could be deemed unreasonable and unacceptable by government managers.

- The C4ISRAF concentrates on the visionary product to be made operational in the future. However, the government reviewers of the contractors’ proposals are often more concerned with what will change in the near term (e.g., over the first five years). Hence the contractors must include a good representation of the initial spirals in the proposal, even though this information is not explicitly called for in the C4ISRAF.
- The contractor must remember that detailed technology presentations are likely to “turn off” higher-echelon government team members. Just because the contractor’s technical team is proud of some technology innovation does not mean that the higher echelons should be swamped with detailed presentations.
- The contractors sometimes have difficulty developing the OV4 command structure diagram since they may fail to grasp how much (or how little) detail is needed. It is probably best if the government develops that diagram.

3.2.3 Government/Contractor Team

After the contract is awarded, the government and the contractor form a team to proceed with the development in a non-adversarial fashion. The products built using the C4ISRAF usually provide an end-vision architecture, with some hints of the first increment in the system’s development and deployment, and often a process has been agreed upon for the development of each increment. However, sufficient time may not exist to “do the first increment right” due to contracting issues. The team therefore must learn important lessons from the first increment to make sure that the second increment correctly addresses the appropriate technical, process, and organizational risks.

3.3 Acquisition Impacts

In a rolling-down-select acquisition, a GVAT is usually involved with the contractors during the fly-off portion of the contract. This GVAT tracks the development of the C4ISRAF products, reads and comments on the CDRLs, and tracks the comments until closure. This group is very familiar with the approaches being taken by the different contracting teams. When the final source selection is made, the GVSSAT may differ from the GVAT.

- The source selection process must clearly define a hand-off process such that the knowledge and understanding of the GVAT is communicated directly to the GVSSAT; for example, by passing along a set of risks associated with each team. The GVSSAT can then use its own judgment as to whether these risks have been mitigated appropriately in the final proposal.
- A rolling down select can take a long time, during which mission changes can occur. In addition, various organizations that were only slightly involved with the acquisition can start to have a major impact on the C4ISRAF products after contract award. Hence the

first stage in the new contract is to reflect these changes in an updated version of the architectural products.

- A demonstration of capabilities is often made during source selection. If the contract was written correctly, and each contractor is required to demonstrate the same capabilities, this information can serve as a good discriminator in the fly-off phase of a rolling down select between multiple contractors. However, if the specifics of the demonstration are not included in the contract, the demonstration can be unsatisfactory and cannot be used as a discriminator. In addition, the contractors should remember that the demonstration may be given to the GVAT, but the GVSSAT, who makes the final decisions, may not be strongly affected by the demonstration.

3.4 Reviewing the Products

The C4ISRAF products built, which serve as a communication vehicle among stakeholders, are often called out in the CDRLs. The government stakeholders then review those CDRLs to approve both the architecture and the impacts of the contractors' tradeoff decisions. A number of issues are associated with reviewing the C4ISRAF products:

- The CDRL reviewers and the GVSSAT come from the existing military stovepipes; thus, their review is biased by the impact of changes on their organization. If the intent of the SoS is to reengineer the operations and systems between the existing military stovepipe organizations (e.g., by changing the force structure or the command chain), some reviewers may object to the change because of organizational bias rather than the effectiveness of the proposed change.
- The CDRL reviewers may be tied too closely to the current CONOPS and technology to effectively review products that are tied to a new vision.
- Some reviewers are interested mainly in targeted fragments of the system that align with their everyday responsibilities. However, the C4ISR architecture documentation tends to capture everything about the SoS and organizes it so that the reader is introduced to everything in an organized, top-down way. For example, one reviewer may be interested mainly in strategic and tactical plans, and how they interoperate, while another may be interested in Unmanned Aerial Vehicle (UAV) usage as sensor platforms. But often the reviewers have no easy way to access the documentation for their specific interests; they have to start by skimming through on a first pass, tagging those elements that are relevant to their interest, and then thoroughly inspecting the tagged elements.
- Since much of the documentation is captured online, the reviewers must be sufficiently familiar with the supporting tool set to perform their review work effectively. Readers of the documents, therefore, require some specialized training on the tool set (substantially less than the architects have received) and some continual exposure to prevent their expertise from getting stale.
- The architectural products are quite voluminous. The more products there are that need to be reviewed, the more work the reviewers must do—especially in a source selection,

