Using the Architecture Tradeoff Analysis Method[SM] to Evaluate a Reference Architecture: A Case Study

Brian P. Gallagher

June 2000

Architecture Tradeoff Analysis Initiative

Technical Note
CMU/SEI-2000-TN-007

Contents

List of Figures

About the Technical Note Series on Architecture Evaluation in the Department of Defense

The Product Line Systems Program is publishing a series of technical notes designed to condense knowledge about architecture evaluation practices into a concise and usable form for the Department of Defense (DoD) acquisition manager and practitioner. This series is a companion to the Software Engineering Institute (SEI) series on product line acquisition and business practices [Bergey 99].

Each technical note in the series will focus on the use of architecture evaluation and, in particular. on applying the SEI's architecture tradeoff analysis technology in the Department of Defense. Our objective is to provide practical guidance on ways to integrate sound architecture evaluation practices into their acquisitions. This series of technical notes will lay down a conceptual foundation for DoD architecture evaluation practice.

Abstract

The software architecture of a system is a major determinant of software quality and one of the earliest artifacts available for evaluation. For a government acquisition organization, the ability to evaluate software architectures can have a favorable impact on the delivered system. This technical note describes the application of the Architecture Tradeoff Analysis Method[SM] (ATAM[SM]) to evaluate a reference architecture for ground-based command and control systems. The use of the term *reference architecture* in the context of this application is presented. A general overview of the ATAM process is provided and the results of the ATAM are explored, including the benefits of performing an ATAM-based architecture evaluation both to the acquirer and to the developer.

[SM] Architecture Tradeoff Analysis Method and ATAM are service marks of Carnegie Mellon University.

1 Introduction

Most modern defense systems rely heavily on software to achieve system functionality. Because software architecture is a major determinant of software quality, it follows that software architecture is critical to the quality of a software-intensive system. For a Department of Defense (DoD) acquisition organization, the ability to evaluate software architectures can reduce the risk that the delivered system will not meet its quality goals.

This technical note describes the application of the Architecture Tradeoff Analysis Method (ATAM) in the evaluation of a government-sponsored reference architecture for a ground-based command and control system. The names of the organizations involved and the reference architecture are not provided since they are not necessary for the discussion. The use of the term *reference architecture* is explored as well as how it relates to a *product line architecture.* A general overview of the ATAM process is provided and the results of the ATAM are explored, including the benefits of performing an ATAM-based architecture evaluation both to the acquirer and the developer.

2 The Context for the Architecture Evaluation

The Architecture Tradeoff Analysis Method was used to evaluate a reference architecture for ground-based command and control systems. To explore this application of ATAM, we first need to define the terms being used and describe the evaluation context. We begin with the definition of software architecture.

2.1 What is Software Architecture?

> *The software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them* [Bass 98].

The software architecture represents the earliest software design decisions. These design decisions are the most critical to get right and the most difficult to change downstream in the system development life cycle. The software architecture is the first design artifact addressing reliability, modifiability, security, real-time performance, and interoperability goals and requirements.

There are many different views of a software architecture, and which one is relevant depends on the stakeholders and the system properties that are of interest. If we consider the analogy of the architecture of a building, various stakeholders such as the construction engineer, the plumber, and the electrician all have an interest in how the building is to be constructed. However, they are interested in different structural units and different structural relationships; they each have a different view of what constitutes the architecture. Each of their views of architecture is valid. Each represents a structure that maps to one of the structural goals of the building, and all views are necessary to represent the architecture of the building fully. Similarly, a software architecture has a variety of stakeholders, including possibly the development organization, the end user, the system maintainer, the operator, and the acquisition organization. Each of these stakeholders has a vested interest in different system properties and goals that are structurally represented by different views of the system. These different properties and goals and their corresponding architectural views are important to understand and to analyze in order to reason about the appropriateness and the quality of the architecture.

The software architecture to be evaluated was actually a reference architecture.

