Complex Systems Informatics and Modeling Quarterly (CSIMQ)
eISSN: 2255-9922
Published online by RTU Press, https://csimq-journals.rtu.lv
Article 263, Issue 47, June/July 2026, Pages 48–83
https://doi.org/10.7250/csimq.2026-47.03

Enhancing Military Architectural Frameworks with Method Chunks: A Modular Approach to Agile C2-System Design

Jan Lundberg[1,2*], Kent Andersson[1], and Janis Stirna[2]

[1] Swedish Defence University, Drottning Kristinas väg 37, 114 28 Stockholm, Sweden
[2] Department of Computer and Systems Sciences, Stockholm University, Borgarfjordsgatan 12, 164 55 Kista, Sweden

jan.lundberg@dsv.su.se, kent.andersson@fhs.se, js@dsv.su.se

Abstract. Military command and control system (C2-system) design is increasingly shaped by emerging technologies such as AI-enabled decision support, multi-sensor fusion and autonomous intelligence, surveillance and reconnaissance (ISR) capabilities. Architecture frameworks such as the NATO Architecture Framework (NAF) provide a structure for describing complex systems, but are less explicit about how to select entry points and sequence modeling activities. This article proposes a modular, context-informed framework, an extension to NAF, that combines *Navigation Guidance* with a repository of seven reusable *Method Chunks* (MCs) to support architectural design activities in a C2-system context. Using Design Science Research (DSR) as the overarching framework, this work is based on Situational Method Engineering (SME) and MAP-based formalization. A demonstration instantiates the framework for integrating a next-generation ISR capability into a C2-system. The resulting architecture and modeling activities link operational workflow to capability framing and cross-level capability dependencies and are consolidated through documentation and traceability artifacts. Collectively, the architecture supports reasoning about mission impact, capability motivation, coordination requirements, and traceability, while keeping later models anchored in a relevant operational context and workflow. The contribution of this work is methodological. The demonstration illustrates how a specific NAF-aligned modeling progression can be instantiated for an ISR integration case.
Keywords: NATO Architecture Framework (NAF), Command and Control System (C2-system), Situational Method Engineering (SME), Design Science Research (DSR), MAP-Based Formalization.

* Corresponding author

Reference: J. Lundberg, K. Andersson, J. Stirna, "Enhancing Military Architectural Frameworks with Method Chunks: A Modular Approach to Agile C2-System Design," Complex Systems Informatics and Modeling Quarterly, CSIMQ, no. 47, pp. 48–83, 2026. Available: https://doi.org/10.7250/csimq.2026-47.03

Additional information. Author ORCID iD: J. Lundberg – https://orcid.org/0009-0008-0485-6005, K. Andersson – https://orcid.org/0000-0001-6104-5788, and J. Stirna – https://orcid.org/0000-0002-3669-832X. PII S225599222600263X. Received: 28 January 2026. Accepted: 8 July 2026. Available online: 31 July 2026.

1 Introduction

Throughout history, military actors have consistently sought to translate emerging technologies into operational advantages. Those who are first to successfully integrate new technologies into capabilities often gain a decisive edge by surprising adversaries and exploiting their lack of preparedness [1]. As the pace of technological innovation continues to accelerate, the ability to adapt effectively to change has become an increasingly critical determinant of military effectiveness [2]. Today, integrating potentially disruptive technologies such as Artificial Intelligence (AI) and Robotics and Autonomous Systems (RAS) into military capabilities is essential for maintaining tactical superiority and ensuring readiness. In response, military command and control systems (C2-systems), central to decision-making, coordination, and operational execution, must evolve to meet the demands of Multi-Domain Operations (MDO), which require synchronization across the air, land, sea, cyber, and space domains [3]. Given the interconnected nature of personnel, technology, organizational structures, and doctrine, it is therefore essential to conceptualize C2-systems as Socio-Technical Systems (STS) when addressing this challenge [4], [5]. Consequently, the increasing complexity of modern military operations underscores the need for methodologies that can guide the design and evolution of C2-systems, aligning technical integration with organizational and doctrinal coherence [6].

Architectural frameworks, such as the NATO Architecture Framework (NAF) [7], provide a structure for describing and analyzing complex System-of-Systems (SoS), ensuring traceability from capability needs to physical implementations. However, existing frameworks often exhibit certain limitations, primarily focusing on *what* should be modeled (viewpoints and representations) while providing limited guidance on *how* to conduct design activities. This gap leaves C2-system designers without adequate methodological support for assessing integration strategies or evaluating operational consequences within dynamic, real-world military contexts. The main problem addressed in this research is the lack of process guidance in NAF for integrating emerging technologies into C2-systems.

Addressing this methodological shortfall forms the foundation of and motivates our research project, guided by Design Science Research (DSR) [8]. Earlier stages of this project established the conceptual and architectural foundation, outlined in Section 3.1, necessary for the development of a robust framework designed to support the integration of emerging technologies into military C2-systems.

Building upon this foundation, this article addresses the methodological requirements captured by the high-level goals (see Table 1, reported in [5]). We operationalize the first full iteration of the envisioned framework by expanding the method repository with the necessary method chunks. For transparency, the two method chunks, outlined already in [9], are also presented. This work moves the research project closer to providing a complete, actionable framework that supports designers and commanders in navigating the complexity of integrating emerging technologies into existing military capabilities.

The remainder of this article is structured as follows. Section 2 reviews the background and related work, including NAF and Situational Method Engineering (SME), a discipline that focuses on creating context-specific methodologies. Section 3 outlines the DSR methodology, i.e., the research approach, detailing how this work contributes to the overall project and how the remaining method chunks were developed. Section 4 presents the proposed structure and detailed content of the newly developed method chunks, linking them directly to the corresponding high-level goals. In Section 5, a demonstration illustrates *how* our envisioned framework can be applied to a specific situation. Section 6 presents a discussion on the practical and theoretical contributions and defines limitations. Finally, Section 7 concludes the discussion and defines directions for future demonstration and evaluation.

2 Background and Related Work

2.1 NAF – NATO Architecture Framework

NAF [7] is a widely adopted architectural framework for structuring, describing, and analyzing complex systems in military contexts. Earlier versions of NAF, e.g., version 3, organized architectures through predefined *views*, such as Operational (NOV), System (NSV), and Capability views (NCV). Each view represented a specific architectural product, often diagrams, matrices, or tabular descriptions, that focused primarily on how information was presented. NAF version 4 replaces this document-oriented structure with a model-oriented framework organized around *viewpoints* and *aspects*, as shown in Figure 1. This two-dimensional structure enables both vertical coherence across abstraction levels and horizontal traceability across architectural concerns.

