8

An Abstracted and Effective Capabilities Portfolio Management Methodology Using Enterprise or System of Systems Level Architecture

Joongyoon Lee and Youngwon Park
Ajou University/SE Technology Ltd.
Republic of Korea

1. Introduction

The purpose of this chapter is to provide an abstracted methodology for executing Capabilities Portfolio Management (hereafter, CPM) effectively based on the Department of Defense Architecture Framework version 2.0 (hereafter, DoDAF V2.0)[1]. A methodology is the specification of the process to follow together with the work products to be used and generated, plus the consideration of the people and tools involved, during a development effort. Based on the definition of methodology of ISO 24744 (ISO, 2007), this chapter provides process, product and modeling related technology as considerations of people and tools involved in CPM. From DoDAF V2.0, the purpose of developing an architecture is for beneficial use of it. A good set of architectural artifacts facilitates the manipulation and use of them in meeting its purposes of use. Systems engineering methodologies evolve to accommodate or to deal with problems in enterprise level, system of systems (hereafter, SoS) level and family of systems (hereafter, FoS) level. And the CPM of the United States Department of Defense (hereafter DoD) is a good example which demonstrates enterprise or SoS level problems. However, the complexity of the metamodel of DoDAF makes it difficult to develop and use the architecture models and their associated artifacts. DoDAF states that it was established to guide the development of architectures and to satisfy the demands for a structured and, repeatable method in evaluating alternatives which add value to decisions and management practices. One of the objectives of DoDAF V2.0 is to define concepts and models usable in DoD's six core processes. DoDAF V2.0 provides a particular methodology in the architecture development process. However, DoDAF as well as other guidelines states requirements for CPM which is one of DoD's six core processes, rather than how to perform CPM. This chapter provides an abstracted methodology for CPM which includes the process, abstrated products and tailored meta-models based on DoDAF Meta Model (hereafter, DM2).

[1]The Department of Defense Architecture Framework (DoDAF) is an architecture framework for the United States Department of Defense, that provides structure for a specific stakeholder concern through viewpoints organized by various views. (quoted from http://en.wikipedia.org/wiki/DODAF)

2. Current issues on system of systems problems

The definition of system of DoDAF V2.0 (DoD, Aug. 2010) has been changed from that of DoDAF V1.5. A system is not just computer hardware and software. A system is now defined in the general sense of an assemblage of components (machine or, human)- that perform activities (since they are subtypes of Performer) and interact or become interdependent. The Federal Enterprise Architecture Practice Guidance (Federal Government, 2007) has defined three types of architecture: enterprise architecture, segment architecture, and solution architecture. "Enterprise architecture" is fundamentally concerned with identifying common or shared assets – whether they are strategies, business processes, investments, data, systems, or technologies. By contrast, "segment architecture" defines a simple roadmap for a core mission area, business service, or enterprise service. "Solution architecture" defines agency IT assets such as applications or components used to automate and improve individual agency business functions. The scope of solution architecture is typically limited to a single project and is used to implement all or part of a system or business solution. From the viewpoint of a system hierarchy, the solution architecture addresses system level problems whereas enterprise architecture and segment architecture address SoS/FoS problems respectively. Systems engineering methodologies have evolved to deal with enterprise or SoS level problems.

The purpose of DoDAF V2.0 is to define concepts and models usable in DoD's six core processes:

1. Capabilities Integration and Development (JCIDS)
2. Planning, Programming, Budgeting, and Execution (PPBE)
3. Acquisition System (DAS)
4. Systems Engineering (SE)
5. Operations Planning
6. Capabilities Portfolio Management (CPM)

The DoD's six core processes are good examples of addressing SoS level problems. However, DoDAF V2.0 and other guidelines state requirements rather than how to perform these processes. This chapter provides a methodology for CPM which contains detailed processes, methods, artifacts and tailored meta-model of DM2.

3. Capability Portfolio Management methodology development guide

ISO/IEC 24744 (ISO, 2007) defines that a methodology specifies the process to be executed, usually as a set of related activities, tasks and/or techniques, together with what work products must be manipulated (created, used or changed) at each occasion possibly including models, documents and other inputs and outputs. So a methodology is the specification of the process to follow together with the work products to be used and generated, plus techniques which are the consideration of people and tools involved, during a development effort.

3.1 Methodology development requirements

3.1.1 Process, methods, tools, and environment concept of methodology element

According to Martin (Martin, 1997), it is important to have a proper balance among process, methods, tools, and environment (PMTE) when performing systems engineering tasks. He defines that a process is a logical sequence of tasks performed to achieve a particular objective, a method consists of techniques for performing a task, and a tool is an instrument when applied to a particular method. While, in ISO/IEC 24744, a method is used as a synonym of methodology, this chapter adopts Martin's PMTE paradigm. So this chapter provides the CPM methodology which has its own process, method (technique), and tool (model or artifacts).

ISO/IEC 24744 (ISO, 2007) also states that a methodology element is a simple component of a methodology. Usually, methodology elements include the specification of what tasks, activities, techniques, models, documents, languages and/or notations can or must be used when applying the methodology. Methodology elements are related to each other, comprising a network of abstract concepts. Typical methodology elements are Capture Requirements, Write Code for Methods (a kind of tasks), Requirements Engineering, High-Level Modelling (kinds of activities), Pseudo-code, Dependency Graphs (notations), Class, Attribute (kinds of model building blocks), Class Model, Class Diagram, Requirements Specification (kind of work products), etc. From this concept, the elements for CPM methodology of this chapter are Capture Requirements (top level CPM requirements), High-Level Model of CPM process (kinds of activities), metamodel (Class Diagram), and Attribute.

3.1.2 Metamodel development requirements

A metamodel is the specification of the concepts, relations and rules that are used to define a methodology. This metamodel should be simple and consistent with the analysis methodology. And the metamodel is a schema for semantic data and a language that supports a particular process, method (technique), and tool (model or artifacts).

