

A Capability-Based Method for System of Systems Architecting in the Net-Centric Environment

P. L. Rui, R. Wang, and H. Yu

Abstract—As the emergence of the Net-Centric Warfare (NCW), the military information system has been evolved from platform-centric to be net-centric, which brings great challenges for System of Systems (SoS) engineering in the net centric environment. A major task of system engineering is to build system architecture. Although classical system engineering deals very well with architecting problems for a single system in which user requirements are well defined, it has no good solutions for SoS architecting problems. In this paper, existing architecture development methods are briefly reviewed, and a novel capability-based method (CBM) for architecture design is proposed, which not only enables SoS architecting with the new kind of capability-based development process, but also ensures consistency to form integrated architectures.

Index Terms—System of systems (SoS) engineering, architecture design, NCW.

I. INTRODUCTION

As the military information system moves through the brave new world of Net-Centric Warfare (NCW) [1] or Net Enabled Operations (NCO) and the evolution of the U.S. Department of Defense (DoD) Global Information Grid to help implement that vision, the importance of engineering system of system (SoS) in the net-centric environment becomes more urgent [2].

A major task of system engineering is to build system architecture. Although classical system engineering deals very well with architecting problems in which user requirements are well understood, technology is evolving slowly, organizational dynamics can be mapped out in advance and external interfaces are stable. The system-of-systems environment is anything but static [3]. This has necessitated an evolution of the architecting approach, intensified focus on capability development rather than satisfy well defined user requirements. System properties (such as changeability, flexibility, agility, etc.), and recognition of the inseparability of technological system and the enterprise developing and operating such systems are the factors should be considered.

Architecture frameworks are methods usually used in architecture modeling. They provide a structured and systematic approach to design systems. To date, there are many architecture frameworks [4]-[8]. According to DoD statements, the most famous DoD Architecture Framework

(DoDAF) was established to guide the development of architectures, to satisfy the demands of a structured, repeatable method for evaluating investments and investment alternatives, as well as the ability to effectively implement organizational change, create new systems, and deploy new technologies.

DoDAF also provides a general guide for architecture description with a high level, 6-step development process: (1) determine the intended use of the architecture, (2) determine scope of architecture, (3) determine data required to support architecture development, (4) collect, organize, correlate, and store architectural data, (5) conduct analyses in support of architecture objectives, and (6) document results in accordance with decision-maker needs. However, it lacks specific methodology for architecture development, so it is difficult to build architecture views or products.

The aim of this paper is therefore to develop a new architecture design method for SoS architecting problems in the net-centric environment. For this purpose, an overview and evaluation of existing architecture development methods is given in section 2. Building from here, a novel capability-based methodology is presented and analyzed in more details in section 3. The paper concludes with a summary of the proposed method and an outlook for further research in section 4.

II. EXISTING METHODS OVERVIEW

There are several methods for architecture development which can be divided into three categories, i.e. structure analysis method (SAM), object-oriented method (OOM) and activity-based method (ABM).

The structured analysis approach for architecture design was presented by Wagenhals L.W. et al in 2000 [9], which is a kind of process-oriented method evolved from an earlier structured analysis and design technique (SADT) in software engineering field. It is based on C4ISR architecture framework and provides a specified 6-phases process for developing operational views (OV) and system views (SV) products. The main idea of the SAM is that, it is started from concept of operations (CONOPS) or operational missions, to sequentially created required architecture products by hierarchical process decomposition from up to bottom. Since it is aimed to design systems with well defined operational missions or requirements, it is difficult to apply to SoS architecting problems in the net-centric environment.

The object-oriented approach was also proposed by Bienvenu M.P. et al in 2000 [10] hich is characterized by utilizing the unified modeling language (UML) architecture technique and notation for architecture design. It describes

Manuscript received July 28, 2012; revised September 3, 2012. This work was supported in part by the National Defense Pre-Research Foundation of China.

The authors are with the 28th Research Institute of China Electronics Technology Group Corporation, Nanjing, China, 210007 (e-mail: cetc28_plrui@163.com)

the operational need, places data in the context of its used, and provides a traceable foundation for system design. It is based on the concepts of data abstraction and inheritance from a service-oriented view. The OOM provides an orderly arrangement of the parts of the business organization and includes a style and method of design through its highly developed notation style. However, since it needs to be derived by unambiguous operational need, usually captured by use case diagrams, it encounters the same problem with the SAM when applied for SoS architecting problems.