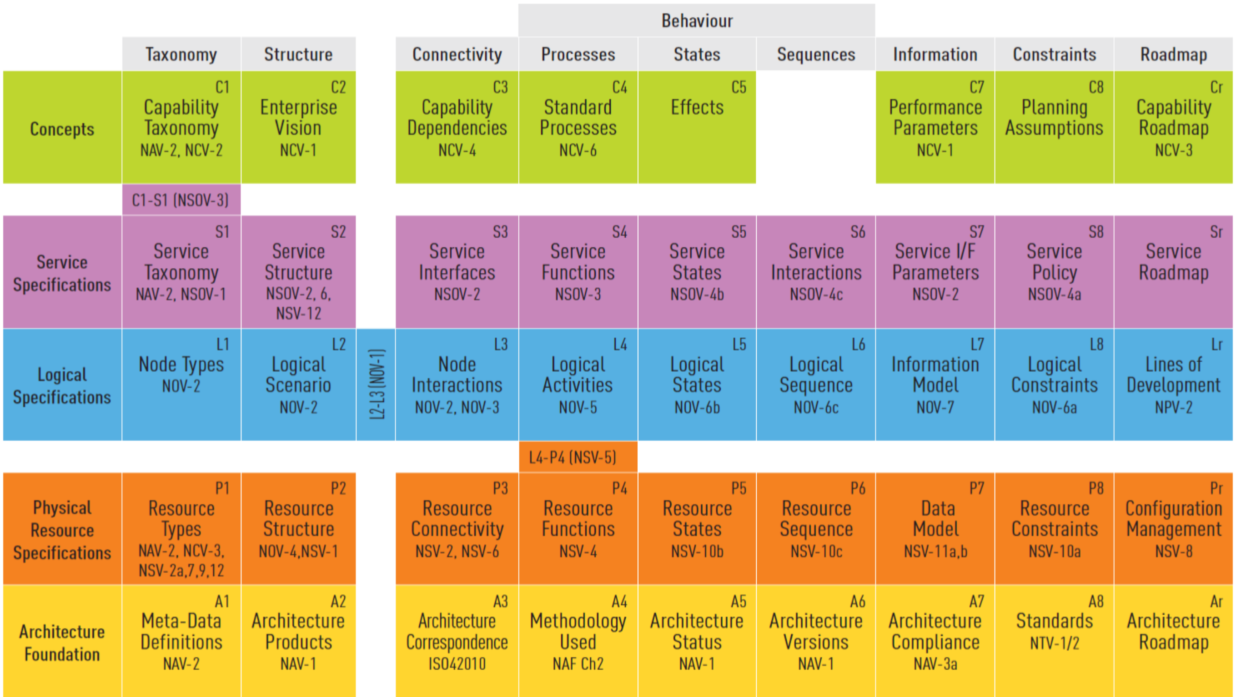

Figure 1. NAF version 4, viewpoints[†]

This grid representation provides a foundation for modeling and analyzing architectures through both vertical and horizontal dimensions. The vertical axis defines five Viewpoints of concern, namely, Concepts (C), Service Specifications (S), Logical Specifications (L), Physical Resources (P), and Architecture Foundation (A), each specifying the type and scope of architectural information to be represented.

The horizontal axis defines nine Aspects of concern, including Taxonomy, Structure, Connectivity, Processes, States, Sequences, Information, Constraints and Roadmap. These aspects determine the kind of concern being addressed. Each intersection of a Viewpoint and an Aspect defines a specific cell (e.g., L4 Logical Activities, P3 Resource Connectivity) representing a distinct architectural perspective. This two-dimensional structure allows designers to model both vertical coherence across abstraction levels and horizontal traceability across architectural concerns.

Although NAF version 4 replaces the view-based structure of NAF version 3, the grid representation retains legacy identifiers such as NSV and NCV for backward traceability. These identifiers indicate which NAF version 3 views have been integrated into the corresponding NAF version 4 Viewpoint/Aspect intersections (e.g., NSV-6 System Information Exchange corresponds

† https://commons.wikimedia.org/wiki/File:NAFv4_Grid.png

to P3 Resource Connectivity). In NAFv4, a View is the model instance created according to the conventions of its Viewpoint, while a Representation specifies how that model is visualized or documented (e.g., diagrams, matrices, etc.).

In this study, the Method Chunks (MCs) operationalize NAF logic by activating specific intersections within this grid representation. Each MC identifies the relevant viewpoints and representations to structure the design intentions of the MC. The proposed MCs therefore expand and improve NAF by linking them to specific components of NAF for a specific design purpose. The MCs have been developed in such a way that they support designers in modeling and analyzing C2-system architectures iteratively.

2.2 Situational Method Engineering (SME) and Method Chunks (MCs)

To complement NAF with methodological guidance, this research applies SME as an approach for constructing and adapting methods to specific design contexts. SME is based on the principle that no single, universal method can address all design contexts. Instead, SME encourages modularity and flexibility, allowing methods to be assembled from smaller, reusable components suited to the particular characteristics of a design situation. Within SME, a method is composed of method components, reusable fragments of methodological knowledge (e.g., modeling techniques or process steps). Combined, this approach enables the construction of a method repository, where designers can select and assemble appropriate components to suit the specific needs of a design context [10], [11], [12].

A core concept in SME is MC, a reusable method fragment that supports a specific design activity. Each MC specifies an intention (what to achieve) and a strategy (how to achieve it) and can be combined or tailored to a specific design need [13], [14].

By applying SME, our research approach ensures that the proposed framework remains adaptable, reusable, and context-sensitive, supporting design activities within military C2-system development.

2.3 Related Work

Architectural frameworks such as NAF [7], DoDAF [15] and TOGAF [16] provide a basis for governing complex and evolving SoS designs, such as military C2-systems. However, comparative studies, such as [17], have primarily focused on structural and representational aspects while often excluding explicit evaluation of methodological guidance. This reinforces the need for complementary research that extends architectural frameworks with method support, a focus addressed in this work.

Several attempts have been made to bridge this gap by introducing dedicated methodological processes into existing architectural frameworks. For instance, [18] introduces a *Fit-for-Purpose Human Views* methodology that enhances DoDAF with a six-step process for translating stakeholder concerns into architecture views, thereby addressing the lack of methodological guidance in existing framework practice.

