Leveraging Network-Centric Strategic Goals in Capabilities[*]

Jo Erskine Hannay[†]and Eli Gjørven[‡]

Abstract

The vision of network-centric operations is to increase operational capabilities through networked collaboration. NATO and its member nations state this vision in strategic documents at a very high level of abstraction. While suitable for giving an overall feel, current documentation renders the steps toward implementing those visions largely unsupported. We outline a method, based on agile requirements engineering, for converting high-level strategic visions into capabilities, whose form lend themselves to incremental implementation. We illustrate the use of this method in two cases that concern both operational capabilities and technical capabilities. We also show how the method enables one to prioritize which capabilities to develop first. We conclude that some form of explicit methodology to span the gap between strategic visions and implementing those visions is necessary.

keywords: Strategic goals, Capabilities, Requirements engineering, Benefits Management

Introduction

Military strategic visions are often written in an informal style structured across many documents. For example, one nation's army declares its vision for network-centric operations (Alberts et al., 1999) in two main documents with some 70 annexes and attachments. The vision is formulated in terms of

[*] Accepted version to Journal of Military Studies (2021), De Gruyter.

[†] Center for Effective Digitalization of the Public Sector (EDOS), Simula Metropolitan Center for Digital Engineering, PO Box, 134, 1325 Lysaker.

[‡] Norwegian Defence Research Establishment (FFI), PO Box 25, 2027 Kjeller.

high-level characteristics that give a comprehensive overview of the vision, along with suggested high-level initiatives toward implementing the vision.

It is, however, not at all obvious how one should operationalize such a vision into workable solutions. In lack of any progress, the result may well be that the vision loses attention, is abandoned, and perhaps replaced by a new more fashionable formulation of essentially the same vision. An example in point is the concept "NATO Network-Enabled Capability" (NNEC)[1] going off fashion, but where the presently in-vogue concept "Federated Mission Networking" (FMN)[2] encompasses the ideas in NNEC via the more fashionably named "Connected Forces Initiative"[3] (NATO Allied Command Transformation, 2013). While reformulating visions may, in some instances, be beneficial, the reason for doing so should not be because no-one was able to use the vision for producing value. Moreover, if it is no easier to operationalize the new vision, then one has come no further.

We argue that the lack of methodological support for transitioning from strategy to working system is a main disabler for operationalizing strategic visions. Various frameworks, such as the NATO Architecture Framework (NAF)[4], The Open Group Architecture Framework (TOGAF)[5] and Capability Driven Design (CDD) (Sandkuhl, 2015), do address the transition from strategy to working system. These frameworks are valuable in that they structure the workflow through all the relevant layers of abstraction. In particular, they suggest what kinds of diagrams are suitable at various levels for describing enterprise architectures, business processes, systems architectures, system processes, etc. Further, and more specifically for network-centric operations, the NATO NEC Command and Control (C2) Maturity Model (N2C2M2) (Alberts et al., 2010) is a framework for determining how to implement C2 within a C2 approach space laid out by concepts of NNEC. The N2C2M2 is designed to be used in conjunction with the NATO Code of Best Practice for C2 Assessment (COBP-C2A) and the C2 Conceptual Reference Model (C2CRM). All of these frameworks give steps and checklists for what to do and what to think about when operationalizing strategic content. However, they give meagre support for eliciting and structuring the information; that is, the actual content, needed at each level. For example, the N2C2M2 speaks of patterns of interaction, but does not give guidelines on how to write such patterns.

As one step to address this, (Hannay et al., 2017c) suggested a require-

[1] `http://www.nato.int/cps/en/natolive/topics_54644.htm`

[2] `https://www.act.nato.int/activities/fmn`

[3] `https://www.nato.int/cps/en/natohq/topics_98527.htm`

[4] `https://www.nato.int/cps/en/natohq/topics_157575.htm?`

[5] `https://pubs.opengroup.org/architecture/togaf9-doc/arch/`

ments engineering approach, where so-called user stories are used to create cohesive story lines through levels of abstraction and detail. This effectively enables step-wise refinement and detailing of high-level descriptions into operational (human) processes and (technical) systems. This story-line approach starts at the level of so-called capabilities and works its way down to specific (human) processes and (technical) systems.

In this article, we supplement that approach with prior steps that *initiate* the story-line approach. In other words, we describe steps to arrive at capabilities written as user stories, from strategic visions. Further, we adapt a technique for assigning benefit estimates to the resulting user stories for the purpose of prioritizing which stories should be implemented first. We present these steps as a *method for operationalizing strategic visions into capabilities.* We apply the method in two cases. The first case concerns the derivation of core IT capabilities from strategic visions of network-enabled operations. The second case concerns the derivation of simulation capabilities for training from the same visions.

Method for operationalizing strategic visions into capabilities

The method we present uses techniques from requirements engineering; a discipline in systems and software engineering aimed at eliciting, capturing and expressing systems requirements from people's incomplete, but evolving, understanding of what is needed from a system under development. In this article, the systems under development are human procedural systems as well as information technological systems.

The basic idea of the method is to make strategic visions concrete by formulating them in terms of capabilities; where a capability is a description of what one aims to achieve while abstracting away the concrete means of how to achieve it.

Capabilities

The concept of *capability* is central to this discussion. A capability defines an ability to perform certain actions to achieve certain outcomes (The Open Group, 2011a,b) and is expressed in general and abstract terms. To implement a capability, a combination of organization, people, processes and technology must be coordinated into a so-called *capability configuration.* But importantly, capabilities as such are persistent and independent of their various implementations. A capability can therefore be seen as an abstract requirement that has a life time in terms of strategic periods, rather than in terms of, say, a development project.

Capability specification

Also central to the method is the explicit formulation of capabilities in a standard format. Several notations can be used to specify capabilities, for example modelling formats such as the Unified Modelling Language (UML)[6] and the Business Process Model and Notation (BPMN)[7]. To remain close to what operational personnel are familiar with, we choose to base our capability specifications on the textual requirements format of user stories from agile development. In particular, we use *epics*, which are high-level low-detailed user stories. In systems and software engineering, epics are commonly used to elicit and specify requirements from stakeholder perspectives in an intuitive manner. The format is designed to express stakeholder use cases, in terms of what stakeholders want to achieve by using a process or a system. In our method, epics will be used to specify how one intends one or several actors to use the capabilities. When implementing capabilities, high-level epics can be detailed and refined into more concrete user story specifications (Hannay et al., 2017c), but in our discussion, we stay at the epics level of abstraction.

Epic format

In our framework, a methodological key to clarifying strategic visions is to phrase capabilities in a structured manner. To this end, we use similar formats for specifying epics to those in (Hannay et al., 2017c,a,b).

Single actor epics

A usual format for specifying epics is as follows:

> Single actor epic: ⟨Actor A⟩ ⟨performs activities a in his/her domain D⟩ by using ⟨capability C⟩ to ⟨perform tasks t⟩

Activities a are what the actor A does in his/her/its work domain D, regardless of any systems support. A can be a group of actors. The capability C gives is a systematic approach to perform work in the form of procedures, processes, CIS systems, etc. The actor uses C for certain tasks t to support the activities a.

[6] https://www.omg.org/spec/UML/

[7] http://www.bpmn.org/

Collaboration epics

Higher network-centric maturity levels are characterised by the ability of different organisations to work together towards common goals. To capture the coordination, collaboration and coherence characterising network-centric operations, we have extended the standard single actor format into a new type of user story format that expresses a *collaboration coupling*, in order to describe and reason about collaboration between several actors explicitly. A collaboration coupling is defined as "two or more actors collaborating in order to reach a common goal". The actors in a collaboration coupling may belong to different organisations, and thus represent different interests and priorities. We extend the epics format to support collaboration stories as follows:

> Collaboration epic: $\langle$Actors A and $B\rangle$ $\langle$perform collaborative activities ca in their common domain $cD\rangle$ by using $\langle$capability $C\rangle$ to $\langle$perform tasks $t\rangle$

The collaborative activities ca are what actors A and B work together on, regardless of any systems support. A and B usually work in two different domains aD and bD which merge into cD. The common domain, cD, may be an *ad hoc* domain that spontaneously constructs around A and B's need for collaboration. A collaboration story specifies the roles of the participating actors in the activity that the actors perform.

