Journal of Defense Modeling and Simulation: Applications, Methodology, Technology
XX(X):1–19
©The Author(s) 2015
DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Architectural work for Modeling and Simulation combining the NATO Architecture Framework and C3 Taxonomy

Jo Erskine Hannay[1]

Abstract

To provide modeling and simulation functionality as services is strategically leveraged in the defense domain and elsewhere. To describe and understand the context—the ecosystem—wherein such services are used and interoperate with other services and capabilities, one needs tools that capture the simulation services themselves as well as the capability landscape they operate in. By using the NATO Consultation, Command and Control (C3) Taxonomy to structure architecture design in the NATO Architecture Framework (NAF), cohesive descriptions of modeling and simulation capabilities within larger contexts can be given. We show how a basic seven-step approach may benefit architecture work for modeling and simulation at the overarching, reference and target architectural levels; in particular for (1) hybrid architectures that embed simulation architectures within a larger service-oriented architecture, and (2) for architectural design of simulation scenarios. Central to the approach is the use of the C3 Taxonomy as a repository for overarching architecture building blocks and patterns. We conclude that the promotion of technical functionality as capabilities in their own right helps delineate simulation environment boundaries, helps delineate services within and outside the boundary and is an enabler for defining the service concepts in cloud-based approaches to Modeling and Simulation as a Service (MSaaS).

Keywords

Taxonomy of Capabilities, Architecture Framework, Reference Architecture, Modeling and Simulation as a Service

1 Introduction

The idea of providing modeling and simulation (M&S) functionality as services raises a number of questions as to exactly what is to be provided; for example, what units of simulation functionality are to be provided to service consumers, what it is to be a simulation service and what is not such a service, and what the boundaries are for a simulation environment[1] in the context of a larger service-oriented federation of systems.

We argue that a capability-based approach to architecture modeling can facilitate decisions on such questions, as well as help structure architectural work for M&S in general. To this end, we will combine the NATO Consultation, Command and Control (C3) Taxonomy[2] of capabilities and the NATO Architecture Framework (NAF)[3].

The concept of 'capability' embodies a notion of persistence relative to other capabilities as well as relative to underlying implementations. This decouples pieces of functionality from each other and decouples functionality from the means to provide it. The intention is that functionality can be described and composed at the abstract level via abstract requirements, interfaces and contracts; which can be done more readily and rapidly than directly at the implementation level. To achieve interoperability over different and changing needs, NATO will modernize its systems portfolio in terms of capabilities. Consequently, capabilities are foundational to the Connected Forces Initiative (CFI)[4;5] which focuses on NATO forces' ability to work together and with partners in complex operations across a variety of environments. The information-management and -technological aspect of CFI is embodied in Federated Mission Networking (FMN), which focuses on command, control and decision-making in operations through improved information-sharing[6].

---

[1] Norwegian Defence Research Establishment (FFI)

Corresponding author:
Jo E. Hannay, FFI, Pb. 25, NO-2027, Norway.
Email: jo.hannay@ffi.no, ph.: +47 63807454, fax: +47 63807115

The complex landscape of the defense domain generates a need to organize the development of functionality into manageable parts[7–9], with a focus on capabilities in order to align with FMN and so-called Capability-Based Planning (CBP)[10]. The C3 Taxonomy incorporates *operational* and *technical* capabilities in the same framework, and it can be used to organize and discipline the development of operational and technical functionality into loosely coupled capabilities and services[9]. NAF on the other hand, has a notion of capability, but the focus is on operational capabilities, whereafter the technology that gives support to the capability is seen as implementation; not as a capability in itself. This difference in how technical functionality is treated entails a mismatch between NAF and the C3 Taxonomy. Both the C3 Taxonomy and NAF are strategically leveraged tools in NATO, and it is vital that the two tools can be used seamlessly in concert. In particular, for better architectural work, we promote the use of the C3 Taxonomy as a repository of architecture building blocks and architecture patterns[11], and it is important to enable the use of such building blocks and patterns directly in NAF diagrams. We therefore outline how to combine the C3 Taxonomy and NAF into a cohesive framework and use the framework in the ensuing discussion for M&S.

We will outline how the framework may be used to structure architecture development for hybrid architectures that embed simulation architectures within a larger service-oriented architecture[11], and we will outline how the framework may aid architectural design of simulation scenarios. Through these two cases, we discuss how the framework may facilitate capability-based decisions on what functionality one wishes to provide as simulation services, capability-based decisions on what the notion of simulation environment might encompass, as well as demonstrate the usefulness of the framework for M&S architecture work in general.

The next three sections outline the C3 Taxonomy, NAF and other architecture notions. Then, Sections 5 and 6 outline how NAF can be extended to cater for the capability structure of the C3 Taxonomy; and subsequently, a seven-step approach is outlined to how the extension may be used to model architecture at the overarching architecture, reference architecture and target architecture levels. Sections 7 and 8 show how the approach may be used for M&S. We conclude in Section 9.

2 The C3 Taxonomy

While strategic planning earlier had a focus on assets (specific material and resources), the focus is now on capabilities. According to The Open Group Service-Oriented Architecture (SOA) Reference Architecture (RA), a capability “represents a requirement or category of requirements that fulfill a strongly cohesive set of needs”[12]. Capabilities are stable over longer periods of time and thereby allow for better planning. The C3 Taxonomy enables the defense community to sort C3 functionality into capabilities; see Fig. 1 for an abstract view. Fig. 2 shows a more detailed view. Central to our discussion is that the taxonomy explicitly declares Operational Capabilities and Communication and Information Systems (CIS) Capabilities; that is, CIS support to operational capabilities is promoted as capabilities in their own right, rather than being seen as implementation specific to given operational capabilities.

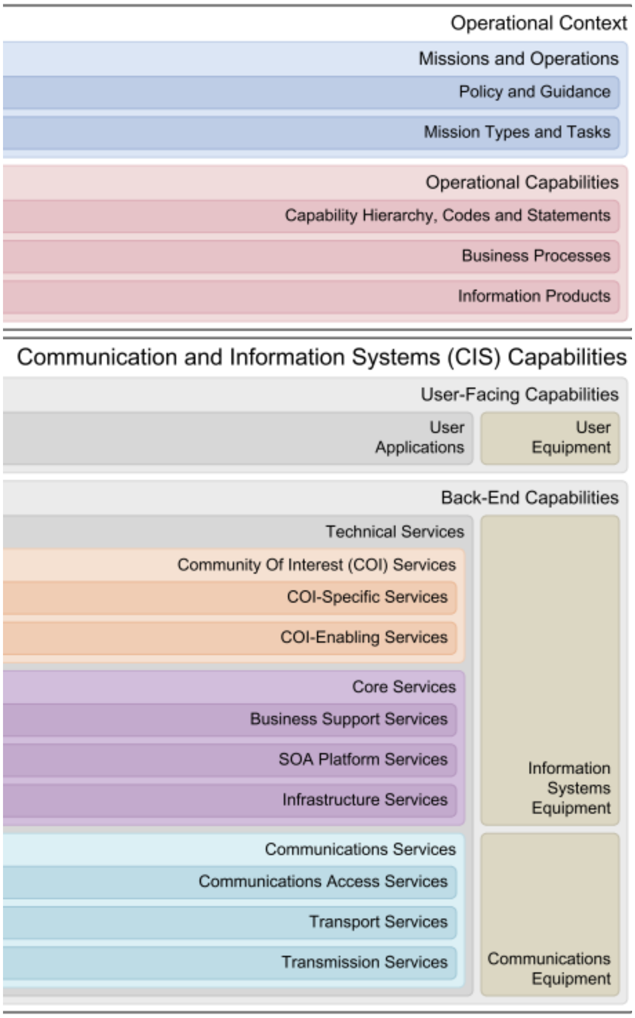

Figure 1. C3 Taxonomy—top-level view[2]

At the operational level of the taxonomy, Business Processes is the central set of capabilities for our discussion. These human-based processes are defined as independent of technology. CIS support for Business Processes presents itself to end users in the form of User-Facing Capabilities geared toward User Applications for specific domains (air, land, maritime, joint, etc.) and communities of interest (modeling and simulation, environment, missile defense,