where there is a limited set of reviewers for the documents from each bidder. The acquirers should not ask for more detail than they are willing to review.

- Some reviewers take the documents and consider them strictly as descriptions of the system under development. Others consider the tasks that users of the system will perform and evaluate the architecture in terms of its ability to support those tasks. The development of scenarios by teams of reviewers can assist with the review problem. For example, at the system level, reviewers may construct whole system scenarios to consider how a broad set of problems (e.g., sensor to shooter) will be handled. These scenarios should include desired global system properties, such as reliability, by injecting faults to determine if operational availability can be achieved. At the software level, the reviewers may also construct scenarios to explore software qualities—and then analyze whether a proposed software structure or integration of COTS software supports these desired qualities. This scenario-based approach is embodied in methods developed and promulgated by the Software Engineering Institute (SEISM): the Architecture Tradeoff Analysis MethodSM (ATAMSM) for software qualities [Kazman 00] and the Quality Attribute Workshop (QAW) for system-level concerns [Barbacci 02].
- During the source selection, getting closure on open issues can be problematic. The GVAT usually reviews the set of products associated with a CDRL from multiple contractors and sends the set of comments back to the contractor. Often just after the team finishes, the next set of CDRLs arrive and must be reviewed on a tight schedule. When the contractor responds to the comments (usually 30 days later), everyone is busy with other things, and only a pro forma review of the responses takes place.

SM SEI, Architecture Tradeoff Analysis Method, and ATAM are service marks of Carnegie Mellon University.

4 Strengths

The benefits of using the C4ISRAF that arose during the interviews are listed below:

- The C4ISRAF “fits the military mind” during source selection by leveling the playing field between contractors, since the contractors build architecture documentation from the same framework. Therefore the government source selection is comparing apples to apples.
- The C4ISRAF OV's define a full-scope system and may go from the government to multiple contractors for further refinement, transition to SV's, or inclusion as part of a statement of work for an RFP. This approach allows subcontractors to build parts of the SoS, while the integration contractor oversees the complete development.
- The C4ISRAF OV's provide information that may form the basis of other documentation, such as the CONOPS and ORD, that is required of acquisition organizations. Managed appropriately through tools providing document generation, the OV's can be used to produce the other documents in a nearly automatic fashion.
- The C4ISRAF allows the contractors to be innovative in their own way. For example, during the fly-off, one may focus on technological improvements, another on operational improvements, and yet another on the first stage of evolution from the current legacy systems. The government then gets to judge which is preferable. Of course this situation can work against the “apples to apples” comparison mentioned above.
- The C4ISRAF is good for describing the force structure, which weapon and sensor capabilities will be needed, how the units in the force structure will communicate, and how these units will be deployed. This description provides good estimates for hardware investments.

5 Challenges and Suggestions

This section presents the challenges of using the C4ISRAF and suggestions for overcoming them.

5.1 Process

For this discussion, process refers to the approaches by which C4ISRAF products and other products are built: sequentially, in parallel, or concurrently. The process should define where reviews and feedback loops should occur, which studies should be conducted, and which white papers should be written to develop reasonable alternative architectures. The process should also include steps that can be taken to identify the major risks and propose mitigation steps.