Steps from strategy to capabilities

Our method, then, is illustrated in Figure 1 and contains four steps.

Step 1: Identify objectives. Review strategic documents to identify goal statements. Formulate measurable objectives from these goal statements. Objectives give the reason for developing capabilities and should be used to determine the expected benefit of capabilities.

Step 2: Identify capability fragments. Review strategic documents to identify text passages that appear to constitute capability fragments for the identified objectives.

Step 3: Declare capabilities. Identify existing or new capabilities from identified fragments. Identified capability fragments are either found to be subsumed in an existing capability already specified or found to give rise to a new capability.

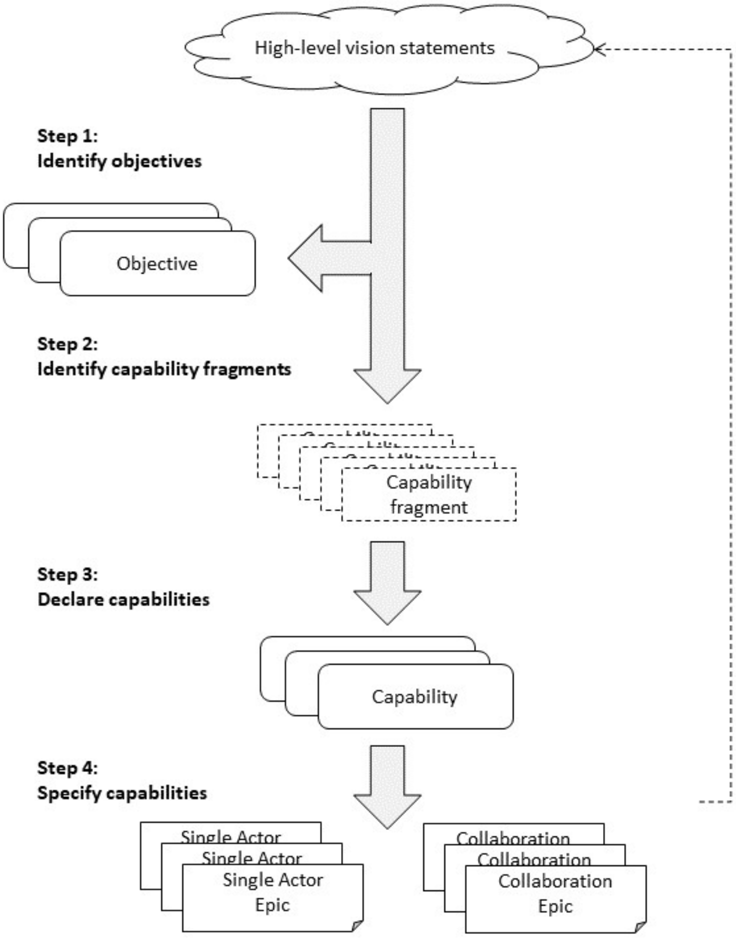


Figure 1: Steps from strategy to capabilities.

Step 4: Specify Capabilities. Phrase the identified and categorised capabilities as single-actor or collaboration epics. Specifying capabilities in terms of epics may lead to the identification of the need for further capabilities. Thus, one may conduct another increment of the method as illustrated by the dashed arrow in the figure.

Specializing the method to the C3 Taxonomy

To support the transition to a capabilities-based portfolio, NATO Allied Command Transformation are developing the Consultation, Command and Control (C3) Taxonomy of capabilities[8]; see Figure 2 for a high-level view.

[8] `https://www.nato.int/cps/en/natohq/topics_157573.htm?`

The taxonomy explicitly includes, in the same picture, the operational context (Operational Context frame) and the computing context (Communication and Information Systems (CIS) Capabilities frame).

Referring to Figure 2, the Operational Capabilities layer declares capabilities independent of technology. The User-Facing Capabilities layer declares capabilities that enable users of Operational Capabilities to interface with Computer and Information Systems (CIS) Capabilities. The Back-End Capabilities declare CIS Capabilities that are hidden from operational users, but that are accessed through User-Facing Capabilities. The *service* concept at the CIS layer embodies notions of integral and loosely coupled components

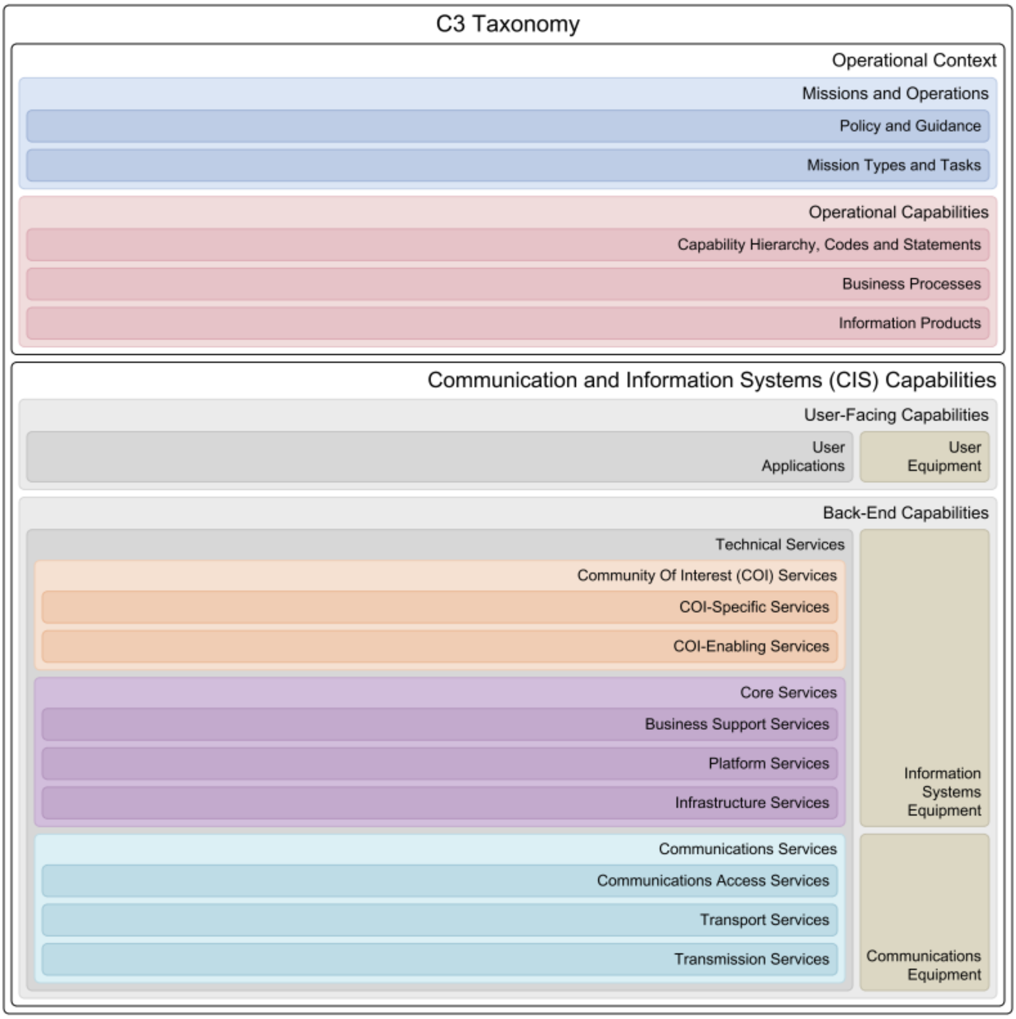


Figure 2: The NATO Consultation, Command and Control (C3) Taxonomy of capabilities.

that deliver clearly delineated functionality to a range of service consumers, which may be CIS user-facing applications or other CIS services. Services are essential in so-called service-oriented architecture (Erl, 2007) and more recently, cloud-based architecture, where CIS services can be accessed at all times by very large ranges of consumers independently of where the consumers are.

