A Paper submitted to the
10th International Command and Control Research and Technology Symposium
"The Future of C2"

Architecture Modeling Approach for Net-Centric Enterprise Services

(C4ISR/C2 Architecture Track)

Yun-Tung Lau, Ph.D.*, Michelle "TJ" King
Science Applications International Corporation (SAIC)
5113 Leesburg Pike, Suite 200
McLean, VA 22041

Walter J. Okon and David Kye
System Engineering Division
Net-Centric Enterprise Service Program Office
Defense Information Systems Agency (DISA)
5275 Leesburg Pike
Falls Church, VA 22041

*POC: Phone: (703) 824-5817
Fax: (703) 824-5836
E-mail: Yun-Tung.Lau@saic.com

Report Documentation Page			*Form Approved OMB No. 0704-0188*

1. REPORT DATE **2005**	2. REPORT TYPE	3. DATES COVERED **00-00-2005 to 00-00-2005**
4. TITLE AND SUBTITLE **Architecture Modeling Approach for Net-Centric Enterprise Services**		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Science Applications International Corporation (SAIC),5113 Leesburg Pike Suite 200,McLean,VA,22041**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR’S ACRONYM(S)
		11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**16**	

Abstract

This paper presents an architecture modeling approach for service-oriented architectures such as the Net-Centric Enterprise Services (NCES). The approach is driven by operational mission threads. It uses Unified Modeling Language and the Department of Defense Architecture Framework to capture, analyze, and present the architecture products. Steps in this approach include:

1. Formulating activity models for a mission thread.

2. Mapping the activities to NCES and existing systems.

3. Developing logical deployment architecture with NCES included.

4. Developing logical data models.

5. Constructing executable architecture models.

This architecture development approach has been applied to NCES mission threads, which cover a wide range of activities in the Warfighting, Intelligence, and Business domains. It provides a direct trace from NCES capabilities to operational requirements and shows how NCES will support various communities of interest. We illustrate the approach using mission threads that are closely related to Command and Control. Examples include Time-Sensitive Targeting, Joint Close Air Support, and Global Strike.

Table of Contents

1 INTRODUCTION 4

2 ARCHITECTURE MODELING APPROACH 5

3 EXAMPLE ANALYSIS RESULTS 6

4 EXECUTABLE ARCHITECTURE MODEL 12

5 SUMMARY 14

6 REFERENCES AND ACRONYMS 15

1 Introduction

The Global Information Grid (GIG) is the globally interconnected, secured end-to-end set of information capabilities, associated processes, and personnel for collecting, processing, storing, disseminating, and managing information on demand to warfighters, policy makers, and support personnel [1].

The GIG as a transformational vision aims at achieving information superiority in a network-centric environment. It enables various systems to interoperate with each other. For the warfighters, it brings "power to the edge" through a Task, Post, Process, Use (TPPU) process [2]. For the business and intelligence communities, it provides the infrastructure for effective information gathering and collaborative operation.

Net-Centric Enterprise Services (NCES) provides a set of Core Enterprise Services (CES) on the GIG to support operational missions conducted by various communities of interest (CoI) in the warfighting, business, and intelligence domains. NCES is built on a service-oriented architecture (SOA), which enables distributed, parallel information sharing, and dynamic collaboration on a ubiquitous network.

In an SOA, a set of loosely coupled services works together seamlessly and securely over a network to provide functionalities to end users. These services have well defined interface definitions. Supported by service management tools at the enterprise level, they are published, discovered, mediated, and consumed in an orderly fashion.

As part of the NCES architecture development, a number of operational mission threads were analyzed. The analyses show the end-to-end traceability from operational activities, to system functions and CES. They ensure that the NCES architecture does fulfill the needs of the domain and CoI users and provide value to them.

This paper presents an architecture modeling approach for NCES mission threads. The approach uses Unified Modeling Language (UML) [3] within the Department of Defense Architecture Framework (DoDAF) [4] to capture, analyze, and present the architecture products. We have also made adaptations to the SOA paradigm (e.g. from “systems” to “services”) [5].

In the following sections, we first give an overview of steps in our architecture modeling approach. Then we walk through the process for a sample mission thread and discuss the executable architecture model (EAM). Finally, we provide a summary of the approach and results.

2 Architecture Modeling Approach

The architecture modeling approach for NCES mission threads consists of the following steps:

1. Starting with the mission thread descriptions formulated from Joint Doctrines, Concept of Operations, and other similar guiding documents, we formulate the activity models (OV-5, OV-6c) for the mission threads or operational scenarios.
2. After validating the activity models with subject matter experts, we map the activities to existing systems and Core Enterprise Services (CES) (SV-5).
3. Next we develop the logical deployment architecture (SV-1) with CES included. The logical architecture may be presented at several levels of detail. Physical or network architecture (SV-2) may also be identified as appropriate. Sequence diagrams (SV-10c) may be used to describe interaction between services or systems in the architecture.
4. The logical data model (OV-7) provides useful information on data flow. It is also important in applying the DoD Net-Centric Data Strategy [6] to CoI services. If the system data exchange matrix (SV-6) and data schema (SV-11) are known, we can derive further estimates on the data sizes and data transmission latency.
5. Finally, we combine all of the above architecture information to construct the executable architecture model (EAM). The EAM is constructed based on the activity model (OV-5), with additional details for data flow, sequencing and timing. One may use discrete event simulation techniques to implement and execute the EAM. The EAM validates the architecture logic and provides quantitative information such as timing data for the end-to-end process.