2.2 What is a Reference Architecture?

A *reference architecture* is the generalized architecture of several end systems that share one or more common domains. The reference architecture defines the infrastructure common to the end systems and the interfaces of components that will be included in the end systems. The reference architecture is then instantiated to create a software architecture of a specific system. The definition of the reference architecture facilitates deriving and extending new software architectures for classes of systems. A reference architecture, therefore, plays a dual role with regard to specific target software architectures. First, it generalizes and extracts common functions and configurations. Second, it provides a base for instantiating target systems that use that common base more reliably and cost effectively.

The concept of a reference architecture is very similar to that of an application framework in the parlance of object technologists and to that of a product line architecture. In fact, the terms *reference architecture* and *product line architecture* are often used interchangeably. A product line architecture is the architecture for a software product line.[1]

Within a government context, the establishment of a reference architecture that can span a broad range of missions, development organizations, and operations concepts is motivated by the desire of end-system customers and integrators. These groups wish to achieve high levels of strategic software reuse and interoperability. The reference architecture provides specifications for both system and component developers.

2.3 The Goals of the Architecture in Question

The reference architecture being evaluated was not developed to support a software product line, but rather to provide a common framework for classes of ground-based command and control systems being built by multiple government and commercial organizations.

The primary goal of this reference architecture is to standardize the interfaces between software components used in ground-based command and control systems. The expectation is that the standardized interfaces will reduce the risk and cost of implementing new systems by facilitating the use of standard components in a "plug-and-play" mode. The stated vision is that vendors will develop components that are compliant with the reference architecture, making the job of a system integrator easier, quicker, and less risky. Figure 1 illustrates this vision.

[1] A *software product line* is a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way.

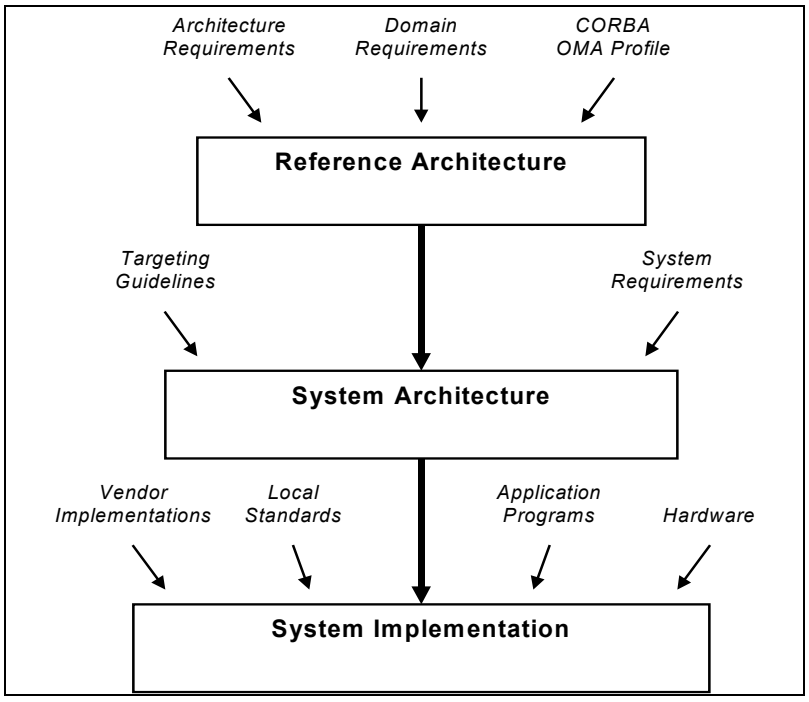

Figure 1: The Intended Use of the Reference Architecture Being Evaluated

2.4 The Acquisition Organization

Because the reference architecture for ground-based command and control systems is envisioned to be a common starting point across many instantiations of systems, acquisition organizations from two distinct government agencies provided funding.

Acquisition management responsibilities were split between the two major agencies, and the business case and generalized requirements for the reference architecture were developed in conjunction with both communities.

2.5 The Development Organization

The two agencies used various methods to encourage developers to participate in the definition of the reference architecture. Some developers participated as part of an ongoing support role to an agency, while others participated as a task on an existing development effort funded by the agency. In addition, vendors interested in providing components compliant with the reference architecture supplied developers to participate in the definition of the architecture using internal funds.

