

# **Using the Architecture Tradeoff Analysis Method to Evaluate a Wargame Simulation System: A Case Study**

Lawrence G. Jones  
Anthony J. Lattanze

*December 2001*

**Architecture Tradeoff Analysis Initiative**

Unlimited distribution subject to the copyright.

**Technical Note**  
CMU/SEI-2001-TN-022

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

Copyright 2002 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 52.227-7013.

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

<b>About the Technical Note Series on Business and Acquisition Guidelines</b>	<b>vii</b>
<b>Abstract</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Context for the Architecture Evaluation</b>	<b>2</b>
2.1 Software Architecture	2
2.2 The Joint National Integration Center	3
2.3 The Wargame 2000 System	3
<b>3 The ATAM</b>	<b>5</b>
3.1 Purpose	5
3.2 ATAM Steps	5
<b>4 The ATAM Evaluation of Wargame 2000</b>	<b>7</b>
4.1 Background	7
4.2 Business and Mission Drivers	7
4.3 System Overview	9
4.4 Architectural Approaches	10
4.5 Utility Tree	11
4.6 Scenario Generation and Prioritization	13
4.7 Overview of the Analysis Process	13
4.8 Scenario Analysis	14
4.8.1 Scenario 1	14
4.8.2 Scenario 6	15
4.9 Sensitivity Points	15
4.10 Other Risks	16
4.11 Post-ATAM Activities	18
<b>5 Conclusion</b>	<b>19</b>
<b>References</b>	<b>20</b>



---

## List of Figures

Figure 1: The Primary Wargame 2000 Subsystems	10
Figure 2: The Utility Tree Concept	11



---

## List of Tables

Table 1: Wargame 2000 Utility Tree	12
Table 2: Prioritized Wargame 2000 Scenarios	13





---

## **About the Technical Note Series on Business and Acquisition Guidelines**

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

Each technical note in the series will focus on applying architecture tradeoff analysis in the Department of Defense. Our objective is to provide practical guidance to early adopters on ways to integrate sound architecture tradeoff analysis practices into their acquisitions. By investigating best commercial and government practices, the SEI is helping to overcome challenges and to increase the understanding, maturation, and transition of this technology.

Together, these two series of technical notes will lay down a conceptual foundation for DoD architecture tradeoff analysis and product line business and acquisition practices. Further information is available on the SEI Product Line Systems Program Web page at [http://www.sei.cmu.edu/activities/plp/plp\\_init.html](http://www.sei.cmu.edu/activities/plp/plp_init.html).



---

## **Abstract**

The software architecture of a software-intensive system greatly determines system quality. When used appropriately, software architecture evaluations can have a favorable effect on a delivered or modified government system. This technical note describes the application of the Architecture Tradeoff Analysis Method<sup>SM</sup> (ATAM<sup>SM</sup>) to a major wargaming simulation system. A government-contractor team is developing the Wargame 2000 system at the Joint National Integration Center (formerly known as the Joint National Test Facility [JNTF]), Colorado. In this technical note, we present the contextual background about the software architecture, the organization, and the system being evaluated. Next, we present a general overview of the ATAM process. Finally, we describe the application of the ATAM to the Wargame 2000 system and present important results and benefits. While architecture evaluation is valuable early in the development life cycle, this case study illustrates that such evaluations are also useful when a system is well into development.

---

<sup>SM</sup> Architecture Tradeoff Analysis Method and ATAM are service marks of Carnegie Mellon University.



---

## 1 Introduction

Because software architecture is a major determinant of software quality, it follows that software architecture is critical to the quality of any software-intensive system. For a Department of Defense (DoD) acquisition organization, the ability to evaluate software architectures before they are realized in finished systems can substantially reduce the risk that the delivered systems will not meet their quality goals.

Over the past several years, the Software Engineering Institute (SEI) has developed the Architectural Tradeoff Analysis Method<sup>SM</sup> (ATAM<sup>SM</sup>) and validated its usefulness in practice [Kazman 00]. This method not only permits evaluation of specific architecture quality attributes (e.g., modifiability, performance, security, and reliability) but also allows engineering tradeoffs to be made among possibly conflicting quality goals.

This technical note describes an ATAM evaluation of a sophisticated wargaming simulation system, Wargame 2000. This system is being developed at the Joint National Integration Center (JNIC),<sup>1</sup> Schriever Air Force Base, Colorado. While the ATAM has proven very useful when applied early in the development life cycle, this case study also demonstrates the method's utility even when a system is well into development.

Following this introduction, Section 2 provides background on software architecture, a description of the JNIC, and an overview of the Wargame 2000 system. Section 3 contains an overview of the ATAM including its purpose and primary steps. Section 4 describes how the ATAM was applied specifically to Wargame 2000 and presents some results.