DOI til publisert versjon / DOI to published version: 10.1177/1548512916670785

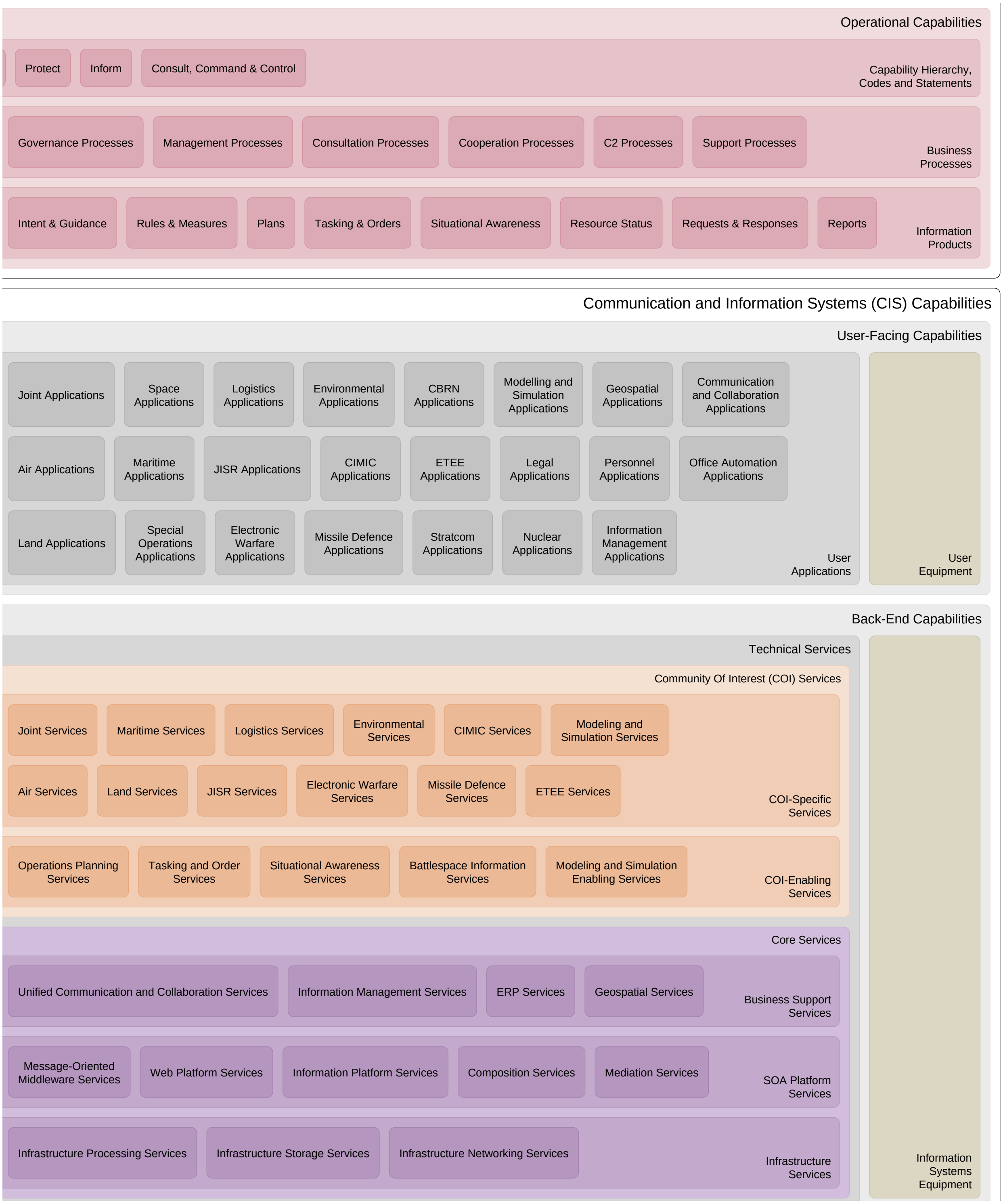

Figure 2. C3 Taxonomy; detail of Operational Capabilities, Front-End Capabilities and Back-End capabilities)[2]

etc.). The Back-End Capabilities may be used to support the user-facing capabilities and are layered into Community of Interest (COI) Services—subdivided into COI-Specific Services and the more generic COI-Enabling Services—and the Core Services and Communication Services layers, both of which provide generic infrastructure capabilities.

DOI til publisert versjon / DOI to published version: 10.1177/1548512916670785

- – Operational Capabilities
    - – Business Processes
        - – C2 Processes
            - – Strategic Communications Processes
            - – Joint C2 Processes
            - – Land C2 Processes
                - – Land Operations Planning Processes
                    - – Land Appreciation and Assessment of Options Process
                    - – Land Orientation Process
                    - – Land CONOPS Developement Process
                        - + Land Guidance Gathering and Reviewing Process
                        - + Land Enemy COAs Development Process
                            - x Determine Enemy Most Likely COA
                            - x Determine Enemy Most Dangerous COA
                        - + Land COAs Development Process
                            - x Develop Tentative COAs
                            - x Consolidate and Synthesise Related COAs
                        - + Land COAs Analysis Process
                            - x Test for Suitability
                            - x Test for Acceptability
                            - x Test for Feasibility
                            - x Test for Exclusivity
                            - x Test for Completeness
                            - x Test for Compliance with NATO Doctrine
                        - + Land COAs Comparison Process
                            - x Compare COAs against Commander's Selection Criteria
                            - x Compare Friendly and Opposing COAs
                            - x Compare COAs Advantages and Disadvantages
                        - + Land Decision Brief Process
                        - + Land CONOPS Production Process
                    - – OPLAN and OPORD Development Process
                    - – Execution Assessment and OPLAN Review Process
                    - – Land Transition Process
            - – Joint Targeting Processes
            - – Time Sensitive Target Processes
            - – Joint ISR Processes
            - – Intelligence Processes
            - – Operations Planning Processes
            - – Air C2 Processes
            - – Special Operations C2 Processes

Figure 3. C3 Taxonomy: Operational Capabilities breakdown for Land Operations Planning Process

There are cross-cutting Information Assurance (IA) and Service Management and Control (SMC) groupings (not shown) that hold functionality, respectively, for safety and security and for service discovery, mediation and quality of service, etc.

The capability taxonomy is hierarchical. For example, at the Operational Capabilities level, one finds the Land Operations Planning Processes as capabilities under C2 Processes in the Business Processes layer as seen in Figure 3. The individual processes (Land Appreciation...Process, Land Orientation Process, etc.) are leaf nodes of the capability breakdown. In a leaf node one finds the overall purpose of that capability and requirements for the capability. Figure 3 shows this capability break-down with two levels of requirements. Capabilities are prefixed by "-", while abstract requirements are prefixed by "+" and the more detailed requirements are prefixed by"x". In general, the capability breakdown structure of the C3 Taxonomy is as follows:

- – <Capability>
    - – <Capability>
        - – ...
            - – <Capability>
                - + <Abstract Requirement>
                    - x <Detailed Requirement>

We will see examples of the capability breakdown structure in User-Facing Capabilities and Back-End Capabilities shortly.

In terms of user stories, the abstract and detailed requirements are here on the level of *epics*— high-level user stories to be elaborated and refined into even more detailed *stories* for production in, e.g., a product backlog[9].

The Taxonomy also holds patterns that suggest how to combine capabilities.

The C3 Taxonomy is work in progress, and there is currently varying levels of detail for the various capabilities. This will be apparent when we apply our approach later, where missing requirements will indicate points to initiate requirements elicitation and innovation. Further, the C3 Taxonomy will likely always be in flux to mirror evolving understanding of capability needs.

3 The NATO Architecture Framework

The NATO Architecture Framework (NAF)[3] provides guidelines on how to describe and document an architecture. It structures architectural work in views:

AV All View—captures overarching aspects of architectures that relate to all views.

CV Capability View—captures analysis of the delivery of military capabilities in line with strategic intent.

OV Operational View—describes activities, operational elements, and information exchanges required for missions.

SOV Service-Oriented View—describes services needed to directly support the operational domain as described in OV.

SV Systems View—describes systems and connections providing for or supporting processes and associates systems resources to the OV and/or the SOV to support the operational activities and facilitate the exchange of information among operational nodes as defined in the OV.

TV Technical View—provides the technical implementation guidelines upon which engineering specifications are based and common building blocks are established.

PV Programme View—describes the relationships between capability requirements and the various programs and projects being implemented.

For our discussion, the CV, OV, SOV and SV are relevant.

4 Architecture types

In NAF, various architecture types are specified:

An *overarching architecture* is a description of the desired configuration of the NATO C3 system necessary to meet NATO's medium to long-term (up to 15 years) capability requirements[3]. We regard the C3 Taxonomy's capabilities— in various shades of detail and refinement according to the capability break-down structure—as *architecture building blocks* (ABBs)[12]. The patterns in the taxonomy then amount to *architecture patterns* (APs)[12].

Dette er en postprint-versjon / This is a postprint version.
DOI til publisert versjon / DOI to published version: 10.1177/1548512916670785

Table 1. Mapping from C3 Taxonomy to NAF

C3 Taxonomy	*NAF*
Capability	Capability
Abstract Requirement [Operational|User-Facing|Back-End] level	Standard [Operational|User-Facing|Back-End] Activity
Detailed Requirement [Operational|User-Facing|Back-End] level	[Operational|User-Facing|Back-End] Activity

The C3 Taxonomy is then a repository of C3 ABBs and APs and is a an overarching architecture for our purposes[11].

A *reference architecture* should support the development of capability packages[3]. A reference architecture is an implementation-independent perspective that captures operational business processes, information products, user requirements, interface specifications and logical architectural patterns[13]. For us, this means an assembly of C3 Taxonomy ABBs (guided by APs) for a particular kind of purpose[11].

A *target architecture* is derived from the related reference architecture and specifies a design at a detail sufficient to direct the acquisition and integration of components to achieve a desired capability. It modifies the relevant baseline architecture and can provide feedback to a reference architecture[3]. One might find that the C3 Taxonomy gives examples of target architecture elements; such as concrete software systems that provide a capability (e.g., systems that provides services), but the main objective of the taxonomy is to give declarations in the overarching and reference architecture perspectives.

5 Mapping the C3 Taxonomy into NAF

The meta model and structure of NAF is in disharmony with that of the C3 Taxonomy. NAF was not designed originally for service orientation and mirrors a more traditional systems architecture style, even though service-oriented views have been added. NAF is suitable for developing operational capabilities, where underlying implementations (capability configurations) are developed with a particular capability in mind. This does not support loose coupling between operational and technical levels. In contrast, the C3 Taxonomy's structure supports the definition and development of loosely coupled and persistent capabilities at both operational and technical levels[9].

To induce that structure in NAF, we suggest a simple extension to NAF and a mapping from the C3 Taxonomy to NAF. The mapping effectively allows the use of the C3 Taxonomy's C3 capability ABBs to be used directly in NAF architecture work. The extension involves a modification to the NAF meta model, which we omit here for space reasons. Table 1 shows the mapping and the implied extensions. The C3 Taxonomy's notion of "Capability" is mapped to NAF's "Capability" construct. Then, the taxonomy's capability breakdown structure in terms of abstract and (more) detailed requirements are mapped to, respectively, "Standard Activity" and "Activity" constructs; the former being more abstract than the latter. Note that NAF only has Operational (Standard) Activity constructs, so since the C3 Taxonomy speaks of Operational, User-Facing and Back-End capabilities, the (Standard) Activity construct is generalized to (Standard) [Operational|User-Facing|Back-End] Activity.

Although technically simple, the extension implies a fundamental change of focus in NAF toward promoting CIS functionality as capabilities in their own right. This is not mixing up operational capabilities and technical functionality. Strictly to the contrary, the purpose of the extension is to ensure loose coupling between operational capabilities and what supports those capabilities in terms of CIS. It is now possible to model in NAF how CIS capabilities support operational capabilities at an implementation-independent level of abstraction. Both operational capabilities and CIS (front-end and back-end capabilities) must subsequently be implemented in terms of corresponding capability configurations.

6 From overarching architecture to target architecture – a guide

Based on the above mapping, we give guidelines on how to compose a reference architecture from an overarching architecture and how to move from reference architecture to a target architecture. We will relate to Figure 4, where overarching architecture elements are depicted in the upper part, reference architecture elements are depicted in the middle part, and target architecture elements are depicted in the lower part of the figure. Our focus is on the capability breakdown structure in an overarching and reference architecture perspective.

(1) Start at the overarching architecture level by declaring capabilities in a CV-2 (Capability Taxonomy) diagram in a nested manner to the level of detail required. The capabilities can be taken from the C3 Taxonomy. New capabilities can be declared if necessary.