The results of the EAM can be used as inputs for further modeling and simulation efforts that include actual network traffic measurements to produce performance parameters (SV-7) for NCES. Figure 1 shows the above steps in architecture modeling for the NCES mission threads.

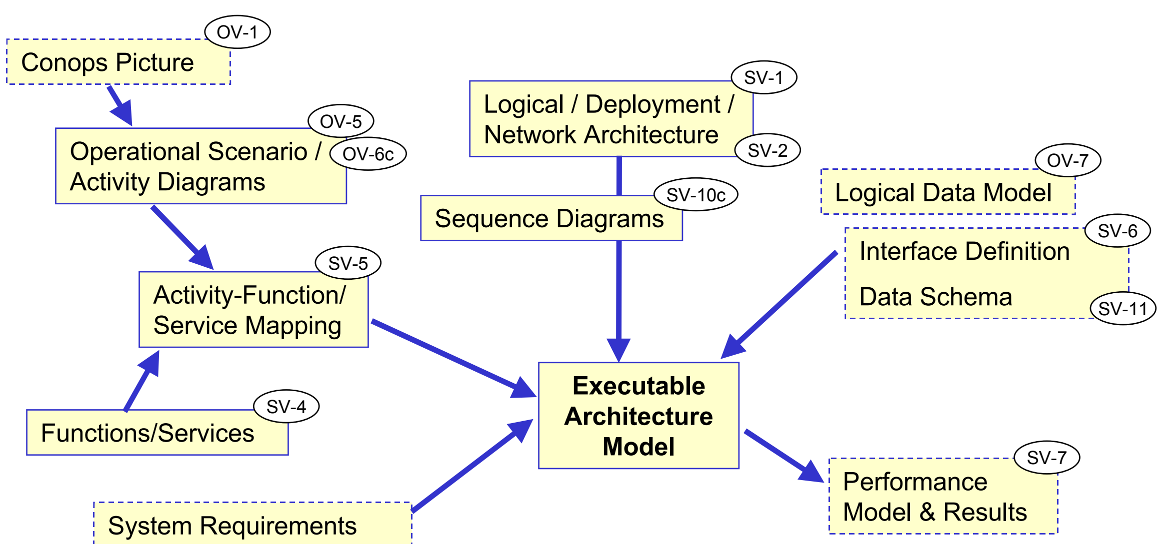

Figure 1 – Architecture Modeling Methodology for NCES Mission Threads

Note that for an SOA, we extend the traditional system data exchange matrix (SV-6) to show data exchange between service consumers and providers. Thus SV-6 is associated with service interface definition. The functional decomposition in the SV-4 gives the functions supported by the CES.

In the next section, we walk through this analysis process for a sample mission thread.

3 Example Analysis Results

Here we use the Time-Sensitive Targeting (TST) mission thread as an example to present our architecture modeling methodology. Time-sensitive targets are targets that pose (or soon will pose) a clear and present danger to friendly forces or are highly lucrative, fleeting targets of opportunity that the Joint Force Command (JFC) has designated as requiring expedited response [7, 8, 9]. Prosecution and destruction of these targets or rendering them harmless represents significant combat capability to help achieve the Joint Force Command's objectives.

The TST Mission Thread is broken into six major phases: Find, Fix, Track, Target, Engage, and Assess, as shown on the left of Figure 2. For each phase we further break it down to activities. The corresponding UML activity diagram for all phases is given on the right of Figure 2. We note that whereas the details of the activities in Figure 2 may not be discernable, it does contain features such as start point, end point, decision branch, loop back, and parallel activities.