The activity-based method was developed by Steven J. Ring et al in 2004 [11] which is based on a set of DoDAF OV and SV elements symmetrically aligned to each other, from which four operational and four system architecture elements provide the core building block foundation of an integrated architecture. The associations between these core elements form the basis of an integrated architecture data specification model, and from these core elements, several DoDAF architecture elements are rendered and several DoDAF products are generated. The main purpose of this method is to enable architects to concentrate on the art of architecture—that is identifying the core architecture elements, their views and understanding how they are all related together and the architecting problems in the net-centric environment stated above are also not considered.

III. PROPOSED CAPABILITY-BASED METHOD (CBM)

A. Supporting Architecture Product

Architecture Framework provides a good basis for developing and presenting architecture descriptions in a uniform and consistent manner. To support the DoD net-centric transformation, a DoDAF v2.0 [4] is proposed, which consists of eight views. To demonstrate our proposed CBM method, four main architecture views, including capability view (CV), operational view (OV), system view (SV) and service view (SvcV), and associated products are selected.

The capability view (CV) captures the enterprise goals associated with the overall vision for executing a specified course of action, or the ability to achieve a desired effect under specific standards and conditions through combinations of means and ways to perform a set of tasks. It provides a strategic context for the capabilities regarding capability evolution.

The operational view (OV) captures the organizations, tasks, or activities performed, and information that must be exchanged between them to accomplish DoD missions.

The system view (SV) captures the information on supporting automated systems, interconnectivity, and other systems functionality in support of operating activities. To incorporate introduction of SOA development in the net-centric environment, the service view (SvcV) is added, which is highly symmetrically aligned to SV.

B. Capability-Based Analysis Process

Before we start discussing CBM, the architecture described above can also be re-viewed in four aspects, i.e. Data, Function, Organization, and Technology infrastructure.

As a general concept of information technology, applications consist of data and functions. In this case, the sub-hierarchy of an application is the shared data and common functions in the overall architecture.

The Data aspect describes the set of data needed to perform data flow and the relationships in the database. The Function aspect describes functions, processes, and activities that act on information to support operations. The Organization aspect consists of the organizational structure of the command, the major operations performed by organizations, the organization breakdown structure, and the distribution of the organizations to locations. The Technology infrastructure aspect consists of the hardware, software, network, telecommunications, and general services that constitute the operational environment in which applications operate.

A capability analysis process is given in Fig. 1. As shown in Fig. 1, it uses a capability-based approach for architecture design instead of a mission-based approach. It starts with creating high-level capability vision, which focuses on capability development and aims to support multiple rather than specified single mission or task. With the capabilities in mind, several concepts of operations (CONOPS) can be derived, which brings together required architecture elements from Function, Organization, and Technology infrastructure aspects.

The architecture then can then be obtained accordingly. For example, the Function Architecture can be derived based on CONOPS by the way of Function Decomposition, i.e. systems or services functions decomposition and activities decomposition.

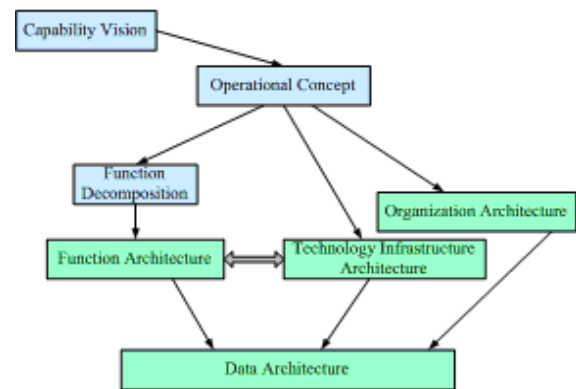


Fig. 1. Capability-based analysis process.

C. Relationships of Core Entities

In order to obtain an integrated, unambiguous, and consistent architecture, CBM also defines one CV, four OV and four SV/SvrV architecture entity objects, which are used to provide the core foundation building block primitives of an integrated architecture. As shown in TABLE I, on the CV side, Capability represents the primary architecture entity. On the OV side, Activity, Operational Node (Op Node), Role, and Information represent the primary entities. On the SV/SvrV side, Function, System/Service Node, System/Service, and Data, which are symmetrically aligned with each other, represent the primary entities.

Relationships among the core architecture entity objects are given in Fig. 2, which are characterized by four three-way associations: “Capability—Activity—Function”, “Activity

—Operational Node—Role”, “Function—System/Service Node—System/Service”, and “Organization Unit—Role—System”. It indicates that each Activity (Function) that produces and consumes Information (Data) is performed at an Operational (System/Service) Node by a Role (System/Service).