(2) From the (abstract) requirements for the capabilities in the C3 Taxonomy, declare (standard) activities in a CV-6 (Activity to Capability Mapping) diagram and link them to the appropriate capability.

Dette er en postprint-versjon / This is a postprint version.
DOI til publisert versjon / DOI to published version: 10.1177/1548512916670785

(3) Combine (standard) activities in an OV-5 (Activity Model) diagram. Swim lanes, or more elaborate Business Process Model and Notation (BPMN) can be used to model activity flow.

Steps (1) and (2) import C3 ABBs into NAF, and Step (3) combines these ABBs; preferably guided by APs.

The difference from regular NAF modeling is that the CV-2 includes capabilities at all three levels; Operational, User-Facing and Back-End. Further, the CV-6 and OV-5 include (standard) activities at all three levels. Regular NAF modeling would only include operational capabilities and activities, and the regular names for CV-6 and OV-5 are, respectively, Operational Activity to Capability Mapping and Operational Activity Model. These are here generalized to include capabilities and activities on all three levels. The standard activities and activities may now be used to model abstract process flow at the capabilities level of abstraction at all three levels showing interactions between levels; enabling architects to combine, orchestrate or choreograph capabilities at all three levels. This spurs capability development not only on the operational level, but also on user-facing and back-end levels[9].

Moving on, abstract descriptions can now be provided for user applications (or light-weight apps; see below) and services. In the C3 Taxonomy, user applications and services are declared at the capability level of abstraction. Thus, an application or a service exists solely in terms of its description—which consists of an interface for syntactic interoperability and a contract for semantic interoperability—and can be provided by varying implementing systems[14]. There is, at this level, a distinct difference between user applications and services and their implementations. It is here useful to generalize the NAF service-oriented view to include applications:

(4) Declare abstract application and service interfaces with dependencies in SOV-2 (App/Service Definition) diagrams.

(5) Based on the chosen reference architecture topology[11], declare capability configuration placeholders in CV-5 (Capability to Organizational Deployment Mapping). These are not implementation specific, but only organizational outlines that delineate the overall topology of the architecture. This step is important in that it declares a capability-based loose coupling for the pending target architecture.

These 5 steps complete the composition of ABBs from the overarching architecture into a reference architecture. Such a composition is specific with regards to functionality, but remains implementation independent and at the capabilities level. Elsewhere, we propose that a wider range of abstraction should be used in architecture work (from architecture ontology, to overarching architecture, to reference architecture, to target architecture) and that the notion of reference architecture should then be used at a more specific level of abstraction than often is the case[11]. This is particularly meaningful when modelling can be done at the implementation-independent level longer, as we are doing here.

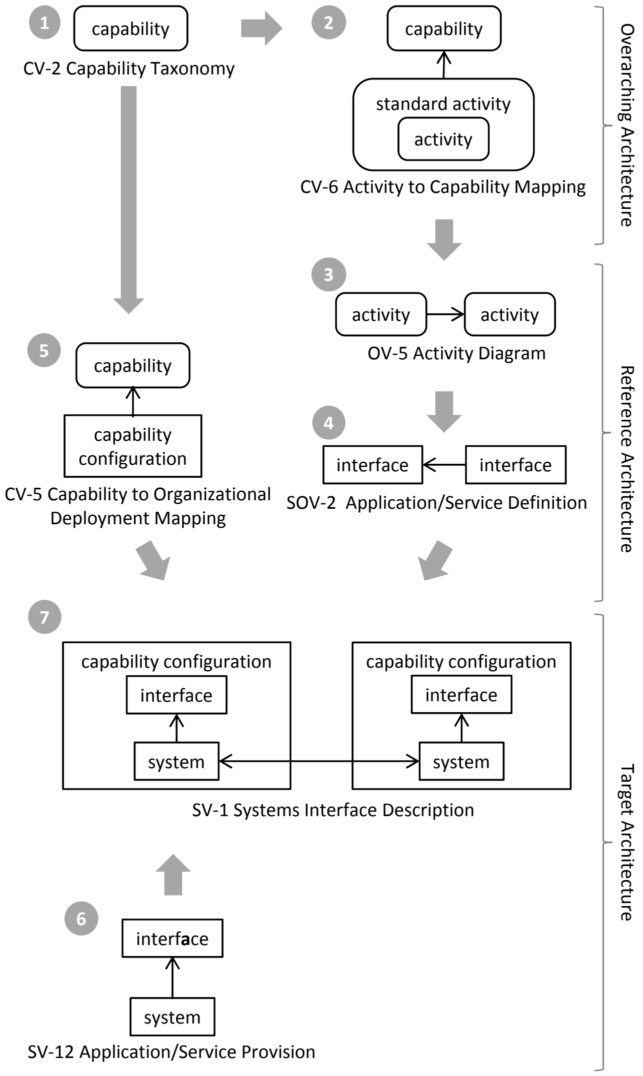

Figure 4. A guide from overarching architecture to target architecture

The next steps transforms the reference architecture into a target architecture.

(6) Determine the systems that will provide applications and services in a SV-12 (App/Service Provision) diagram. These systems may exist, may have to be wrapped into providers or developed from scratch.

(7) Instantiate the capability configuration placeholders with the providers from the SV-12. This gives you

Dette er en postprint-versjon / This is a postprint version.
DOI til publisert versjon / DOI to published version: 10.1177/1548512916670785

the systems architecture in a SV-1 (Systems Interface Description) diagram.

The NAF diagrams suggested in these seven steps are a core set of architectural descriptions. From these, other NAF diagrams can be worked out as desired.

7 Hybrid architecture—simulation architectures within a SOA

We now apply and elaborate the approach in designing an architecture where a simulation architecture resides intact in a larger service-oriented architecture[11]. As a case, we use a concept developed at FFI for demonstrating the feasibility of enhancing the wargaming process in operations planning with simulation support. The concept goes under the name of "Simulation-Supported Wargaming for Analysis of Plans" (SWAP). The idea is to facilitate better analysis of course of actions (COAs) by enabling planners or Operational Planning Groups (OPGs) to simulate COAs to learn about feasibilities and consequences of COAs. The objective is not only to develop better plans, but also to gain insight into factors that may be important if and when the plan fails[15]. A fuller description of the concept and its initial implementation can be found elsewhere[16].

In its implementation, a SWAP Web Application should present CIS functionality for brigade-level planning and wargaming. The Brigade OPG should be able to use the SWAP Web Application to refine the high-level plan received from higher command and develop COAs for individual battalions. A synchronization matrix should be refined and developed in parallel. The OPG should be able to use a route planning service to lay out provisional routes for battalions according to terrain and various criteria for coverage, speed, etc. When ready, the OPG should be able to submit a COA to interactive simulation, and get metrics from the simulation for use in analysis. The simulation should provide entity movement and actions at company level. Where possible, SWAP should use appropriate interoperability standards and associated middleware.

National Defense IT Policy prescribes harmonization with NATO's CFI and FMN programs and the accompanying C3 Taxonomy. We therefore consult the C3 Taxonomy for capability declarations and requirements.

7.1 Step (1): Capabilities

We start by picking ABBs from the C3 Taxonomy's capability hierarchy. The structure of the taxonomy allows for smaller well delineated development projects within the scope of a large portfolio[9], and for our purposes we select a modest number of capabilities and activities.

For operations planning, a relevant operational capability is the Land CONOPS Development Process (Figure 3), relevant user-facing capabilities are the Land COA Development Application and Land COA Analysis Application (Figure 5), and relevant back-end capabilities are the Courses of Action Services and Synchronization Matrix Services. In addition, we need Terrain Analyzer Services for route planning, and Battlespace Simulation Services, Simulation Control Services and Simulation Composition Services for simulation. Relevant core services are Geospatial Web Map Services and Message Brokering Services. See Figure 6. Accordingly, we declare these capabilities in a CV-2 Capability Taxonomy diagram; see Figure 7. C3 Taxonomy color coding is applied to NAF diagrams throughout.

- User-Facing Capabilities
  - User Applications
    - Land Applications
      - Land Operations Planning Applications
        - Land COA Development Application
          + Develop COA
            x Develop Courses of Action
            x . . .
          + . . .
        - Land COA Analysis Application
          + Conduct Structured COA Wargaming
          + Analyse COA Wargaming
          + . . .
        - . . .

Figure 5. C3 taxonomy: Excerpt of User-Facing Capabilities breakdown for Land Operations Planning Applications

7.2 Step (2): (Abstract) requirements—(Standard) Activities

The next step is to map (standard) activities to capabilities. The purpose of the mapping is to model what activities a capability enables an actor to perform. As mentioned above, we find these activities in the C3 Taxonomy in the form of capability requirements. For layout reasons, we here treat only standard activities in Step (2) and introduce underlying activities under Step (3).

There are a number of requirements that are relevant. For our purposes, we choose to focus on the operational abstract requirements Land COAs Development Process and Land COAs Analysis Process under the CONOPS Development Process capability (Figure 3). We focus on the user-facing abstract requirements Develop COA under the Land COA Development Application and the user-facing abstract requirements Conduct Structured COA Wargaming and Analyze COA Wargaming under the Land COA Analysis Application (Figure 5).

Relevant back-end abstract requirements are Estimate Coverage and Estimate Routing under Terrain Analyzer Services; Develop COAs, Wargame COAs and Validate COAs under Courses of Action Services and Define Phases under Synchronization Matrix Services (Figure 6).