- Challenge 1. The C4ISRAF should *not* define a mandatory process to be followed by all parties. Instead, each organization developing C4ISRAF products should develop its own. But the lack of sufficient examples of good processes is problematic, especially in the transition from OVs to SVs.
- Challenge 2. Building the OV products first and the SV products second can eliminate some of the trade space between operations and systems. In some cases this method forces an expensive system solution rather than a small operational change.
- Challenge 3. Criteria must be developed about the level of abstraction necessary for the products being built. It is often easier to include detail rather than thoughtfully create more abstract products. But unnecessary detail should be avoided.

Suggestions: Suggestions for users of the framework include these five:

1. Develop some example processes with supporting rationale, showing how development could proceed.
2. Establish what should be developed for each defined milestone in the life cycle. This information should include products within each view and the level of detail expected for each product. It should also include any necessary products that are not included in the C4ISRAF view products.
3. Establish how to proceed between stages. This information will include the documentation that the organization must provide, the reviews to conduct, and guidance for completing the steps of subsequent stages.
4. Determine how to minimize throwaway products. A common complaint among developers is the effort they must invest to complete a set of products only to see those

products languish in subsequent steps. Without an established process, developers cannot know how their products will be used. The process should spell out a minimum set of documentation to produce and carry forward for use in downstream stages.

5. Represent all resources explicitly required for the support of the system. Some support resources may be shared with other organizations, and the constraints imposed by these organizations should be reflected in the C4ISR/AF products. Otherwise, once a contract is awarded, changes will be forced on the architecture due to these unconsidered constraints.

5.2 Evolutionary Development and Deployment

Challenge 4. The C4ISR/AF products define the visionary system of the future and take little or no account of the way in which the current legacy systems will be evolved and retired in incremental deployments to eventually achieve this vision. Yet the whole success of many SoSs depends on using development spirals, incremental deployment to the field, incremental upgrades to some legacy systems, and the incremental retirement of others. Of course everyone knows this and attempts to define these spirals in some manner. But the C4ISR/AF does not provide a uniform way of capturing this information. This deficiency is further complicated by the fact that reviewers are often more interested in how the current systems are going to evolve and retire over the next five years than in the visionary system, which they regard as more tenuous. And there is the further complication that technology undergoes unexpected changes over a 10-to-20-year period, and COTS and NDI products are not predictable beyond 5 years.

Challenge 5. Some of the legacy software systems will also have to be evolved and/or retired. Yet no product defines which legacy systems exist and describes how they should be treated. This need is especially important for systems providing logistical support to the new system and to legacy systems under the control of a different program office.

Suggestion: Some form of master evolution plan (MEP) is needed to show how the system will evolve with each development spiral and delivery increment. The MEP should document at least these four types of information:

1. the legacy system and weapon's platform retirement profile at each delivery increment. In general, whole systems should be retired, since building "software bridges" to enable partial system retirement is, in the long run, "throwaway work" that can be quite expensive.
2. the operational capability improvements at each delivery increment
3. the new systems being introduced at each delivery increment
4. the technical risks being addressed by engineering analysis, mission effectiveness studies, and technology prototyping for each development spiral

Note: Although the MEP should be built for the time period to achieve the visionary system, as each delivery is deployed, changes will occur in the next increment in response to changes to the current increment. The decisions as to what to include in the next cycle will always be a mix of mission needs and contracting priorities.

5.3 Technical View

- Challenge 6. The TV of the C4ISR/AF requires the contractors to list the standards being used. In general the contractors list all standards (and COTS and NDI products) they can conceive of using but do not necessarily say how these standards will be used. Hence the TV consists of thousands of standards, some interdependent and others incompatible. Listing all possible standards ensures that any required standards defined within such reference models as the Joint Technical Architecture (JTA) and the Common Operating Environment (COE) are included, but at a loss of clearly defined requirements.
- Challenge 7. Tradeoffs between COTS, GOTS, and custom development software are needed and can have a significant impact on cost, schedule, and various quality attributes. Since no explicit software architecture products exist, there is no direct place to discuss such issues.