Beyond defense-related architectures, method engineering principles have also been applied to tailor Enterprise Architectures (EAs) to a specific context. In [19], the authors propose a method base for EA management, explicitly motivated by the insight that EA methods often neglect situational factors such as organizational goals or context. Similarly, in [20], the authors develop a situational method for enterprise-architecture documentation that selects and frames documentation approaches depending on the project situation. Additionally, in [21], the authors introduce the SEA (Situational Enterprise Architecture) approach that formalizes EA components, using TOGAF, to enable context-driven selection of method components rather than an inflexible adoption. Similarly, [22] examines the implementation of TOGAF and NAF within the Norwegian Armed Forces and uncovers methodological limitations in both frameworks. While they provide a general process architectural view, they provide limited support for guiding *how* designers should proceed through design activities or adapt methods to specific contexts. Collectively, these works highlight a common challenge across architectural frameworks, the absence of explicit, situationally adaptable method support to guide how architectures are developed and linked to stakeholders' needs.

The International Council on Systems Engineering (INCOSE) [23] explicitly encourages the principle of process tailoring, emphasizing that methods, activities, and tools must be adapted to a specific context. This perspective aligns with the core principles of SME, which argue that no single method can address all design situations [14]. SME operationalizes adaptability through modular method construction, enabling designers to assemble and tailor method components to specific contexts. When combined, the systems engineering principle of process tailoring and the modular construction logic of SME provide a baseline for expanding context-driven methodological guidance to frameworks such as NAF. We argue that this would enable designers to adapt methods to specific design activities.

Building on these insights, this study advances beyond prior work. While earlier studies have recognized the lack of methodological guidance, none have systematically integrated the principles of SME or the use of MCs as a method extension within frameworks such as NAF. To this end, our research: (1) attaches MCs explicitly to NAF viewpoints and representations, (2) provides MAP-based navigation guidance focusing on emerging-technology integration decisions, and (3) demonstrates utility in a military C2-system integration scenario.

In the next section, we outline our research approach to address the stated methodological gap.

3 Method

3.1 Design Science Research Methodology

This research project follows the DSR methodology, which structures design activities into five iterative phases: (1) explicate problem, (2) define requirements, (3) design and develop artifact, (4) demonstrate artifact, and (5) evaluate artifact [24]. Each phase builds on insights from the previous phase, but the central mechanism is continuous feedback and refinement, both between consecutive phases and within iterations of the full cycles. Maintaining traceability between the results from evaluation in a relevant context, with new requirements on the solution, and with artifact development, is key to scientific validity. The work presented here contributes parts of a first iteration, with a focus on the demonstration phase. The research process used in this work and the related publications are outlined in Figure 2.

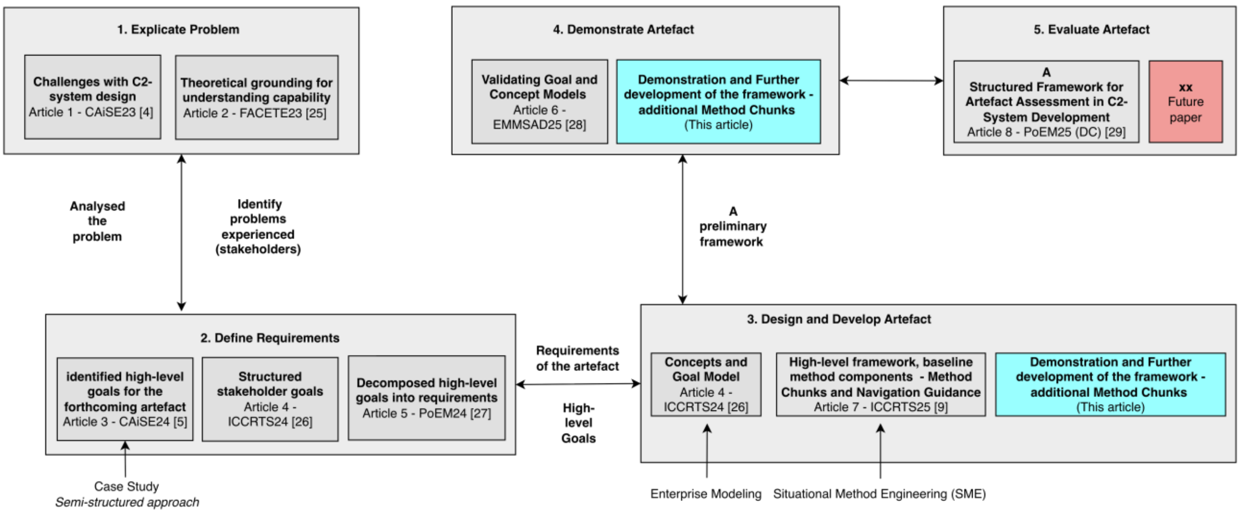

Figure 2. Overview of the research process and related publications (submitted/future/planned)

In the overarching research project, the *Explicate problem* phase was addressed in Article 1 [4], which identified challenges associated with military C2-system design, and in Article 2 [25], which provided a theoretical baseline for understanding *capability* within complex socio-technical systems (STS), such as military C2-systems. Together, these studies argued that existing architectural frameworks, such as NAF, provide strong modeling support but limited methodological guidance for integrating emerging technologies into military C2-systems.

The Define requirements phase was addressed through Article 3 [5] and Article 4 [26], where stakeholder analysis, concept modeling, and goal modeling were used to elicit high-level design goals. The high-level goals were refined through enterprise modeling techniques in Article 5 [27], where they were decomposed into specific design requirements. These activities resulted in a set of eleven high-level goals (see Table 1) and a set of design requirements that together formed the foundation for the next step, the artifact construction.

Table 1. A complete table of high-level goals for the envisioned framework, first reported in [5]

Overarching Goal	**A conceptual framework supporting the integration of emerging technologies into military C2-systems.**
G1	Be able to leverage from a high system level to a lower, i.e., adopting a holistic approach.
G2	Be able to evaluate a C2-system holistically, i.e., a multi-dimensional evaluation method/model that assesses not just technology, but also processes, organizational aspects, and interoperability.
G3	Be able to simulate the impact of new technology.
G4	Be able to integrate earlier versions and interact with higher and lower levels.
G4. B	Be able to assess the necessity of the integration process by evaluating specific criteria.
G6	Be able to handle both incremental and disruptive development.
G8	Be able to handle unspecified capability requirements − even if the unspecified.
G9	Be able to perform a comprehensive risk/opportunity assessment.
G10	Be able to evaluate the synergies between different domains (air, land, sea, cyber, space) within a SoS.
G11	Be able to provide continual feedback and improvements.

The *Design and develop artifact* phase began with Article 4 [26] in which the initial concept and goal models were introduced. The first operational instantiation of the framework was presented in Article 7 [9], introducing a *Navigation Guidance* (see Figure 3), which supports designers in selecting and sequencing Method Chunks according to the design context, stakeholder intentions, and architectural objectives, together with the first two Method Chunks (MC1 and MC2). The present article further extends the artifact by adding five additional Method Chunks, thereby completing the initial set of Method Chunks. At the same time, the Navigation Guidance is extended to support the selection and sequencing of all seven Method Chunks, and each Method Chunk is explicitly linked to corresponding NAF viewpoints and representations.