For simulation, back-end abstract requirements Initialize Simulation Scenario and Simulate Scenario under Battlespace Simulation Services are relevant (Figure 6). Further, Simulation Control Services (provided by simulation systems) and

Dette er en postprint-versjon / This is a postprint version.
DOI til publisert versjon / DOI to published version: 10.1177/1548512916670785

- Back-End Capabilities
  - Technical Services
    - Community of Interest (COI) Services
      - COI-Specific Services
        - Land Services
          - Terrain Analyzer Services
            + Estimate Coverage
              - x Provide means to identify coverage, non-coverage, areas for known sensor types, taking into account terrain, weather and vegetation
              - x Provide means to identify coverage, non-coverage, areas for known communication types, taking into account terrain, weather and vegetation
            + Estimate Routing
              - x Provide means to identify the optimal route achieving desired effects of concealment, communication, and timeliness
            + . . .
        - M&S Infrastructure Services
        - . . .
      - COI-Enabling Services
        - Operations Planning Services
          - Courses of Action Services
            + Develop COAs
              - x Support the development of COAs
              - x Support the update of COAs
              - x Provide information suitable for map sketches, overlays and standard NATO military symbols
              - x . . .
            + Wargame COAs
              - x Provide the means to evaluate a COA against the most likely/most dangerous COA of opposing forces
              - x Provide functionality to determine the sequels of a COA to be evaluated during wargaming
              - x Provide the means to use the wargaming results to refine the COAs and correct its deficiencies
              - x Provide the means to identify the (dis)advantages of each COA after wargaming
              - x . . .
            + Validate COAs
              - x Provide the means to test that the COA is achievable within allocated resources
              - x Provide the means to test that the COA answer the who, what, when, where, why and how questions
              - x . . .
            + . . .
          - Synchronisation Matrix Services
            + Define Phases
              - x Provide functionality to develop a timeline of the planned tasks within a COA
              - x Provide functionality to develop a timeline of the planned objectives within a COA
              - x . . .
            + . . .
          - . . .
        - Modeling and Simulation Enabling Services
          - Battlespace Simulation Services
            + Initialise Simulation Scenario
              - x . . .
            + Simulate Scenario
              - x . . .
            + . . .
          - Simulation Control Services
            + . . .
          - Simulation Composition Services
            + . . .
          - . . .
        - . . .
    - Core Services
      - Business Support Services
        - Geospatial Services
          - Geospatial Web Map Services
          - . . .
      - SOA Platform Services
        - Message-Oriented Middleware Services
          - Message Brokering Services
            + Manage Subscriptions
              - x . . .
            + Exchange Messages
              - x . . .
            + . . .

Figure 6. C3 taxonomy: Excerpt of Back-End Capabilities breakdown

Simulation Composition Services (provided by, e.g., a HLA RTI[17]) are necessary, but we do not detail these into activities here and include the capabilities themselves as place holders for activities in diagrams.

Relevant core service back-end abstract requirements are those for Geospatial Web Map Services and Manage Subscriptions and Exchange Messages under Message Brokering Services.

These abstract requirements become Standard [Operational|User-Facing|Back-End] Activities in a CV-6 Activity to Capability Mapping diagram; see Figure 8. The CV-6 is originally an operational activities to capabilities mapping, but we extend it to cover user-facing and back-end activities as well. The new «Standard User-Facing Activity» and «Standard Back-End Activity» stereotypes reflect the fact that we model user-facing and back-end capabilities, in addition to operational capabilities.

Dette er en postprint-versjon / This is a postprint version.
DOI til publisert versjon / DOI to published version: 10.1177/1548512916670785

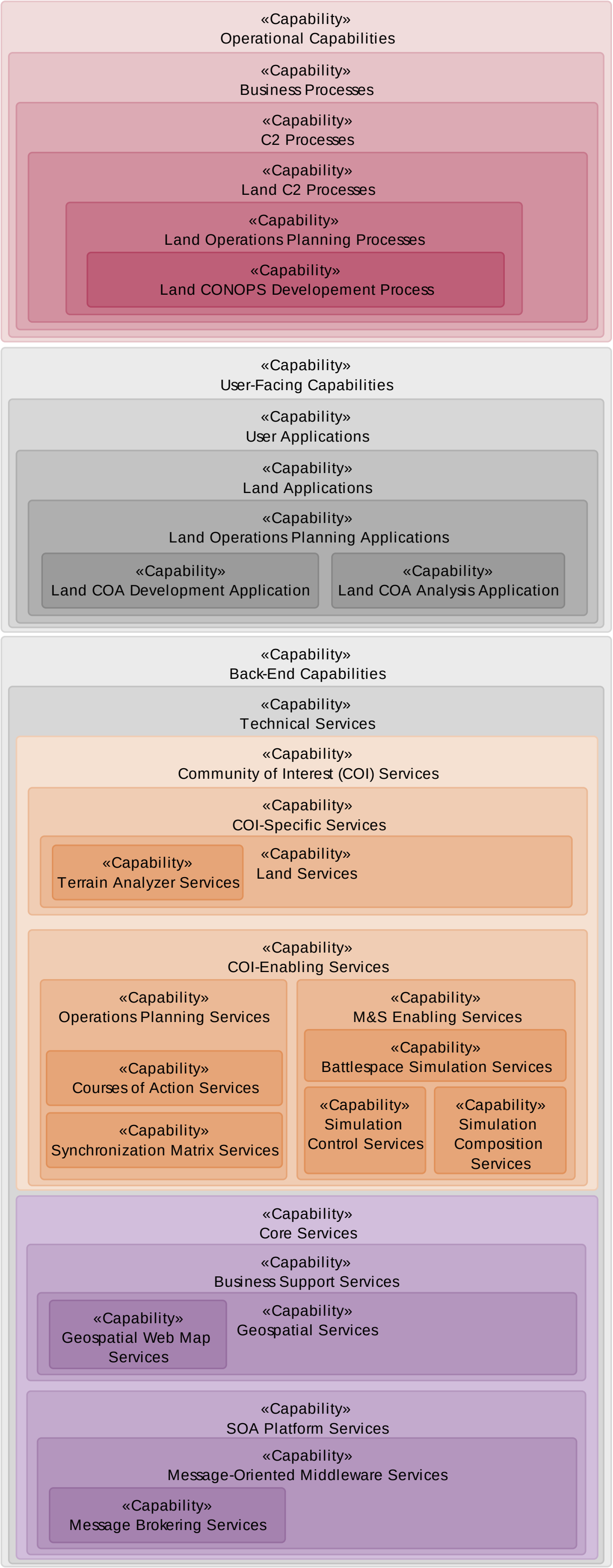

Figure 7. CV-2 Capability Taxonomy: C3 Taxonomy capabilities for land operations planning

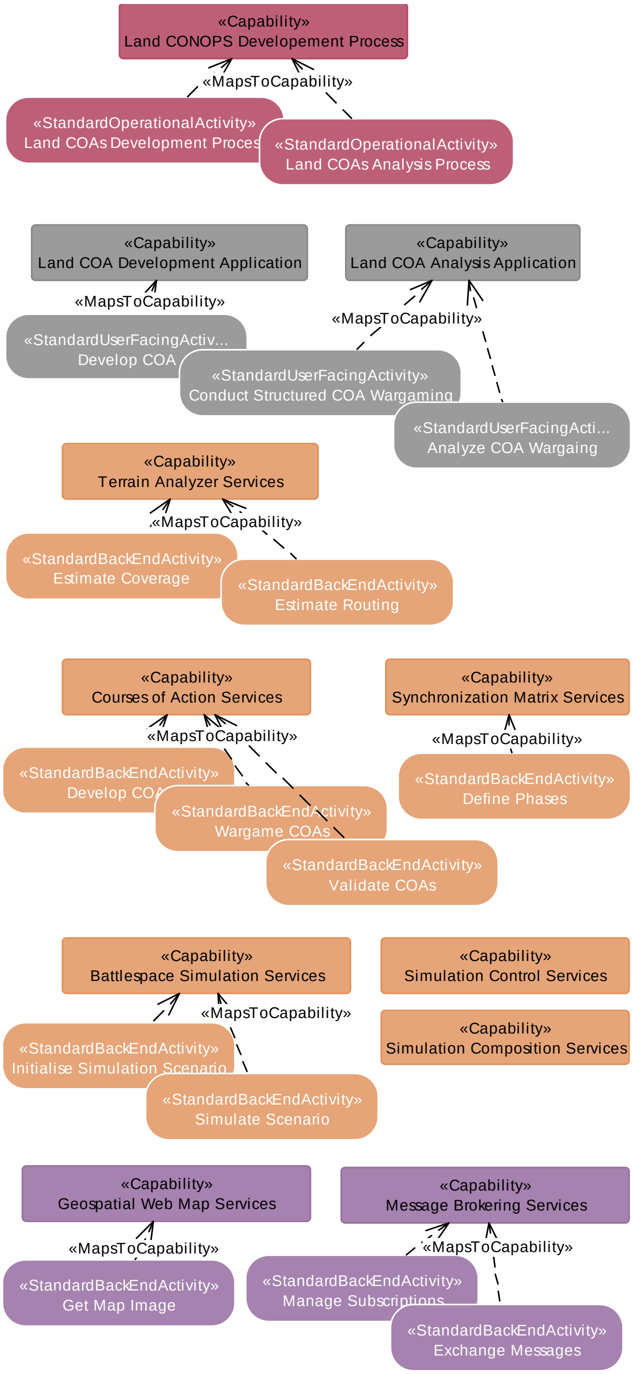

Figure 8. CV-6 Activity to Capability Mapping: Abstract requirements from C3 Taxonomy as [Operational|User-Facing|Back-End] Standard Activities.

7.3 Step (3): Combine Activities

Standard activities are high-level activity groups that are broken down into more detailed activities. We find the detailed activities as more detailed requirements in the C3 Taxonomy (Figures 3, 5, 6).