Probability and set theory have axioms of mutually exclusive and collectively exhaustive (hereafter, MECE) concepts, and decomposition concepts. This means no overlap, no omission of concept and complete decomposition of a concept also. Axiomatic design theory (Suh, 1990) states that the design axiom No.1 is the independence axiom, "Maintain the independence of functions (not affecting other functions)" and the design axiom No. 2 is the information axiom, "Minimize the information content of the design (functionally uncoupled design)." These are the same MECE principle concept of different viewpoints, one is a set viewpoint and the other is a functional design viewpoint. A past study (Lee and Park, 2009) adopted this concept to the metamodel design. The study pointed out that if the metamodel design satisfies the MECE principle, the classes within the metamodel is distinguished from each other clearly, the model composes a complete set of semantic, and relates to each other clearly. The metamodel requirements of this study are summarized in Table 1.

No.	Metamodel requirements
1	Each class of the metamodel should be mutually exclusive and collectively exhaustive concepts as defined in the axiomatic design theory.
2	Metamodel should be consistent, integrated and balanced among processes and methods to achieve the greatest benefits from the disciplined systems engineering practice.
3	The requirement space and the solution space shall be separated strictly as the systems engineering teaches to ensure the solution space is open for multiple candidates.
4	The attributes of the metamodel should result in effective benefits from the viewpoint of SoS architecting and its usage (e.g. CPM).

Table 1. Metamodel requirements

And the study (Lee & Park, 2009) also proposed five rules for developing metamodel and those metamodel development requirements are presented in Table 2.

No.	Metamodel development requirements
1	Create the minimum number of data groups
2	Do not overlap concept across data groups
3	Make the relation names among groups clear and meaningful.
4	Make the relations among the groups to represent systems engineering methodology.
5	Include the operational viewpoint and system viewpoint category while creating groups.

Table 2. Metamodel development requirements

Current proposed DM2 shows many similar type of classes which violates Lee & Park 's metamodel requirement No.1.

As mentioned before, the metamodel must be consistent, integrated and balanced between process and methods to achieve the greatest benefits from the good systems engineering practice. The systems engineering method teaches that the requirement space and the solution space shall be divided strictly. These attributes of the metamodel resulted in effective benefits from the viewpoint of building architecture (e.g. SoS architecting) and the usage (e.g. CPM).

3.2 Capability Portfolio Management methodology requirements

3.2.1 Capability Portfolio Management requirements

DoDD 7045.20 (DoD, Sep. 2008) defines that capability portfolio management (CPM) is the process of integrating, synchronizing, and coordinating Department of Defense capabilities needs with current and planned DOTMLPF[2] investments within a capability portfolio to better inform decision making and optimize defense resources and capability portfolio is a collection of grouped capabilities as defined by JCAs[3] and the associated DOTMLPF programs, initiatives, and activities. The top level requirement of CPM is that CPMs shall provide recommendations regarding capability requirements to capability investments. And other requirements for recommending capability requirement to the Heads of the DoD Components, and to the Deputy's Advisory Working Group (DAWG) are that the CPM should evaluate capability demand against resource constraints, identify and assess risks, and suggest capability trade-offs within their capability portfolio. DoDD 7045.20 (DoD, Sep. 2008) provides CPM requirements and responsibilities but does not provide process and method.

---

[2] DOTMLPF is an acronym used by the United States Department of Defense. DOTMLPF is defined in the The Joint Capabilities Integration Development System (JCIDS) Process. The JCIDS process provides a solution space that considers solutions involving any combination of doctrine, organization, training, materiel, leadership and education, personnel and facilities (DOTMLPF).

[3] Joint Capability Area (JCA) - Collections of like DOD capabilities functionally grouped to support capability analysis, strategy development, investment decision making, capability portfolio management, and capabilities-based force development and operational planning. (http://www.dtic.mil/futurejointwarfare/cap_areas.htm).

3.2.2 Current status of Capability Portfolio Management methodology

DM2 of DoDAF V2.0 provides Conceptual Data Model (hereafter, CDM), Logical Data Model (LDM), and Physical Exchange Specification (PES). LDM provides data groups (classes) and their usage in DoD's six core processes including CPM. DoDAF V2.0 provides metamodel which support method but does not provide process and methods itself for CPM. Table 3 shows DM2 CDM core concepts which represent the relation among DM2 Data Groups and DoD's six core processes. Table 3 also shows twenty five data groups that are used to develop architectures across DoD's six core processes including CPM.

No	DM2 Data Groups	JCIDS Cap. Mgmt.	JCIDS Iteop. & Supp.	DAS	PPBE	CPM	SE / SOSE	Ops. Planning	Class usage level
1	Activity	6	2	3	3	5	3	5	27
2	Agreement	2	3	3	0	0	3	3	14
3	Capability	6	3	4	3	6	3	3	28
4	Condition	4	2	1	2	2	2	4	17
5	Data	3	4	2	1	3	3	2	18
6	DesiredEffect	6	1	3	0	4	2	6	22
7	Guidance	1	3	3	2	2	3	4	18
8	Information	4	4	2	1	3	3	4	21
9	Location	2	2	1	2	2	1	5	15
10	Materiel	3	1	3	2	2	4	4	19
11	Measure	6	4	4	2	4	3	2	25
12	MeasureType	6	4	4	2	4	3	2	25
13	Organization	2	1	1	2	2	1	5	14
14	Performer	4	4	4	4	4	5	4	29
15	PersonType	3	2	2	1	2	2	4	16
16	Project	0	0	4	2	3	2	3	14
17	Resource	3	3	2	1	3	3	4	19
18	Rule	2	4	3	1	2	4	2	18
19	Service	2	3	3	2	3	5	0	18
20	Skill	3	2	2	1	2	2	4	16
21	Standard	2	6	3	1	2	4	2	20
22	System	2	5	5	5	5	6	3	31
23	Vision	2	0	2	1	3	0	1	9
24	ArchitecturalDescription	4	5	3	2	4	5	3	26
25	Constraint	2	4	3	1	2	4	2	18
	Class usage level	80	72	70	44	74	76	81	-
Legend 6: Critical role, 5: Substantial role, 4: Significant role, 3: Moderate role 2: Supporting role, 1: Minor role, 0: No role									

Table 3. Relation among DM2 Data Groups and DoD's six core processes

The study (Lee & Park, 2009) points out that DoDAF metamodel is too complex to use and proposed more simplified metamodel to enhance usability. Fig. 1 shows many classes used in DM2. There are still many classes which generate complexity when architecting.