---

<sup>SM</sup> Architecture Tradeoff Analysis Method and ATAM are service marks of Carnegie Mellon University.

<sup>1</sup> The organization was formerly known as the Joint National Test Facility (JNTF).

---

## 2 Context for the Architecture Evaluation

### 2.1 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 decisions affect reliability, modifiability, security, real-time performance, and interoperability goals and requirements. They are the most critical to get right and the most difficult to change downstream in the development life cycle. The right software architecture can pave the way for successful system development. The wrong architecture often results in a system that fails to meet critical requirements and incurs high maintenance costs.

There are many relevant 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. Although they are interested in different components and different relationships, each of their views is valid. Each represents a structure that maps to one of the construction 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 represented by different structural views of the system. These different properties and goals and their corresponding architectural views are important to understand and to analyze. They provide the basis for reasoning about the appropriateness and quality of the architecture.

Some common architectural views include [Clements 96]

- the conceptual (logical) view, which represents system functions, key system abstractions, their dependencies, and data flows
- the module (development) view, which represents the decomposition of the system's functionality. This can include objects, procedures, functions and their relationships.
- the process (coordination) view, which represents processing threads, their synchronization, and data flows

- the physical view, which represents hardware including processors, storage, and external devices or sensors, along with the communications paths that connect them

Other software architectural views are described in *Software Architecture in Practice* [Bass 98].

## **2.2 The Joint National Integration Center**

The Joint National Integration Center (JNIC) is located at Shriever Air Force Base, Colorado. In their own words,

The Joint National Integration Center supports the evolutionary development and incremental fielding of an overarching ballistic missile defense system for the Nation. To do this, the JNIC performs interoperability tests; develops models and simulations; hosts and supports missile defense-related wargames; and provides missile defense exercise support, system-level engineering support, and related analyses.

The JNIC is best described as a government-owned contractor-operated facility. A small government staff representing all three services leads a large group of supporting contractors. The contractors execute their program of work under agreed-upon task orders. Wargame 2000 block developments are covered by specific task orders.

A prerequisite for using evaluative methods in any acquirer-supplier relationship is that the parties agree to perform such an evaluation. The best way to ensure that an evaluation is performed is for the acquirer to specify the evaluation requirements in the original acquisition agreements, which requires awareness of the methods and early planning. However, specifying the evaluation requirements early often makes it difficult to apply helpful technology that becomes apparent after a contract is let.

The JNIC used another approach to accomplish the same objective. The Wargame 2000 program manager has the dual advantage of a task order with great flexibility and an outstanding relationship and shared vision with his supplier. The JNIC acquirer-supplier team was made aware of the benefits of an ATAM evaluation, and they mutually agreed that applying it was “common sense.” Thus, the only constraint was to schedule the ATAM evaluation so as not to interfere with a scheduled wargame.

## **2.3 The Wargame 2000 System**

The Wargame 2000 system is the centerpiece simulation tool supporting the JNIC mission. Wargame 2000 is primarily used to

- examine, develop, and model concepts of operation, doctrine, and Tactics, Techniques and Procedures (TTP) using human-in-control experiments in a simulated combat environment that includes the real world confusion caused by the “fog-of-war”
- serve as a rapid prototyping facility for interoperability studies

Wargame 2000 accomplishes this through realistic representations of threats, sensors, weapons, displays, terrain, weather, logistics, and other factors that affect decision timelines or stress the human-in-control interfaces. Its architecture must be robust, highly reliable, portable across platforms, scalable to handle ever-growing scenario sizes, and modifiable to accommodate different analysis needs and exercise conditions. We will describe the system and its architecture in more detail in Section 4.



---

## 3 The ATAM

### 3.1 Purpose

The purpose of the ATAM is to assess the consequences of architectural decision alternatives in light of quality attribute requirements [Kazman 00]. The major goals of the ATAM are to

- elicit and refine a precise statement of the architecture's driving quality attribute requirements
- elicit and refine a precise statement of the architectural design decisions
- evaluate the architectural design decisions to determine if they satisfactorily address the quality requirements

The method does not need to produce detailed analyses of any measurable quality attribute of a system (such as latency or mean time to failure) to succeed. Instead, success is achieved by identifying trends.

The ATAM achieves these goals by involving a wide group of stakeholders including managers, developers, maintainers, testers, reusers, end users, and customers. It is meant to be a risk mitigation method, a means of detecting areas of potential risk within the architecture of a complex software-intensive system. An evaluation team experienced in software architecture and the method leads this group of stakeholders to ensure that the right questions are asked 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

These areas can be made the focus of future efforts in terms of prototyping, design, and analysis.

### 3.2 ATAM Steps

The heart of the ATAM consists of the following nine steps:

1. **Present the ATAM:** The evaluation team presents a quick overview of the ATAM steps, techniques used, and outputs from the process.
2. **Present the business drivers:** The system manager briefly presents the business drivers and context for the architecture.