The various layers of the C3 Taxonomy are populated by capabilities in a tree structure. Through the taxonomy's web page, one can view successively more specific capabilities by clicking on a given capability. For example, clicking on "Business Processes" will eventually reveal, among others, "Domain-specific Operations", in which one will find "Land Operations" that contains "Land C2 Processes", "Land Operations Planning Processes", "Land Operations Assessment Processes", etc.

Although the C3 Taxonomy is being developed for NATO capabilities, individual nations also use the structure of the taxonomy for structuring their national capabilities. This is useful simply for the sake of structuring, but also to integrate national capabilities into the NATO context. However, it may also be the case that smaller-scale national capabilities might not find its counterpart, or its place in a capability group, in the C3 Taxonomy.

When identifying capabilities in our method's Step 3, one would look for relevant capabilities in the taxonomy that might fit. If not found, one would try to find the appropriate neighbourhood in the taxonomy in which to place the new capability. Indeed, simply being explicit on at what layer (operational, user-facing, back-end) the capability under investigation belongs to is valuable, since this helps delineate a capability more clearly and helps to keep it decoupled from concerns that should rather be dealt with in other types of capability. All this should help to avoid monolithic systems that "stove-pipe" through the layers, rather than share common capabilities.

The incremental aspect of the method has special significance for the C3 Taxonomy. Following (Hannay et al., 2017c), increments consists of capabilities and epics at successively layers of the C3 Taxonomy; starting perhaps at the operational level and moving down in the user-facing CIS capabilities and back-end CIS capabilities. Figure 3 illustrates this idea: Declaring and specifying capabilities at the operational level triggers a need to identify, declare and specify capabilities at the user-facing CIS level, which triggers a need to identify, declare and specify capabilities at the back-end CIS level. (In this discussion, we relate only to the Community of Interest (COI) back-end capabilities.) Following the structure of the C3 Taxonomy should ensure the development of technical functionality as capabilities in their own right, rather than developing technical functionality solely on the basis of a given operational capability specification. Thus, each epic, whether operational or

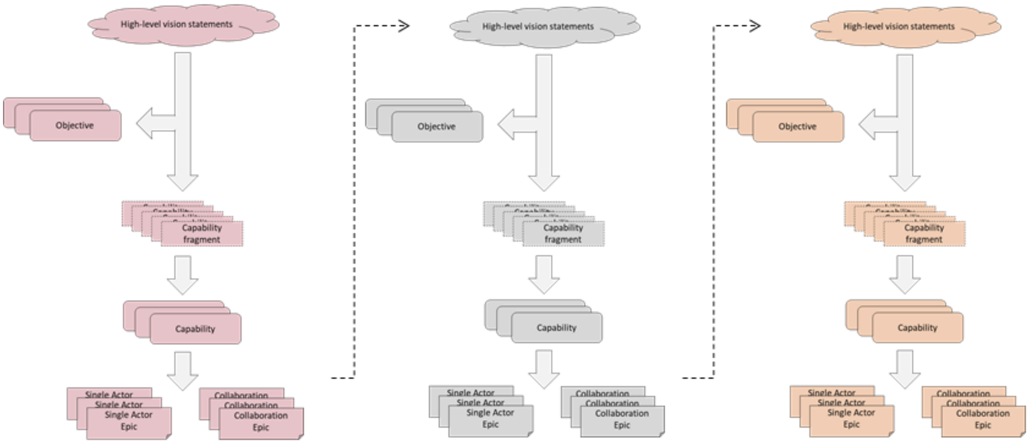


Figure 3: Increments of the method when using the C3 Taxonomy. Operational level (pink), user-facing level (grey), back-end COI level (orange).

technical, gives rise to a path of implementation into concrete processes and technical functionality that are loosely coupled to each other; see (Hannay et al., 2017c) for details. The purpose of having integral technical capabilities is to counter stove-piped technical solutions tailored specifically to a given operational capability.

Since each level has capabilities in their own right, one must consult the respective strategic visions for each level at each increment. For the same reason, each level has its own objectives. It may be the case that a nation's strategic visions is only stated at what amounts to the operational level of the C3 Taxonomy. In that case, other relevant documents that contain strategic visions for CIS capabilities should be consulted; for example, strategic documents in the NATO portfolio, or if not found there, general standards from bodies such as the Object Management Group, the Open Group, etc.

The C3 Taxonomy is under constant development and reevaluation, and many of the high-level capability groups are sparsely populated or empty. Individual analyses that come out of using our method can be provided as input to further development of the taxonomy. When we identify capabilities in the two cases below, some will be similar to existing ones in the taxonomy, while others will not exist. The essential point in our analysis is to distinguish between the operational, user-facing and back-end layers.

Network-Centric Operations

The strategic vision with which we will illustrate our method is that of Network-Centric Operations (Alberts et al., 1999). To express this vision, NATO developed its Network Enabled Capability (NNEC) Maturity Levels.

NNEC maturity levels

The NNEC Maturity levels express process maturity. They are, as are a variety of other maturity models, based on Humphrey's Capability Maturity Model (CMM) (Humphrey, 1988, 1989) for software development processes, originally developed for the U.S. Department of Defense (Domingo, 2010). The NNEC maturity level definitions are given in the upper part of Figure 4. The evolution from NNEC maturity level 1 to NNEC maturity level 5 is characterized by increasing capabilities to operate in a coordinated, collaborative and coherent manner, based in increasing organizational flexibility and dynamic sharing of resources. All this is based on sufficient communication; that is, on sufficient network "enabledness".

The NML levels relate to the operational capabilities level of the C3 Taxonomy. There is also a corresponding set of maturity levels for CIS capabilities. The lowermost part of Figure 4 summarises the NATO Networking and Information Infrastructure (NII) System Maturity Levels (SML) (NATO Consultation, Command and Control Agency, 2007).

Cases

We now illustrate the use of single actor epics and collaboration epics together to convert high level NNEC visions into capabilities.

The two cases we present are based on actual analyses completed for a nation's armed forces. Our recount is filtered to become unclassified and intended for illustration only.

In both cases, we start out with the following high-level vision from the nation's plan for the development of network-enabled capabilities toward NML 4:

> *Increased operational capability through networked collaboration, where the purpose is to use all resources in a more flexible and coordinated way in order to achieve higher efficiency.*

Specifically, we focus on the following goal:

> *Goal 2: Planning and execution: Increased ability to fast planning, flexible task execution and the use of integrated processes.*

NML1 – Standalone Organizations operate more or less independently. No commonly agreed interoperability processes are in place. Interoperability when needed is worked out in an ad-hoc fashion. Success depends on the experience, competence and resourcefulness of the people involved and not on the use of proven processes and procedures.

NML2 – Deconflict Organizations are separated in space or time to avoid interfering with one another. A basic level of agreed, interoperability processes are in place. These processes support the sharing of planning information, at a sufficient level of detail, to deconflict planned operations of units operating in close proximity to one another. These processes also support the updating of planning information during the conduct of a mission.

NML3 – Coordinate Organizations coordinate their activities to improve overall operational effectiveness. Additional interoperability processes are in place to support the exchange of information needed to coordinate operational plans and coordinate the operational execution of these plans.

NML4 – Collaborate Organizations work together in a collaborative fashion to further improve operational effectiveness. Additional interoperability processes are in place to support collaborative working arrangements among organizations during all phases of a mission. These processes support the rapid sharing of needed information to support collaborative working arrangements.

NML5 – Coherent Organizations are able to maximize overall operational effectiveness by working together in a coherent manner. Interoperability processes allow organizations to work together in a seamless manner. These processes support the sharing of information needed to support this manner of working.

SML1 – Isolated Standalone applications, databases and multiple types of communication networks. Interoperability is supported through liaison personnel and the use of 'air-gap' interfaces to share data.