The *Demonstrate artifact* phase was initiated in Article 6 [28], where the underlying goal and concept models were demonstrated through stakeholder workshops and validated regarding their coherence, comprehensibility, and practical relevance. These activities provided empirical evidence supporting subsequent refinement of the framework. The current article further contributes to this phase by demonstrating the extended framework in a realistic military C2-system design context.

The *Evaluate Artifact* phase extends beyond validation by assessing the practical utility of the complete framework. Building on the structured evaluation approach proposed in Article 8 [29], the framework will be evaluated in which participants apply the Navigation Guidance and Method Chunks within a realistic C2-system context. Findings from this evaluation will provide the basis for the final refinement of the framework.

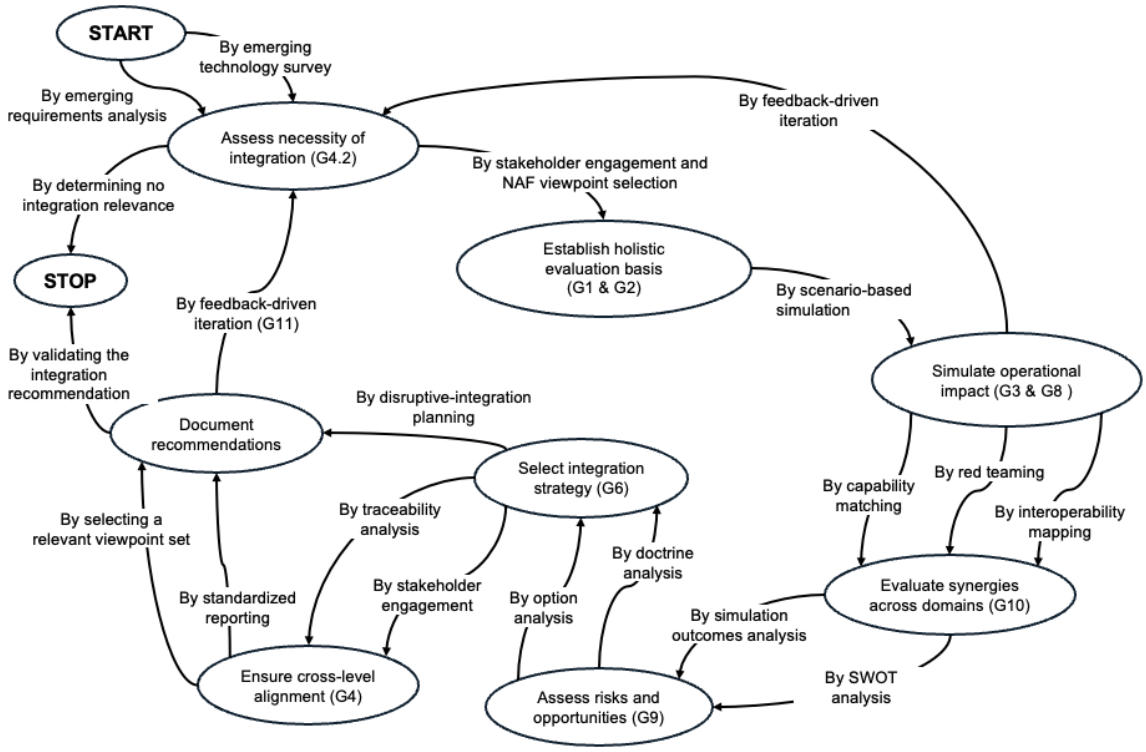

Figure 3. Navigation Guidance, first reported in [9]

3.2 Further Construction of the Framework

The study reported in this article builds on the established Navigation Guidance and the two initial MCs [9], extending the framework with five additional MCs. Together, they form a design process that fulfills stakeholders' high-level goals and the design requirements identified in earlier research [5], [27]. Each MC operationalizes one or more intentions from the navigation guidance, ensuring that the design process maintains a logical progression**.**

The construction of MCs follows SME principles [10], [30], [31], enabling systematic design of MCs that are both reusable and context-specific. Each MC is designed as an independent yet interconnected component, defined by its purpose, situation of use, and expected outputs. To visualize and model the internal logic, MAP formalism [32], [33] was applied, structuring each MC as a sequence of *intentions* and *strategies*. Combined, these principles enable iterative refinement, allowing designers to adapt and configure the design process to the context of a given scenario.

All MCs are explicit linkages to NAF [7] viewpoints and representations, ensuring architectural coherence and traceability throughout the design process. Here, the MAP-based internal processes define the workflow logic, while the NAF linkages to viewpoints and representations specify the modeling activities and the architectural artifacts to be produced. We argue that this combined application of SME and MAP will transform the design requirement elicitation process into a process-driven design framework, thereby enhancing NAF with structured methodological guidance.

4 MAP-Guided Method Chunks for Emerging Technology Integration

4.1 Framework Overview

This research proposes a modular method extension to NAF that adds a situational guidance layer, expressed as MAP-based [32], [33] navigation guidance and operationalized by reusable Method

Chunks (MCs), while retaining NAF viewpoints and representations as the product backbone. At this stage, all MCs are developed in equal detail for the sake of completeness, despite their differing levels of abstraction. Each MC is defined using a uniform structure, similar to that in [11], [13], [34], and is specified as a combined process and product unit, outlining (1) a sub-map describing navigable intentions and strategies (process part) and (2) a structured table specifying situation of use, inputs/outputs, and the linked NAF viewpoints and representations (product part). Following SME principles [10], [30], [31], the MCs constitute modular, context-aware method components that can be selected, adapted, or combined depending on designers' needs and the situational context of C2-system design. Each MC description begins with an introductory overview explaining the purpose, scope, and role of the MC in relation to the overarching high-level goals [5] and the Navigation Guidance [9]. For readability, we provide a short narrative for each MC that explains when it is selected, what decision question it corresponds to, and which NAF products the MC produces. This narrative is followed by a structured MC table, following the principles in [13], summarizing each MC's intentions, inputs, outputs, and contextual dependencies, as well as its linkage to previously validated framework requirements [28]. To ensure architectural traceability, each MC description explicitly links the MC outputs to specific NAF viewpoints and representations. These linkages demonstrate how designers can actively use NAF viewpoints as modeling tools, and how NAF representations correspond to the formal architectural artifacts produced or updated through each product part. In this way, the MCs expand NAF by connecting specific design intentions to the appropriate architectural representations, allowing designers to model and refine C2-architectures as iterative, method-driven processes. Each MC concludes with a detailed process description presented in MAP formalism [32], [33]. This formalism expresses the MC's intentions and strategies, outlining alternative pathways for achieving each design goal depending on the situational context.