3 The ATAM

3.1 Purpose

The purpose of the ATAM is to assess the consequences of architectural decision alternatives in light of quality attributes. The method ensures the right questions are asked early to discover

- *risks*: alternatives that might create future problems in some quality attribute
- *sensitivity points*: alternatives for which a slight change makes a significant difference in a quality attribute
- *tradeoffs*: decisions affecting more than one quality attribute [Kazman 98][2]

The ATAM is intended to analyze an architecture with respect to its quality attributes, not its functional correctness.

The ATAM involves a wide group of stakeholders (including managers, developers, maintainers, testers, reusers, end users, and customers) in an effort to surface the relevant stakeholders' quality goals for the system. The ATAM is a method for mitigating architecture risk, a means of detecting areas of potential risk within the architecture of a complex software-intensive system, and not a precise mathematical analysis. As such, the ATAM can be done early in the software development life cycle, and it can be done inexpensively and quickly.

It does not need to produce detailed analyses of any measurable quality attribute of a system (such as latency or mean time to failure) to be successful, but it instead identifies trends where some architectural parameter is correlated with a measurable quality attribute of interest.

3.2 ATAM Steps

The ATAM process consists of the following 10 steps:

1. Present the ATAM: a quick overview by the evaluation team of the ATAM steps, techniques used, and outputs from the process.
2. Present the business drivers: a brief presentation by the system manager describing the business drivers and context for the architecture.
3. Present the architecture: the architect's presentation of the architecture.

[2] The ATAM has been improved since the application described in this report [Kazman 00].

4. Identify architectural styles: an itemization of styles discovered as a result of the previous step.
5. Generate the quality attribute utility tree: identification, prioritization, and refinement of the most important quality attribute goals in the form of a utility tree.
6. Elicit and analyze architectural styles: a probing of the architectural styles in light of the quality attributes in order to identify risks, sensitivity points, and tradeoffs.
7. Generate seed scenarios: a representation of the stakeholder's interest to understand quality attribute requirements.
8. Brainstorm and prioritize scenarios: addition of scenarios from stakeholders and an understanding of their relative importance.
9. Map scenarios onto styles: continuing to identify risks, sensitivity points, and tradeoffs while noting styles and components within styles that are affected by each scenario.
10. Present out-brief and/or write report: recapitulation of the execution of the ATAM steps, results, and recommendations.

4 The Evaluation

4.1 Timing of the Evaluation

At the time of the ATAM evaluation, the development team for the reference architecture had progressed through the following three phases:

1. Understand the problem and scope the effort.
2. Document the reference architecture and develop high-level Interface Design Language (IDL).
3. Complete approximately one-fourth of the IDL for the 12 sub-domains.

4.2 Phase 1 of the Evaluation

The ATAM method is best applied in two phases. The first phase involves the initial connection of the evaluation team leads with the architects and the beginning of the exploration and analysis. Phase 1 allows the evaluation team to get acquainted with the architect(s), the system purpose, and the architecture and to begin a preliminary analysis of the architecture. Phase 1 also permits the architects to become familiar with the ATAM and understand the information they must supply for the complete evaluation to proceed.

An SEI ATAM evaluation team was selected for this evaluation. The team leads met with reference architecture architects and provided templates for architecture presentation to be given as part of the evaluation exercise. A draft utility tree and seed scenarios were produced. A template was also given to the architecture manager to be used in the presentation of the business drivers.

4.3 Phase 2 of the Evaluation

All of the ATAM steps are executed during Phase 2, which typically lasts two days. Any preliminary analysis resulting from Phase 1 is presented.

For this evaluation, an assembly of approximately two dozen stakeholders gathered for the two-day event.[3] These stakeholders represented the two government acquisition organizations as well as a variety of government contractors with a vested interest in the reference architecture. Some of these contractors had been involved with the definition of the reference architecture. Some hoped to build components that would be compliant with the reference architecture.

[3] The typical use of ATAM involves a much smaller group of stakeholders. In this case, the two government acquisition agencies felt it was necessary to involve a broader set.