Dette er en postprint-versjon / This is a postprint version.
DOI til publisert versjon / DOI to published version: 10.1177/1548512916670785

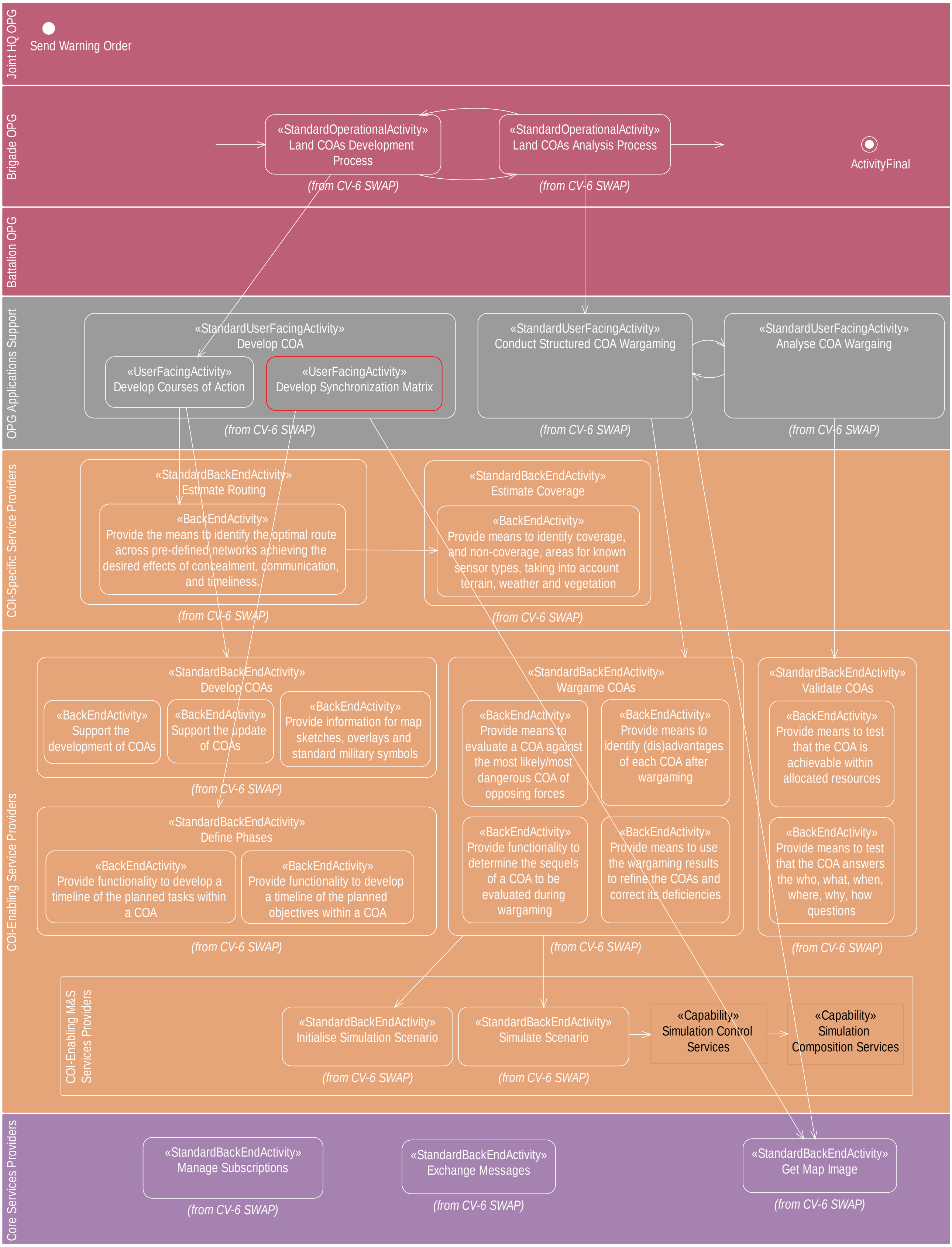

Figure 9. OV-5 Activity Model: C3 Taxonomy (more) detailed requirements as [Operational|User-Facing|Back-End] Activities.

Dette er en postprint-versjon / This is a postprint version.
DOI til publisert versjon / DOI to published version: 10.1177/1548512916670785

Figure 9 introduces the activities that we want to include in our modeling—actually part of Step (2)—and shows simple process flow partitioned according to performers.

The flow goes as follows: While going through the stages of the Land CONOPS Development Process capability, the Brigade OPG performs the Land COAs Development Process and Land COAs Analysis Process standard operational activities. (Since we are not innovating at the operational level at this stage, we do not detail standard operational activities into operational activities here.) To perform the first of these standard operational activities, the OPG, or, more likely, applications support staff to the OPG, perform the Develop Courses of Action and Develop Synchronization Matrix user-facing activities within the Develop COA standard user-facing activity. The Develop Synchronization Matrix is not declared within Develop COA in the C3 Taxonomy, so it appears with a red frame to indicate an innovation. Then to perform the second standard operational activity, the Conduct Structured COA Wargaming standard user-facing activity is undertaken. In the C3 Taxonomy, there are currently no detailed requirements under Conduct Structured COA Wargaming, so the corresponding standard activity has no activities associated to it in the diagram. At this point, innovation could take place in terms of writing more detailed requirements/activities, but we leave that for later. The Conduct Structured COA Wargaming standard user-facing activity iterates with the Analyze COA Wargaming standard user-facing activity.

While performing the Develop Courses of Action user-facing activity, calls are made to a back-end activity that Provide(s) means to identify the optimal route... which in turn, calls upon a back-end activity that Provide(s) the means to to identify coverage..., in order to get initial route plans and time estimates for troop movement in the COA. These estimates are then used in setting up the synchronization matrix, which is facilitated by calls to the back-end activities in the Define Phases standard activity. The Brigade OPG receives a high-level intent from the echelon above (Joint HQ in this case) and constructs various COAs from this. To do this, back-end activities from the Develop COAs standard back-end activities are called, as well as the standard back-end activity Get Map Image from the Core Service Providers.

When performing the Conduct Structured COA Wargaming standard user-facing activity, back-end activities within the Wargame COAs standard back-end activity are called that support wargaming functionality. The Wargame COAs back-end activities call the Initialize Simulation Scenario and Simulate Scenario standard back-end activities to employ simulations for wargaming.

The Analyze COA Wargaming standard user-facing activity, makes calls to Validate COAs back-end activities.

The two standard back-end activities Manage Subscriptions and Exchange Messages are broker functionality; that is, information flow is mediated by a broker, and all activities therefore communicate via the broker rather than directly with each other. This type of interaction is left out from the process flow here.

The new «User-Facing Activity» and «Back-End Activity» stereotypes reflect the fact that we model capabilities at user-facing and back-end levels, in addition to operational levels. For space reasons Figure 9 is a process flow sketch only. More extensive process flow modeling, using control flows, information products, etc., would be done in actuality. Further modeling would also expose more extensive process flow at technical levels than shown here.

Modeling process flow in terms of capability requirements on all three levels enables SWAP to be defined in terms high-level requirements—overarching ABBs in a reference architecture—more extensively before indulging in implementation. This is vital for architecting a SOA.

The C3 Taxonomy has APs which show how capabilities are intended to work together. However, our OV-5 process flow is intended to be more detailed on the one hand, and does not necessarily adhere to the patterns on the other hand. For example, the Wargame COAs calls to simulation is not specified in the taxonomy. As this use of the OV-5 is at the level of capabilities and high-level requirements, this innovates and supplies ABBs and APs to the C3 Taxonomy.

7.4 Step (4): Interfaces

The capability breakdown evident in the C3 Taxonomy exhibits applications at the user-facing levels and groups of services at the back-end levels. The question arises as to at what level of granularity one should declare individual applications and services.

For the user-facing level, the C3 Taxonomy capability structure ends in individual applications. However, to meet demands on flexibility and mobility we argue that lighter-weight applications at the granularity level of standard activity, or even finer at the level of activity, is appropriate. Thus a (standard) user-facing activity may give rise to a light-weight application; here denoted an *app*—a self contained unit of user-facing functionality—a *mobile* app downloadable on a mobile devise being a special case[18]. At the back-end level, the C3 Taxonomy capability structure ends in service groups. Here as well, the appropriate level of granularity is (standard) activities, and a (standard) back-end activity may then give rise to a service. However, the decision on granularity is not clear cut, and different levels of granularity may be appropriate for different designs, since the amount of functionality one chooses to include in both applications and services may vary widely.

Figure 10 shows abstract interfaces for the CIS capabilities of our design. The (standard) activities from the OV-5 are mirrored in app and service interfaces. An exception is the Simulation Service Interface that exposes all

Dette er en postprint-versjon / This is a postprint version.
DOI til publisert versjon / DOI to published version: 10.1177/1548512916670785

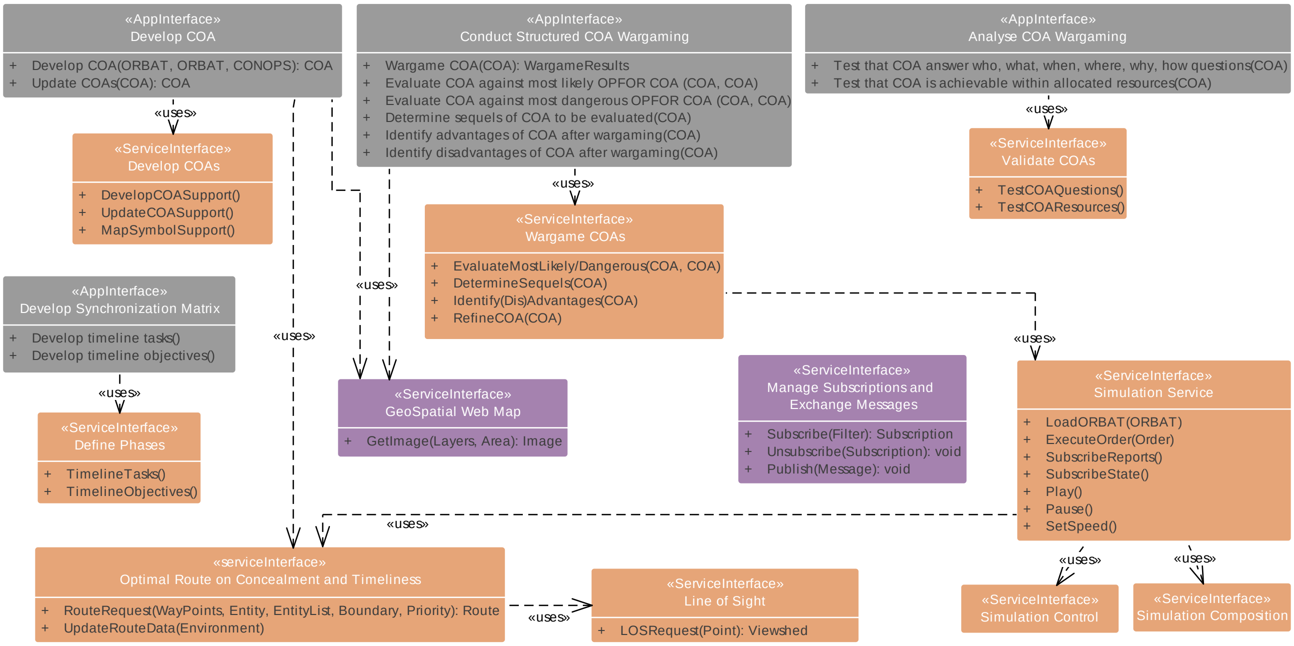

Figure 10. SOV-2 App/Service Definition Diagram.

the simulation functionality in one interface so as to provide this unit as "simulation as a service". This is a reference architecture topology choice that encapsulates and embeds specialized simulation architectures into a larger SOA[11].

