10TH INTERNATIONAL COMMAND AND CONTROL RESEARCH AND TECHNOLOGY SYMPOSIUM

THE FUTURE OF C2

C4ISR/C2 Architecture

The Integrated Executable Architecture Model Development by Congruent Process, Method, and Tools

Dr. Joongyoon Lee
SE Technology Ltd.
IAE, 633-2, Koan-Ri, Packam-Myon, Yongin-City, Kyonggi-Do, 449-863, Korea

Ms. Minsin Choi
SE Technology Ltd.
IAE, 633-2, Koan-Ri, Packam-Myon, Yongin-City, Kyonggi-Do, 449-863, Korea

Ms. Jaeduck Jang
Systems Engineering Dept., Ajou University
IAE, 633-2, Koan-Ri, Packam-Myon, Yongin-City, Kyonggi-Do, 449-863, Korea

Prof. Youngwon Park
Systems Engineering Dept., Ajou University
IAE, 633-2, Koan-Ri, Packam-Myon, Yongin-City, Kyonggi-Do, 449-863, Korea

Col. Jinsoo Jang
Ministry of National Defense of ROK.
#1 Youngsan-Dong 3ga, Youngsan-Gu, Seoul, Korea

Col. Bonghong Ko
Ministry of National Defense of ROK
#1 Youngsan-Dong 3ga, Youngsan-Gu, Seoul, Korea

Point of Contact ; Dr. Joongyoon Lee
Telephone/ Office ; +82-31-330-7226
Telephone/ Mobile ; +82-16-9237-7111
Fax; +82-31-330-7118
E-mail; leejy@se2u.com

The Integrated Executable Architecture Model Development by Congruent Process, Method, and Tools

Dr. Joongyoon Lee
SE Technology Ltd.
IAE, 633-2, Koanri, Packammyon, Yongin, Kyonggido, 449-863, Korea

Ms. Jaeduck Jang
Systems Engineering Dept., Ajou University
IAE, 633-2, Koanri, Packammyon, Yongin, Kyonggido, 449-863, Korea

Col. Jinsoo Jang,
Ministry of National Defense of ROK.
#1 Youngsan-Dong 3ga, Youngsan-Gu, Seoul, Korea

Ms. Minsin Choi
SE Technology Ltd.
IAE, 633-2, Koanri, Packammyon, Yongin, Kyonggido, 449-863, Korea

Prof. Youngwon Park
Systems Engineering Dept., Ajou University
IAE, 633-2, Koanri, Packammyon, Yongin, Kyonggido, 449-863, Korea

Col. Bonghong Ko
Ministry of National Defense of ROK
#1 Youngsan-Dong 3ga, Youngsan-Gu, Seoul, Korea

Abstract

There are interoperability problems in C4I systems of ROK because Army, Navy, Air Force, Joint Chiefs of Staff of ROK have been developed their own C4I systems independently for their own needs. An *integrated executable architecture* model was developed to solve the interoperability problems. This paper shows congruently integrated *process, method and tools* that are applied for developing *integrated executable architecture* model. This paper concentrates on using architecture templates as a communication tool to support front-end stage of architecting process. This approach was effective and efficient in solving interoperability problems of C4ISR systems by identifying IERs and deliver key interface requirements for each of the C4ISR system and shooter system.

1. Introduction

There were interoperability problems in C4I systems of Republic of Korea (ROK) because Army, Navy, Air Force, and Joint Chiefs of Staff of ROK have been developed their own C4I systems independently for their own needs. Moreover, there were requirement changes and incident requirements while developing C4I systems.