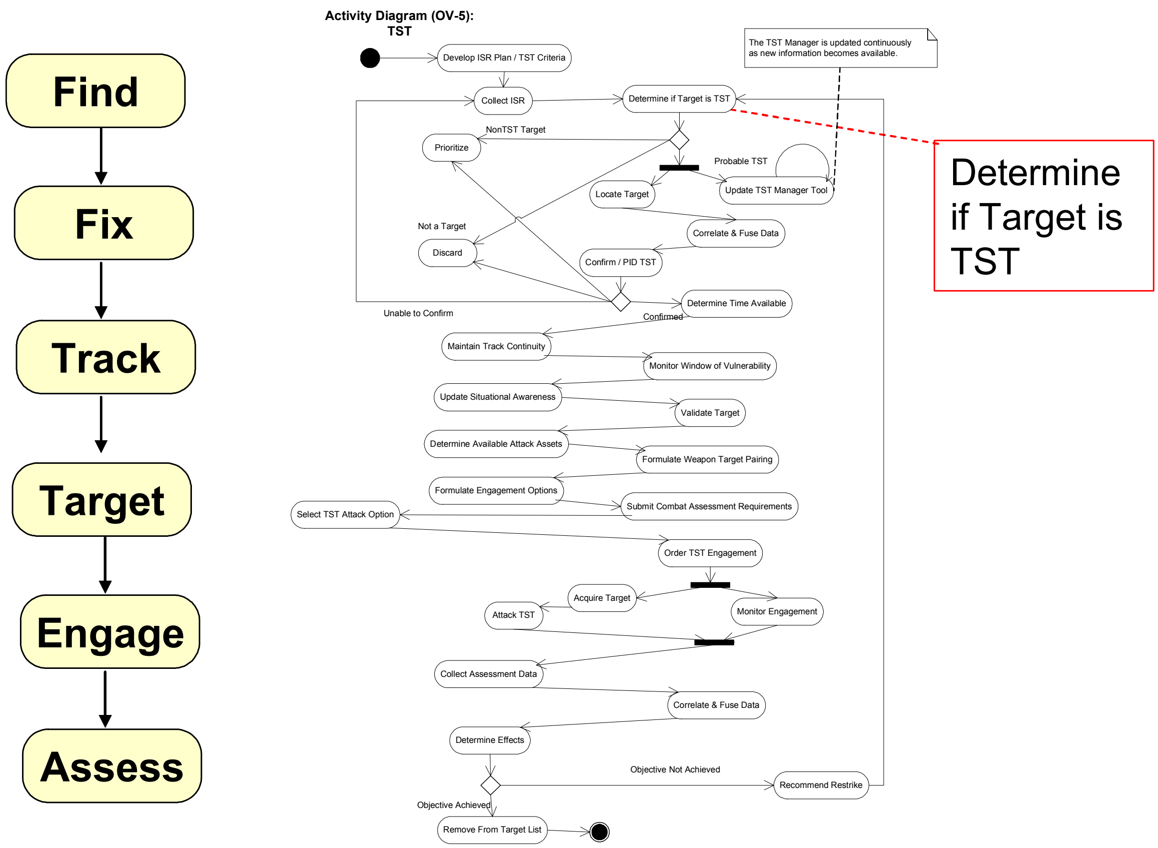

Figure 2 – Operational Activity Diagram (OV-5) for TST

As a supplement to the activity diagram, we also develop a list of description for the activities. An example is shown in Table 1 for the first three activities in the "Find" Phase.

Activity	**Description**
Develop ISR Plan/TST Criteria	Based on JFC TST guidance and initial preparation of battlespace (IPB), develop an Intelligence, Surveillance and Reconnaissance (ISR) plan to deploy sensors to identify potential TSTs.
Collect ISR	Use all available sensor data to identify, locate, and prioritize emerging targets.
Determine if Target is TST	Use further ISR to determine if emerging target is a TST. The final decision is made by the TST Cell Chief.

Table 1 – Example Operational Activity Descriptions ("Find" Phase)

If desired, one may further develop sequence diagrams for some of these activities to understand how the actors interact with certain operational nodes. The activity diagrams, activity descriptions, and sequence diagrams together constitute a full activity model, which should be reviewed and validated with subject matter experts (TST experts, in this case). This ensures that the key steps in the model are captured correctly.

After validating the activity model, we map the operational activities to the Core Enterprise Services (CES) provided by NCES and to other systems (such as CoI-specific applications). This mapping produces the Operational Activity to System Function Traceability Matrix (SV-5). An excerpt of the matrix is given in Table 2. Note that CES that work in the background or apply across all operational activities appear as grayed out rows. Examples are Configuration and Administration for various services.