3. Present the architecture: The architect presents an overview of the architecture.
4. Identify architectural approaches: Itemize the architectural decisions discovered in the previous step.
5. Generate the quality attribute utility tree: Identify, prioritize, and refine the most important quality attribute goals in a utility tree format.
6. Analyze architectural approaches: Probe the architectural approaches in light of the quality attributes in order to identify risks, sensitivity points, and tradeoffs.
7. Brainstorm and prioritize scenarios: Create and analyze scenarios that represent the various stakeholders' interests to understand quality attribute requirements and their relative importance.
8. Analyze architectural approaches: Continue to identify risks, sensitivity points, and tradeoffs while noting the impact of each scenario on the architectural approaches.
9. Present results: Recapitulate the ATAM steps, outputs, and recommendations.

These steps are typically carried out in two phases.<sup>2</sup> The first phase (Phase 1) is architect-centric and concentrates on eliciting and analyzing architectural information. This phase is run as a miniature version of the ATAM with a small group of technically oriented stakeholders concentrating on Steps 1-6. The second phase (Phase 2) is stakeholder-centric, elicits points of view from a more diverse group of stakeholders, and verifies the results of the first phase. This involves a larger group of stakeholders and builds upon the work of the first phase.

---

<sup>2</sup> The complete methodology also has a planning and preparation phase (Phase 0) and a follow-up and finalization phase (Phase III) that are beyond the scope of this report.

---

## **4 The ATAM Evaluation of Wargame 2000**

### **4.1 Background**

At the time that this ATAM evaluation was conducted (November 2000), the Wargame 2000 system was relatively far along in the development cycle. Thus a primary goal of this ATAM evaluation was to obtain a measure of confidence in the system's software architecture for future block releases. While an ATAM-based evaluation is often useful in this circumstance, its greatest utility occurs earlier in the design process when sufficient architecture definition has occurred, yet it is still early enough to redirect development, if necessary.

The Wargame 2000 system is a highly complex real-time simulation system, and the team could explore only some of the architectural issues during its two-day ATAM evaluation. However after seeing the value of the method, the JNIC team continued to explore additional priority scenarios once the ATAM evaluation was concluded.

### **4.2 Business and Mission Drivers**

One of the most important steps when performing an ATAM-based evaluation is to elicit the business and mission goals driving the development of the architecture. Often stakeholders will use traditional qualities in unconventional ways to describe their business and mission goals. For example, stakeholders described "integrability" in the context of Wargame 2000 as "the ability to easily connect new and dissimilar 'parts' over time." A more traditional definition would address the process of moving from implemented components to a fully functioning system. It also would address the ease with which the parts of a system can be assembled into a whole. While the evaluation team would not change the terms used, it would attempt to capture as much descriptive prose as possible to clarify the qualities being described.

During the first meeting, the evaluation team met with a group of primary stakeholders, mostly senior technical personnel and project management. They listened as the contractor's project manager of Wargame 2000 presented business and mission goals and primary system architecture drivers. Based on this information and subsequent discussions, the evaluation team synthesized the information given here.

As discussed in Section 2, Wargame 2000 puts human operators in a realistic simulated combat command and control environment. In such a simulation, operators are regarded as part of the simulation rather than external to it. This drives the need to have a system with the resolution and fidelity to allow analysts to answer the "Big Five Questions":

1. What did the operator know? This drives the need for realistic displays and complete display contents.
2. When did the operator know it? This drives the need for end-to-end timing, accurate within a second.
3. What did the operator do? This drives the need for a realistic set of controls, rules of engagement (ROE), TTP, decision aids, and command structure coordination.
4. When did the operator do it? This drives the need for deterministic and reliable response to operator input.
5. What effect did their actions have on the battle? This drives the need for detailed information storage and retrieval.

These points are essential issues for any exercise in which Wargame 2000 is used. It is the “mantra” that drives the Wargame 2000 development and operational strategies. To achieve sufficient resolution and fidelity to address these questions, *performance* emerges as *the* prime business/mission driver for Wargame 2000. From JNIC’s perspective, performance has multiple facets:

- The phrase, “We speak in the currency of margin,” was often heard during the ATAM. The term *margin* means how much faster than real time the system can run in a worst-case scenario for any particular exercise. The more margin that can be provided by the system, the more features that can be offered to the customer to support future requirements.
- The system must meet statistical deadlines. Therefore, Wargame 2000 can be described as a “firm” real-time system.
- The Wargame 2000 system must undergo preparation prior to a simulation that typically involves many months. The time it takes to prepare the system for an exercise is referred to as “exercise turnaround time.” The system must be able to meet new exercise turnaround time demands. Starting in 2001, demand was anticipated to increase to one simulation (exercise) a month. Previously, users were running one simulation a year. While this is perhaps not a traditional interpretation of performance, it certainly is an important driver for Wargame 2000.