SML2 – Functional Stovepipes Dedicated applications built to support different functions. These applications are often developed as separate systems, utilizing their databases and many cases their own networks. Multiple types of communication networks are involved. Little joint situational awareness and little collaboration due to incompatibility of applications, networks and data. Interoperability is supported through 'air-gap' interfaces and some physical connectivity between systems.

SML3 – Communicate and Inform Improvements in basic communications and information sharing capabilities. Improvements in communications will involve migration towards a single type of network for voice, data and video traffic and will include improved interoperability between static, deployable and mobile networks utilizing data link technology. Improvements in information sharing involve the ability of independent applications to exchange and use independent data components. Interoperability is generally improving, involving the ability of independent applications to exchange and use independent data components.

SML4 – Collaborate and Plan exploits improvements in shared situational awareness to improve decision making processes. Advanced collaborative tools will be used to support large scale vertical and horizontal collaboration and to support advanced adaptive planning processes within and across COIs. Interoperability will generally improve and include domain data and meta data models and the development of procedures for COIs. Improved data and meta data sharing among independent applications will allow theses applications to work together in an integrated fashion.

SML5 – Sense and Respond Proliferation of sensor capabilities and information sharing capabilities at all levels, and by continued improvements in functional capabilities to enable extremely rapid and agile responses to changing circumstances and fleeting opportunities. This phase involves major improvements in interoperability at the tactical level allowing any sensor to work with effector and any commander in near real time.Interoperability will generally improve, where a top-level perspective including enterprise data and meta data models and procedures will exist, and where data will be seamlessly shared among applications able to work together across domains in an integrated fashion.

Figure 4: NATO Network-Enabled Capability (NNEC) Maturity Levels (NML) and the NATO Networking and Information Infrastructure (NII) System Maturity Levels (SML) (abridged).

This goal has four sub-goals, of which we used text from two of them; namely, sub-goal 2.1 for Case 1 and sub-goal 2.2 for Case 2, to identify capability fragments. The sub-goals appear below, in each of the two cases. Although stemming from the same overall goal, the two cases are sufficiently different to warrant inclusion in this discussion. The two cases were, in fact, initiated independently of each other, and it was through our analysis that the link to a common goal was uncovered.

The strategy documents state that the NML and SML statements (Figur 4) should be seen as part of the general basis for its visions.

Case 1 – Planning and execution of operations

Sub-goal 2.1 is expressed summarily as: *The ability to plan and execute joint operations on several levels has been established.*

Figure 5 summarises the result of the analysis: Objectives, epics and capabilities (processes) are identified at the operational level (pink) in increment 1. Then, in increment 2, objectives, epics and capabilities (applications) are identified at the user-facing CIS level (grey). Increment 3 gives objectives, epics and capabilities (services) at the back-end CIS level (orange). Each level has its own objectives (omitted from figure to avoid clutter).

Notice how capabilities are used across more situations (epics) at deeper levels. This is in line with capabilities becoming increasingly generic at deeper levels of the C3 Taxonomy.

We now detail Case 1 in its three increments.

Increment 1 (operational level)

1-1: Identify objectives. From the above NML4 high-level vision, we formulated the following objectives:

- Increase portion of collaborative fast planning by a factor a
- Increase collaborative use of resources by a factor b

The measures a and b and their metrics are placeholders for actual measures, which must be established when projects or development programs are initiated to implement the above visions into systems.

1-2: Identify capability fragments. Two capability fragments were identified based on sub-goal 2.1:

- A *planning* capability fragment: The ability to plan joint operations on several levels.

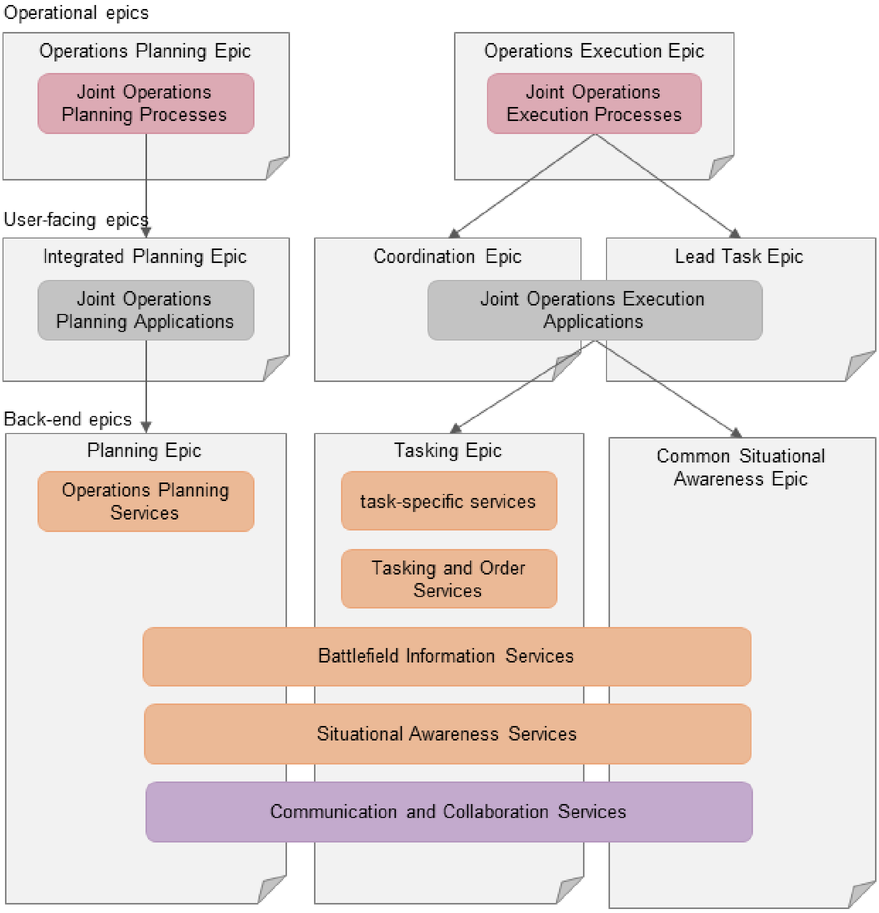

Operational epics
Operations Planning Epic
Joint Operations Planning Processes
Operations Execution Epic
Joint Operations Execution Processes
User-facing epics
Integrated Planning Epic
Joint Operations Planning Applications
Coordination Epic
Lead Task Epic
Joint Operations Execution Applications
Back-end epics
Planning Epic
Operations Planning Services
Tasking Epic
task-specific services
Tasking and Order Services
Common Situational Awareness Epic
Battlefield Information Services
Situational Awareness Services
Communication and Collaboration Services

Figure 5: Case 1: Epics and the capabilities they use.

- An *execution* capability fragment: The ability to execute joint operations on several levels.

1-3: Declare capabilities. From the two capability fragments, we declared two capabilities belonging at the operational capabilities layer of the C3 Taxonomy: *Joint Operations Planning Processes* and *Joint Operations Execution Processes*.

1-4: Capability specification. The main instrument for identifying capabilities is in the formulation of epics. We formulated epics by studying established national tactical scenarios for more concrete perceptions of how the capabilities should be used.

The two operational capabilities were identified in the following collaboration epics:

> Operations Planning Epic: Operational leaders and tactical leaders plan joint operations by using the *Joint Operations Planning Processes* to integrate plans for individual disciplines
>
> Operations Execution Epic: Tactical leaders LA and LB of units A and B in separate domains lead the coordinated execution of a plan involving A and B by using *Joint Operations Execution Processes* to
>
> - coordinate the movements of units A and B
> - lead A and B's execution of tasks

Increment 2 (user-facing CIS level)

2-1: Identify objectives. SML4 (Figure 4) implies that "interoperability" and "information sharing" are objectives to be reached through domain data and meta-data models and through improved data and meta-data sharing among independent applications. We therefore formulate the following objectives at the user-facing CIS level:

- Increase information sharing by a factor c
- Increase interoperability at user levels by a factor of d

Again, the measures c and d and their metrics are placeholders for actual measures to be established at project initiation time.