Activities to Function Mapping for Time-Sensitive Targeting (SV-5)	FIND Phase	Develop ISR Plan/TST Criteria	Collect ISR	Determine if Target is TST	Fix Target	Discard	Prioritize	FIX Phase	Locate Target	Correlate and Fuse Data	Confirm /PID TST	Determine time available	TRACK Phase	Maintain Track Continuity on the TST	Monitor Window of Vulnerability	Update Situational Awareness	TARGET Phase	Validate Target	Determine Available Attack Assets	Match Available Attack Assets	Formulate Weapon Target Pairing	Formulate Engagement Options	Submit Combat Assessment Requirement	Select TST Attack Option	ENGAGE Phase	Order TST Engagement	Monitor Engagement	Attack TST	ASSESS Phase	Collect Assessment Data	Determine Effects	Correlate and Fuse Data	Retarget	Remove From Target List
		Legend: P=Portlets via Browser, C=Client App (Thick), S=Service-to-Service, M=System-to-Service. Cells in gray are common across Operational Activities.																																
Human Actor Only																												X						
Human Computer Interface												X			X																			
Core Enterprise Services																																		
Messaging Services																																		
Publish/Subscribe/Receive Enterprise Messages		S												S		S		S					S	S							P			
Publish/Subscribe/Receive Alerts by Topic		P	P	P	P	P	P				P					P							P	P		P				P				
Publish/Subscribe/Receive Notifications		P	P	P		P	P				P					P		P					P	P		P				P				
Configuration & Admin																																		
Discovery Services																																		
Content Discovery		P	M	P	P	P	P		M																		S							
Service Discovery		P	M						M																		S							
Metadata Discovery																																		
Person Discovery			S															S	S	S	S	S	S	S										
Taxonomy/Ontology Mgmt																																		
Metadata/Schema Reg & Mgmt																																		
Collaboration Services																																		
Establish Collaboration Session		P	P	P							P				P	P		P	P	P	P	P	P	P		P				P			P	
Publish/Subscribe/Receive Presence Info		P	P	P							P					P		P	P	P	P	P	P	P		P				P			P	

Table 2 – Operational Activity to System Function Traceability Matrix (SV-5) for TST

The SV-5 given in Table 2 serves two purposes. First the mapping shows explicitly which CES or systems support which operational activities. For instance, in TST, DoD Enterprise Collaboration services such as instant messaging (IM), text chat, and video over IP will enable the TST CoI to rapidly resolve conflicts and restrictions to the prosecution of TSTs. The ability to publish information (via the enterprise service bus of the Messaging service) will allow rapid horizontal and vertical dissemination of TST data to all interested parties. Mediation services will enable the transformation of data into standardized formats and aggregate information from disparate sources. Similarly, one may use the SV-5 to perform a gap analysis to identify unsupported operational activities.

The second purpose of the SV-5 is to help identify potential interactions or data exchanges between services and systems. These interactions are performed via service or system interfaces. This leads us to the logical architecture, which is depicted by the System Interface Description (SV-1). For a Service-Oriented Architecture, the System Interface Description shows how the service consumers and providers connect to each other to support the operational missions.

Figure 3 shows a notional System Interface Description for TST. This diagram depicts only a few of the many connections between the TST CoI services, the CoI Portal, and the CES. For instance, users may utilize the Collaboration service to resolve deconfliction issues in real time. Remote data providers may use the Discovery service to register their data for federated search. Additionally, remote systems such as a Track Manager, or a Restricted Target List (RTL) data provider can use the Messaging service to automatically populate the data server of a TST Manager. Note that for simplicity, Figure 3 shows only those connections with services that have directly interfaces with end users (customer facing services). We have omitted those connections with foundational services that work mostly in the background (such as IA/Security service).