Suggestion: More of the engineering analysis and decisions should be documented to show how these standards will be applied and grouped together. For example, if both a Portable Operating System Interface (POSIX) standard and Windows are being proposed as operating systems, then an effort must be made to define which type of computational platforms will be used and where, and which operating systems are used in each platform type.

Note: The proposed DoDAF has already strengthened the requirements for the TV section.

5.4 Software Issues

- Challenge 8. The C4ISR/AF was not intended to document software views, nor was it intended to explain or explore the software styles that were proposed for use in the system. Yet software is the technology on which integration and interoperability depends. Even at the highest level of system description, some concept of these styles should be captured; for example, publish/subscribe, client/server, pipe/filter, master/slave, load sharing, peer-to-peer, and hub/spoke.
- Challenge 9. No products exist that define software development plans or the processes to be used in software development. Yet these plans and processes have a tremendous impact on the success of a program; software budgeting and scheduling need such representations.

Suggestions: The following four suggestions address the software issues:

1. The important software views should be included in the architecture at the early stages to capture the important concepts and styles. They can be included by combining the IEEE-1471 viewpoint concepts [IEEE 00], and the “views and beyond” approach outlined in *Documenting Software Architectures: Views and Beyond* [Clements 03]. This combination will define which software views are necessary for the identified stakeholders.
2. Stakeholders can identify the needs for software views. For example, the OV-6c Operational Event – Trace Description may connect well to use case or concurrency views in software. The OV-5 Activity Diagrams and SV-5 Activity to System Function Traceability Matrix provide the opportunity to introduce sequence diagrams that could also trace to software concurrency views. SVs will include roles and responsibilities, including the transition from manual-to-manual to manual-to-system and system-to-system interactions. The choice among alternative interactions may include hardware/software tradeoffs.
3. The QAW can be used to develop system scenarios, from which themes can be derived and the important software and system engineering risks established [Barbacci 02].
4. Some combination of the three suggestions above should be incorporated into an engineering process for developing the software architecture drivers.

5.5 Crosscutting Quality Attributes

Challenge 10. In an SoS context, the structure of the architecture is driven not by its functionality, but rather by the quality attributes (such as performance, availability, security, and modifiability) that cut across the systems. Major architectural decisions are based on these quality attributes; for example, the distribution of functionality and data; the communication between components; the start-up, shut-down, and failover procedures. Yet the C4ISR/AF addresses quality attribute issues only in terms of bandwidth between nodes. It presumes that the system engineers will take the appropriate quality attributes into account, but it does not recommend how to determine which quality attributes are important and how to ensure that the architecture supports them.

One of the interviews revealed that the contractor was using performance (latency and network bandwidth) as an architectural driver. The system used the Rational Rose tool set to document the architectural products and used an annotation scheme to capture the relevant performance parameters. These parameters were then automatically transferred to an Excel spreadsheet, which performed calculations to determine if the latency and bandwidth criteria were being satisfied.

Suggestion: The SEI developed the ATAM for evaluating software architectures. The ATAM uses groups of stakeholders to determine the important quality attributes; create and prioritize scenarios related to these quality attributes; capture all of this information in a utility tree; and develop risks, tradeoffs, sensitivity points, and trends associated with the architecture as it relates to the high-priority scenarios. This method could be extended and used for system architecture. The SEI also developed the QAW that generates and prioritizes system scenarios before a system or software architecture has been created. If a risk, sensitivity point, or tradeoff is found in the software architecture, it can be “bubbled up” to the related C4ISRAF products where its effect on the SoS can be further analyzed.

5.6 Training and Testing

Challenge 11. Training and testing environments can have significant impacts on the architecture. For example, for “human and hardware in the loop” training, the software being deployed must be directly usable in the training simulation environment. Similarly, the deployed software must fit directly into the various testing environments, which should be sufficiently powerful to minimize the amount of testing required (e.g., flight testing). Yet the C4ISRAF provides no direct discussion of either of these environments.