The high-level goal "Be able to provide continual feedback and improvements" (G11) is implemented as an overarching process rather than as a separate MC. In accordance with the Navigation Guidance (Figure 3), the overall process is iterative, meaning that each intention can and should trigger a return path when new insights, inconsistencies, or validation results require reconsideration of previous steps. This structure should be understood as a generic intention, governing the adaptive nature of the overarching process rather than defining a distinct design activity. Each MC therefore embeds a feedback loop where results are evaluated, lessons are captured, and refinements are introduced in future design cycles. The feedback loop ensures that the process supports continuous improvement across all stages of C2-system design. Combined, the MCs form an iterative process in which outputs from earlier chunks serve as structured inputs to subsequent ones. Together, the seven MCs (MC1-MC7) outlined in this section operationalize the intentions defined in the Navigation Guidance. Each MC contributes a distinct and connected function. MC1 and MC2 establish technological relevance and assess operational impact. MC3 ensures architectural coherence across command levels. MC4 and MC5 extend the analysis to multi-domain interactions and to risk/opportunity considerations. MC6 synthesizes these insights into a feasible integration strategy. Finally, MC7 ensures documentation, traceability, and iterative refinement. The navigation guidance provides the procedural logic, connecting the MCs, while the NAF anchors each MC within specific viewpoints and representations. This alignment secures traceability across the C2-system design process.

Some structural refinements have been applied to MC1 (integration assessment) and MC2 (scenario-based simulation), as compared to what was reported in [9], to ensure consistency within this article. In particular, each MC now links its outputs and deliverables to the corresponding NAF viewpoints and representations. The refinements in this article are limited to the addition of a link between the MC deliverables and the corresponding NAF viewpoints and representations, as discussed in Sections 4.2 and 4.3.

4.2 Updated Method Chunk 1 (MC1) – Integration Assessment

Note that MC1 and MC2 were initially introduced in [9]. In this work, we apply structural refinements to them (e.g., linking outputs to NAF viewpoints) to ensure consistency of presentation of MCs in this article. Information about MC1 is amalgamated in Table 2 and Figure 4.

Table 2. MC1 – Integration Assessment with linked NAF [7] Viewpoints and Representations

Product Part	Linked NAF [7] Viewpoints and Representations
Prioritized list of technologies for integration	- C1, Capability Taxonomy - C3, Capability Dependencies *Representations include:* - Hierarchy diagrams (NAF, p.75) - Dependency diagrams (NAF, p.79)
Operational concept & capability linkage	- L2 - Logical Scenario - C3, Capability Dependencies *Representations include:* - Composite structure diagram (NAF, p.103) - Dependency diagrams (NAF, p.79)
System functionality assessment	- P4 - Resource Functions *Representations include:* - Collaboration diagram (NAF, p.121)

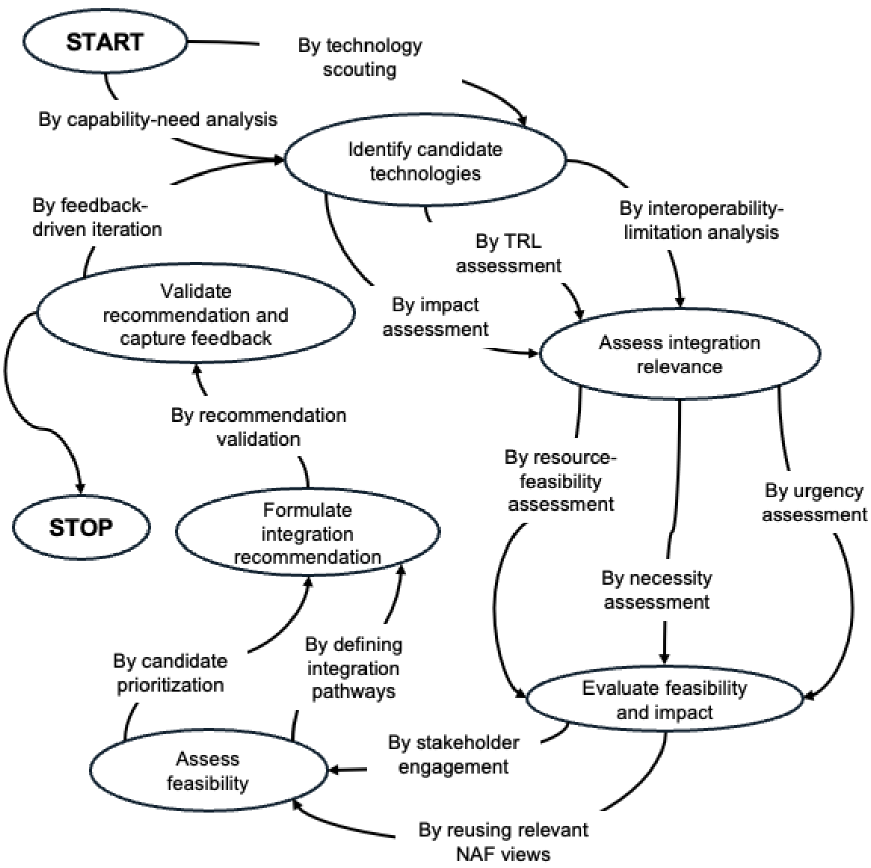

Figure 4. Method Chunk 1 (MC1) – Integration Assessment

4.3 Updated Method Chunk 2 (MC2) – Scenario-Based Simulation

Information about refined MC2, introduced in [9], is provided in Table 3 and Figure 5.