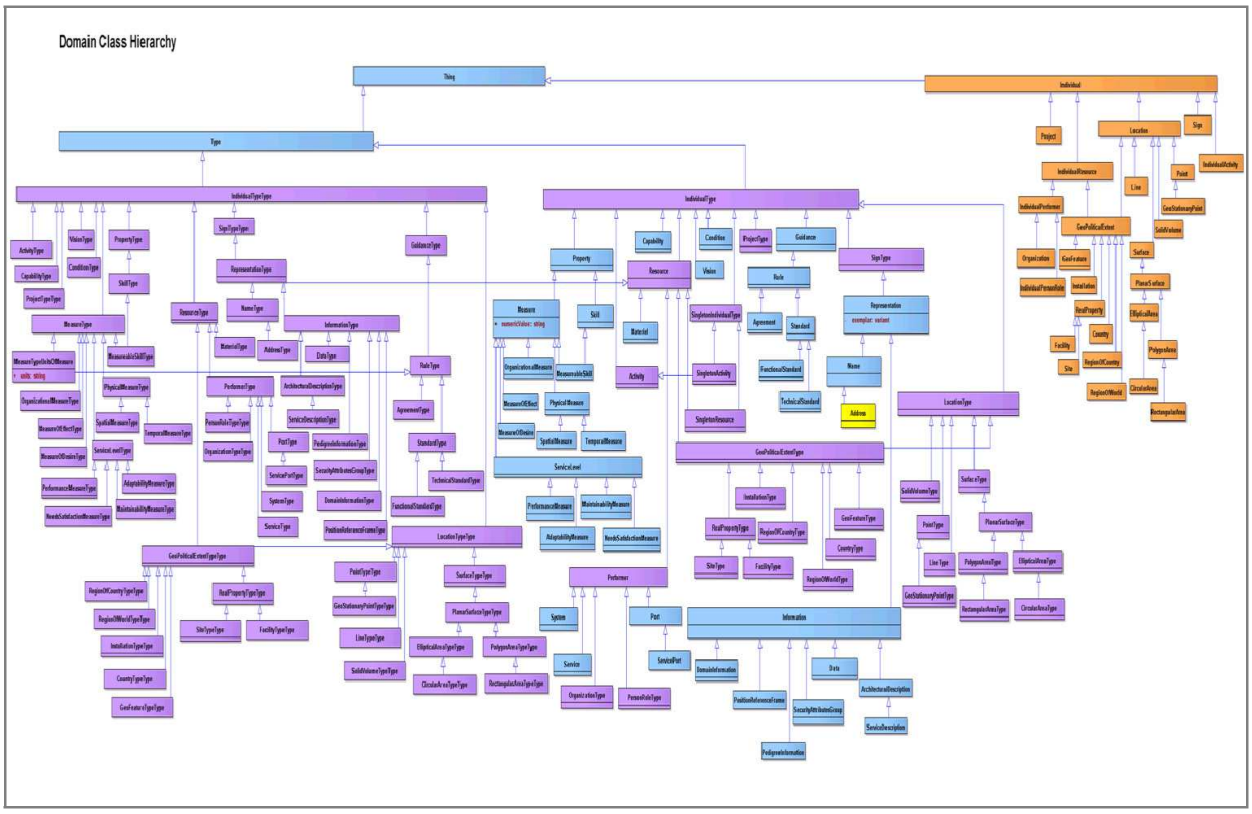


Fig. 1. Notional class hierarchy of DM2

In order to overcome complexity and enhance usability of metamodel, Lee & Park proposed another metamodel based on DoDAF 2.0 JCIDS overlay protocol. Fig. 2 shows Lee & Park's proposal for CDM. The study articulate that the proposed metamodel is the product of an integrating effort that combines the MECE principles, systems engineering principles. The study also demonstrates that it is a simple and effective process to develope and use the artifacts of an architecture.

The CDM of current DM2 of DoDAF V2.0 is similar to the proposed Lee & Park's metamodel. Fig. 3 shows DM2 CDM overlay with the Lee & Park's proposed metamodel. Table 4 shows the relation between classes of DM2 CDM and Lee & Park proposed metamodel. From the contents viewpoint the total of eighteen classes of DM2 CDM are matched with classes of Lee & Park's proposed metamodel. Unmatched classes of DM2 CDM with Lee & Park's are seven classes as follows: Data, Information, Agreement, Location, Skill, MeasureType, and PersonType. To maintain consistency with DM2 CDM, Lee & Park's metamodel complemented with these 7 classes. Three classes of Data, Information, and Location are added, two classes of Agreement and Skill are excluded for the reason of not directly related to the CPM and the other two classes of MeasureType and PersonType go for attribute of Measure and Person. Based on these analysis results the metamodel of CPM methodology could maintain consistency conceptually with DM2 CDM.