Suggestion: Ensure that the architecture includes both training and testing environments, and that scenarios for these environments are generated.

5.7 Tool Support

The C4ISRAF requires that many products be built using graphical notations and tables, which usually results in the development of hundreds of pages of graphical and tabular documentation and supporting text. Tools are necessary to ensure consistency and establish version control between the various products.

Unfortunately no commercial tool sets were available to support the development of products when most of these programs started, although one tool, Popkin’s System Architect (PSA), now supports the C4ISRAF products [Popkin 03]. All the contractors interviewed used some form of graphical tool sets to build the C4ISRAF products. Some started late enough to use PSA. Others used a government-furnished equipment (GFE) tool set that built only part of the product set and used commercial tools based on the Unified Modeling Language (UML) for other products. In general the experience with using tools was very significant to all the contractors, was discussed at length in the interviews, and is recorded below.

Challenge 12. Using immature tools can squander resources. In a source selection environment, the government is often tempted to ensure a level playing field by having the competing contractors use the same tool set, which makes it easier to compare their efforts. This tool set could still be used for many different purposes; for example, simulating mission effectiveness; developing architecture documentation; producing reliability,

maintainability, and availability studies; simulating performance loading on networks; or evaluating man-machine effectiveness.

- If the tool set is GFE, the government must ensure that the contractors can get sufficient training initially and have timely access to resolving problems. An immature GFE tool set can lead to endless frustration on the part of everyone involved. While having all the contractors use the same tool set is appealing and may be seen as leveling the playing field, the difficulties associated with its use may overcome the initially perceived benefits.
- If the government intends to use its own tool and have the contractors deliver some data to represent their concept, once again the tool set and the data definition must be rock solid, and the government must work closely with all contractors starting early in the source selection to ensure that this approach will work.

Challenge 13. The C4ISRAF products tend to be built in a number of parallel activities, and the products must be integrated at some merge points. During the integration, the names of the software elements used must be unified and integrated across the various products, and consistency must be enforced. Often tool sets do not support this integration, so it must all be done manually, which is time-consuming and error prone.

Challenge 14. Many of the graphical tool sets available in the marketplace are based on UML, which was initially built to represent software designs using object-oriented technology. The C4ISRAF is based on the Integrated Computer-Aided Manufacturing (ICAM) Definition (IDEF) representations, which are not currently represented in the UML model. Other important architectural products are not included in most UML tool sets, such as

- some software architectural views (layering diagrams, context diagrams)
- many C4ISRAF IDEF diagrams
- programmatic diagrams, such as schedules

Note: UML is in the process of being extended to include many of the IDEF graphical representations. However, no plan currently exists to resolve the inconsistencies between the two types of representations.

Challenge 15. Sequence diagrams are routinely used to elaborate software use cases in UML-based development. These diagrams usually consider the event-based, ordered interaction between actors (users and/or remote interfaces) and classes or objects that represent the software automating the system. The diagrams detail that interaction. One C4ISRAF product can be represented as a sequence diagram showing the interactions among operational activities, actors, and systems. However, often when developing system use cases, the automation versus manual split has not yet been made; in this case the sequence diagrams can be used to show the interaction between actors or between the actors and the different systems.

Suggestions: The following eight suggestions address the challenges of tool support:

1. The government should not require the contractors to use a tool set that is not supported commercially. The government should encourage the contractor to use a familiar tool set based on a standard, thus ensuring that the contractors are mature users of the tool set and will not be swamped with start-up issues.
2. The C4ISRAF product builder should develop a set of rules and procedures for building fragments of the architecture independently so that they can be integrated smoothly.
3. The C4ISRAF product builder should be prepared to include some tools (e.g., spreadsheets and scheduling tools) that are not included in the graphical tool set for building the architecture. If possible, the product builder should develop wrappers to move data semi-automatically between the various tool sets.
4. The C4ISRAF product builder should be prepared to annotate the tool set using colors, shaped comment boxes, and other techniques in a uniform way to add information for the reader. The developer should bear in mind the problem mentioned earlier concerning stakeholders' capabilities to look at only those parts of the system within their immediate concern. The viewpoints of each category of stakeholder should also be considered in the annotation scheme.
5. The C4ISRAF product developer should develop a good understanding of views and information needs to raise the level of debate above that of the representation. The draft version of the DoDAF establishes UML as the desired representation method. However, the draft states that UML has been adapted to cover architectural views and differs in intent from software views using the same representation methods (e.g., use cases, class hierarchies). The focus should be more on what information the builders of OV and SV products must produce rather than the exact form of expression.
6. The C4ISRAF product developer should use tools to accomplish more than a convenient information storage mechanism. The tools should also enforce guidance with respect to
 - adherence for a specific process
 - product releases
 - product support
 - user validation
 - system build, test, and acceptance
7. Many different tools will be used to build all the C4ISRAF products and other products associated with requirements, mission effectiveness simulation, schedules, software, releases, and so forth. Hence the C4ISRAF product builder should have a strong configuration management system to coordinate releases across the complete set of tools.
8. The graphical products should be built using a tool set that is targeted at these products; such a tool set invariably captures all the data associated with building the products. Thus the concept of a data dictionary is not needed, since the tool set's data repository serves this purpose.

6 Conclusions

Our review of C4ISRAF experiences captured results and impressions from the application of the framework on several ACAT I programs. While the framework's use varied from program to program, we discovered several recurring themes pointing to issues for improvement on new programs. Some of these issues are addressed under DoDAF, but until that new framework sees widespread adoption, the C4ISRAF concerns should be addressed.

- Organizations need to develop a process for using and building the C4ISRAF products and other products. This process will control the flow of building and reviewing the architecture and relating the architecture to the system. Cross-service workshops or discussion forums can bring a shared understanding to the issues related to the architecture. Lessons learned can also provide needed guidance in this area.
- Program teams must tailor the C4ISRAF products they use. Reviewers cannot be inundated with information that they have neither the time nor expertise to comment on. Tailoring may include annotating products, providing more or less detail within specific products, or adding guidelines for understanding specific products. Program teams may also consider adding other views such as those that address software concerns. They should ask only for the details that can be reviewed effectively by the available reviewers within the specified time frame.
- Tool usage requires specific rules for successful application within a program. Out of the box, many COTS tools are not sufficient to address the needs of architects or reviewers. Even when tailored, without training, the architects will not use tools consistently across a program, and different contractors will apply the tools in different ways. Organizations may consider the use of different tool sets by contractors, provided the contractors are given adequate guidance in defining the information they must provide. The government must also expect some guidance on using the tools to understand and review the products.
- A waterfall approach to developing the views (operational followed by system) does not allow for sufficient tradeoffs between these views. Organizations can adopt some iterative OV/SV development that allows for greater parallelism and the ability to make refinements or corrections at the developer level without having to "go up the chain" once a product has been approved.
- Crosscutting issues such as performance, availability, modifiability, and security are at least equal to functionality in determining the architecture. Scenario-based methods offer one approach to dealing with these issues. Architecture evaluation in the form of an ATAM or architectural requirements elicitation in the form of a QAW can help with exploring these issues and assuring stakeholders they are being addressed.