2-2: Identify capability fragments. Based on the identified capability fragments at the operational level, we derived the following capability fragments at the user-facing capability level:

- A *planning-support* capability fragment: The ability to support users in planning joint operations on several levels.
- An *execution-support* capability fragment: The ability to support users in executing joint operations on several levels.

2-3: Declare capabilities. We declared two user-facing CIS capabilities *Joint Operations Planning Applications* and *Joint Operations Execution Applications*.

2-4: Capability specification. The user-facing capabilities were specified in the following user-facing epics:

> Integrated Planning Epic: Tactical leaders LA and LB of units A and B in separate domains integrate plans for individual disciplines by using *Joint Operations Planning Applications* to perform
>
> - problem and mission analysis
> - courses of action development, validation and decision
> - concept and plan development
> - assessment and plan review

Following (Hannay et al., 2017c), the $\langle$ ... tasks t in $C\rangle$ epics at one level become the $\langle$ ... activities a in the their domain $D\rangle$ in related epics at lower levels in the taxonomy. Thus, for the Operations Planning Epic, the capability *Joint Operations Planning Process* is used to "integrate plans for individual disciplines", and the latter is then the domain activity in the Integrated Planning Epic.

Similarly, the Operations Execution Epic gave rise to one user-facing capability used in two user-facing epics, one for each of two sub-task required when executing an operation:

> Coordination Epic: Tactical leaders LA and LB of units A and B in separate domains coordinate the movements of units A and B by using *Joint Operations Execution Applications* to
>
> - create common situational awareness between A and B
> - exchange information about the plan execution and the battlefield
> - coordinate task execution between A and B

> Lead Task Epic: Tactical leaders LA and LB of units A and B in separate domains lead A and B's execution of tasks by using *Joint Operations Execution Applications* to
>
> - create and issue tasks which timely execute A and B's part of the plan
> - track the execution of the tasks in A and B

Increment 3 (back-end CIS level)

3-1: Identify objectives. The SML4 description in 4 states the use of advanced collaborative tools, supporting vertical and horizontal collaboration. Based on this system-level goal, we formulate:

- Obtain information in real time
- Obtain information at Mbit/s capacity

3-2: Identify capability fragments. Based on the identified capability fragments at the user-facing level, we derived the following capability fragments at the back-end capability level:

- A *planning-service* capability fragment: The provision of services that provide back-end functionality for planning operations on several levels.
- An *execution-service* capability fragment: The provision of services that provide back-end functionality for executing joint operations on several levels.

3-3: Declare capabilities. We declared six back-end CIS capabilities: *Operations Planning Services (OPS)*, *Tasking and Order Services (TOS)*, *Battlefield Information Services (BIS)*, *Situation Awareness Services (SAS)*, *Communication and Collaboration Services (CS)*, and task-specific services. The task-specific services is a placeholder for services that will be uncovered at later stages of development and detailing.

3-4: Capability specification. Further elaborations on the tactical scenarios gave rise to three back-end epics. The Planning epic, was derived from the *Joint Operations Planning Applications*. This epic describes the steps in a planning process, and the functional services that users rely on to perform the steps:

> Planning Epic: A service consumer performs
> - problem and mission analysis
> - courses of action development, validation and decision
> - concept and plan development
> - assessment and plan review
>
> by using
> - *Operations Planning Services (OPS)*
> - *Battlefield Information Services (BIS)*
> - *Situation Awareness Services (SAS)*
>
> to get text, images, video, map overlays, messages, voice, virtual telecommunications

From the *Joint Operations Execution Applications*, two back-end epics were derived. The first, Common Situational Awareness Epic, addresses the problem of several operational leaders which need to establish common situational awareness between them and with other leaders, and the functional services they use to achieve this:

Common Situational Awareness Epic: A service consumer

- creates common situational awareness between A and B
- exchanges information about the plan execution and the battlefield

by using

- *Battlefield Information Services (BIS)*
- *Situation Awareness Services (SAS)*

to get text, images, video, map overlays, messages, voice, virtual telecommunications

The other back-end epic derived from the *Joint Operations Execution Applications*, is the Tasking Epic, describing the steps involved in creating and performing the tasks which executes plan, and the supporting functional services:

Tasking Epic: A service consumer

- creates and issues tasks which timely execute A and Bs part of the plan
- tracks the execution of the tasks in A and B
- coordinates task execution between A and B

by using

- task-specific services
- *Tasking and Order Services (TOS)*
- *Battlefield Information Services (BIS)*
- *Situation Awareness Services (SAS)*

to get text, images, video, map overlays, messages, voice, virtual telecommunications

The services in the above epics rely on *Communication and Collaboration Services* which is indicated in Figure 6 as a C3 Taxonomy Core Service capability (violet), but which we do not elaborate on further here.

Case 2 Training

Sub-goal 2.2 can be summarised as “Training and operations is integrated, and performed in collaboration networks”. Figure 6 summarises the result of the analysis with three increments (objectives omitted). Notice again how capabilities are used across more situations (epics) at deeper levels.

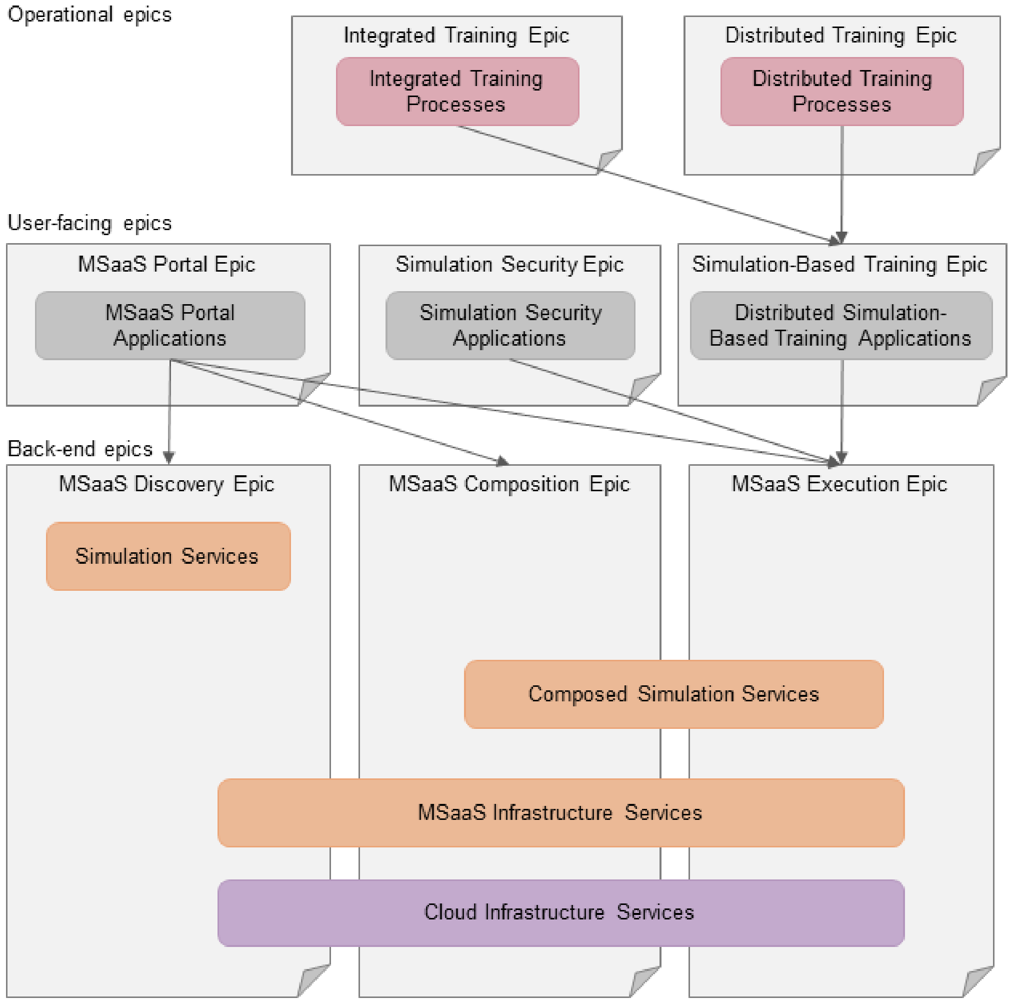

Operational epics
Integrated Training Epic
Integrated Training Processes
Distributed Training Epic
Distributed Training Processes
User-facing epics
MSaaS Portal Epic
MSaaS Portal Applications
Simulation Security Epic
Simulation Security Applications
Simulation-Based Training Epic
Distributed Simulation-Based Training Applications
Back-end epics
MSaaS Discovery Epic
Simulation Services
MSaaS Composition Epic
MSaaS Execution Epic
Composed Simulation Services
MSaaS Infrastructure Services
Cloud Infrastructure Services

Figure 6: Case 2: Epics and the capabilities they use.