Table 3. MC2 – Scenario-Based Simulation with linked NAF [7] Viewpoints and Representations

Product Part	Linked NAF [7] Viewpoints and Representations
Simulation objective (defined objectives and scope)	- C1, Capability Taxonomy - L2, Logical Scenario - L2-L3, Logical Concept *Representations include:* - Hierarchy diagrams (NAF, p.75) - Composite structure diagram (NAF, p.103) - Graphic/Rich Picture (NAF, p.112)
Simulation logs	- L4 – Logical Activities - P4 – Resource Functions *Representations include;* - Activity and Collaboration Diagram (NAF, p.105)
Scenario-specific risk/opportunity report	- P4 – Resource Functions *Representations include;* - Collaboration diagram (NAF, p.121)
Impact analysis & report (transforms simulation results into architectural knowledge)	- A8 – Standards *Representations include;* - Diagram (NAF, p.139)

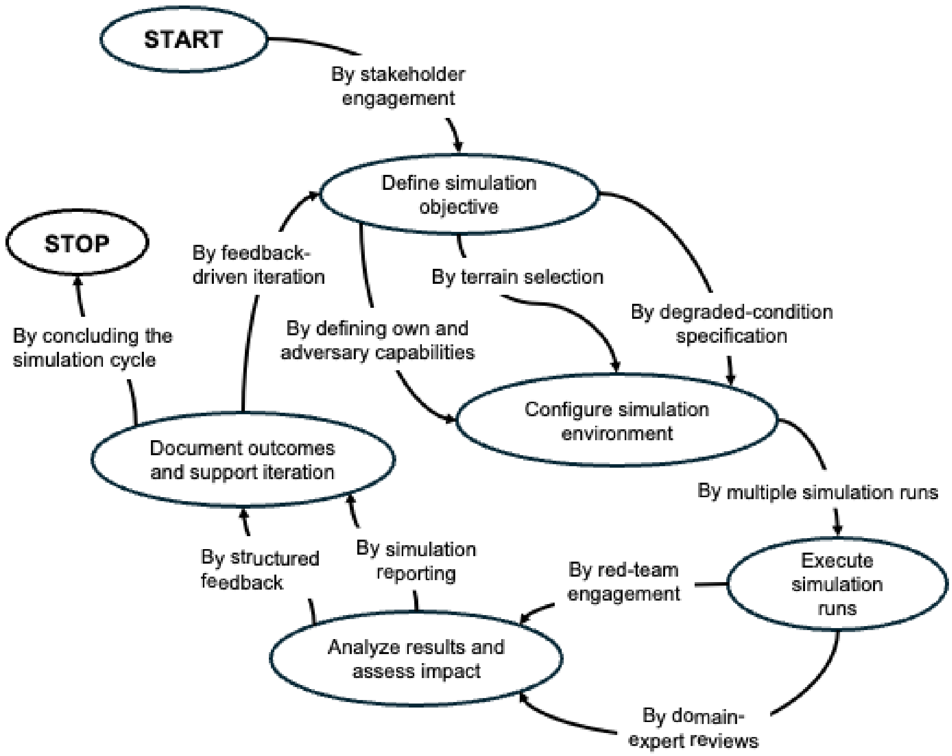

Figure 5. Method Chunk 2 (MC2) – Scenario-Based Simulation

4.4 Method Chunk 3 (MC3) – Cross-Level Integration Coherence

MC3 (Table 4, Table 5, and Figure 6) offers support for maintaining architectural coherence across different command levels, ensuring that technological, procedural, and organizational elements remain interoperable across tactical, operational, and strategic levels, and that the system supports evolving generations of C2-systems. MC3 is to be used when integration involves new technologies that must coexist with earlier versions or generations of technology. By managing both structural and procedural dependencies, MC3 ensures that integration efforts reinforce, rather than disrupt, the coherence of the C2-system.

MC3 can be reused independently in various design and development contexts and is particularly applicable when the outcomes from MC1 (Integration Assessment) and MC2 (Scenario-Based Simulation) reveal inconsistencies or interoperability gaps. MC3 therefore complements MC1 and MC2 by ensuring that candidate technologies and processes remain aligned across all command levels. MC3 applies the MAP formalism to structure its process, guiding designers through a sequence of intentions and strategies.

Each intention (Figure 6) corresponds to NAF viewpoints and representations summarized in Table 5, ensuring that every process step activates a specific (architectural) model component. The process starts with the intention to *identify cross-level integration dependencies*, which aims to explore structural, procedural, and informational interdependencies between tactical, operational, and strategic levels of capabilities as defined in the C1 viewpoint of NAF [7].

Strategies include stakeholder engagement to elicit dependencies between echelons (military units) and technical systems, a model-based analysis of existing architecture repositories, and a simulation review using outcomes from previous simulations to capture cross-level dependencies. The C3 and L4 viewpoints are applied to represent capability relationships and activity flows, forming the basis for capability dependency assessment.

The next intention, to *assess architectural and procedural coherence*, focuses on describing how technical systems and processes interact across all command levels. The aim is to ensure that data and decision flows are synchronized. Strategies include technical interface modeling to define system-to-system communication paths, procedural workflow mapping to align C2 processes, and technical interoperability testing to evaluate robustness under degraded conditions, e.g., bandwidth limitations. P3 and P4 viewpoints are used to capture how communication links and resource functions operate between echelons, ensuring tactical, operational, and technical system interfaces remain coherent.

Table 4. MC3 – Cross-Level Integration Coherence

Id and Name	MC3 – Cross-Level Integration Coherence
Type	Aggregated (composed of interrelated intentions and strategies that together ensure vertical alignment across command levels and temporal alignment across capability evolution and iterations of C2-systems).
Situation	Applied after scenario-based simulation (MC2) if results indicate potential inconsistencies or interoperability challenges across hierarchical command levels or between legacy and new system components.
Intention	Secure alignment across tactical and operational levels and analyze capability dependencies to uncover strengths, weaknesses, and relationships in the system. This MC represents the integration logic derived from high-level goal G4 – *Be able to integrate earlier versions and interact with higher and lower levels.*
Reuse Situation	Can be reused independently in other *system-of-systems* (SoS) contexts where dependency analysis or backward interoperability/compatibility is required. Particularly useful for iterative updates.
Product Part (including linked NAF [7] Representations)	- Alignment requirement mapping – documentation of cross-level integration needs. NAF Representations C1, C3, and L2. - Cross-level dependency mapping – documentation of procedural interdependencies. NAF Representations L4, L5, and P3. - Alignment validation report – stakeholder feedback and dependency results to confirm that architectural, procedural, and technical elements remain coherent and interoperable across all command levels. NAF [7] Representations P8, A8, and Ar.
Process Part	Described in the process description and in MAP, see Figure 6.
Linked Requirements (R)	R4.1 – R4.16, R1.7 – R1.10, R2.2, R2.4. These requirements are reported in [35]

Table 5. NAF Views and Representations activated by MC3

NAF Viewpoint	Viewpoints and Representations in NAF [7]	Purpose in MC3 with respect to the NAF [7]
Concept	C1, Capability Taxonomy (NAF, pp.74–75) C3, Capability Dependencies (NAF, pp.78–79)	Defines capability hierarchies in C1 with generalization relations. Defines capability dependencies in terms of logical composition with C3 across command levels to establish a foundation for integration.
Logical	L2 - Logical Scenario (NAF, pp.102–103) L4 - Logical Activities (NAF, p.105) L5 - Logical States (NAF, p.106)	Models operational scenarios with L2 by creating composite structure diagrams; activities and information exchange with L4 (activity diagram) to identify procedural and informational dependencies affected by candidate technology integration. Models the state of a node with L5 (state diagram).
Physical	P3 - Resource Connectivity (NAF, pp.119–120) P4 - Resource Functions (NAF, pp.121–122) P8 - Resource Constraints (NAF, p.126)	Defines system design, communication links between subsystems (P3) for a capability configuration, and ensures interoperability and consistency across system levels. Models the functionality of resources with P4, by means of an activity diagram, functional decomposition, or collaboration diagram. Adds resource constraints expressed in natural language or OCL as comments (P8).
Architecture Foundation	A8 – Standards (NAF, pp.139–140) Ar - Architecture Roadmap (NAF, p.141)	Establishes standards compliance (A8), configuration traceability, and long-term alignment across system generations by creating a timeline annotated with architecture releases and metadata (Ar).