Appendix A: Summary of the Architectural Products

Table 1: Architectural Products [C4ISR/AF 97]

Applicable View	Framework Product	Framework Product Name	General Description
All Views	AV-1	Overview and Summary Information	Scope, purpose, intended users, environment depicted, analytical findings
	AV-2	Integrated Dictionary	Data repository with definitions of all terms used in all products
Operational	OV-1	High-Level Operational Concept Graphic	High-level graphical/textual description of operational concept
	OV-2	Operational Node Connectivity Description	Operational nodes, operational activities performed at each node, connectivity and information exchange needlines between nodes
	OV-3	Operational Information Exchange Matrix	Information exchanged between nodes and the relevant attributes of that exchange
	OV-4	Organizational Relationships Chart	Organizational, role, or other relationships among organizations
	OV-5	Operational Activity Model	Operational activities and relationships among activities, inputs, and outputs. Overlays can show cost, performing nodes, or other pertinent information.
	OV-6a	Operational Rules Model	One of the three products used to describe the operational activity sequence and timing; identifies business rules that constrain operation
	OV-6b	Operational State Transition Description	One of three products used to describe the operational activity sequence and timing; identifies business process responses to events
	OV-6c	Operational Event-Trace Description	One of three products used to describe the operational activity sequence and timing; traces actions in a scenario or sequence of events and specifies the timing of events
	OV-7	Logical Data Model	Documentation of the data requirements and structural business process rules of the OV.

Table 1: Architectural Products [C4ISR/AF 97] (continued)

Applicable View	Framework Product	Framework Product Name	General Description
Systems	SV-1	Systems Interface Description	Identification of systems and system components and their interconnections, within and between nodes
	SV-2	Systems Communications Description	Physical nodes and their related communications laydowns
	SV-3	Systems-Systems Matrix	Relationships among systems in a given architecture; can be designed to show relationships of interest (e.g., system-type interfaces, planned vs. existing interfaces).
	SV-4	Systems Functionality Description	Functions performed by systems and the information flow among system functions
	SV-5	Operational Activity to Systems Function Traceability Matrix	Mapping of systems back to operational capabilities or of system functions back to operational activities
	SV-6	Systems Data Exchange Matrix	Provides details of systems data being exchanged between systems
	SV-7	Systems Performance Parameters Matrix	Performance characteristics of each system's hardware and software elements, for the appropriate time frame
	SV-8	Systems Evolution Description	Planned incremental steps toward migrating a suite of systems to a more efficient suite, or toward evolving a current system to a future implementation
	SV-9	Systems Technology Forecast	Emerging technologies and software/hardware products that are expected to be available in a given set of time frames, and that will affect future development of the architecture
	SV-10a	Systems Rules Model	One of three products used to describe systems activity sequence and timing—constraints that are imposed on systems functionality due to some aspect of systems design or implementation
	SV-10b	Systems State Transition Description	One of three products used to describe systems activity sequence and timing—responses of a system to events
SV-10c	Systems Event-Trace Description	One of three products used to describe systems activity sequence and timing—system-specific refinements of critical sequences of events and the timing of these events	
SV-11	Physical Schema	Physical implementation of the Logical Data Model's information (e.g., message formats, file structures, physical schema)	

Table 1: Architectural Products [C4ISR/AF 97] (continued)

Applicable View	Framework Product	Framework Product Name	General Description
Technical	TV-1	Technical Standards Profile	Extraction of standards that apply to the given architecture
	TV-2	Technical Standards Forecast	Description of emerging standards that are expected to apply to the given architecture, within an appropriate set of time frames

Appendix B: Acronym List

ACAT I	acquisition category I
ATAM	Architecture Tradeoff Analysis Method
AV	all view
C4ISR	command, control, communications, computer, intelligence, surveillance, and reconnaissance
C4ISRAF	Command, Control, Communications, Computer, Intelligence, Surveillance, and Reconnaissance Architecture Framework
CDRL	contract data requirements list
CFI	call for improvement
COE	Common Operating Environment
CONOPS	concept of operations
COTS	commercial off-the-shelf
DoDAF	Department of Defense Architecture Framework
GFE	government-furnished equipment
GOTS	government off-the-shelf
GVAT	government analysis team
GVSSAT	government-source-selection technical analysis team
IDEF	Integrated Computer-Aided Manufacturing (ICAM) Definition
JTA	Joint Technical Architecture