TABLE I: CORE ARCHITECTURE ENTITY OBJECTS

Views	Entity Objects	General Description
CV	Capability	The planned capabilities needed to develop.
OV	Activity	A set of operational actions which consumes or outputs information.
	Operational Node	The collection of similarly related activities usually at a place or location.
	Role	Resources characterized by a set of knowledge, skills and abilities assigned to humans.
	Information	Resource flow exchanged between Opnodes.
SV/Svr V	Function	A set of system/service actions which consumes or output data.
	System/Service Node	Collection of similarly related system/service functions usually at a place or location.
	System/Service	Material resources are described in terms of performance characteristics.
	Data	Resource flow exchanged between System/Service nodes.

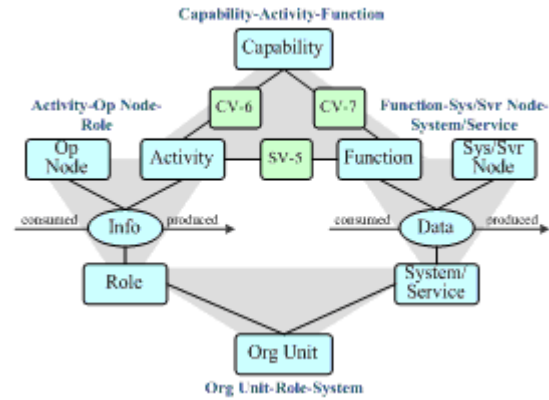


Fig. 2. Relationships of core entities.

In the three-way association between the OV entities, the intersection of the association between an Operational Activity and an Operational Node is a Role. Likewise, in the three-way association between the SV entities, the intersection of the association between a System/Service Function with a System/Service Node is a System/Service. The intersection of the association between a Role and a System/Service is the Organizational Unit.

Furthermore, the association of Organizational Units with Roles already exists in OV-4. The association of Capability and Operational Activity exists in CV-6. The association of Capability and System/Service Function exists in CV-7. The association of Operational Activity and System/Service Function exists in SV-5.

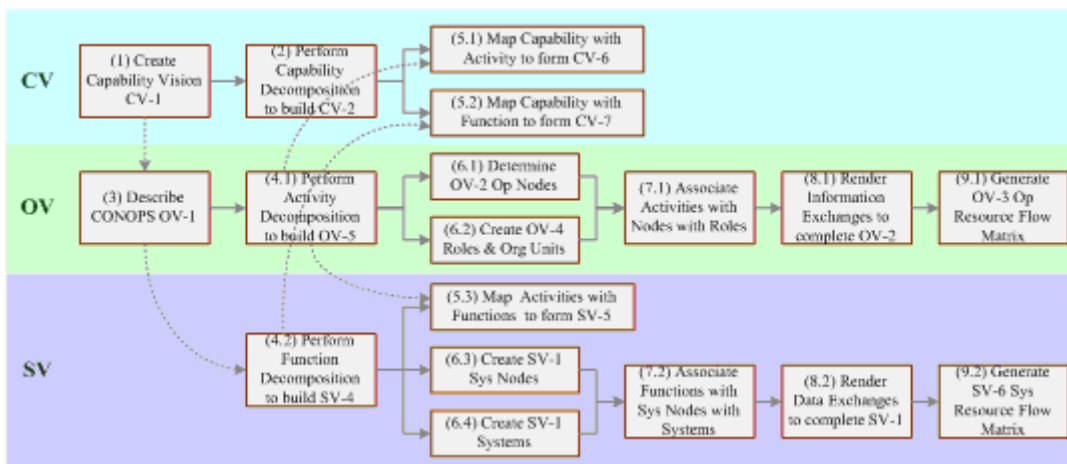


Fig. 3. Primary architecture products development steps.

D. Products Development Steps

An architecture development workflow, which combines the capability-based analysis process and the four three-way associations, is demonstrated in Fig. 3. The workflow consists of 9 steps:

Step 1: Create a strategic vision regarding capability development to form CV-1.

Step 2: Perform capability decomposition to build CV-2 and the core entity—Capabilities.

Step 3: Describe CONOPS based on the capability vision. There may be several CONOPS in order to reflect the goal of capability development.

Step 4: Perform activity decomposition to build OV-5 and

the core entities—Activities and Information on the OV side. At the same time, perform function decomposition to build SV-4 and the core entity—Functions on the SV side.