7.5 Step (5): Capability configuration placeholders

In NAF, a capability configuration is a configuration of material and personnel that implements a given capability. The central point to this in our approach is that we now have user-facing and back-end capabilities that are given their own capability configurations. This is in contrast to traditional NAF modeling where technical functionality is modeled not as capabilities in their own right, but as part of capability configurations for operational capabilities.

In this step, we use capability configurations as abstract placeholders to be filled out with actual configurations in a target architecture. These placeholders indicate the intended units of deployment and responsibility that reflect a given reference architecture topology[11]. Figure 11 shows how this is done for SWAP, where the topology reflects boundaries within which given interoperability standards are valid.

For terrain services, a decision has been made to have a terrain services repository on the Web Processing Service (WPS) standard[19] and to use an (existing) repository for Web Map Services (WMS)[20]. There is a decision to use the High Level Architecture (HLA) standard[17] with the Realtime-Platform-Reference Federation Object Model (RPR-FOM)[21] extended with Low-Level Battle Management Language (LLBML)[22]. One can note also that the configuration for Battlespace Simulation Services is denoted "Simulation Environment" in line with the term in the Distributed Simulation Engineering and Execution Process (DSEEP)[1]. Further, the broker capability is designated for C2IS-simulation (C2SIM) interoperability, which implies communication over the Military Scenario Definition Language (MSDL) standard[23] and Coalition Battle Management Language (C-BML) standard[24]. Thus, the capability configuration placeholders, together with the interfaces from the previous step, act as boundaries in which interoperability standards are valid. Here, this promotes a reference architecture topology where a simulation environment interoperating over HLA interoperates with C2 planning systems over C2SIM standards[11].

7.6 Step (6): App/Service provision

This step is the first target architecture step. Actual systems are chosen as providers of apps and services. Figure 12 shows the app and service provision for an implementation of SWAP. Three points can be noted with this implementation. First, part of the Develop COA app functionality is provided by a legacy command and control information system (C2IS). This is currently necessary so that OPGs can use their regular systems for developing plans. The SWAP Web Application has a twofold purpose in this context: It supplements the C2IS with simulation functionality, and it is also a C2IS web application prototype for demonstrating lighter field deployment.

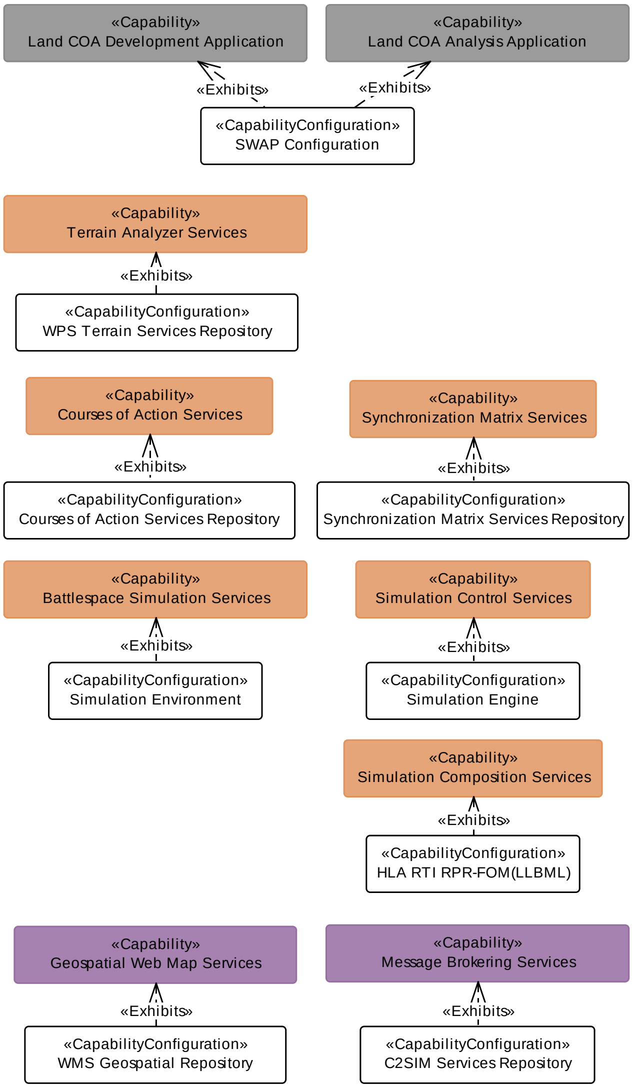

Figure 11. CV-5 Capability to Organization Deployment Mapping Diagram.

Secondly, the Develop COAs, Define Phases, Wargame COAs and Validate COAs service functionalities are not implemented in terms of service providers. Instead this functionality is embedded in the Develop COA, Conduct Structured Wargaming and Analyze COA Wargaming apps, making these apps rather obese. To develop appropriate service providers would be a point for further work.

Third, the Simulation Service is provided by the Agent System which exposes the federation as a service over C2SIM standards, and the MAESTRO Time Management system which exposes basic simulation control in terms of pausing and speed management of the simulation.

7.7 Step (7): Target systems architecture

Step (7) puts it all together. In Figure 13, one can see the capability configurations instantiated with systems that provide apps and services. The simulation environment is a HLA federation as a service provided by the Agent System and the MAESTRO Time Management system. One can further see that there are several capability configurations that interoperate over standards, rather than one single capability configuration implementing an operational capability. By reflecting the C3 Taxonomy's structure through these seven steps, a loosely coupled capabilities-based system has evolved. Further, it is clear where additional development can be undertaken to make the architecture even more loosely coupled. We have indicated the target architecture's communication standards and protocols (Advanced Message Queue Protocol (AMQP)[25], Restful Web Services (REST) over HTTP, JSON[26] over WebSocket[27], GeoJSON[28]) that will carry the standards and protocols decided upon in Step (5).

8 Scenario Development

We now discuss how the approach to combining NAF and the C3 Taxonomy may benefit scenario development. When developing systems that integrate simulations—such as for the SWAP concept—one must at some stage design the scenario that is to be simulated. Such scenarios are in many cases snap shots or derivatives of real operational scenarios.

Scenario development as put forth in recommended guidelines[29] starts by defining or sampling a real-world operational scenario, from which a conceptual scenario is designed for developing the simulation environment. The conceptual scenario contains the M&S concepts for representing the operational scenario. Then, from the conceptual scenario, an executable scenario is developed. Various tools are suggested for describing scenarios. For NAF, OV diagrams are suggested for describing the operational scenario, and SV diagrams are suggested for describing the conceptual and executable scenarios. This couples the conceptual and executable scenarios as specific implementations of the operational scenario. The work of designing and developing the conceptual and executable scenarios are seen as the domain of the M&S expert.

In contrast, we can use the approach suggested here to decouple the conceptual scenario from the operational scenario by defining the conceptual scenario in terms of capabilities in their own right. When designing or setting up simulations for operations, mission rehearsal or training, it is important to compose scenarios readily and rapidly. Further, training staff should be able to compose conceptual and executable scenarios directly, without the aid of M&S experts. In the words of a commander of the U.S. Army Training Support Center: "Nowhere is the

Dette er en postprint-versjon / This is a postprint version.
DOI til publisert versjon / DOI to published version: 10.1177/1548512916670785

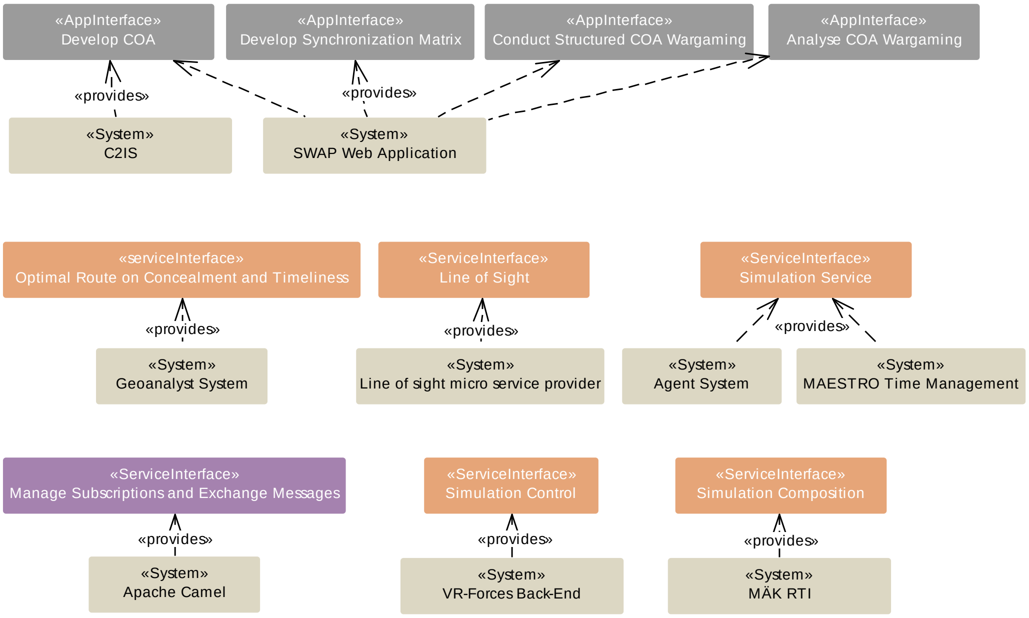

Figure 12. SV-12 Application/Service Provision Diagram.

need for rapid environment shaping more important than in mission rehearsal for real-world operations. In this case, both speed (i.e., rapid scenario generation) and accuracy are paramount to prepare military forces for imminent deployment and the conduct of operations. Current joint M&S is more suited to an 18-month JELC [Joint Exercise Life Cycle] vice the much shorter period required for rapid mission rehearsal (i.e., as short as 3–7 days). To achieve shorter planning cycles prior to training events, joint M&S solutions must be far more flexible than they have been—they must also be composable at the trainer level [...] The trainer is closest to the unit and understands the training objectives best, [...] the various factions in the battlespace, and the limiting factors that will shape the unit's operations. By allowing the trainer to frame the scenario directly and enabling rapid, and intuitive scenario generation, large-scale manpower savings can be achieved and unit training objectives are more likely to be best served"[30].

This approach to scenario development where conceptual scenarios can be composed "on the fly" relies strongly on simulation functionality being available in terms of designated capabilities rather than being hidden in specific implementation or implementation designs.