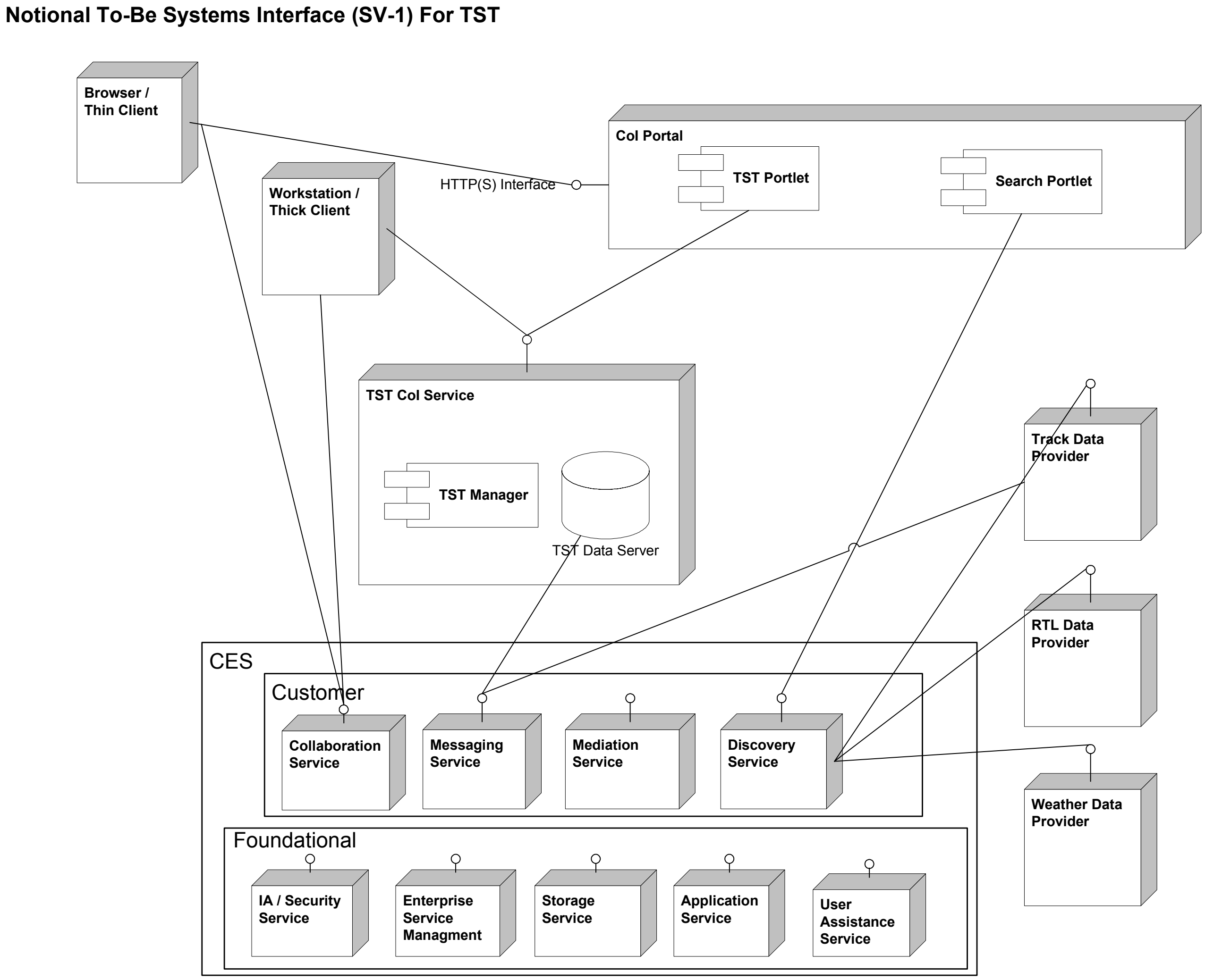

Figure 3 - Notional System Interface Description (SV-1) or Logical Architecture for TST

To see how the logical architecture operates, one may also develop Systems Event-Trace Description (SV-10c) in the form of sequence diagrams for various scenarios of a mission thread. Figure 4 shows the scenario of a Component Commander who has forces in the vicinity of a TST that is being targeted. Using the Discovery service and Messaging service, the Commander learns about the situation and locates the TST Cell Chief to coordinate deconfliction. Details of the steps in the figure are given below.

1. Updated TST data, such as geographic location, time available, etc., are sent via Messaging service to the Track Data Provider.

2. The Messaging service notifies the Discovery service that TST data have been updated.
3. The Discovery service finds a user profile which matches the new data, and triggers the Messaging service to send a notification to the user. The user is a Component Commander with forces in the immediate vicinity of the TST.
4. The Messaging service sends a notification to the Component Commander.
5. The Component Commander retrieves the message.
6. The Component Commander initiates a search for the detailed TST information.
7. When the information is returned, the Component Commander initiates a search for the TST Cell Chief in order to perform target deconfliction.