An additional driver is that Wargame 2000 must support evolving mission needs. This might require adding a new function, representing a new sensor, or modifying the user interface. An important philosophy of the project has been to involve the customer in identifying needs as early as possible. The project evolved the system to meet those needs by first providing basic functionality, then incrementally extending that functionality. This calls for an architecture that is flexible enough to accommodate broad classes of changes.

Other general business and mission drivers were identified. We present these essentially unchanged from original stakeholder wording to illustrate that strict conformance to standard or traditional definitions of quality drivers is not necessary. The important thing is that the terms are meaningful to the stakeholders. These drivers include (in no particular order)

- **Modifiability/Configurability/Composability:** This is the ability to change the system to incorporate new scenarios, entities, and entity behavior; this includes changes in breadth (new entities) and depth (higher fidelity).
- **Integrability:** The system must be able to easily connect new and dissimilar “parts” over time.
- **Flexibility:** The system must be flexible in terms of changing direction by incorporating new requirements and mapping new functionality to an implementation schedule.
- **Interoperability:** Wargame 2000 must be able to connect to and communicate with other simulations. Wargame 2000 must be able to support rapid prototyping for interoperability studies conducted primarily through the use of simulations. To these ends, the system must be compliant with the High Level Architecture Standard (HLA).
- **Scalability:** The system must be able to instantiate models up to the limits of memory and processor capacity.
- **Reliability:** The cost to prepare for and conduct an exercise is very high. As one stakeholder said, “On game day, its got to work.”
- **Maintainability:** Good system documentation is necessary for long-term supportability. Rational Rose has been used; however Rational/Unified Modeling Language (UML) has been shown to be very weak at capturing discrete events.
- **Reuse:** Code reuse itself has not helped at all. Design and knowledge reuse has proven helpful and should be maximized as much as possible.
- **Fidelity:** This refers to the ability to accurately model the details of the objects that give the entities in the simulation some level of realism. This includes modeling the “fog of war,” the uncertainty factor that needs to be part of a realistic simulation.

### **4.3 System Overview**

The SEI ATAM team was given a broad overview of the system during the first phase. Only the high-level details of the system will be provided here.

Figure 1 illustrates the three main subsystems (the boxes) and the data and control paths (the arrows) of Wargame 2000 that were the focus of this ATAM evaluation. The Missile and Air Defense Simulator (MADSIM) is a “human-in-the-loop” control simulator. It provides

- parallel discrete event simulation
- distributed simulation
- mission space system representation

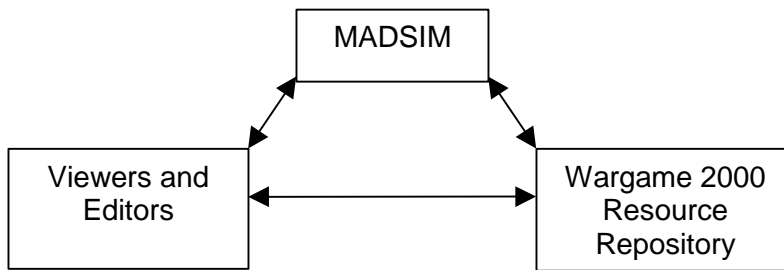


Figure 1: The Primary Wargame 2000 Subsystems

The Concept of Operations (CONOPS) for MADSIM includes

- testing battlefield doctrine
- human-in-control experiments
- simulation of realistic combat environments to include the fog of war

As mentioned in the previous section, the MADSIM must provide the resolution and fidelity required to answer the “Big Five Questions.” The MADSIM subsystem garnered most of the attention of the stakeholders and the evaluation team.

The Viewers and Editors subsystem provides the interface for operators, and the Wargame 2000 Resource Repository acts as a data store for the system.

While there were many driving architectural concerns for Wargame 2000, the system architect’s main concern was meeting system performance demands imposed by a real-time human-in-the-loop simulation. Performance continued to be a recurring theme during the entire ATAM evaluation.

#### 4.4 Architectural Approaches

Within the Wargame 2000 system, several architectural approaches were used by the architect to meet the business and mission drivers. Among them are the following:

- The client-server approach was used to decouple the event generator from the rest of the system. This offered flexibility for upgrading to a more capable event generator. Client-server was also used between the user interface (UI) and the rest of the system, offering more flexibility in changing/upgrading the UI. There was also distributed access to the data repository.
- Parallel discrete event simulation was used as an overall strategy to generate events for the simulation. This approach is common to many simulation systems.
- A publish/subscribe mechanism was used throughout the system as an approach to further decouple various components within the MADSIM subsystem.
- An event-based message passing mechanism was used as an approach to indicate that various simulation events had occurred.