We used architecture as an effective means to control the requirement fluctuation quickly and to enhance the interoperability between C4I systems of ROK. This paper shows congruent process, method and tools which used for developing integrated executable architecture model. We adopted the systems engineering process as architecting process, the DoDAF was used as method, and the computer-aided systems engineering(CASysE) tool

CORE® and architecture templates were used as tools. This paper shows that how the systems engineering process, DoDAF, architecture template and CASysE tool CORE® could be used congruently for building integrated executable architecture. This paper focused especially on the architecture templates which were used for communication between stakeholders. We have mapped Requirement Definition Process and Solution Definition Process which defined in EIA-632 to Operational View and System View of DoDAF. According to the process, the products of DoDAF were arranged to develop the architecture efficiently. In the course of this architecting program, the most difficult task was elicitation of right operational information at the front-end step of the architecting process. This problem made lots of architecture development effort obsolete. To overcome this problem, two architecture templates were developed and used to elicit information and communicate with stakeholders. One is operational architecture template and the other is system architecture template. The architecture templates effectively bridge the users and developers who use different languages. The information, collected by the architecture templates, was precise to reduce the rework of architecting efforts. The executable architecture which was constructed by the CASysE tool CORE® made easy to enhance the operational concepts in the course of construction and analyzing the architecture model. Finally, we assured the interoperability level 3(LISI 3) by identifying IERs and their attributes between C4I systems of ROK. This paper shows congruent process, method and tools that are applied for developing integrated executable architecture model.

2. Purpose, Scope, and Products

The DoDAF guides architecture description process by the generic six-step. [DoDAF Ver.1 Vol.1, 2003] The architecture developement program was performed according to this guide. As step 1, the intended use of the architecture description was defined. The main purpose of the architecture of ROK is to identify and resolve the interoperability problems within C4I systems of Army, Navy, Air Force, Joint Chiefs of Staff of ROK. The selected target objective is to achieve Levels of Information Systems Interoperability (LISI) 3 for the warfighter domain. Secondly, we defined the scope of architecture description. Table 1 shows the scope of architecture description.

Table 1 The scope of architecture description

Class	Contents of scope
Mission	5 missions which have significant real time requirments
Time Frame	AS-IS(2000-2007) which include currently developing C4I systems
Organization	Army, Navy, Air Force, Joint Chiefs of Staff of ROK
Geography	Korean territory includes land, sea, and space.
System Context	The warfighter components and JCCS-K of USFK are jointed as external systems.

As step 3, the information, which the architecture description needs to capture, is determined based on the intended use. Table 2 shows the determined data items. As step 4, the products to be built determined. Most of the products which related to the As-Is architecture were selected. The results are listed in the Table 3

Table 2 Information of architecture description

Views	Information of architecture description
Operational view	▪ Operational node ✓ Attribute; ID Number, Name, Description ▪ Operational activity ✓ Attribute; ID Number, Name, Description ▪ Need line ✓ Attribute; ID Number, Name, Description ▪ Operational information ✓ Attribute; ID Number, Name, Description, Language, Transaction type, Periodicity, Timeliness, Integrity, Security level, Security type, Dissemination control, Target LISI
System view	▪ System/Components ✓ Attribute; ID Number, Name, Description ▪ System function ✓ Attribute; ID Number, Name, Description ▪ Interface ✓ Attribute; ID Number, Name, Description ▪ Link ✓ Attribute; ID Number, Name, Description, Capacity, Delay ▪ System information ✓ Attribute; ID Number, Name, Description, Language, Media type, Size/Unit, Accuracy, Transaction type, Periodicity, Timeliness, Integrity, Security level, Security type, Protection, Dissemination control, Target LISI
Technology standard view	▪ Service ✓ Attribute; ID Number, Name, Description ▪ Standard ✓ Attribute; ID Number, Name, Description, Time frame, Type