Following this, the intention to *validate coherence* seeks to ensure that the candidate systems support tactical and operational goals without causing technical, structural, or procedural inconsistencies. Strategies include conducting alignment assessments across the Concept, Logical, and Physical viewpoints, model-based verification to confirm consistency and synchronization of information flows, and doctrinal alignment analysis to ensure that integration outcomes remain compatible with existing tactical and operational procedures. The A8 viewpoint presents the overall standards for the architecture.

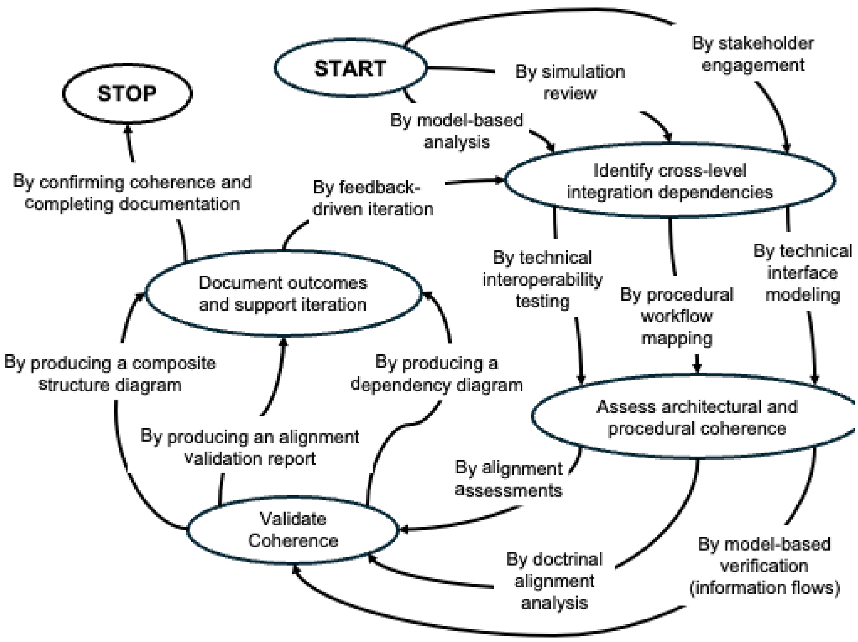

Figure 6. Method Chunk 3 (MC3) – Cross-Level Integration Coherence

Finally, the process moves to the intention to *document outcomes and support iteration*, which consolidates findings and secures traceability for future iterations. Strategies include developing a composite structure diagram and dependency diagram, as well as an alignment report that summarizes coherence assessments. This step also embeds a feedback mechanism that enables refinement and reuse in subsequent design iterations. The Ar and A8 viewpoints are applied to provide a structured baseline for continued improvement. The transition into *Stop* represents the completion condition of the MC, indicating that the process concludes once coherence has been confirmed and documentation has been completed. These steps are visualized in Figure 6.

4.5 Method Chunk 4 (MC4) – Multi-Domain Synergy Evaluation

MC4 (Table 6, Table 7, and Figure 7) offers methodological support for identifying and assessing synergies across multiple military domains: land, air, sea, cyber, and space. It helps designers evaluate how technological or procedural changes in one domain can improve or constrain capabilities in others. MC4 provides a structured approach to map inter-domain interactions and discover opportunities for cross-domain integration that improve tempo and interoperability.

MC4 is to be used when evaluating emerging technologies intended for joint operations (operations including multiple domains) or when optimizing existing capabilities across domains. It builds on MC1 (Integration Assessment) by focusing on cross-domain effects rather than within-domain performance. By modeling domain relationships and interdependencies, MC4 supports a SoS understanding of the C2-system environment.

The process (Figure 7) begins with the intention to *identify cross-domain relationships*, focusing on how capabilities and systems in one domain influence or depend on others. This intention can be realized through several strategies. The first is capability mapping, in which the C3 viewpoint and the C1 viewpoint are used to describe and visualize hierarchical and functional links between domains. The second strategy is to detect information flow, where designers analyze how information flows spread through the C2-systems, exposing potential cross-domain bottlenecks. The third strategy involves cross-domain dependency analysis, engaging stakeholders to validate the dependency structures and reveal coordination challenges. Together, these strategies form a baseline understanding of inter-domain connectivity and mutual dependencies.

Table 6. MC4 – Multi-Domain Synergy Evaluation

Id and Name	MC4 – Multi-Domain Synergy Evaluation
Type	Aggregated (combines multi-domain capability evaluation, inter-domain dependency mapping, and interoperability analysis into one evaluative method component).
Situation	Applied if designers need to evaluate candidate technologies (or procedures) that may affect more than one domain (e.g., land and air integration, space and electronic warfare convergence). Relevant in early concept and capability development phases to assess potential with multi-domain integration.
Intention	Evaluate inter-domain synergies and dependencies to identify opportunities for enhanced joint effectiveness and interoperability. This MC represents the evaluation logic derived from high-level goal G10 – *Be able to evaluate the synergies between different domains.*
Reuse Situation	Can be reused independently in other *SoS* evaluation situations.
Product Part (including linked NAF Representations)	- Cross-Domain synergy matrix – maps relationships and potential benefits between domains. NAF Representations C3 and L4. - Interoperability gap analysis – identifies communication, procedural, or architectural gaps limiting cross-domain coordination. NAF Representations P3 and P4. - Joint capability opportunity assessment – evaluates where multi-domain collaboration can enhance total capability. NAF Representations C1 and L2. - Synergy evaluation (summary) – consolidates findings, risks, and recommendations for designers. NAF Representations A8 and Ar.
Process Part	Described in the process description and in MAP, see Figure 7.
Linked Requirements (R)	R10.1 – R10.7, R3.4 – R3.6, R2.4, R4.13 – R4.16. These requirements are reported in [35].