Step 5: Associated Capabilities, Activities, and Functions based on the corresponding three-way association to form CV-6, CV-7, and SV-5, respectively.

Step 6: Create the core entities—Op Nodes and Roles on the OV side, and System Nodes and System on the SV side.

Step 7: Associate activities with Nodes with Roles, auto form three-way associations between Activities, Operational Nodes and Roles. At the same time, associate Functions with System Nodes with Systems, and auto form three-way associations between Functions, System Nodes and Systems.

Step 8: Render information exchanges to complete OV-2 and data exchanges to form OV-1.

Step 9: Auto generate OV-3 operation resource flow matrix and SV-6 system resource flow matrix.

It should be noted that the corresponding products for SvrV can be easily obtained by the same workflow as shown on the bottom SV side. Once the primary products made up the foundation of the architecture are all completed. A capability driven, integrated architecture can be resulted. Other architecture products can be created on the basis of these primary products on user demand.

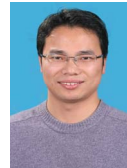
IV. CONCLUSIONS

The implications behind Net Centric Warfare (NCW) or Net Enabled Operations (NCO) bring great challenges for architecting system of system (SoS) in the net centric environment. This has necessitated an evolution of the architecting approach considering SoS properties, such as dynamic user requirements. In this paper, the Capability-Based Method (CBM) is presented, which is characterized by capability-driven for developing fully integrated, unambiguous, and consistent architecture views or products. Further researches will be done to validate the effectiveness of the proposed framework and its associated development process.

REFERENCES

- [1] D. S. Alberts, "Information Age Transformation: Getting to a 21st Century Military," *Washington, DC, CCRP Publications*, pp.7-8, 2002
- [2] A. Meilich, "System of Systems (SoS) Engineering & Architecture Challenges in a Net Centric Environment," in *IEEE/SMC Int. Conference on System of Systems Engineering*, April 2006, pp. 5-9.
- [3] C. Reggie, "The Changing Role of Requirements and Architecture in Systems Engineering," in *Proceedings of the 2006 IEEE/SMC Int. Conference on System of Systems Engineering*, Los Angeles, CA, USA - April 2006.
- [4] *DoD Architecture Framework Version 2.0 Volume I, II, III*. Department of Defense Architecture Framework Working Group, 2009.

- [5] *MoD Architecture Framework Version 1.0*, Ministry of Defense, August 2008
- [6] A. Z. John, "A Framework for Information System Architecture," *IBM System Journal*, vol. 26 no. 3, pp. 276-292, 1987
- [7] *C4ISR Architecture Framework Version 2.0*, Department of Defense. Architecture Framework Working Group, 18 December 1997
- [8] A. H. Levis, "Architecting Information System (lecture notes)," George Mason University, 2000
- [9] L. W. Wagenhals, I. Shin, and D. Kim, et al. "C4ISR Architectures II: Structured Analysis Approach for Architecture Design," *System s Engineering*, vol. 3, no. 4, pp. 248-287, 2000
- [10] M. P. Bienvenu, I. Shin, A. H. Levis, "C4ISR Architectures III: An Object-Oriented Approach to Architecture Design," *System Engineering*, vol. 3, no. 4, pp.288-312, 2000
- [11] J. R. Steven, N. Dave, T. Jim, and H. Stanley, "An activity-based methodology for development and analysis of integrated DoD architectures," in *2004 Command and Control Research and Technology Symposium*, 2004



P. L. Rui was born in An Hui, the People's Republic of China, in 1979. He received his B.S. degree from Dept. of Electrical Engineering, An Hui Normal University, in 2002, and M.S. and Ph.D degrees from Dept. of Electrical Engineering, Nanjing University of Science and Technology in 2007. His research interests include system engineering, system architecting and data fusion.



R. Wang was born in Shan Dong, the People's Republic of China, in 1983. He received his B.S. degree from Mechanical Manufacturing and Automation, Nanjing University of Aeronautics and Astronautics, in 2005, and M.S. degree from Mechanical and Electrical engineering, Nanjing University of Aeronautics and Astronautics, in 2007. His research interests include system architecting and software engineering.



H. Yu was born in Shan Xi, the People's Republic of China, in 1974. He received his B.S. degree from College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, in 1996, and M.S. and Ph. D degrees in 2002. His research interests include Enterprise architecting, Information Resource Planning and system engineering.