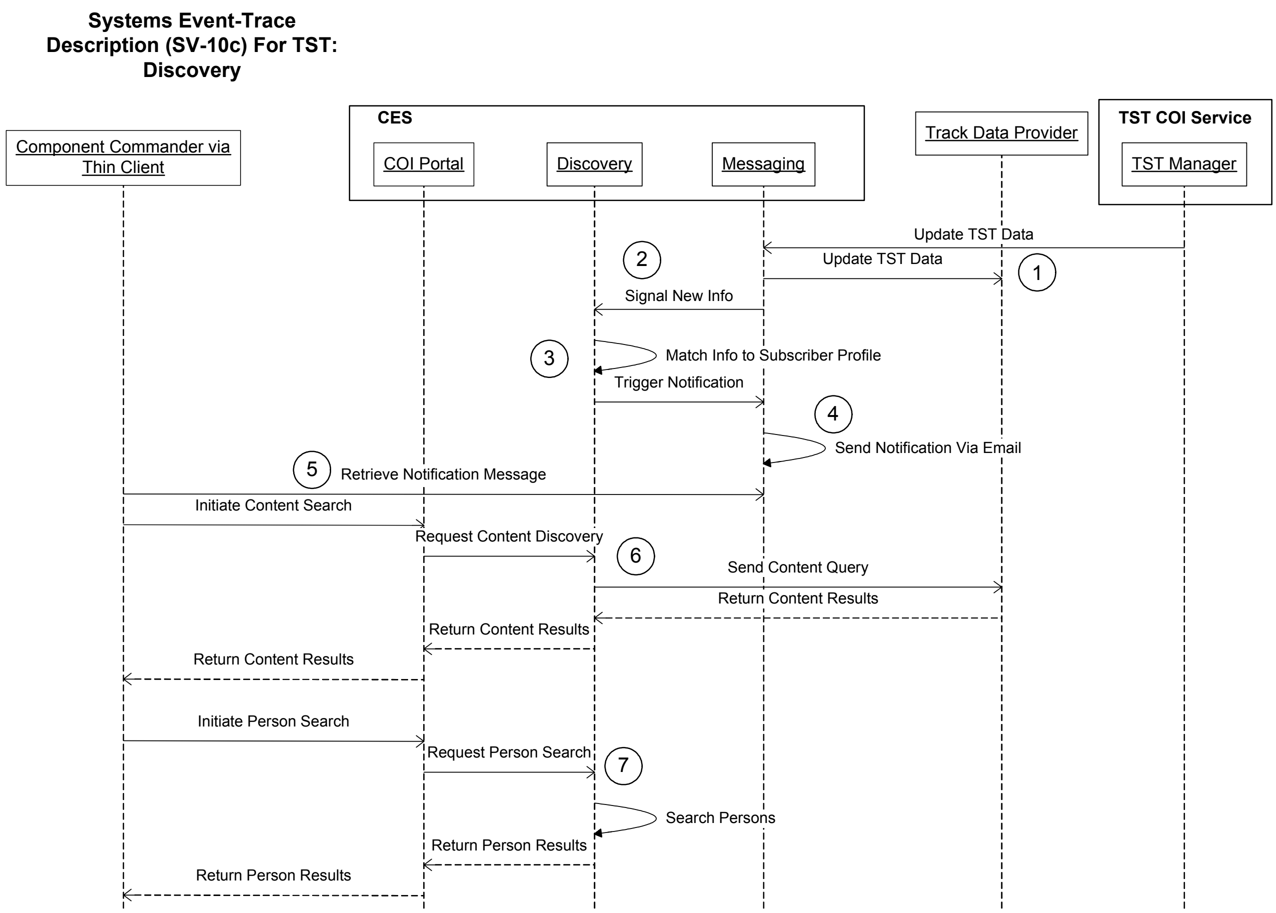

Figure 4 – Notional Systems Event-Trace Description (SV-10c) for TST

All interactions supported by the SV-1 involved data exchanges. For structured data, the logical data model (OV-7) describes the key data types used by the relevant CoI. In a net-centric environment, these data types may be exposed to authorized users of other communities on the GIG.

For the TST CoI, the central object or data type is Target (Figure 5), which is created with some initial attributes populated during the Find Phase. ISR collection requests and results are captured by the Collection Request object, which is linked to the main Target object. The Coordination Data type contains the status of coordination (in review, approved, denied, etc.) across components. The Weather Info object contains pertinent details about the weather affecting the prosecution of the Target.