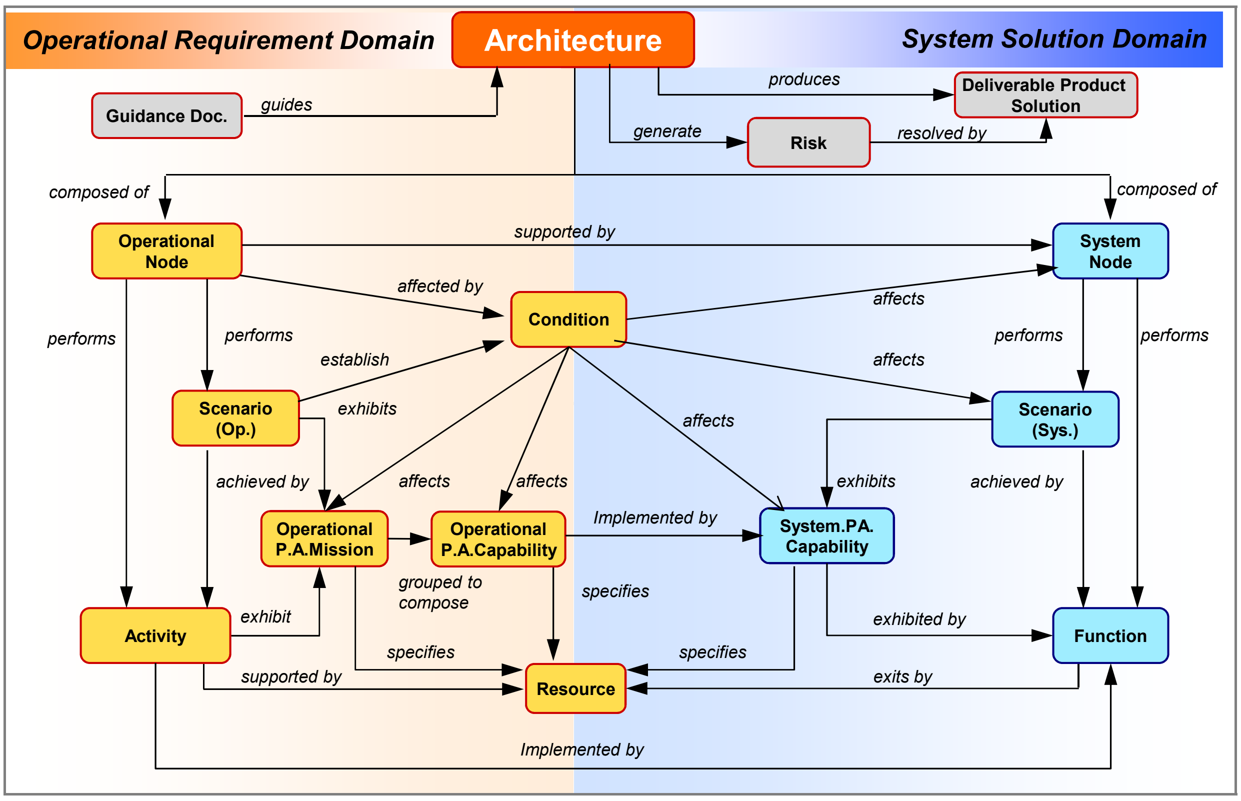


Fig. 2. Lee & Park proposed metamodel for capability based assessment (CBA) methodology

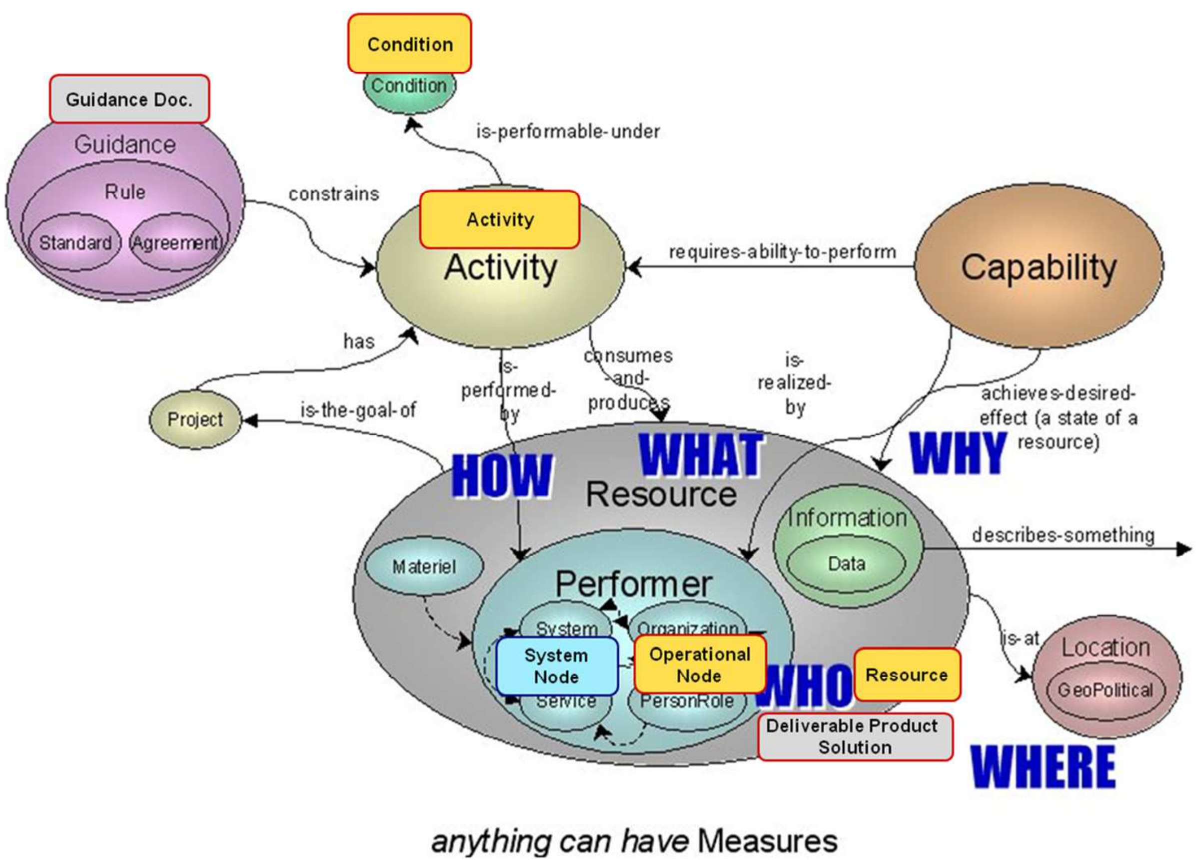


Fig. 3. DM2 CDM overlay with Lee & Park proposed metamodel

DM2 CDM No.	Lee & Park proposed classes / Classes of DM2 CDM core concepts	Operational Node	Scenario	Activity	Op. Perf. Attribute (Oriented)	Op. Perf. Attribute (Capability Oriented)	System Node	Scenario (Sys)	Function	System Perf. Attribute (Capability Oriented)	Architecture	Guidance Document	Risk	Deliverable Product Solution	Condition	Resource	Executables	Ref
13	Organization	○														●		
14	Performer	●																
19	Service		●	○				○	○									
1	Activity			●														
6	DesiredEffect				●													
3	Capability					●				○								
11	Measure				○	○				○								
23	Vision										○							
24	Architectural Description										●							
7	Guidance											●						
21	Standard											●						
10	Materiel													●		●		
22	System													●		●		
17	Resource	○					○							●		●		
16	Project													○				
4	Condition														●			
18	Rule														●			
25	Constraint														●			
2	Agreement																	Class
5	Data																	Class
8	Information																	Class
9	Location																	Class
12	MeasureType																	Attribute
15	PersonType																	Attribute
20	Skill																	Class

Table 4. Relation between classes of DM2 CDM and Lee & Park proposed metamodel

And process viewpoint of methodology status, CBA guides (DoD, Dec. 2006) have relatively detailed information about CBA process and methods. The CBA process and related information could be used to perform CPM but that is not sufficient for CPM method. The following part provides CPM process, product and method which manipulate information of the product and supporting metamodel.