Table 3 Selected architecture products to be built

View	Prod. ID	Name	Sel.
AV	AV-1	Overview and Summary Information	O
	AV-2	Integrated Dictionary	O
OV	OV-1	High-Level Operational Concept Graphic	O
	OV-2	Operational Node Connectivity Description	O
	OV-3	Operational Information Exchange Matrix	O
	OV-4	Organizational Relationships Chart	O
	OV-5	Operational Activity Model	O
	OV-6	Operational Activity Sequence and Timing Descriptions	O
	OV-7	Logical Data Model	O
SV	SV-1	Systems Interface Description	O
	SV-2	Systems Communications Description	O
	SV-3	Systems-Systems Matrix	O
	SV-4	Systems Functionality Description	O
	SV-5	Operational Activity to Systems Function Traceability Matrix	O
	SV-6	Systems Data Exchange Matrix	O
	SV-7	Systems Performance Parameters Matrix	X
	SV-8	Systems Evolution Description	X
	SV-9	Systems Technology Forecast	X
	SV-10	Systems Functionality Sequence and Timing Descriptions	O
	SV-11	Physical Schema	O
TV	TV-1	Technical Standards Profile	O
	TV-2	Technical Standards Forecast	X

3. Architecting Strategy

Martin described process, methods, tools and environment (PMTE) paradigm which represent a mental model used to screen data and filter new information with regard to systems engineering process, methods, tools and environment. [Martin, 1997] Martin insists that PMTE elements must be consistent with each other while doing systems engineering. From this point of view, this paper introduces congruent process, methods and tools to develop the integrated executable architecture model. Figure 1 shows the PMTE elements which were used in architecture development program