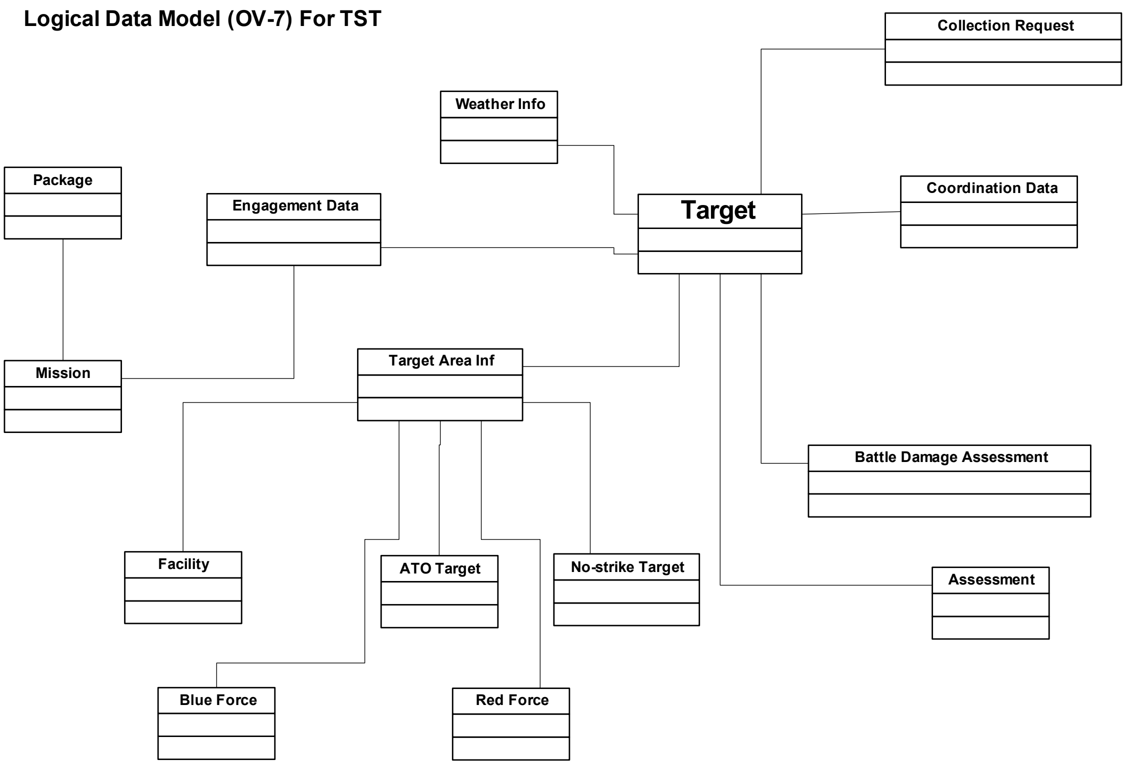

Figure 5 – Logical Data Model (OV-7) for TST

Each Target has Target Area Information, which contains information on Facilities, Air Tasking Order (ATO) Targets, No-Strike and Restricted targets, Blue and Red Forces, etc. within a certain radius of a Target.

During the Target phase, a Target will be paired up with a weapon (Engagement Data), which is associated with a Mission, which is grouped with other missions into Packages.

The Battle Damage Assessment (BDA) and Assessment data types contain information about the results of attacking the Target.

The logical data model is also useful for defining the system data exchanges (SV-6) and physical data schema (SV-11). All of the above architecture views eventually contribute to the Executable Architecture Model, which we discuss next.

4 Executable Architecture Model

The Executable Architecture Model (EAM) provides an end-to-end executable approach to validating the architecture logic. The EAM closely follows the Operational Activity Model (OV-5) with additional details for data flow, sequencing and timing. Because of the similarities, here we only show an excerpt of the model and a sampling of the execution results.

Figure 6 shows the first three activities of the EAM for TST. It is the same as the activity diagram in Figure 2, except that the input and output data objects are indicated. The execution of the EAM is simply activating the activities in order. The activity "Determine if Target is TST" will be triggered only when the TST Data is available. Similarly, other activities in the EAM may be activated only when certain triggering data are present. The EAM also contains activities that may occur in parallel.

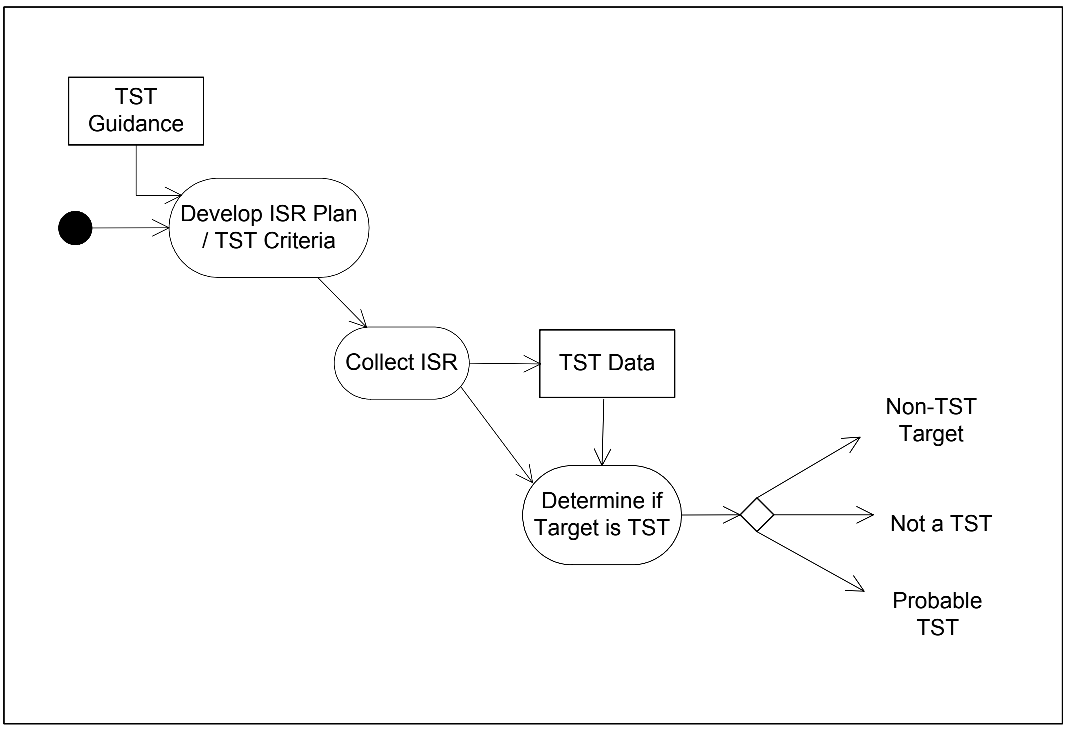

Figure 6 – The first three activities of the Executable Architecture Model for TST

Each activity has certain associated parameters, such as timing. The timing data are based on inputs from the subject matter experts and the best judgment of the model designers. For those activities that are fairly regular, a single time is assigned. For activities that may vary greatly in their duration, a distribution can be used. As a result, each execution of the EAM may result in different time duration.