4. Proposal of Capability Portfolio Management methodology

As mentioned before, a methodology specifies the process to be executed, usually as a set of related activities, tasks and/or techniques, together with work products possibly including models, documents. CPM methodology has its own process, method (technique), and product (model or artifacts) as the tool category of Martin's PMTE. According to these requirements, the CPM methodology of this chapter shows CPM process, product and model related technique. The CPM process consists of a set of activities/tasks. Each step of activity has corresponding output product and model related technique which is used to build model and/or generate the output products.

In order to facilitate further discussions, key terms quoted from DoDD 7045.20 capability portfolio management (DoD, Sep. 2008) are defined as follows. Capability portfolio is a collection of grouped capabilities as defined by JCAs and the associated DOTMLPF programs, initiatives, and activities. And CPM is the process of integrating, synchronizing, and coordinating capability requirements with current and planned DOTMLPF investments within a capability portfolio to better inform decision making and optimize defense resources. From this definition, CPM can make a balanced capability requirements to maximize mission effects within limited resources and the capability requirements are originated from a group of capabilities defined by JCAs.

4.1 Capability Portfolio Management process

CPM requirement is to provide recommendations regarding capability requirements to capability investments. So CPM process has to generate balanced capability requirements. The capability requirements should be generated with DOTMLPF investments within a capability portfolio (a collection of grouped capabilities as defined by JCAs).

To achieve these CPM requirements a proposed process is composed of following 5 activities: (1) Define top level missions and develop scenarios (2) Build trace relation among elements of JCA, universal joint task list (hereafter, UJTL) and activity and identify mission essential task list (hereafter, METL) of DoD (3) Develop capabilities and the related conditions and resources (4) Analyze mission effectiveness and derive (transform) capability requirements (5) Derive integrated & balanced capability requirements. And more detailed tasks are listed in Table 5.

4.2 Capability Portfolio Management method and product

In order to provide CPM method and product which could be a model or artifact. This part provides descriptions, products and model related techniques for each task of CPM process.

Activities of CPM process		Tasks of CPM Process	
A.1	Define top level missions and develop scenarios	T.1	Define top level missions
		T.2	Define states & modes for each missions
		T.3	Develop mission threads for each states & modes
		T.4	Design operational scenarios for each missions
A.2	Build trace relation among JCA, UJTL and activity and identify METL	T.5	Trace each activity to UJTL
		T.6	Check alignment JCA, UJTL and allocated activity
		T.7	Identify METLs for each mission scenario
A.3	Develop capabilities and related conditions and resources	T.8	Develop capability instance which aligned to activity (attributed in METLs)
		T.9	Develop condition instances for each activity (attributed in METLs)
		T.10	Develop resource instances for each activity (attributed in METLs)
A.4	Analyze mission effectiveness and derive(transform) capability requirements	T.11	Analyze operational effectiveness (MOEs) for each operational missions (e.g. Joint Operating Concepts, hereafter, JOC)
		T.12	Analyze operational effectiveness (MOEs) for functional missions (e.g. Joint Functional Concepts, hereafter, JFC)
A.5	Derive integrated & balanced capability requirements	T.13	Allocate operational element to supporting systems element
		T.14	Synthesize operational performances to system performances
		T.15	Optimize resources to maximize MOEs for a capability
		T.16	Define integrated capability requirements

Table 5. Activities and tasks of proposed CPM process

4.2.1 Define top level missions

- Description: Defining top level mission is a process to define top level missions of an enterprise to provide the point of reference or directions which CPM aims to attain.
- Product: Top level mission statement of an enterprise
- Model related technique: Mission is a kind of activity and the mission activity is the top level activities of an activity hierarchy. And, the level attribute of the mission activity should be set as 'Mission level'.

4.2.2 Define states & modes for each mission

- Description: Defining states & modes for each mission is a process to define states and modes in which top level missions of an enterprise are implemented, and this process provides categories (e.g. war-time & piece-time) of thinking for developing threads which encompass tasks to be analyzed.
- Product: States & modes definition
- Model related technique: States & modes are kind of activity and the abstraction level of this activity is below the mission activities of an activity hierarchy. Thus the level attribute of the states & modes activity should be set as 'States & modes'.

4.2.3 Develop mission threads for each states & modes

- Description: From CJCSI 6212.01E (DoD, Dec. 2008) definition, a mission thread could be defined as an operational and technical description of the end to end set of activities and systems that accomplish the execution of a mission. Developing mission threads means producing a list of threads which needed for each states & modes to accomplish a mission. Each thread composed of a series of required tasks.
- Product: Mission threads
- Model related technique: Mission threads are kind of activity and this activity is the below the states & modes activities of an activity hierarchy. And so, the level attribute of the thread activity should be set as 'Thread'.

4.2.4 Design operational scenarios for each mission

- Description: Designing operational scenarios for each mission is a process to define a series of activities in each thread. Through this process, the end to end sets of activities of operational nodes are designed, and states of mission accomplishments are designed by integrating those threads
- Product: Operational scenario template
- Model related technique: The activities within an operational scenario are kind of activity and these activities are the leaf-node activities of an activity hierarchy. Thus the level attribute of the leaf-node activity should be set as 'Leaf-node'. The leaf-node activity could directly allocate to supporting entity e.g. operational node and system node.

4.2.5 Trace each activity to Universal Joint Task List

- Description: Tracing each activity to UJTL is a process allocating each activity to the related tasks listed in UJTL which contains information that identifies conditions and standards to analyze Leaf-node-level activities
- Product: Activity to UJTL traceability table
- Model related technique: UJTL class is required and the elements of UJTL class (tasks) are allocated by leaf-node activities of scenario.