All steps of the ATAM were executed during the two-day period. The architects used the templates provided in their presentation of the architecture. The draft utility tree was presented and then refined by the larger convened group. Scenarios were generated and prioritized by the entire group. Some key aspects of this evaluation were the business drivers and the architectural artifacts that were available.

4.3.1 Business Drivers

One of the most important steps when performing an ATAM-based architecture evaluation is to elicit from the acquirer a clear articulation of the business case driving the development of the architecture. During this evaluation, a representative from one of the acquisition agencies described the business goals motivating the development effort and hence identified the primary drivers (e.g., high availability, time to market, or high security) for the architecture. The presentation of the business goals included these key points:

- The goal of the reference architecture is to enable system builders to integrate systems on time, within costs, while meeting performance needs; the primary objective is to drive down the cost of command and control systems. The strategy involves the use of object-oriented technology and Common Object Request Broker Architecture (CORBA) to achieve this goal.
- The focus is on defining interfaces to sub-domains with an emphasis on interoperability among the sub-domains as opposed to reusing sub-domains directly.
- The definition of the standard interfaces gives vendors a target and permits competition among vendors in developing products for the same sub-domain.
- System builders should be able to use the reference architecture to integrate the sub-domains quickly. Under current practices, integration costs are very high. Point solutions are too expansive and too late.
- Customers want open systems.
- The reference architecture will lower the cost and cycle time for system integrators.
- The reference architecture will help meet future challenges by enabling
  - constellations of systems
  - "plug and play," regardless of location of component (autonomous versus under direct control)

The final point was aimed at being able to demonstrate the "goodness" of the reference architecture to other government agencies.

4.3.2 Architectural Artifacts Available for the Evaluation

The artifacts available to describe the ground-based command and control reference architecture included of a set of definitions for large-grained components, IDL interfaces for these components, and CORBA mechanisms used to connect the components. A number of use-cases were also presented during the ATAM evaluation, and their realization in the architecture was briefly discussed. As noted earlier, the reference architecture specification was not complete at the time of the evaluation.

5 Evaluation Results

5.1 The Findings

The 2-day ATAM evaluation produced a list of 22 risks, which were grouped into the following 6 risk themes:

1. The architecture provides no mechanisms for the integrator to reason about quality attribute tradeoff (for example, real-time performance versus reliability).
2. The definition of the reference architecture did not include key stakeholder requirements for real-time performance, integrability, modifiability, and availability.
3. There are major tradeoffs between affordability, real-time performance, and modifiability that have been explicitly left to the integrator to address.
4. There are no overarching quality attribute models (for example, real-time performance models, availability models).
5. There is a lack of common support for managing data.
6. The direct reliance on CORBA as a distributed system poses a risk for CORBA evolution or change to a different middleware.

Based upon the ATAM evaluation, the evaluation team concluded that little had been done in the definition of the reference architecture to address the quality goals of the ground-based command and control systems to be supported by this architecture. The fulfillment of those goals would be left to the integrators, leaving the likely result that components built to be compliant with the reference architecture would indeed "plug," with no real guarantee that they would "play." In other words, the business drivers for the reference architecture are at risk with the given architecture definition.

5.2 The Benefits

An architecture helps the developer and customer reason about a proposed system during its development and evolution. A structured approach to eliciting that reasoning early in development increases the system developer's probability that a system built conforming to the architecture will meet the needs of its customer base.

The two days spent using the ATAM to analyze the reference architecture for ground-based command and control systems pointed out potential deficiencies that may have taken months, perhaps years, to uncover at a greatly increased cost to the acquirer.

The ATAM evaluates an architecture with respect to its quality goals. In this case, the goals were performance, security, availability and modifiability. It became evident fairly quickly

that the proposed architecture did not provide any architectural mechanisms to aid the integrator in ensuring that the intended quality goals were met.

The team of developers and acquirers of the reference architecture came away from the two-day ATAM with a clear understanding that more work needed to be done to ensure that not only were the interfaces syntactically correct, but that the components were integrated semantically as well.