Increment 1 (operational level)

1-1: Identify objectives. From the above vision statements, we more or less directly, formulated the following objectives:

- Increase training with systems for actual operations by a factor g
- Increase local collaborative training by a factor h

for placeholder measures g and h.

1-2: Identify capability fragments.

Two capability fragments were identified based on sub-goal 2.2:

- The *integrated training* capability fragment: The ability to integrate training with operational systems.
- The *distributed training* capability fragment: The ability to train collaboration from different locations.

1-3: Declare capabilities.

Based on the two capability fragments, two corresponding operational capabilities where identified; the *Integrated Training Processes* and *Distributed Training Processes*.

1-4: Capability specification.

Similarly as for the analysis for sub-goal 2.1, we formulated epics by studying established national tactical scenarios.

We formulated the following operational epics:

> Integrated Training Epic: Operational leaders and tactical leaders LA and LB in different domains A and B train (multi-national) collaboration by using *Integrated Training Processes* to engage in training using the systems they use in real operations
>
> Distributed Training Epic: Operational leaders and tactical leaders LA and LB in separate domains train (multi-national) collaboration by using *Distributed training processes* to engage in collaboration training from home base

Increment 2 (user-facing CIS level)

2-1: Identify objectives. The nation's strategy on network-enabled capabilities did not address simulation technology specifically, but SML4 and relevant documents such as the NATO Modelling & Simulation Master Plan (NATO Modelling and Simulation Group, 2012) imply the following objectives:

- Increase distributed simulation-based training by a factor i
- Decrease time to set up simulation-based training by a factor j

for placeholder measures i and j.

2-2: Identify capability fragments. From the MSaaS reference architecture (Hannay and van den Berg, 2017), in addition to the above strategic documents, we identified these capability fragments:

- A *simulation-based training* capability fragment: The ability to support users to partake in simulation-based training
- A *simulations building* capability fragment: The ability to support exercise managers to build simulation-based training.

2-3: Declare capabilities. We declared three user-facing CIS capabilities *Distributed Simulation-Based Training Applications*, *Simulation Security Applications* and *MSaaS Portal Applications*.

2-4: Capability specification. The user-facing capabilities were specified in the following user-facing epics:

> Distributed Simulation-Based Training Epic: Operational leaders and tactical leaders LA and LB in different domains engage in distributed training using the systems they use in real operations by using *Distributed Simulation-Based Training Applications* to engage in training where operational systems interact with simulations

The next two epics are for exercise managers, which are not personnel under training. Figure 6 has operational epics only for personnel under training, and the two epics therefore mark the start of the story-line for the exercise managers.

The first epic here pertains to the challenges of information flow between operational systems and (distributed) simulation systems that operate in different security domains (Möller et al., 2012). For example, when coupling a highly classified fighter jet flight simulator with an army tactical engagement simulator for training close air support to land forces, the high fidelity simulation data necessary to train flight skills would be contained within the flight simulator, and only information at the lower classification level of the army simulation system would be shared.

> Simulation Security Epic: Exercise managers E1 and E2 in separate domains manage security issues in (distributed) simulations by using *Simulation Security Applications* to set up cross-domain solutions and multilevel security

The next epic engages the modelling & simulation as a service (MSaaS) paradigm (Hannay and van den Berg, 2017; van den Berg et al., 2018), which envisions the rapid composition of simulations from a repository of common standardized simulation services. An MSaaS portal gives tool support for (technically skilled) exercise managers to build such simulations.

> MSaaS Portal Epic: Exercise managers E1 and E2 in separate domains tailor distributed simulation-based training for the training objectives at hand by using *MSaaS Portal Applications* to discover appropriate *Simulation Services*, compose them to *Composed Simulation Services* and execute them

Increment 3 (back-end CIS level)

3-1: Identify objectives. Documents such as the Allied Framework for Modelling and Simulation as a Service (MSaaS) Governance Policies (NATO Standardization Office, 2018) imply the following objectives for our purposes:

- Decrease simulation build time by a factor k
- Increase location-independent availability of simulations by a factor l
- Increase time-independent availability of simulations by a factor m

for placeholder measures k, l and m. Simulation build time is the time it takes to put together a simulation from scratch or from components; perhaps using an off-the-shelf general-purpose simulation framework. Today, this build time is often unacceptably long.

3-2: Identify capability fragments. Based on the the MSaaS reference architecture (Hannay and van den Berg, 2017), we derived the following capability fragments at the back-end capability level:

- A *simulations enabling services* capability fragment: The provision of services that provide back-end support functionality for simulations.
- A *simulations services* capability fragment: The provision of services that provide simulations.

3-3: Declare capabilities. We declared the capabilities *MSaaS Infrastructure Services*, *Simulation Services*, *Composed Simulation Services* and *Cloud Infrastructure Services*.

3-4: Capability specification. We formulated the following back-end epics. They rely on the MSaaS reference architecture (Hannay and van den Berg, 2017), which declares simulation functionality as simulation services to be composed into composed simulation services that give (distributed) simulations. The discovery, composition and execution is facilitated by MSaaS Infrastructure Services (Hannay et al., 2020).

> MSaaS Discovery Epic: service consumers discover appropriate *Simulation Services* by using *MSaaS Infrastructure Services* to search for simulation functionality in cloud repositories using service descriptions and metadata
>
> MSaaS Composition Epic: service consumers compose *Simulation Services* to *Composed Simulation Services* by using *MSaaS Infrastructure Services* to put together simulation functionality using service descriptions and metadata
>
> MSaaS Execution Epic: service consumers run *Composed Simulation Services* with cross-domain solutions and multilevel security by using *MSaaS Infrastructure Services* to execute simulations in (distributed) cloud environments

MSaaS relies on *Cloud Infrastructure Services* (Kratzke and Siegfried, 2020; Kratzke, 2018; Kratzke and Quint, 2017), which is indicated in Figure 6 as a C3 Taxonomy Core Service capability (violet), but which we do not elaborate on further here.