4.2.6 Check alignment Joint Capability Area, Universal Joint Task List and allocated activity

- Description: The Joint Capability Area Management System (JCAMS) of DoD provides JCA linkages to Universal Joint Tasks. The allocated activities to UJTL should be checked by the alignment with JCA from the viewpoint of semantics. From the viewpoint of semantics, tracing relation between activity-UJTL-JCA should be meaningful.

- Product: Traceability table of Activity-UJTL-JCA
- Model related technique: JCA class is required and the elements (contents) of JCA could be traceable to UJTL. Then the traceability from leaf-node activity via UJTL to JCA is established.

4.2.7 Identify Mission Essential Task Lists for each mission scenario

- Description: METLs are decided through a process to identify key tasks, which directly contribute to achieve mission effectiveness, among leaf-node level activities of a mission scenario. The designated activities as METL have a role to develop capability requirements. The activities designated as METL are base activities for following analysis of CPM methodology.
- Product: METL List
- Model related technique: The activity class needs the importance attribute. And so, the activity importance attribute of the METL activity should be set as 'METL'.

4.2.8 Develop capability instance which aligned to activity

- Description: The activities which are identified as METLs should be carried out CBA separately and develop appropriate capabilities in the light of JCAs. The developed capability is an instance of capability class which are traced to activity instances. The developed capabilities will be traced to the functions of systems or other requirements of DOTMLPF.
- Product: Traceability table of Activity - Capability
- Model related technique: Capability class is required aside from JCA class and the capability class should have relation with JCA and activity class.

4.2.9 Develop condition instances for each activity

- Description: For the purpose of carrying out CBA, proper conditions for missions are developed and allocated to activities which are identified as METLs. The developed conditions are instances of the conditions of UJTL.
- Product: Traceability table of Activity – Condition
- Model related technique: Condition class is required aside from UJTL class and the UJTL class has 'provide relation' with Condition class.

4.2.10 Develop resource instances for each activity

- Description: Required resources (DOTMLPF) are defined to fulfill relevant activities. Those resources realize capabilities to support related activities.
- Product: Traceability table of Activity – Resource – Capability
- Model related technique: Resource class is separately required with other performer type classes e.g. organization and system. The resource class has relation with activity and capability class. Resource class has resource type of DOTMLPF. Especially the Resource class typed with organization is equivalent to organization class and resource class typed with materiel is equivalent to system class.

4.2.11 Analyze operational effectiveness for each operational mission

- Description: Performance levels for each activity are analyzed to estimate the performance level of activities to produce best mission effectiveness within constrained resources under the given condition for activities to accomplish operational missions. It is required to determine performance indicators or standards of each activity to achieve mission effectiveness. Especially this analysis works are performed in aspect of operational missions in this step. And so, the performance levels of activities for each operational mission (e.g. JOC) are measurements of operational effectiveness (MOEs).
- Product: Operational mission effectiveness and performance of activity table
- Model related technique: Within activity class, mission level attributed activity element could have sub attribute of operational mission type and functional mission type. And to display and analyse the performance of scenario, a performance measure class is required. And according to the type of activity of mission scenario, the attribute of a measure could be operational effectiveness (reflect JOC), functional performance (reflect JFC) or system performance.

4.2.12 Analyze operational effectiveness for functional missions

- Description: Performance levels for each activity are analyzed to estimate optimized performance level of activities which produce best operational mission effectiveness within 'constrained resources' which support activities to accomplish functional missions. From the viewpoint of opposite direction, the performance level of activities performing a functional mission should be optimized to enhance the total performance of the activities performing various operational missions. This opposite directional task will be explained at 4.2.15 again. It is required to determine performance indicators or standards of each activity to achieve mission effectiveness. Especially in this step, the analysis works are performed from the viewpoint of functional missions relative to the operational missions. And so, the performance levels of activities for each functional mission (e.g. JFC) are measurements of operational effectiveness (MOEs).
- Product: Functional mission effectiveness and performance of activity table
- Model related technique: Within Activity class, mission level attributed Activity element could have sub attribute of operational mission type and functional mission type. And to display and analyse the performance of scenario, a Performance Measure class is required. And according to the type of activity of mission scenario, the attribute of a measure could be operational effectiveness (reflect JOC), functional performance (reflect JFC) or system performance.

4.2.13 Allocate operational element to supporting systems element

- Description: This phase changes operational viewpoint to system viewpoint. And this phase allocates defined organization, operational nodes, activities and input/output information to systems, system nodes, system functions and input/output data. Lessons learned from systems engineering imply that system elements are not considered before this step, and instead, requirements are defined in operational viewpoint, then operational requirement are converted into system viewpoint in order to support operational requirements.

- Product: Organization vs. system relation table, Operational node vs. system node relation table, operational activity vs. system function relation table, Operational information vs. system data relation table.
- Model related technique: To reflect the principle of systems engineering which divide requirement space and the solution space strictly, the following relations are built. Organization class is supported by system class. Operational node class is supported by system node class. Activity class is supported by function class. Operational information class is supported by system data class.

4.2.14 Synthesize operational performances to system performances

- Description: This step aims that operational performances, which are derived from operational activities, are changed to system performance, which are derived from system functions. A system function is employed to support several operational activities. Those operational performances are synthesized into an optimized system performance from the view point of cost-effectiveness.
- Product: Operational performances vs. system performances matrix
- Model related technique: Measure class of operational performances type is traced to measure class of system performances type

4.2.15 Optimize resources to maximize operational effectiveness for a capability

- Description: This is the most peculiar phase of CPM process. Perform cost-effectiveness analysis repeatedly to achieve maximum effectiveness under the condition of limited resources. And define the capability requirements, which are the requirements for all resources to encompass DOTMLPF, and those resources are traced to one capability under certain items of JCA. The resulted performances of resources are synthetically maximize return on invest (ROI) for the relevant capability.
- Product: Capability vs. Resources matrix
- Model related technique: Capability class has been realized by relation with DOTMLPF type of resources.

4.2.16 Define integrated capability requirements

- Description: According to the definition of capability, the capability elements e.g. desired effects of various missions, a set of tasks and combination of means & ways are defined for a capability using performance measures of activity, function and resources. The elements contributing critically to the resulted capability should be marked.
- Product: Capability recommendation document
- Model related technique: 'Capability decisive element' attribute required for classes of resource, activity and function.