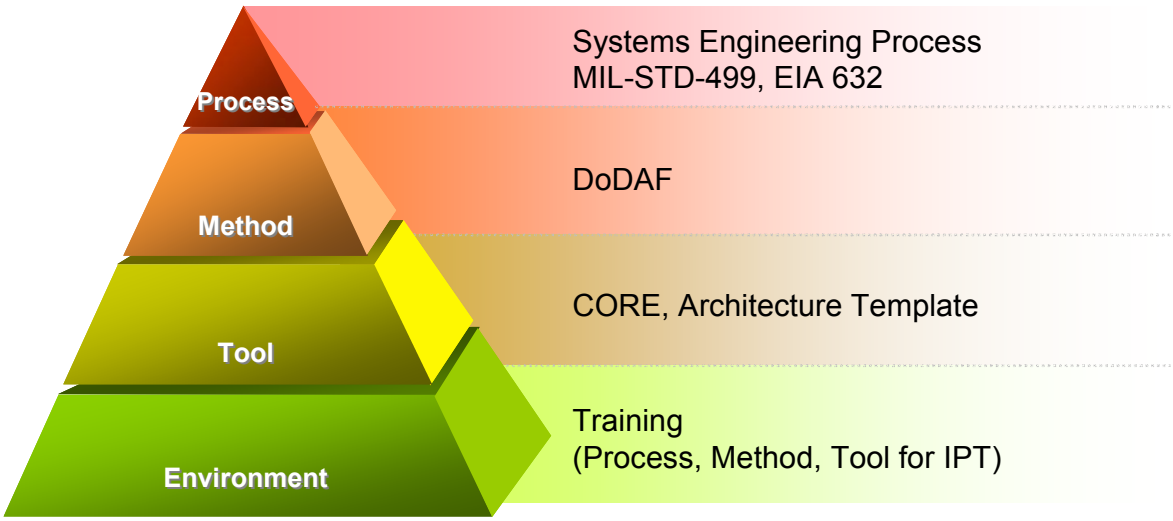

Figure 1 PMTE paradigm for architecture model development

3.1 Architecting Process and Method

MIL-STD-499 described the systems engineering process as requirement analysis, functional analysis/allocation, synthesis and system analysis & control.[MIL-STD-499B, 1994] This systems engineering process evolved to EIA 632. The EIA 632 described system design process as requirements definition process and solution definition process.[EIA 632, 1998] Every system has an architecture and the architecture of a system can be recorded by an architectural description.[IEEE 1471, 2000] At the concept stage of system development, lots of systems engineering products are architectural descriptions. An operational concept document can be mapped to operational view of the architecture. A system specification can be mapped to system view of the architecture. DoDAF WG suggested a data-centric build sequence using DoDAF product. They said that the product order takes advantage of the related nature of the products and the dependencies among products. [DoDAF WG Vol.III, 2003] And Harry showed that the architecting process can be mapped with systems engineering process of MIL-STD-499B.[Harry, 2002] To support system design process of EIA 632 with DoDAF method, this paper suggests another data-centric build sequence. The suggested data-centric build sequence was used for developing the As-Is architecture model of C4I systems of ROK. Figure 2 shows the suggested data-centric build sequence. Figure 2 shows that the operational view(OV) products form requirement definition process and the system view(SV) products form solution definition process. The shaded boxes of Figure 2 are used to indicate principal products for building architecture model. In figure 2, OV-1 and OV-4 was used as input information, and then the principal products(OV-5 and OV-6) was developed. If the principal products were developed than the other products(OV-2, OV-3, and OV-7) could be developed with little additional effort because most of operational architecture information is contained in the principal products and the remaining information is some more attributes of architecture elements of principal products. Because of these dependencies among products, the integrated capability of architecting tool is important. The same concept was applied to the solution definition process. The OV-5 was used as detailed operational requirements and the SV-4 was

developed to meet the operational requirements of OV-5. Developing the principal systems architecture products(SV-4 and SV-10) was major task of system design. The SV-5 and SV-6 was used to verify that all the operational requirements were implemented in system architecture or not.

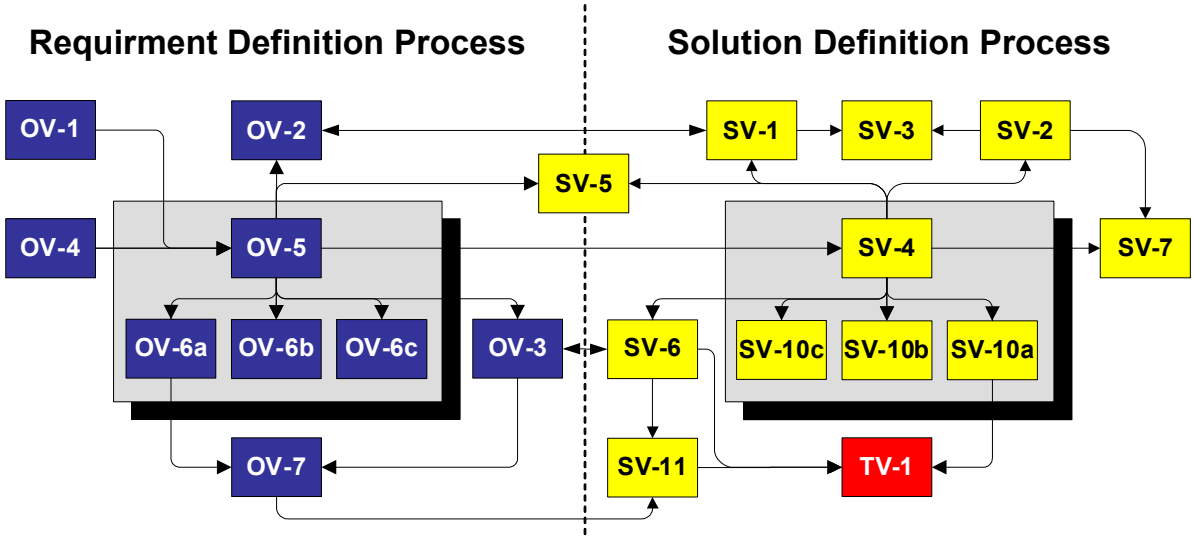

Figure 2 Data-centric system design process

3.2 Architecting Tools

A tool is an instrument that, when applied to a particular method, can enhance the efficiency of a task. The purpose of a tool should be to facilitate the accomplishment of the "Hows". [Martin, 1997] We used the CORE® of the Vitech Co. as tool, to facilitate the accomplishment of the architecture design process and DoDAF methods. That is to say, the CORE® equipped with an systems engineering and DoDAF schema which support the systems engineering process and DoDAF.