Prioritising epics on objectives

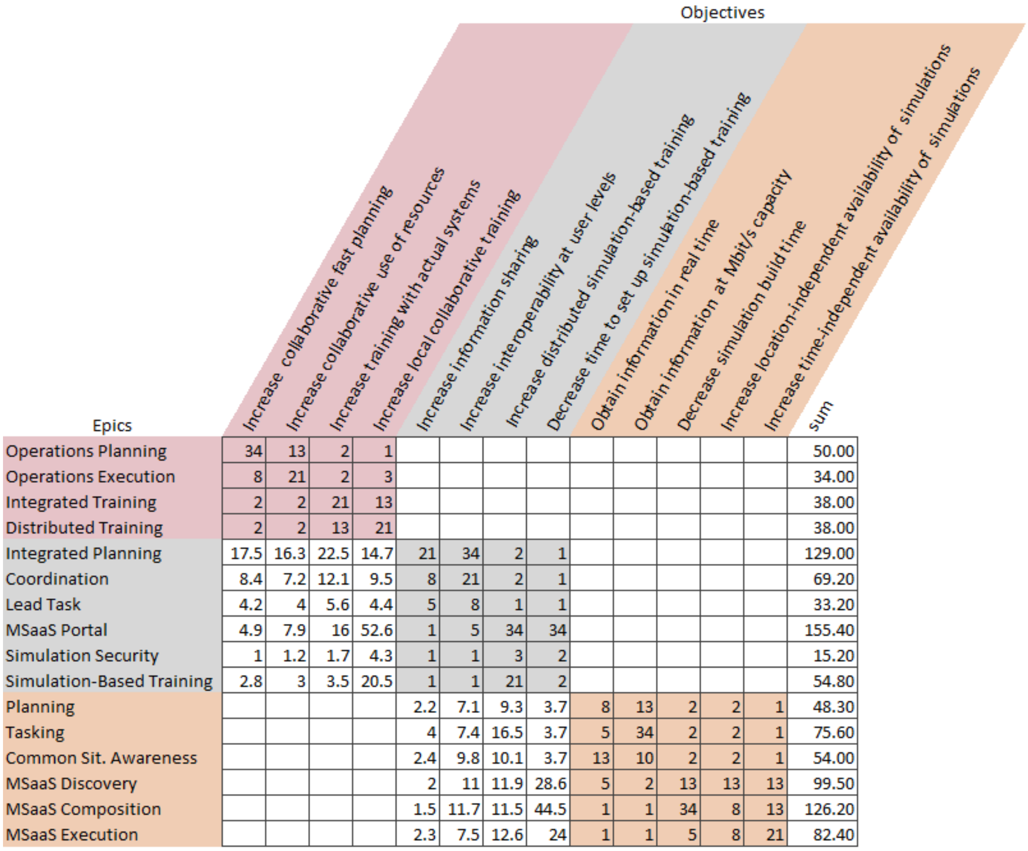
Epics in the form above should represent minimum viable products (Lenarduzzi and Taibi, 2016), in the sense of giving a unit of integral functionality that can be deployed and that gives value to the organisation. To prioritise which epics to develop in what order, one can assess how much epics contribute to a given set of objectives (Hannay et al., 2017a,b).

Here, we adapt that assessment method to work with our layered structure of epics. Although this has not yet been done by the nation's strategic staff, Figure 7 illustrates, with fictitious numbers, how our epics can be assessed according to how much they are expected to contribute to our objectives. (For now, ignore the numbers on white background.) Using the Fibonacci numbers for relative assessment as is common from *planning poker*

(Cohn, 2005), one can assign *benefit points*.[9] For example, the Operations Planning Epic is estimated to contribute much more (34) to the objective "Increase collaborative fast planning", than the other operational epics (8, 2, 2, respectively). Epics at a given level are assessed on the corresponding level's objectives as indicated in the pink, grey and orange quadrants in the table.

Further, we can also account for the fact that objectives at lower levels will usually have an effect on objectives at higher levels. Figure 8 illustrates how objectives at the user-facing level have effects on objectives at the operational level and how objectives at the back-end level have effects

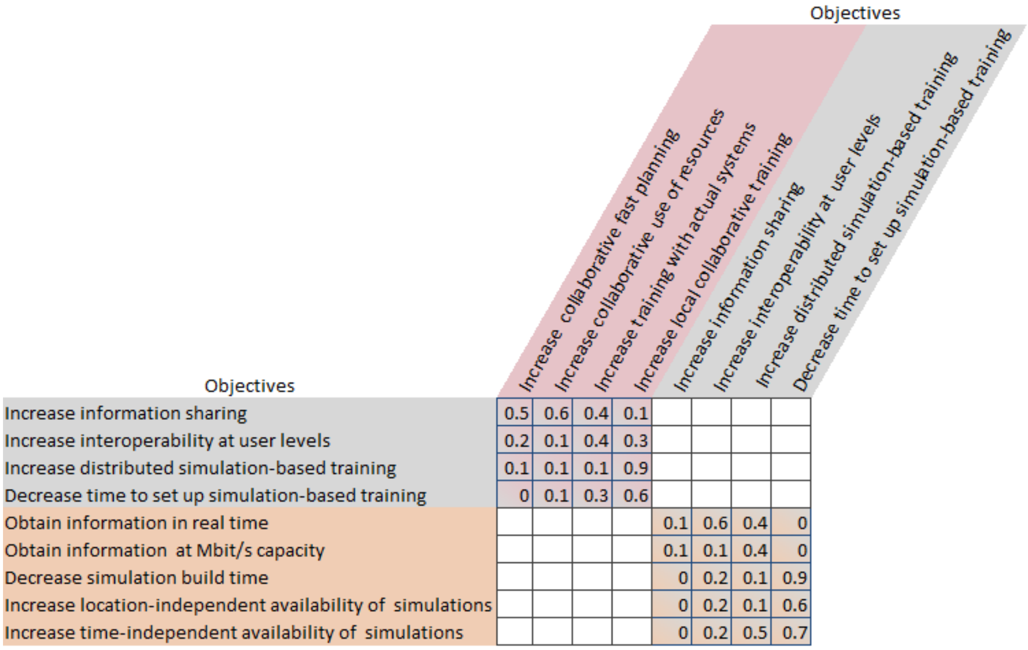
[9] Planning poker is a group estimation technique where stakeholders present their cost estimates by displaying cards from a deck featuring the Fibonacci numbers. The highest and lowest bidder explain their rationale and another bidding round commences. The process continues until bids are consistent, signalling that group consensus has been reached. The technique has lately been adopted to benefit estimation as well.



Objectives

Epics	Increase collaborative fast planning	Increase collaborative use of resources	Increase training with actual systems	Increase local collaborative training	Increase information sharing	Increase interoperability at user levels	Increase distributed simulation-based training	Decrease time to set up simulation-based training	Obtain information in real time	Obtain information at Mbit/s capacity	Decrease simulation build time	Increase location-independent availability of simulations	Increase time-independent availability of simulations	sum
Operations Planning	34	13	2	1										50.00
Operations Execution	8	21	2	3										34.00
Integrated Training	2	2	21	13										38.00
Distributed Training	2	2	13	21										38.00
Integrated Planning	17.5	16.3	22.5	14.7	21	34	2	1						129.00
Coordination	8.4	7.2	12.1	9.5	8	21	2	1						69.20
Lead Task	4.2	4	5.6	4.4	5	8	1	1						33.20
MSaaS Portal	4.9	7.9	16	52.6	1	5	34	34						155.40
Simulation Security	1	1.2	1.7	4.3	1	1	3	2						15.20
Simulation-Based Training	2.8	3	3.5	20.5	1	1	21	2						54.80
Planning					2.2	7.1	9.3	3.7	8	13	2	2	1	48.30
Tasking					4	7.4	16.5	3.7	5	34	2	2	1	75.60
Common Sit. Awareness					2.4	9.8	10.1	3.7	13	10	2	2	1	54.00
MSaaS Discovery					2	11	11.9	28.6	5	2	13	13	13	99.50
MSaaS Composition					1.5	11.7	11.5	44.5	1	1	34	8	13	126.20
MSaaS Execution					2.3	7.5	12.6	24	1	1	5	8	21	82.40

Figure 7: Benefit points: Epics' contributions to objectives. Operational level (pink), user-facing level (grey), back-end level (orange). Fictitious example.



Objectives	Increase collaborative fast planning	Increase collaborative use of resources	Increase training with actual systems	Increase local collaborative training	Increase information sharing	Increase interoperability at user levels	Increase distributed simulation-based training	Decrease time to set up simulation-based training
Increase information sharing	0.5	0.6	0.4	0.1				
Increase interoperability at user levels	0.2	0.1	0.4	0.3				
Increase distributed simulation-based training	0.1	0.1	0.1	0.9				
Decrease time to set up simulation-based training	0	0.1	0.3	0.6				
Obtain information in real time					0.1	0.6	0.4	0
Obtain information at Mbit/s capacity					0.1	0.1	0.4	0
Decrease simulation build time					0	0.2	0.1	0.9
Increase location-independent availability of simulations					0	0.2	0.1	0.6
Increase time-independent availability of simulations					0	0.2	0.5	0.7

Figure 8: Lower-level objectives' contributions to higher-level objectives. Fictitious example.

on objectives at the user-facing level. For example, any effect that an epic might have on the metric of "Increase information sharing" is expected to also give a 0.5 per unit effect on the metric of "Increase collaborative fast planning". This gives rise to the indirect contributions of epics on objectives as illustrated by the numbers in the white off-diagonal quadrants in Figure 7. (For example, the user-facing CTS Integrated Planning Epic contributes indirectly $17.5 = 0.5 \times 21 + 0.2 \times 34 + 0.1 \times 2 + 0 \times 1$ to the operational objective "Increase collaborative fast planning".