Some supplemental architectural approaches included

- direct access shared memory
- wrapping/facades for some commercial-off-the-shelf (COTS) products
- distributed processing
- an object request broker (ORB) to mediate client-to-server interface

#### 4.5 Utility Tree

The utility tree provides a vehicle for translating the quality attribute goals articulated in the business drivers presentation to “testable” quality attribute scenarios. The utility tree has four levels with the root node labeled “Utility.” Figure 2 graphically illustrates the utility tree concept. Utility is an expression of the overall “goodness” of the system in terms of the higher level nodes. Quality attributes (performance, modifiability, etc.) are placed immediately under Utility. However, to say simply that performance is an important quality is too vague for meaningful analysis. Therefore, each quality has one or more attribute refinements. These reflect the quality attribute-specific stimuli and responses that the architecture must address. For example, performance might be broken down into “minimize worst-case latency for invoice generation,” “maximize average throughput to the authentication server,” and “preserve priority ordering of customer requests.” To further define the notions of quality attributes and attribute refinements, scenarios are used. A scenario is a characterization of the attribute concern that represents a use or modification of the architecture. The scenario is applied not only to determine whether the architecture supports a functional requirement, but also to predict the architecture’s support of system qualities such as performance, reliability, modifiability, and so forth. Scenarios represent a fundamental analysis aid in the ATAM and are the leaves of the utility tree.

Note that a utility tree is not an attempt at defining a rigorous taxonomy of quality attributes. Its purpose is to elicit a definition of system quality requirements in a practical, operational sense that stakeholders can understand. Table 1 summarizes the utility tree developed for Wargame 2000.

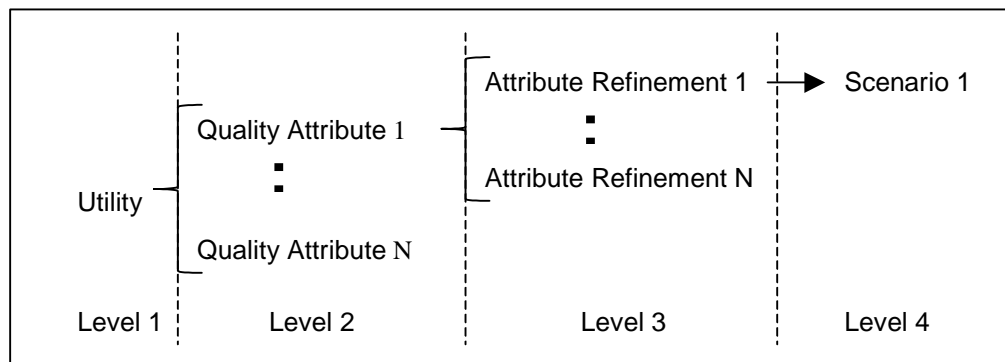


Figure 2: The Utility Tree Concept

Table 1: Wargame 2000 Utility Tree

Quality Attribute	Attribute Refinement	Scenarios
Reliability/Availability	Up when ready to game	Simulation controller initiates simulation execution (a game), starts subordinate processes, loads parameter files, and simulation runs from T=-infinity to T=0 within 10 minutes.
	Stays up during game and test	Runs scenarios without crashes: no hardware and/or software failures. Terminates normally.
	Reasonable response to out-of-plan scenario	Ability to change the location and orientation of assets and still meet the performance, reliability, and credibility criteria.
Performance	Initialize quickly per specification	Must perform all initialization activities within 10 minutes.
	Meet real-time requirements	Run simulations with no instantaneous lags greater than five seconds, no average lags greater than three seconds.
		Run simulations with debug enabled.
Data collector needs to keep up with real time	Finish data collection within 30 seconds of simulation termination.	
Turnaround Time	Set-up Time/Composability	Set-up and test time should be less than two weeks.
	Test time	Set-up and test time should be less than two weeks.
	Analysis Time	Calculate the average depth of penetration of the blue battle space in under four man-hours.
Credibility/Validation	Quality/Correctness	Increase model fidelity of interceptors and sensors to match EADSIM major time-line events within 10%.
Accommodate New/Changed Requirements	From game-to-game	Change joint firing doctrine rule set in the automated weapon target assigner in time for the next game.
	From build-to-build	Add cruise missile infrastructure into next software release.
Flexibility	Reconfigurability	Reconfigure Wargame 2000 from running a theater ballistic missile defense simulation to a national missile defense simulation; changing hardware and software in one week. <sup>3</sup>
Maintainability	Replace existing database	Replace existing database with an object-oriented database within 5-10 staff years.

<sup>3</sup> Since this ATAM, defense policy has eliminated the distinction between theater and national missile defense.