The subject of executable architectures is one area of the future evolution of the DoDAF. "Executable architecture" refers to the use of dynamic simulation software to evaluate architecture models. These executable architectures could be simulated directly from the architecture models to analyze and to support decision making. Following several purposes can be achieved with these specialized tools. (1) The architecture model itself can be verified for internal self-consistency. (2) Operational concepts can be simulated, observed dynamically, verified and refined. (3) Operational plans can be examined and assessed. (4) Tradeoffs between systems can be assessed. (5) Architecture measures can be evaluated (given that metrics have been defined), which can support cost-benefit analyses and quantitative acquisition decisions. [DoDAF WG Vol.I, 2003] The CORE® has the capability to build executable architecture models. Due to the dynamic simulation capability of CORE®, the analysis of architecture was effective and efficient. Besides above benefits, the executable nature of architecture was very helpful to reduce analysis time, and to enhance fidelity of the

architecture.

An architecture description is defined to be an integrated architecture when products and their constituent architecture data elements are developed such that architecture data elements defined in one view are the same (i.e., same names, definitions, and values) as architecture data elements referenced in another view.[DoDAF WG Vol.I, 2003] The integrated architecture capability is generically implemented in CORE®. If the principal products were developed than the other products could be developed with little additional effort because of the integrated nature. The integrated architecture capability of CORE® was very helpful to reduce architecting effort and to secure consistency throughout all the architecture models during architecture development program of ROK.

However, it is not enough in the actual architecture development program. Even though the CORE® has the capability of integrated executable architecture, it is not easy to be used by the various stakeholders. The CORE® is a specialized tool and can only be effectively used by trained systems engineer. We need another tool because it is hard to communicate using CORE directly with various stakeholders-e.g. C4ISR system users. In actual architecting program, elicitation of customer requirement is important and not an easy task. At an elicitation stage, the operational concept fluctuates. This fluctuation makes burden or obsolete the modeling effort. So, we need another tool to elicit the right architectural information in a short time with stakeholders. Lee and Park developed and used a requirement template to elicit and refine the requirements at front-end stage of requirement definition process. [Lee and Park, 2004] This paper introduces the *architecture templates* which can be used for elicitation and refinement the architectural information within stakeholders. The *architecture templates* built from 4WHV which represents Who, Where, What, When, How, and Verb and information attributes which described by DoDAF. Figure 3 shows the heading of operational architecture template. Figure 4 shows the heading of system architecture template.

Operational Behavior Data							Operational Information Attribute Data								
When	Who	From/To	What	How	Verb	Proces. Time (Sec)	Langua ge	Trans. Type	Periodic ity	Timelin ess (Sec)	Integrity Check	Secu. Level	Secu. Type	Disemin. Ctrl	LISI
Op Seq.	Operational behavior text.					Op. time	Operational Information Attribute Data								

Figure 3 Heading of operational architecture template

System Function Data							System Information Attribute Data											
When	Who	From/ To	What	How	Verb	Proces. Time (Sec)	Lang uage	Media Type	Size	Unit	Trsct. Type	Perio dicity	Timeli ness	Integrity Check	Secu Levl	Secu Type	Dissemin. Ctrl.	LISI
Sys. Process sequence	System behavior text					Sys. process time	System Information Attribute Data											

Figure 4 Heading of system architecture template

The *operational architecture template* guides what information has to be gathered and how to write it precisely. Users can write the operational behavior text by the user language at the *operational architecture template*. We elicited right operational requirements quickly by using

the *operational architecture template*. And attribute data columns of template guide to the precise description of the operational information by the users. This was effective in refining operational requirements. The *system architecture template* has the same concept but was mainly used to the system developer. The spreadsheet styles of these *architecture templates* are familiar with and can be trained and used easily.

4. Developing Architecture

Gathering the architecture data and building the requisite products is the forth step of the generic six-step architecture description process of DoDAF. At the start point of this forth step, we used operational plan documents and discussions. Generally those input documents do not have enough information to build architecture model. As stated already, the *operational architecture template* was used to elicit and refine operational architecture information. Figure 5 shows an example of *operational architecture template* which contains operational architecture data.