4.3 Metamodel for Capability Portfolio Management

From the proposed CPM process, and based on DM2 CDM and Lee & Park's metamodel, the additionally required classes (Entity type), attributes of classes and relations for each task are identified. The additionally identified classes, relations, and attribute are used to complement metamodel for CPM methodology. Table 6 shows the additionally required classes, relations and attributes.

No.	Tasks of CPM Process	Required class	Required attribute & relation
T.1	Define Top Level Missions	Activity	Activity_type-attribute: Mission
T.2	Define States & Modes for each missions	Activity	Activity_type-attribute: State&Mode
T.3	Develop Mission Threads for each States & Modes	Activity	Activity_type-attribute: Thread
T.4	Design Operation Scenarios for each missions	Activity	Activity_type-attribute: Scenario
T.5	Trace each Activity to UJTL	UJTL	UJTL traces_to Activity relation
T.6	Check alignment JCA, UJTL and allocated Activity	JCA	JCA categorize UJTL relation
T.7	Identify METLs for each mission scenario	UJTL	Activity_type-attribute: METL
T.8	Develop Capability instance which aligned to Activity (attributed in METLs)	Capability	Capability requires_ability_to_perform Activity relation
T.9	Develop Condition instances for each Activity (attributed in METLs)	Condition	Activity performable_under Condition relation
T.10	Develop Resource instances for each Activity (attributed in METLs)	Resource	Resource _type-attribute: DOTMLPF, Capability realized_by Resource relation
T.11	Analyze Operational Effectiveness (MOEs) for each operational missions (e.g. JOC)	Measure	Measure_type-attribute: MOE for JOC
T.12	Analyze Operational Effectiveness (MOEs) for functional missions (e.g. JFC)	Measure	Measure_type-attribute: MOE for JFC
T.13	Allocate operational element to supporting systems element	Data	Organization supported_by System relation, OperationalNode supported_by SystemNode relation, Activity supported_by Function relation, Information supported_by Data relation
T.14	Synthesize operational performances to system performances	Measure	Measure_type-attribute: Operational performance Measure_type-attribute: System performance Measure traces_to Measure relation
T.15	Optimize resources to maximize MOEs for a capability	Capability	Capability realized_by Resource relation
T.16	Define integrated capability requirements	Document	Document documents AllClass

Table 6. Proposed CPM Process and required classes and attributes for CPM Process

Like the study (Lee & Park, 2009) proposed metamodel, CPM metamodel should be developed in accordance with the metamodel requirement and metamodel development requirements of Table 1 and 2. And also CPM metamodel should be aligned with DM2 CDM for interoperability with DoDAF V2.0. Table 7 shows the proposed CDM classes for CPM which is aligned with classes of DM2 CDM and additionally added classes originated from Lee & Park' metamodel and the CPM task analysis. The additional JCA and UJTL classes comes from CPM task analysis of Table 6 and System Node and Function classes reflect the systems engineering concept of strict separation of requirement space and solution space.

DM 2 No.	Classes of DM2 CDM core concepts	CPM usage level	relation with proposed classes	Proposed CDM classes for CPM
3	Capability	6	correspond to	Capability
1	Activity	5	correspond to	Activity
22	System	5	correspond to	System
6	DesiredEffect	4	correspond to	Measure (Effect attributed)
11	Measure	4	correspond to	Measure
12	MeasureType	4	correspond to	Measure (MeasureType attributed)
14	Performer	4	correspond to	Operational Node
24	Architectural Description	4	correspond to	Architecture
5	Data	3	correspond to	Data
8	Information	3	correspond to	Information
16	Project	3	correspond to	Project
17	Resource	3	correspond to	Resource
19	Service	3	correspond to	Activity (Service attributed)
23	Vision	3	correspond to	Measure (Vision attributed)
4	Condition	2	correspond to	Condition
7	Guidance	2	correspond to	Guidance
9	Location	2	correspond to	Location
10	Materiel	2	correspond to	Resource (Materiel attributed)
13	Organization	2	correspond to	Resource (Organization attributed)
15	PersonType	2	correspond to	Resource (Person attributed)
18	Rule	2	correspond to	Guidance
25	Constraint	2	correspond to	Condition
20	Skill	2	correspond to	Resource (Skill attributed)
21	Standard	2	correspond to	Guidance
2	Agreement	0	correspond to	Guidance
-	N/A	-	-	System Node
-	N/A	-	-	Function
-	N/A	-	-	JCA
-	N/A	-	-	UJTL

Table 7. Relation between DM2 CDM core concepts and Lee's CDM classes for CPM

And according to the metamodel development requirements, classes are related and named meaningfully and reflect operational requirement space and system solution space. Fig. 4 shows the resulted CDM for CPM methodology.