## 4.6 Scenario Generation and Prioritization

The scenario elicitation process allowed stakeholders to contribute scenarios that reflected their concerns and understanding of how the architecture would accommodate their needs. In practice, a particular scenario may have implications for many stakeholders: for a modification, one stakeholder may be concerned with the difficulty of a change and its impact on performance, while another may be interested in the cost of the change. Scenarios were collected by a round-robin brainstorming activity with all of the JNIC stakeholders.

During the Wargame 2000 ATAM evaluation, 31 scenarios were generated. After brainstorming scenarios, stakeholders consolidated similar scenarios to prevent dilution of votes during prioritization. After the consolidation process, 29 scenarios remained. These scenarios were prioritized and eight scenarios emerged that were thought to have the greatest impact. Table 2 lists the final set of consolidated, edited, and prioritized scenarios that formed the basis for subsequent analysis.

Table 2: *Prioritized Wargame 2000 Scenarios*

Priority	Scenarios
1	Build the graphical parameter file front end. (This graphical set-up tool allows the user to click on an entity on the screen and change its characteristics). The tool provides the ability to change the characteristics of a sensor or other simulation entity, such as radar aperture or loop gain of a radar.
2	Apply heavy data throughput (objects, tracks), load to the graphical user interface (GUI) workstation, and be able to analyze the performance.
3	Run an entire simulation without crashes; no hardware or software failures; terminates normally.
4	Simulation set-up and test time is less than two weeks.
5	A pause in the simulation occurs signifying a timing problem. Provide the ability to identify the source(s) of the problem.
6	Provides a virtual gaming site that includes online, offsite, and multiple games running simultaneously and that allows participants to play from their own command centers.
7	A new post-processor is added that produces a summary report. Be able to predict the impact to the system's performance.
8	Be able to make a change to the simulation entity base class and be able to understand the effect/impact on the other objects.

## 4.7 Overview of the Analysis Process

In the second phase, scenarios are prioritized and the ATAM evaluation team facilitates the analysis of the scenarios. Scenarios are analyzed in detail by walking through each one to evaluate its effects on the architecture. The ATAM evaluation team facilitates the ensuing stakeholder discussion to surface architecturally based risks, sensitivities, and tradeoffs. The stakeholders contribute to the analysis by discussing issues regarding the architecture from their points of view.

The examples of analysis that follow are typical of the kind of analysis that will occur for each scenario that is generated during an ATAM-based evaluation. These examples will illustrate how scenarios feed analysis, which then identifies risks, sensitivity points, and tradeoffs. However, ATAM-based evaluations also have other benefits that are not explicitly part of the process but are side benefits. A few of these side benefits will be illustrated as well.

## **4.8 Scenario Analysis**

### **4.8.1 Scenario 1**

We will first present the analysis of the highest priority scenario to illustrate the analysis and discovery process that ATAM-based evaluations provide. Recall Scenario 1 from Table 2:

Build the graphical parameter file front end. (This graphical set-up tool allows the user to click on an entity on the screen and change its characteristics). The tool provides the ability to change the characteristics of a sensor or other simulation entity, such as radar aperture or loop gain of a radar.

Parameter files were created to allow generic objects in the simulation to be instantiated into specific objects during initialization. This approach permitted maximum flexibility in configuring simulations. This supports the following business goal given in Section 4.2:

Modifiability/Configurability/Composability: the ability to change the system to incorporate new scenarios, entities and entity behavior, described as breadth (new entities) and depth (higher fidelity)

The architectural choice to use text-based parameter files to meet this business goal eventually led to tremendous complexity. A risk that emerged from the ATAM was that there was no capability to configure parameter files easily and consistently. This risk directly impacted the following business goal (paraphrased from Section 4.2):

The system must be able to meet new “exercise turnaround time” demands, including an anticipated frequency of one exercise per month.

The architectural tradeoff is between initialization speed and configuration complexity. This tradeoff is critical since text-based parameter files allow the system to be initialized very quickly in preparation for a simulation. However, the tradeoff is that parameter files affect the ability to quickly turn the system around for the next simulation impacting the above business goal. Again, this tradeoff was made explicit to all stakeholders and was documented during the ATAM evaluation.

### **4.8.2 Scenario 6**

Another scenario involved interoperability. Recall Scenario 6 from Table 2:

Provide a virtual gaming site that includes online, offsite, and multiple games running simultaneously and that allows participants to play from their own command centers.

In order to facilitate interoperability with other simulations, compliance with the High Level Architecture (HLA) was a requirement for Wargame 2000. By our definition of architecture in Section 2, HLA is perhaps better described as a set of standards to facilitate simulation reuse and interoperability. The HLA was developed under the leadership of the Defense Modeling and Simulation Office (DMSO) to support reuse and interoperability across the large numbers of different types of simulations developed and maintained by the DoD. HLA compliance in Wargame 2000 is achieved via an HLA gateway to interface with other systems that are HLA compliant.