Operational Behavior Data							Operational Information Attribute Data								
When	Who	From/To	What	How	Verb	Proces. Time (Sec)	Langua ge	Trans. Type	Peridioc ity	Timelin ess (Sec)	Integrity Check	Secu. Level	Secu. Type	Diseminati on Ctrl	LISI
T1						60	K/E	Direct	NS	60	1	2	6	OCC	2
T2						30	K/E	Direct	NS	30	1	2	6	OCC/FSC	3
T3						10	-	-	-	-	-	-	-	-	-
T4						10	-	-	-	-	-	-	-	-	-
T5						10	K/E	Direct	NS	10	1	2	6	ASOC	2
T6						10	K/E	Direct	NS	10	1	2	6	X-ATKer	2
T7						60	-	-	-	-	-	-	-	-	-
T8						10	K/E	Direct	NS	10	1	2	6	OCC/FSC	2

CLASSIFIED

Figure 5 An operational architecture template with elicited data

The *architecture template* significantly reduces the concept fluctuation within stakeholders in a short time. During the architecture development program of ROK, it took several times of 3 or 4 hours discussion to make a consensus and describe precisely about operational architecture data. If there was a consensus of *operational architecture template*, then a systems engineer could build model of the operational architecture with small effort. And then another month was taken for analysis and business process reengineering(BPR) using simulation capability of CORE®. Like this way, the operational requirements were specified. The system architecture model building process has different aspect with that of operational architecture because system developers have specialties in that area and use technical language. The integrated product team(IPT) members of system developer side can understand quickly the specified operational requirements, by using both the OV products and operational model directly. They can fill *system architecture template* by themselves. After the *system architecture template* was completed, system architecture model was built, analyzed and enhanced. Figure 6 shows the work process performed in architecture development. Figure 6 shows a system architecture template filled with system architecture data.

System Function Data							System Information Attribute Data											
When	Who	From/ To	What	How	Verb	Proces. Time (Sec)	Lang uage	Media Type	Size	Unit	Trsct. Type	Perio dicity	Timeli ness	Integrity Check	Secu Levl	Secu Type	Dissemin. Ctrl.	LISI
T1						50	K/E	TEXT	40	Kb	Direct	NS	50	1	2	6	ArmyC4I	2
T2						2	K/E	TEXT	40	Kb	Direct	NS	50	1	2	6	ArmyC4I	2
T3						10	-	-	-	-	-	-	-	-	-	-	-	-
T4						1	K/E	TEXT	40	Kb	Direct	NS	1	1	2	6	KJCCS	3
T5						1	K/E	TEXT	40	Kb	Direct	NS	1	1	2	6	KJCCS	3
T6						2	-	-	-	-	-	-	-	-	-	-	-	-
T7						1	K/E	TEXT	40	Kb	Direct	NS	1	1	2	6	KJCCS	3

CLASSIFIED

Figure 6 A system architecture template filled with system architecture data

We used architecture templates to close the gap between sophisticated integrated executable architecture development tool and various stakeholders. Figure 7 shows the work process from OPLAN text to *integrated executable architecture* model. Figure 7 indicate that architecture templates bridge the gap between the stakeholders, which is especially user and developers, and systems engineering tool.