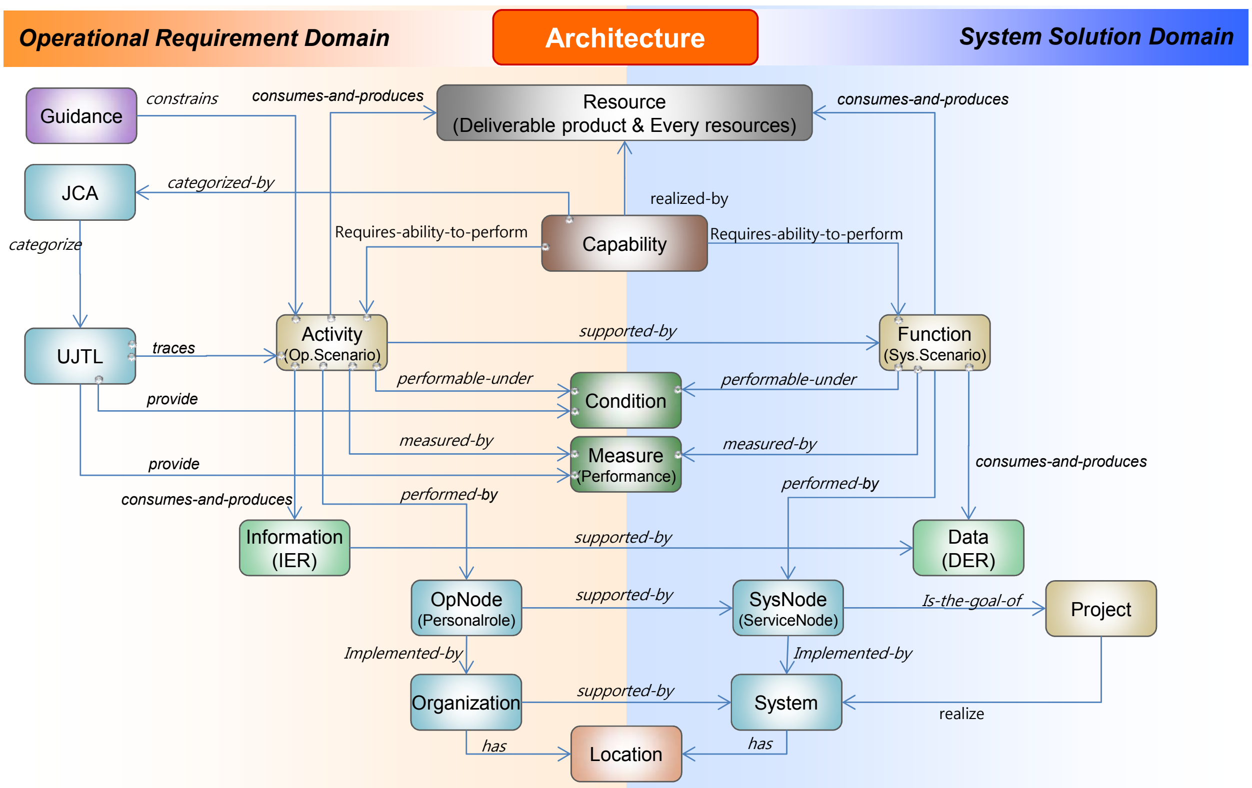


Fig. 4. Proposed CDM for CPM methodology

5. Conclusion

The purpose of this paper is to provide an abstracted metamodel for use in CPM effectively based on DoDAF V2.0. The proposed CPM methodology provides a process, tasks of the process, products, and model related technique which supports the generation of products in accordance with the methodology definition of ISO/IEC 24744. To promote the usability, the proposed methodology suggest a detailed CPM process. Additionally, in order to be an effective and efficient methodology, the CPM metamodel is developed in accordance with the MECE principles, systems engineering principles which was proposed earlier by Lee & Park's metamodel requirements. And to obtain the interoperability with DoDAF V2.0, the proposed CPM methodology is developed in accordance with DM2 CDM.

However, the current proposed abstracted metamodel remains on a theoretical and logical level and requires validation experimentally or in field applications. In the near future, the proposed metamodel must be validated for application use. However, the proposed CPM methodology is expected to be helpful in practice in the field.

6. References

DoD (Dec., 2006), *Capabilities-Based Assessment (CBA) User's Guide*, Ver.2, Sep. 25, 2011, Available from: www.dtic.mil/futurejointwarfare/strategic/cba_guidev2.pdf

DoD (Sep., 2008), *DoDD 7045.20 capability portfolio management*, Sep. 25, 2011, Available from: http://www.dtic.mil

DoD (Dec., 2008), *CJCSI 6212.01E Interoperability and supportability of information technology and national security systems*, , Sep. 25, 2011, Available from: https://acc.dau.mil

DoD (Aug., 2010), *The DoDAF Architecture Framework*, Ver. 2.02, Sep. 25, 2011, Available from: http://cio-nii.defense.gov/sites/dodaf20/products/DoDAF_v2-02_web.pdf

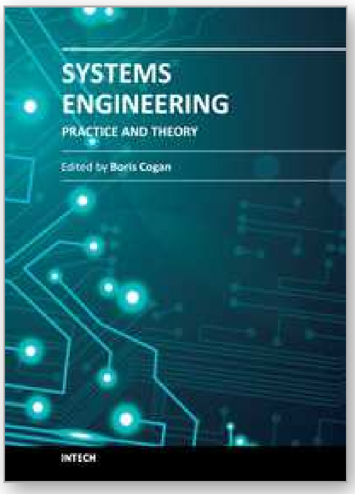
Federal Government (November 2007), *FEA Practice Guidance*, Available from: www.whitehouse.gov/sites/default/files/omb/assets/fea_docs/FEA_Practice_Guidance_Nov_2007.pdf

ISO (2007), *ISO/IEC 24744 Software Engineering - Metamodel for Development Methodologies*, ISO, Retrieved from: www.iso.org/iso/iso_catalogue.htm

Lee, J. & Park, Y. (2009), A Study on the Abstracted Metamodel of DoDAF 2.0 for CBA Methodology Execution, *International Conference on Software Engineering, 2009 10th ACIS Artificial Intelligences, Networking and Parallel/Distributed Computing*, ISBN: 978-0-7695-3642-2, Daegu, Korea, May 27- 29, 2009

Martin, J. (1997), Process concept, In : *Systems engineering guidebook: a process for developing systems and products*, pp. 51-56, CRC, ISBN 0-8493-7837-0, Boca Raton

Suh, N. (1990), Design axioms and corollaries, In: *The Principles of Design*, pp.47-48, Oxford University Press, ISBN 0-19-504345-6, New York



Systems Engineering - Practice and Theory

Edited by Prof. Boris Cogan

ISBN 978-953-51-0322-6

Hard cover, 354 pages

Publisher InTech

Published online 16, March, 2012

Published in print edition March, 2012

The book "Systems Engineering: Practice and Theory" is a collection of articles written by developers and researches from all around the globe. Mostly they present methodologies for separate Systems Engineering processes; others consider issues of adjacent knowledge areas and sub-areas that significantly contribute to systems development, operation, and maintenance. Case studies include aircraft, spacecrafts, and space systems development, post-analysis of data collected during operation of large systems etc. Important issues related to "bottlenecks" of Systems Engineering, such as complexity, reliability, and safety of different kinds of systems, creation, operation and maintenance of services, system-human communication, and management tasks done during system projects are addressed in the collection. This book is for people who are interested in the modern state of the Systems Engineering knowledge area and for systems engineers involved in different activities of the area. Some articles may be a valuable source for university lecturers and students; most of case studies can be directly used in Systems Engineering courses as illustrative materials.