While this is a good idea in principle, a risk emerged due to this requirement. The system architect expressed concerns about the HLA real-time interface and indicated that it would not be able to meet the user's performance expectations. The broader lesson is that while an architectural standard may intend to promote a quality such as interoperability, a standard alone cannot guarantee qualities. In this case, a mandate to achieve interoperability alone was not enough, and interoperability for Wargame 2000 had a performance constraint not addressed by the standard. While the HLA standard may define an interface for interoperability, it says very little about other qualities (such as performance) that may be essential for correct interoperability. Moreover, it is the particular implementation of HLA that ultimately determines system performance, not necessarily the standard itself. Some implementations may be efficient; some may not. Although implementing this standard in the system may have been mandated, it proved to be a significant risk to the following business goal:

Interoperability: Wargame 2000 must be able to connect to other simulations. Wargame 2000 must be able to support rapid prototyping for interoperability studies conducted primarily through the use of simulations. To these ends, the system must be compliant with the High Level Architecture Standard (HLA).

### **4.9 Sensitivity Points**

ATAM also includes the discovery of sensitivity points during scenario analysis. A sensitivity point is an architectural decision that has a strong effect on a particular quality attribute. Scenario evaluations uncovered two sensitivity points. The first sensitivity point follows:

*Performance is sensitive to careful adherence to implementation development guidelines.*

Performance was an issue of great concern for the system architect. Not only was it the architect's goal to have enough capacity to meet the functional and performance requirements on the day of delivery, there had to be enough reserve capacity to support future requirements. Experience proved that the simulation could suffer from poor performance due to violation of various coding guidelines. For example, initializing simulated sensors (e.g., radars) all at the same time during system start-up drives performance down. Hence, there is a coding guideline to use random number generation to determine sensor initialization time.

The system architect indicated the need to present these guidelines to the developers every six months or so to reeducate developers and bring new engineers on line. Sometimes, this task (the presentation) was not accomplished due to other mission pressures and performance problems. Violations of these guidelines were discovered during test.

The second sensitivity point uncovered during the ATAM evaluation could potentially affect system performance:

*Increasing reliability comes at a cost of performance.*

There is an ambiguously stated requirement that the system must not fail during a game (simulation exercise). There were quantitative requirements directly addressing reliability or availability in the documents reviewed or in the presentations made during the ATAM. One approach that was used to provide a higher degree of reliability included the use of monitors. In some cases, a monitor was a human operator checking displays. In other cases, software monitors were used to check the status of various components. Software monitors used techniques like "heart-beat monitoring," checking to ensure that tasks were running and ensuring that various components were available. The sensitivity point emerged. While these techniques were fairly easy to implement in the system to provide higher reliability, they required committing resources, particularly in network usage, thereby affecting performance. While this sensitivity point may not be enlightening from a purely technical standpoint, it was not obvious to all stakeholders. To one group of stakeholders, the impact of requesting more reliability via monitors was not obvious until the ramifications were illuminated through the ATAM exercise. This is a case where the ATAM exercise broadened communication among stakeholders.

#### **4.10 Other Risks**

While ATAM is a method that is intended to uncover technical risks, sensitivity points, and tradeoffs, it often uncovers programmatic issues as well. For example, there were architectural issues with the data interface used to change, update, and create parameter files. Additionally, underlying programmatic risks were uncovered during the ATAM evaluation involving the configuration management of parameter files. A parameter file is a type of configuration file for Wargame 2000 used during system initialization. The original motivation for this requirement was to allow a single parameter file change to be propagated to all of the places that it affects rather than having to make multiple changes, in multiple

places, using different methods. However, discussions with stakeholders revealed a larger issue related to the ease of use of the interface to manage parameter files and the need to better manage the change process for initialization parameters. Parameter file changes are currently made by a combined manual and automated process. There is a tool for generating parameter files that is neither documented nor user friendly. Some changes are made manually, often in multiple places in the individual parameter files. The current configuration management process does not track and coordinate the manual and automated changes effectively.

Another programmatic risk was related to the creation of a graphical tool to create, update, change, and manage the parameter files. The task to create this tool was consistently deferred and generally received a low priority. One of the benefits of an ATAM evaluation is increased communication among stakeholders. During the ATAM evaluation, the importance of effectively managing parameter files and the necessity of a tool to support this became clear and a priority for all stakeholders.

ATAM also had another positive benefit in the area of architectural representation. In the course of the architectural presentation and subsequent scenario analysis, several detailed architectural discussions ensued. It was clear that some of the architectural drawings were at the source of some stakeholder confusion. To refine the architectural representation, the ATAM team worked with the architect to help clarify some of the architectural drawings and diagrams. Some of these diagrams were created during the session. Other diagrams amplified and updated existing Wargame 2000 design documentation. While advice and suggestions were made to improve the Wargame 2000 architectural design documentation, this is not a primary focus of an ATAM. Clarifying these artifacts improved communications among stakeholders. During this exercise, a possible change to the existing architecture was discovered that could reduce the amount of traffic on the network. In discussions with JNIC management after the ATAM evaluation, JNIC management indicated that the improved representation artifacts were very beneficial.