MEP	master evaluation plan
NDI	non-developmental item
ORD	operational requirements document
OV	operational view
PSA	Popkin's System Architect
POSIX	Portable Operating System Interface
QAW	Quality Attribute Workshop
RFP	request for proposal
SA	software architecture
SoS	system of systems
SV	system view
TRD	technical requirements document
TV	technical view
UAV	Unmanned Aerial Vehicle
UML	Unified Modeling Language

References

URLs valid as of the publication date of this document

- [Barbacci 02]** Barbacci, Mario R.; et al. *Quality Attribute Workshops, 2nd Edition* (CMU/SEI-2002-TR-019, ADA405790). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2002. <<http://www.sei.cmu.edu/publications/documents/02.reports/02tr019.html>>.
- [Clements 03]** Clements, Paul; et al. *Documenting Software Architectures: Views and Beyond*. Boston, MA: Addison-Wesley, 2003.
- [C4ISRAF 97]** C4ISR Architecture Working Group. *C4ISR Architecture Framework, Version 2.0*. Washington, DC: Department of Defense, 1997.
- [DoDAF 03]** DoD Architecture Framework Working Group. *DoD Architecture Framework Version 1.0*. Washington, DC: Department of Defense, 2003. <http://www.teao.saic.com/jfcom/ier/documents/DOD_architecture_framework_volume1.doc> and <http://www.teao.saic.com/jfcom/ier/documents/DOD_architecture_framework_volume2.doc>.
- [IEEE 00]** Institute of Electrical and Electronics Engineers. *IEEE Recommended Practice for Architectural Description of Software-Intensive Systems* (IEEE Std 1471-2000). Piscataway, NJ: IEEE Computer Press, 2000.
- [Kazman 00]** Kazman, Rick; Klein, Mark; & Clements, Paul. *ATAM: Method for Architecture Evaluation* (CMU/SEI-2000-TR-004, ADA382629). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2000. <<http://www.sei.cmu.edu/publications/documents/00.reports/00tr004.html>>.
- [Popkin 03]** Popkin Software. *System Architect's C4ISR Option Provides Support for the DoD Architecture Framework*. <<http://government.popkin.com/frameworks/c4isr.htm>> (September 2003).

REPORT DOCUMENTATION PAGE			<i>Form Approved</i> OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE September 2003	3. REPORT TYPE AND DATES COVERED Final		
4. TITLE AND SUBTITLE DoD Experience with the C4ISR Architecture Framework		5. FUNDING NUMBERS F19628-00-C-0003		
5. AUTHOR(S) William G. Wood and Sholom Cohen				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213		8. PERFORMING ORGANIZATION REPORT NUMBER CMU/SEI-2003-TN-027		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116		10. SPONSORING/MONITORING AGENCY REPORT NUMBER		
11. SUPPLEMENTARY NOTES				
12A DISTRIBUTION/AVAILABILITY STATEMENT Unclassified/Unlimited, DTIC, NTIS		12B DISTRIBUTION CODE		
13. ABSTRACT (MAXIMUM 200 WORDS) The Department of Defense (DoD) is mandating the use of the Command, Control, Communications, Computer, Intelligence, Surveillance and Reconnaissance Architecture Framework (C4ISR/AF) for large-scale software-intensive systems. The authors conducted eight interviews with personnel who have used the C4ISR/AF in acquisition projects. The intent of the interviews was to find the strengths and weaknesses of the C4ISR/AF, so that this information could be related to future users of the framework. This technical note discusses the context for using the C4ISR/AF, the observations made during the interviews about its use, and the strengths and challenges of using it. Suggestions for overcoming these challenges also are included.				
14. SUBJECT TERMS architecture framework; C4ISR; Command, Control, Communications, Computer, Intelligence, Surveillance, and Reconnaissance Architecture Framework		15. NUMBER OF PAGES 36		
16. PRICE CODE				
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	