Our approach and the seven steps can be used to this end. Figure 14 depicts schematically how this can be done. At the left is architectural work in terms of C3 Taxonomy capability layers as in the previous section. Operational capabilities use layers of technical capabilities possibly including simulation capabilities. To the right is architectural work in terms of C3 Taxonomy capability layers for scenario development. Operational capabilities used in a military scenarios use, on the one hand, layers of technical capabilities as before and, on the other hand, are reflected in synthetic capabilities that can be run in a simulation. The former is the operational scenario and the latter is the conceptual scenario[29].

As an example, we can consider a scenario to be run during simulation-supported wargaming for operations planning as described in the previous section. The scenario to be simulated is determined by the COA to be tested. Here, the overall plan intent is given from the Joint HQ to the brigade OPG who then devises COAs for its battalions. The COA thus describes battalion movement and actions. To develop a scenario, one therefore starts by modeling that real-world operational process.

For Step (1) in our approach, one can find the appropriate capabilities in the C3 Taxonomy under Land Operations Execution Processes and then under Land Targeting Processes where one finds the Decide, Detect, Track, Deliver and Asses capabilities that give principles applicable across echelons for a systematic approach to enable the right target to be effected with the appropriate system at the right time and place. The battalion commanders would use these principles when performing their orders in an operation; in this case, those envisioned in a COA. They would, in turn, pass orders down to squadron and company commanders who would also employ the targeting processes. The battalion commanders would use user-facing capabilities in terms of functionality on their C2IS, etc., and squadron commanders would use user-facing capabilities in terms of functionality on their battle management systems, etc. The appropriate capabilities for this can be found under

Dette er en postprint-versjon / This is a postprint version.
DOI til publisert versjon / DOI to published version: 10.1177/1548512916670785

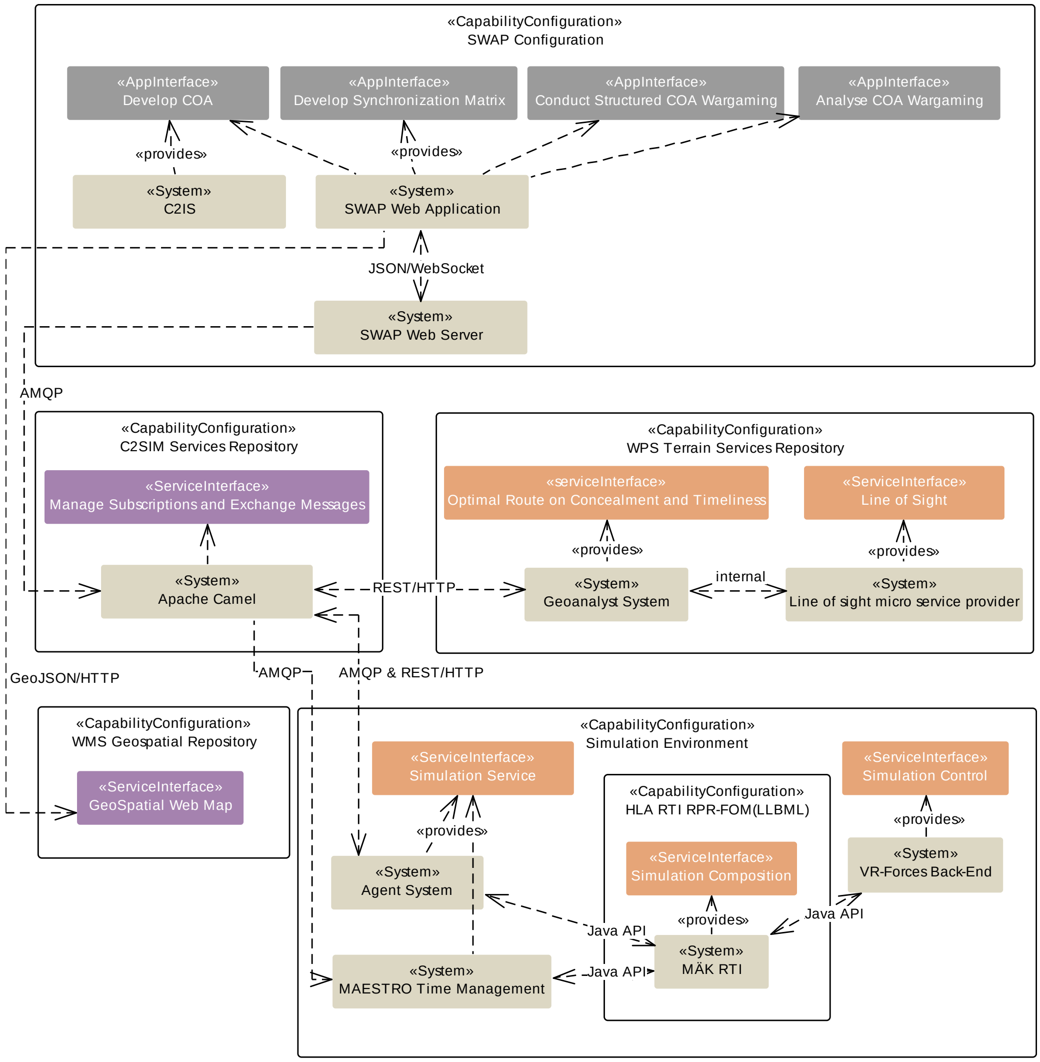

Figure 13. SV-1 Systems Interface Description.

Land operations Execution Applications where one finds the Land Operations Execution Monitoring Application and Recognized Ground Picture Application capabilities. One can envision those capabilities using Estimate Routing and Estimate Coverage, so that C2IS and battle management systems can give commanders route planning and line of sight functionality on their devices during operations.

We omit the CV-2 and CV-6 for space reasons, but sketch the OV-5 from Step (3) in Figure 15, where capabilities are depicted as placeholders for activities. This OV-5 is styled in line with Figure 14(b). The targeting processes are given for the Battalion Commander and Squadron Commander. When performing these processes the commanders use the Land Operations Execution Monitoring Application and Recognized Ground Picture Application (only calls from Deliver Process shown). These applications call route planning and line of sight activities. To model the real-world scenario—i.e., the operational scenario[29]—further; e.g., in order to understand it sufficiently for transforming it into a conceptual scenario for simulation, one can complete the seven steps. We omit this here.

For developing the conceptual scenario, the real-world scenario has to be transformed into synthetic terms. Jumping straight to Step (3), the right-hand part of Figure 15 shows synthetic back-end capabilities that reflect the operational targeting processes. Since these do not currently exist in the C3 Taxonomy they are shown with red frames. The synthetic reasoning underlying these processes can be done by various technologies—here Context-Based Reasoning (CxBR)[31] is given as an example. To model that commanders have access to route planning and line of sight functionality, the synthetic processes call the Estimate Routing and Estimate Coverage activities. Note that the transition between synthetic echelons here implies that battalion-level

Dette er en postprint-versjon / This is a postprint version.
DOI til publisert versjon / DOI to published version: 10.1177/1548512916670785

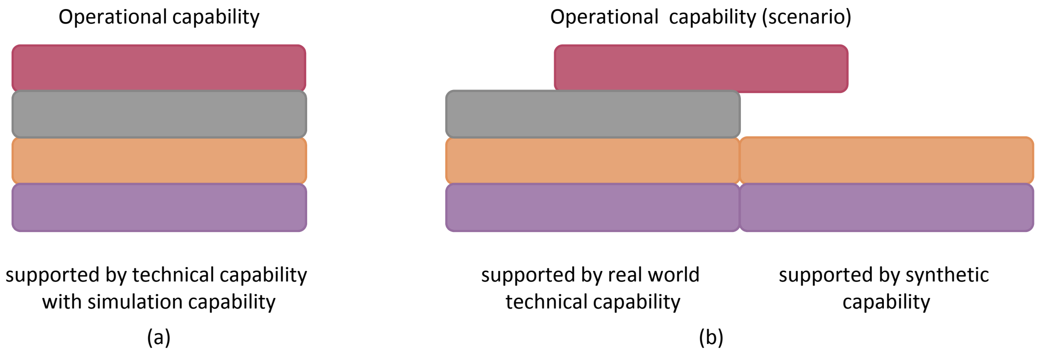

Figure 14. (a) Operational capabilities and supporting technical capabilities with simulation support. (b) Operational capabilities for scenario used in simulation and supporting technical capabilities for real world and for synthetic world.

orders of battle (ORBAT) and orders are decomposed into squadron-level ORBAT and orders. This decomposition could also be defined as a service.

To model the conceptual scenario further, the remaining steps may be employed; again, we omit this here, but indicate the result by showing an excerpt from the resulting SV-1 diagram from Step (7) in Figure 16. One can see that synthetic targeting is a separate capability with its own configuration outside the simulation environment. This opens for the possibility of using the synthetic targeting capabilities as operational support as well as for controlling simulated agents. In the simulation environment, the agent system uses the synthetic targeting system, and the appropriate agents have been added to the simulation engine (VR-Forces Back-End as an example in this case). The synthetic targeting system receives WPS data over REST/HTTP from route planning and line of sight service providers.

9 Discussion and Conclusion

We have used a simple seven-step approach for inducing the C3 Taxonomy capability structure into NAF to structure architectural work pertaining to M&S. The C3 Taxonomy's capabilities and corresponding requirements are architecture building blocks at the overarching architecture level. Our seven steps compose these architecture building blocks into more specific reference architectures that can be used to design target architectures. Thus, the C3 Taxonomy functions as a comprehensive repository of architectural building blocks useful for M&S architectural work. M&S overarching and reference architectures need not list all conceivable services, but can instead reference the C3 Taxonomy as a repository.

There is ongoing debate on what a simulation environment encompasses. For example, the term "C2SIM federation" is used to denote a "simulation environment that contains at least one C2 system, and that uses a C2SIM data exchange model"[32]; more specifically, A C2SIM federation encompasses parts of a traditional simulation environment (e.g., a HLA federation) and parts of an operational environment. The DSEEP Multi-Architecture Overlay (DMAO)[33] speaks of "non-conforming member applications" (that do not interoperate via the simulation architecture interoperability protocols/standards), and "integrating non-conforming member applications into the simulation environment", that might seem to imply that the scope of the simulation environment is extended to things that do not interoperate via simulation architecture protocols/standards. In our example, the reference architecture topology is determined by the boundaries in which interoperability standards reign. Thus, the simulation environment in our example encompasses strictly that which interoperates over HLA.