The benefits of performing an ATAM are clear: early identification of risks, sensitivity points, and tradeoffs before design decisions are made and become costly to change. As one participant from the government team noted after the results of the ATAM were briefed, “Why **wouldn’t** a program want to do an ATAM-based evaluation?”

6 Conclusion

This note has described the application of the ATAM to conduct an architecture evaluation during the development of a government-sponsored reference architecture for ground-based command and control systems. The context for the evaluation was given. A general overview of the ATAM process was presented, and finally, the results of this ATAM-based evaluation were examined as well as the perceived benefits to both the acquirer and the developer for performing the evaluation.

The SEI is collaborating with several government acquisition organizations to explore the appropriate use of the ATAM within these organizations, to help them adopt architecture evaluation practices, and to help them include the appropriate language in a request for proposal (RFP) to ensure that architecture evaluation is an integral part of evaluating proposals. As experience is gained, we will continue to share our lessons learned in future technical notes in this series.

References

[Bass 98] Bass, L., Clements, P., and Kazman, R. *Software Architecture in Practice*. Reading, Mass.: Addison Wesley, 1998.

[Bergey 99] Bergey, J.; Fisher, M.; Jones, L.; and Kazman R. *Software Architecture with ATAM in the DoD System Acquisition Context* (CMU/SEI-99-TN-012, ADA 377450).Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, September 1999. Available WWW <URL: http://www.sei.cmu.edu/publications/documents/99.reports/99tn012/99tn012abstract.html>.

[Clements 99] Clements, Paul & Northrop, Linda. *A Framework for Software Product Line Practice, Version 2.0* [online]. Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, August 1999. Available WWW <URL: http://www.sei.cmu.edu/plp/framework.html>.

[Kazman 98] Kazman, R.; Barbacci, M.; Klein, M.; Carriere, S. J.; and Woods, S. G. "Experience with Performing Architecture Tradeoff Analysis," 54-64. *Proceedings of the 21st International Conference on Software Engineering (ICSE 21),* Los Angeles, Cal., May 1999.

[Kazman 00] Kazman, R.; Klein, M.; and Clements, P. *ATAM: Method for Architecture Evaluation* (CMU/SEI-2000-TR-004). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, September 2000. Available WWW <URL: http://www.sei.cmu.edu/publications/documents/00.reports/00tr004.html>.

Feedback and Contact

Comments or suggestions about this document or the series of technical notes on architecture evaluation in the Department of Defense are welcome. We want this series to be responsive to the needs of DoD and government personnel. To that end, comments concerning this technical note, inclusion of other topics, or any other issues or concerns will be of great value in continuing this series. Comments or suggestions should be sent to

Linda Northrop, Director
Product Line Systems Program
lmn@sei.cmu.edu

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
(Leave Blank)	June 2000	Final

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
Using the Architecture Tradeoff Analysis Method to Evaluate a Reference Architecture: A Case Study	F19628-00-C-0003

6. AUTHOR(S)

Brian P. Gallagher

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213	CMU/SEI-2000-TN-007

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116	

11. SUPPLEMENTARY NOTES

12A DISTRIBUTION/AVAILABILITY STATEMENT	12B DISTRIBUTION CODE
Unclassified/Unlimited, DTIC, NTIS	

13. ABSTRACT (MAXIMUM 200 WORDS)

The software architecture of a system is a major determinant of software quality and one of the earliest artifacts available for evaluation. For a government acquisition organization, the ability to evaluate software architectures can have a favorable impact on the delivered system. This technical note describes the application of the Architecture Tradeoff Analysis Method[SM] (ATAM[SM]) to evaluate a reference architecture for ground-based command and control systems. The use of the term *reference architecture* in the context of this application is presented. A general overview of the ATAM process is provided and the results of the ATAM are explored, including the benefits of performing an ATAM-based architecture evaluation both to the acquirer and to the developer.

14. SUBJECT TERMS	15. NUMBER OF PAGES
architecture evaluation, Architecture Tradeoff Analysis Method[SM] (ATAM[SM]), product line acquisition, software architecture	15

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

[SM] Architecture Tradeoff Analysis Method and ATAM are service marks of Carnegie Mellon University.