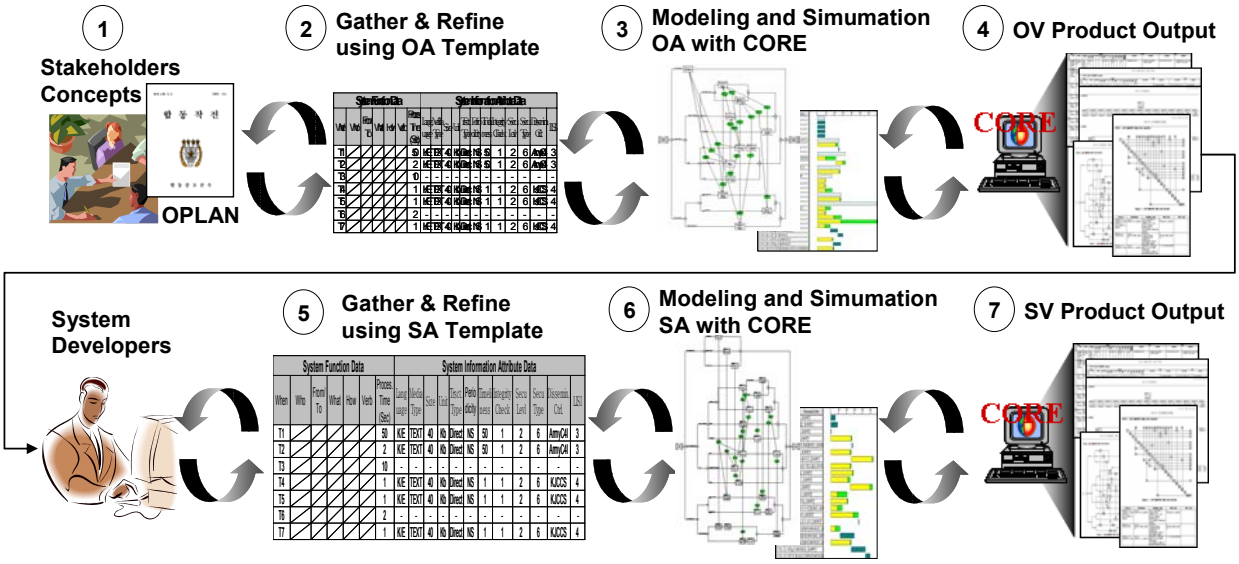

Figure 7 Work process for architecture model development

5. Results

This paper introduced congruently integrated *process, method, and tools* for architecture model development. According to the system design process of EIA 632, this paper suggests a data-centric build sequence which was tailored from that of DoDAF. The DoDAF was used as a method to glue the process with tools. As architecting tool, the architecture templates and specialized systems engineering tool CORE® were used. The *operational architecture template* was effective in eliciting requirements, to reduce requirement fluctuation, and to adopt incident requirement quickly. Due to the integrated architecture capability of CORE® the architecture development effort reduced drastically. And due to the executable architecture capability of CORE®, the performance of architecture was enhanced effectively. This paper also suggests smooth work process by using *architecture templates* which close the gap between stakeholders

and the sophisticated systems engineering tool CORE®. The congruently integrated *process, method, and tools* were effective and efficient in solving interoperability problems by identifying IERs and deliver key interface requirements and key functional performance requirements to each system of warfighter domain. Finally, we assured the interoperability level 3(LISI 3) of C4ISR systems including shooter of ROK through the architecture model.

References

[DoDAF WG Vol.I, 2003] DoDAF WG, DoD Architecture Framework Ver.1.0 Vol. I: Definitions and Guidelines, DoD, Aug. 2003.

[DoDAF WG Vol.III, 2003] DoDAF WG, DoD Architecture Framework Ver.1.0 Vol. III: Deskbook, DoD, Aug. 2003.

[EIA 632, 1998] Electronic Industries Alliance(EIA), EIA-632: Processes for Engineering a System, Electronic Industries Alliance, USA, 1998 .

[Harry, 2002]Harry Crisp, Naval Collaborative Engineering Environment, Navy, 2002.

[IEEE 1471, 2000]Institute of Electrical and Electronics Engineers(IEEE), IEEE Std 1471-2000, IEEE Recommended Practice for Architectural Description of Software-Intensive Systems, The Institute of Electrical and Electronics Engineers, 2000.

[Martin, 1997] Martin, J. N., Systems Engineering Guidebook: A Process for Developing Systems and Products, CRC Press, USA, 1997.

[MIL-STD-499B, 1994] JOSD, MIL-STD-499B, Military Standard Systems Engineering, 1994

[Lee and Park, 2004] Lee and Park, Requirement Architecture Framework(RAF), INCOSE 2004.