We do not conclude on what a simulation environment should, or should not, include, but we do hold that the concept of simulation environment should constitute an implementation of a capability; i.e., should be a capability configuration. Then, what exactly a simulation environment encompasses is determined by how one wishes to define the corresponding capability; in other words, what M&S (and other functionality) one wishes to provide as a unit of service. Thus, if one desires a simulation environment that, e.g., includes C2IS, then this should be the result of conscious decision at the capability level (which now includes both operational and technical capabilities). The integrated framework then allows architecture development as described in the seven steps down to a corresponding capability configuration.

This ability to delineate and develop the service boundaries of larger parts of simulation-(related) functionality in a sound and traceable manner is crucial for cloud-based views of services such as dealt with in the Modeling and Simulation as a Service (MSaaS) concept. There, larger pieces of functionality are provided as a service, than what is often offered in terms of so-called "micro services".

Dette er en postprint-versjon / This is a postprint version.
DOI til publisert versjon / DOI to published version: 10.1177/1548512916670785

Figure 15. OV-5 Activity Model for scenario.

Our three-tiered capability-based approach can thus facilitate the design and development of simulation as capabilities. But it can also help factor out simulation-relevant functionality that is not itself a simulation into separate capabilities. For example, synthetic processes that reflect operational processes can be used for agents in simulations but can also be used for real-world operational support. Further, the focus on capabilities, enables the development of services that can be used in multiple modes; for example a route planning service can be used when devising plans during operations planning, it can be used in actual operations, and it can be used in simulations when simulating an agent with—or without—the route planning service as a real-world capability.

Consulting the C3 Taxonomy as an overarching architecture repository is helpful in delineating functionality in terms of capabilities, applications and services. Nevertheless, the taxonomy is work in progress, and to this end, our approach will identify gaps and modifications that may inform updates to the taxonomy. Two modes of innovation are particularly pertinent: One can define new capabilities and capability breakdowns, and one can define

DOI til publisert versjon / DOI to published version: 10.1177/1548512916670785

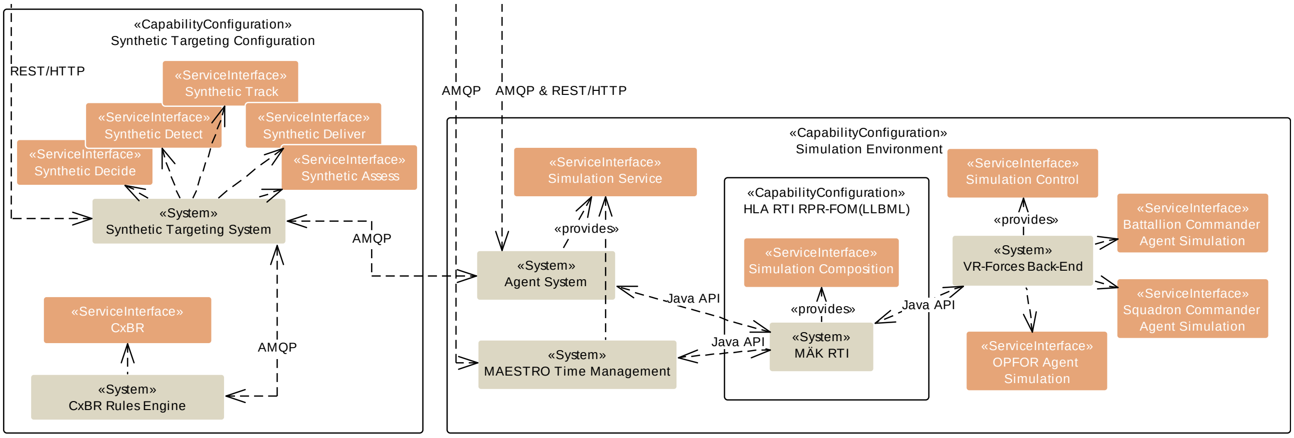

Figure 16. SV-1 Systems Interface Description for simulation environment and supporting synthetic targeting configuration after scenario development.

new patterns for composing the C3 Taxonomy's capabilities and requirements and refine those that are defined for the taxonomy already. This is an important aspect in integrating M&S and MSaaS in NATOs FMN and CFI programs.

Our seven-step approach and its implications for M&S and MSaaS have not been validated by extensive application by practitioners. This needs to be done in due course. For now, the approach is intended as a guide to integrate two strategically leveraged architectural tools that do not, in the outset, fit well together, for architectural work in the context of M&S. Ongoing work in the NATO Task Group (MSG-136) on *Modelling and Simulation as a Service (MSaaS)* aims to provide further refinement and proof of concept to the ideas put forth here.

Acknowledgements

The author is deeply grateful to the SWAP concept development team, to Karsten Brathen and to members of the NATO Task Group (MSG-136) on *Modelling and Simulation as a Service (MSaaS)—Rapid deployment of interoperable and credible simulation environments* for input and fruitful discussions.

References

1. IEEE Standards Association. 1730-2010 – IEEE recommended practice for Distributed Simulation Engineering and Execution Process (DSEEP), 2010.
2. NATO Communications and Information Agency. The C3 Taxonomy. `http://www.act.nato.int/article-8a`, 2016. Accessed January 2016.
3. North Atlantic Treaty Organization. NATO Architecture Framework v4.0 Documentation (draft). `http://nafdocs.org/`, 2016.
4. North Atlantic Treaty Organization. The Connected Forces Initiative (CFI). `http://www.nato.int/cps/en/natohq/topics_98527.htm`, 2015. Accessed November 2015.
5. North Atlantic Treaty Organization. The Secretary General's Annual Report 2014. `http://www.nato.int/cps/en/natohq/opinions_116854.htm`, 2015. Accessed November 2015.
6. NATO Allied Command Transformation. Federated Mission Networking (FMN). `http://www.act.nato.int/fmn`, 2015. Accessed November 2015.
7. Elonen S and Artto KA. Problems in managing internal development projects in multi-project environments. *Int'l J Project Management* 2003; 21: 395–402.
8. Cooper RG, Edgett SJ and Kleinschmidt EJ. *Portfolio management for new products*. Perseus Books, 1998.
9. Hannay JE, Brathen K and Mevassvik OM. Agile requirements handling in a service-oriented taxonomy of capabilities. *Requirements Engineering* 2016; Online `http://dx.doi.org/10.1007/s00766-016-0244-8`.
10. The Open Group. TOGAF Version 9.1 Enterprise Edition, 2011. Document no. G116.
11. Hannay JE, Brathen K and Mevassvik OM. Reference architecture for simulations in a service-oriented environment. *Systems Engineering (major revision)* 2016; .
12. The Open Group. SOA Reference Architecture Technical Standard, 2011. Document no. C119.
13. Goosens JA. C3 capability development in a service-oriented world. Technical report, NATO Allied Command Transformation, 2013.
14. The Open Group. SOA Ontology, Version 2.0 Open Group Standard, 2014. Document no. C144.
15. Hannay JE, Brathen K and Hyndøy JI. On how simulations can support adaptive thinking in operations planning. In *Proc. NATO Modelling and Simulation Group Symp. on M&S Support to Operational Tasks Including War Gaming, Logistics, Cyber Defence (STO-MP-MSG-133)*.

16. Bruvoll S, Hannay JE, Svendsen GK et al. Simulation-supported wargaming for analysis of plans. In *Proc. NATO Modelling and Simulation Group Symp. on M&S Support to Operational Tasks Including War Gaming, Logistics, Cyber Defence (STO-MP-MSG-133)*.
17. IEEE Standards Association. 1516-2010 – IEEE Standard for modeling and simulation (M&S) High Level Architecture (HLA), 2010.
18. Churchill D, Lu J, Chiu TKF et al. (eds.) *Mobile Learning Design: Theories and Application*. Lecture Notes in Educational Technology, Springer, 2015. ISBN 9789811000270.
19. Open Geospatial Consortium Inc. OpenGIS Web Processing Service (WPS) interface standard, 2007.
20. International Organization for Standardization. ISO 19128:2005 geographic information – Web Map Server interface, 2005.
21. Simulation Interoperability Standards Organization. SISO-STD-001.1-2015 – Standard for Real-time Platform Reference Federation Object Model (RPR FOM), version 2.0, 2015.
22. Alstad A, Mevassvik OM, Nielsen MN et al. Low-level battle management language. In *Proc. 2013 Spring Simulation Interoperability Workshop (SIW)*. Simulation Interoperability Standards Organization.
23. Simulation Interoperability Standards Organization. SISO-STD-007-2008 – Standard for Military Scenario Definition Language (MSDL), 2008.
24. Simulation Interoperability Standards Organization. SISO-STD-011-2014 – Standard for Coalition Battle Management Language (C-BML) phase 1, version 1.0, 2014.
25. Organization for the Advancement of Structured Information Standards. Advanced Message Queuing Protocol (AMQP) version 1.0, 2012.
26. Ecma International. ECMA-404 – The JSON Data Interchange Format, 2013.
27. Fette I and Melnikov A. The WebSocket Protocol—request for comments: 6455. On Internet Engineering Task Force (IETF) pages, `http://tools.ietf.org/pdf/rfc6455.pdf`, 2011. Accessed September 2012.
28. Butler H, Daly M, Doyle A et al. The GeoJSON Format Specification, 2008.
29. NATO Science and Technology Organisation. Guideline on scenario development for (distributed) simulation environments. Technical Report STO-TR-MSG-086-Part-II AC/323(MSG-086)TP/562, 2015.
30. Edgren MG. Cloud-enabled modular services: A framework for cost-effective collaboration. In *Proc. NATO Modelling and Simulation Group Symp. on Transforming Defence through Modelling and Simulation—Opportunities and Challenges (STO-MP-MSG-094)*.
31. Gonzalez AJ, Stensrud BS and Barret G. Formalizing context-based reasoning: A modeling paradigm for representing tactical human behavior. *Int'l J Intelligent Systems* 2008; 23: 822–847.
32. Heffner K, Mevassvik OM, Gautreau B et al. A proposed process and toolset for developing standardized C2-to-simulation interoperability solutions. In *Proc. NATO Modelling and Simulation Group Symp. on Integrating Modelling & Simulation in the Defence Acquisition Lifecycle and Military Training Curriculum (STO-MP-MSG-126)*.
33. IEEE Standards Association. 1730.1-2013 – IEEE recommended practice for distributed simulation engineering and execution process multi-architecture overlay (DMAO), 2013.

Dette er en postprint-versjon / This is a postprint version.
DOI til publisert versjon / DOI to published version: 10.1177/1548512916670785