Taken together, the direct and indirect contributions give a benefit score for each epic in the sum column in Figure 7. In this fictitious example, if one wanted to produce as much value as soon as possible, one might start by implementing the MSaaS Portal Epic, followed by the Integrated Planning Epic and the MSaaS Composition Epic. Note that when stakeholders state dependencies as in Figure 8, the effect can be that lower level capabilities should be developed first, as in this example. However, stakeholders might express lower dependencies and higher integral benefit of, say, operational epics, which would reflect more technology-independent benefits associated with the operational capabilities. This might lead to a conclusion that certain operational epics should be developed first, regardless of technology. Such elaborations are crucial in making conscious and deliberate choices regarding the order of development.

If one also has cost estimates for epics, perhaps in the form of story points (Cohn, 2005), one can combine benefit points and story points into benefit/cost index and realise those epics first that give the highest benefit to cost ratio (Hannay et al., 2017b). This enables stakeholders and sponsors to engage in benefits management (Ward and Daniel, 2015).

Finally, notice how the two cases represent modest collections of epics that implement clearly delineated specific sub-goals at a time in a vision. This, we state quite distinctly, is the only way to proceed when developing large defence portfolios (Hannay et al., 2017c).

Final Remarks

The main contribution of this work is to establish a link from unstructured strategic documents to formatted descriptions in an existing structure; in this case an existing taxonomy of capabilities.

A further contribution is to show how techniques from agile practices, requirements engineering and benefits management can help to delineate manageable-sized portfolios of functionality that can be realised to yield high benefit early. We view methodological efforts such as these as vital for the success of implementing strategic visions. Experience shows that the lack of such methods may lead to substantial waste of effort and failure to accomplish the vision and goals of defence in a timely manner.

The method described in this article is intended to aid in operationalising strategic visions. The key to this is the phrasing of a vision into capabilities; where a capability is a well-delineated piece of operational or technical functionality. To help identify capabilities, tactical scenarios can be studied to phrase narratives (epics) that indicate how capabilities are used.

We demonstrated how to use the method in two cases. The two cases also provide an indication of the usability of the method. Further experience is needed to establish and refine usability. However, at this point, we would claim that *any* method that helps structure visions and strategic ideas into manageable parts for implementation, is better than what is the case in many projects and programs, namely to hold ones breath and rely on some sort of good fortune in crossing the chasm between strategic vision and implementation.

Acknowledgements

The authors are grateful to Trude Bloebaum and the reviewers for comments that helped improve this article. This research is funded in part by the

Research Council of Norway under project no. 282081 “MixStrEx”.

References

D. S. Alberts, J. J. Garstka, and F. P. Stein. *Network centric warfare: developing and leveraging information superiority.* CCRP Publication Series. DoD Command and Control Research Program, 1999.

D. S. Alberts, R. K. Huber, and J. Moffat. NATO NEC C2 maturity model. Technical report, DoD Command and Control Research Program, 2010.

M. Cohn. *Agile Estimating and Planning.* Prentice Hall, 2005.

A. Domingo. *On the Development of a Maturity Model and Maturity Levels for NATO Network Centric Capability*, 2010.

T. Erl. *SOA principles of Service Design.* Prentice Hall, Upper Saddle River (NJ), USA, 2007.

J. E. Hannay and T. W. van den Berg. The NATO MSG-136 Reference Architecture for M&S as a Service. In *Proc. NATO Modelling and Simulation Group Symp. on M&S Technologies and Standards for Enabling Alliance Interoperability and Pervasive M&S Applications (STO-MP-MSG-149).* NATO Science and Technology Organization, 2017. paper 13.

J. E. Hannay, H. C. Benestad, and K. Strand. Benefit points—the best part of the story. *IEEE Software*, 34(3):73–85, 2017a.

J. E. Hannay, H. C. Benestad, and K. Strand. Earned business value management—see that you deliver value to your customer. *IEEE Software*, 34(4):58–70, 2017b.

J. E. Hannay, K. Brathen, and O. M. Mevassvik. Agile requirements handling in a service-oriented taxonomy of capabilities. *Requirements Engineering*, 22(2):289–314, 2017c.

J. E. Hannay, T. W. van den Berg, S. Gallant, and K. Gupton. Modeling and simulation as a service infrastructure capabilities for discovery, composition and execution of simulation services. *J. Defense Modeling and Simulation: Applications, Methodology, Technology*, 2020. `https://doi.org/10.1177/1548512919896855`.

W. S. Humphrey. Characterizing the software process: A maturity framework. *IEEE Software*, 5(2):73–79, 1988. doi: 10.1109/52.2014.

W. S. Humphrey. *Managing the Software Process.* SEI Series in Software Engineering. Addison-Wesley Professional, 1989.

N. Kratzke. A brief history of cloud application architectures. *Appl. Sci.*, 8 (8), 2018. paper 1368.

N. Kratzke and P.-C. Quint. Understanding cloud-native applications after 10 years of cloud computing – A systematic mapping study. *J. Systems and Software*, 126:1–16, 2017.

N. Kratzke and R. Siegfried. Towards cloud-native simulations—lessons learned from the front-line of cloud computing. *J. Defense Modeling and Simulation: Applications, Methodology, Technology*, 2020. https://doi.org/10.1177/1548512919895327.

V. Lenarduzzi and D. Taibi. MVP explained: A systematic mapping study on the definitions of Minimal Viable Product. In *2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 112–119, Aug 2016.

B. Möller, S. Croom-Johnson, T. Hartog, W. Huiskamp, C. Verkoelen, G. Jones, , and M. Bennet. Security in NATO collective mission training—problem analysis and solutions. In *Proc. 2012 Spring Simulation Interoperability Workshop (SIW)*, number 12S-SIW-032. Simulation Interoperability Standards Organization, 2012.

NATO Allied Command Transformation. Report of the NATO Network Enabled Capability (NNEC) conference 23 to 25 april 2013. Technical report, 2013.

NATO Consultation, Command and Control Agency. Report on the development of the initial NNEC maturity levels and its applications. Technical Report NC3A-BE/CIS/IPT-1/07/001, 2007.

NATO Modelling and Simulation Group. *NATO Modelling and Simulation Master Plan (version 2.0)*, 2012. Doc. no. AC/323/NMSG(2012)-015.

NATO Standardization Office. AMSP-02 Allied Framework for Modelling and Simulation as a Service (MSaaS) Governance Policies, edition (a), version 1.0. Technical Report STANREC 4794, 2018.

K. Sandkuhl. Investigating the potential of capability-driven design and delivery in an SME case study. In A. Persson and J. Stirna, editors, *Advanced Information Systems Engineering Workshops*, volume 215 of

Lecture Notes in Business Information Processing, pages 3–14. Springer, 2015.

The Open Group. *SOA Reference Architecture Technical Standard*, 2011a. Doc. no. C119.

The Open Group. *TOGAF Version 9.1 Enterprise Edition*, 2011b. Doc. no. G116.

T. W. van den Berg, W. Huiskamp, R. Siegfried, J. Lloyd, A. Grom, and R. Phillips. Modelling and Simulation as a Service: Rapid deployment of interoperable and credible simulation environments – an overview of NATO MSG-136. In *Proc. 2018 Winter Simulation Innovation Workshop*, number 18W-SIW-018. Simulation Interoperability Standards Organization, 2018.

J. Ward and E. Daniel, editors. *Benefits Management: How to increase the business value of your IT projects, Second Edition.* John Wiley Sons, Ltd, second edition, 2015.