#### **4.11 Post-ATAM Activities**

As previously noted, there is often not time during a two-day ATAM evaluation to examine all the scenarios of potential interest. Thus, a typical recommendation is that the organization continue the analysis process after the ATAM evaluation. The JNIC team took this recommendation to heart. All the priority issues raised were incorporated into the project management process and are either now closed or have been included as items to be addressed within current and future task orders. Furthermore, the SEI is working to transition a self-sustaining ATAM capability to JNIC so that it can become a routine part of the JNIC development process.

---

## 5 Conclusion

In this technical note, we have discussed applying the ATAM during the development of a large government-sponsored simulation system. The note presents a general overview of the ATAM process and the results of this ATAM evaluation. It also presents benefits that both the acquirer and developer received.

A post-evaluation survey of the participants showed that the JNIC ATAM evaluation was considered a success. A number of specific benefits were reported:

- The stated goals of the ATAM evaluation were met.
- Nearly all participants learned of risks of which they were previously unaware.
- Nearly all participants said they would be able to act directly on evaluation results.
- There was very open communication among all stakeholders, including government and contractor.
- The evaluation allowed a focus on the entire system rather than narrow or short-term concerns.

Lessons applicable to ATAM evaluations in general were also uncovered:

- While an ATAM evaluation is often promoted as appropriate for use early in the development life cycle, it can also have value when a system is well into development. Additional specific benefits illustrated by the JNIC ATAM evaluation include
  - improved architectural documentation
  - new understanding of the role architecture can serve to improve stakeholder communications
  - improved understanding of architectural issues for future versions of the system
- An ATAM evaluation can be successfully applied in a government-owned contractor-operated environment. Two important factors leading to success were the flexibility of the task order contract and the excellent relationship between the government and the contractor.
- Even though there is typically not enough time to analyze all scenarios during a two-day ATAM evaluation, it is possible for the participants to continue the analysis without the coaching of the evaluation team.

The JNIC and the SEI have had a long-standing strategic collaboration to apply emerging software technologies. The JNIC provides an excellent example of how a government organization can incorporate these technologies to solve real problems and improve their mission effectiveness.

---

## References

- [Bass 98]** Bass, L.; Clements, P.; & Kazman, R. *Software Architecture in Practice*. Reading, Massachusetts: Addison Wesley, 1998.
- [Clements 96]** Clements, P. & Northrop, L. *Software Architecture: An Executive Overview* (CMU/SEI-96-TR-003, ADA 305470). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, February 1996. Available WWW: <<http://www.sei.cmu.edu/publications/documents/96.reports/96.tr.003.html>>.
- [Kazman 00]** Kazman, R.; Klein, M.; & Clements, P. *ATAM: Method for Architecture Evaluation* (CMU/SEI-2000-TR-004, ADA382629). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2000. Available WWW: <<http://www.sei.cmu.edu/publications/documents/00.reports/00tr004.html>>.



<b>REPORT DOCUMENTATION PAGE</b>			<i>Form Approved</i> <i>OMB No. 0704-0188</i>	
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 February 2002		3. REPORT TYPE AND DATES COVERED Final
4. TITLE AND SUBTITLE Using the Architecture Tradeoff Analysis Method to Evaluate a Wargame Simulation System: A Case Study			5. FUNDING NUMBERS F19628-00-C-0003	
6. AUTHOR(S) Lawrence G. Jones, Anthony J. Lattanze				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213			8. PERFORMING ORGANIZATION REPORT NUMBER CMU/SEI-2001-TN-022	
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 software architecture of a software-intensive system greatly determines system quality. When used appropriately, software architecture evaluations can have a favorable effect on a delivered or modified government system. This technical note describes the application of the Architecture Tradeoff Analysis Method <sup>SM</sup> (ATAM <sup>SM</sup> ) to a major wargaming simulation system. A government-contractor team is developing the Wargame 2000 system at the Joint National Integration Center (formerly known as the Joint National Test Facility [JNTF]), Colorado. In this technical note, we present the contextual background about the software architecture, the organization, and the system being evaluated. Next, we present a general overview of the ATAM process. Finally, we describe the application of the ATAM to the Wargame 2000 system and present important results and benefits. While architecture evaluation is valuable early in the development life cycle, this case study illustrates that such evaluations are also useful when a system is well into development.				
14. SUBJECT TERMS Architecture Tradeoff Analysis Method <sup>SM</sup> (ATAM <sup>SM</sup> ), software architecture evaluation, wargame simulation			15. NUMBER OF PAGES 33	
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	