Figure 7 gives the result of a sample run of the TST EAM. It represents a smooth or sequential execution of the TST process with no returns to previous steps. The total execution time was 1855 seconds, or 31 minutes.

Each row in Figure 7 corresponds to an activity in the OV-5. The length of the bar in a row indicates the duration of the activity. The timescale shown on top is in seconds. The shaded bars represent a group of activities, which appear below those bars.

Figure 7 – A Sample Run for the TST Executable Architecture Model

In general, the EAM can provide different timing results depending on the number of iterations and loops in the model, and the probability of different branches. One may also use the EAM to study the effects of automating manual steps with services. Typically, automated services (possibly with workflow support) allow functions to complete more quickly or in parallel, thereby shortening the end-to-end execution time.

5 Summary

In this paper, we have presented an end-to-end architecture modeling approach for Service-Oriented Architectures. In addition to the DoDAF and UML notation, the approach includes an Executable Architecture Model that provides quantitative aspects of the architecture.

We have applied the approach to Net-Centric Enterprise Services (NCES) mission threads, which include for the Warfighting domain:

- Time-Sensitive Targeting (TST),
- Focused Logistics/Deployment (Flog/D),
- Joint Close Air Support (JCAS),
- Global Strike (GS),
- Blue Force Tracking (BFT),

and for the Business domain:

- Military Personnel Pay (MilPay) & Disbursement.

Certain details of the architecture models have been given in this paper for the TST mission thread as an example. A high-level summary of Operational Activity to System Function Traceability Matrix (SV-5) for the above mission threads is given in Table 3, which shows the utilization of Core Enterprise Services that interface directly with end users (customer facing services).

Mission Threads	TST	Flog/D	JCAS	GS	BFT	MilPay/D	Subtotal
Messaging Services	9	6	9	5	1	1	**30**
Discovery Services	5	8	3	4	1	0	**21**
Collaboration Services	14	5	11	8	0	1	**39**
Mediation Services	8	7	3	3	1	0	**21**
User Assistant Services							
Application Services							
IA/ Security Services							
Enterprise Services Mgmt							
Storage Services							

Average number of activities using a service

Shaded area for foundational services

Table 3 –Summary of Potential Usage of Core Enterprise Services by Mission Threads

Clearly, Collaboration and Messaging services are of the highest priority, followed by Discovery and Mediation. The shaded area in the lower part of Table 3 covers the foundational services, which work in the background to provide the basic SOA functions.

Finally, the architecture modeling approach elucidated here is part of an overall system engineering process for building an SOA. Results from the architecture modeling can help the subsequent steps, including design, development, and testing. For example, the logical data model is useful in designing the service interfaces and the associated data schemas. The sequence diagrams help clarify implementation logics and develop corresponding test cases. The Executable Architecture Model provides quantitative information, which along with the system requirements, logical architecture and network information, enables us to determine the optimal deployment architecture. The strong emphasis of the operational activities in this approach also ensures end-to-end traceability between functional requirements and the services in the SOA.

6 References and Acronyms

1. Global Information Grid Capstone Requirements Document, JROCM 134-01, August 2001 <https://jdl.jwfc.jfcom.mil>.
2. Net-Centric Operations and Warfare (NCOW) Reference Model, Draft version 1.1, October, 2004.
3. Unified Modeling Language (UML), Version 1.5, March 2003 <http://www.uml.org/#UML1.5>.
4. DoD Architecture Framework (DoDAF) Version 1.0 < http://www.defenselink.mil/nii/doc/>.
5. Service-Oriented Architecture and the C4ISR Framework, *CrossTalk* magazine, Vol. 17, September 2004.
6. DoD Net-Centric Data Strategy, April 30, 2003 <http://www.defenselink.mil/nii/doc/>.
7. Multi-Service Tactics, Techniques and Procedures (MTTP) for Time-Sensitive Targets (TSTs), Air, Land, Sea Applications Center (ALSC), April 2004 (FOUO).
8. Time-Sensitive Target Mission Thread (Draft), OASD(NII), August 2004 (FOUO).
9. Joint Publication 3-60, Joint Doctrine for Targeting, 17 January 2002.

The acronyms used in this paper are listed below.

ATO	Air Tasking Order
BDA	Battle Damage Assessment
C2	Command and Control
CES	Core Enterprise Services
CoI	Community of Interest
DoD	Department of Defense
DoDAF	DoD Architecture Framework
EAM	Executable Architecture Model

ESM	Enterprise Services Management
FOUO	For Official Use Only
GIG	Global Information Grid
GIG ES	GIG Enterprise Service
IPB	Initial Preparation of Battlespace
ISR	Intelligence, Surveillance and Reconnaissance
JFC	Joint Force Command
NCES	Net-Centric Enterprise Services
NCOW	Net-Centric Operations and Warfare
SOA	Service-Oriented Architecture
RTL	Restricted Target List
TST	Time-Sensitive Targeting
UML	Unified Modeling Language