Table 7. NAF Views and Representations activated by MC4

NAF Viewpoint	Viewpoints and Representations in NAF [7]	Purpose in MC4
Concept	C1, Capability Taxonomy (NAF, pp.74–75) C3, Capability Dependencies (NAF, pp. 78–79)	Used to visualize inter-domain capability relationships and to identify where capability overlaps or dependencies enable cross-domain synergy (C3). Structures capabilities within and across domains by generalization relationship in C1 and capability composition in C3.
Logical	L2 - Logical Scenario (NAF, pp.102–103) L4 - Logical Activities (NAF, p.105)	Models structure within and across domains, forming a baseline for mapping potential synergies. Models operational scenarios (L2) and joint operational processes, showing how activities in one domain contribute to another (L4).
Physical	P3 - Resource Connectivity (NAF, pp.119–120) P4 - Resource Functions (NAF, pp.121–122)	Models communication and coordination links between systems and subsystems across domains, e.g., sensor-to-shooter chains (P3). Describes how cross-domain system functions enhance joint operational effects in the form of a collaboration or activity diagram (P4).
Architecture Foundation	A8 – Standards (NAF, pp.139–140) Ar - Architecture Roadmap (NAF, p.141)	Outlines capability development standards (A8), strategy, and governance regarding improvements to cross-domain capabilities. A timeline clarified cross-domain capability development (Ar).

The process continues with the intention to *assess cross-domain synergies*, aiming to describe how multi-domain coordination creates operational effects. Strategies here include (1) joint process modeling, in which the L4 viewpoint is used to represent linked activities and functional sequences across domains, using NATO Allied Joint Publications (AJPs) as a baseline, and (2) scenario-driven modeling, applying the L2 viewpoint to capture the dynamics and challenges of domain interactivity. These strategies visualize the flow of information and resources across domains, highlighting potential areas for improvement.

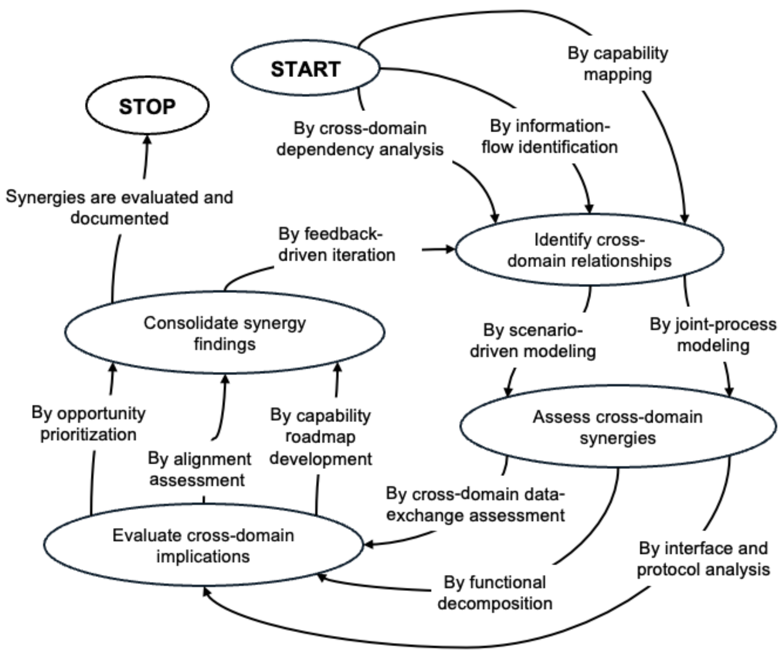

Figure 7. MC4, Multi-Domain Synergy Evaluation

Next follows the intention to *evaluate cross-domain implications*, ensuring that systems and organizations can interact effectively. This is achieved through three strategies. First, by interface and protocol analysis, supported by the P3 viewpoint, used to evaluate system-to-system communication paths and data exchange reliability. Second, by functional decomposition, using the P4 viewpoint, to identify overlapping or redundant functions across the domains to uncover integration conflicts. Third, by cross-domain data-exchange assessment that models how different systems can handle shared information, focusing on latency, accuracy, and protocol alignment. Together, these strategies assess whether existing and emerging systems can interoperate under realistic timing and mission conditions.

Finally, the intention to c*onsolidate synergy findings* integrates results into actual recommendations. The first strategy, opportunity prioritization, identifies the most promising synergies based on operational benefit and feasibility. The second, alignment assessment, applies the A8 viewpoint to ensure interoperability regarding procedural and technical requirements. The third, capability roadmap development, uses the Ar viewpoint, to define phased implementation pathways, coordinated across domains. The process concludes when all domain interactions have been evaluated and documented, ensuring enhanced understanding of cross-domain interoperability benefits and constraints.

4.6 Method Chunk 5 (MC5) – Risk and Opportunity Assessment

MC5 (Table 8, Table 9, and Figure 8) provides support in identifying and analyzing risks and opportunities associated with integrating new technologies, methods, or organizational changes in C2-systems. This allows designers to evaluate vulnerabilities, assess potential gains, and determine possible trade-offs between performance improvement and integration complexity. While MC4 focuses on identifying multi-domain synergies, MC5 extends this by assessing and measuring potential trade-offs between opportunity gains and associated risks.

MC5 is to be used when dealing with high uncertainty, e.g., rapid technological and tactical evolution. MC5 builds on the outcomes of MC2 and MC3, using these results as inputs to forecast how specific integration decisions could affect overall system capability. MC5 provides insights for designers to align integration strategies with operational risk and opportunity potential.

Table 8. MC5 – Risk and Opportunity Assessment

Id and Name	MC5 – Risk and Opportunity Assessment
Type	Aggregated (combines risk identification, system-state modeling, constraint analysis, and mitigation planning into one integrated assessment component)
Situation	Applied after cross-level and multi-domain assessments (MC3 and MC4) to further assess vulnerabilities, resilience, and potential opportunity gains as a result of planned integration.
Intention	Identify, analyze, and manage risks and opportunities affecting C2-system performance, including system resilience and adaptability. This MC represents the logic derived from high-level goal G9 – *Be able to perform a comprehensive risk/opportunity assessment.*
Reuse Situation	Can be reused independently in any context where C2-systems require structured risk and opportunity assessment under uncertainty.
Product Part (including linked NAF Representations)	- Risk and opportunity mapping – a catalogue of identified risks and opportunities, using C1 and C3. - System uncertainty model – a representation of system behavior under degraded or adversarial conditions, applying L2 and L5. - Constraint and trade-off analysis – assessment of functional limitations and resource dependencies using P4 and P8. - Risk mitigation and opportunity exploitation plan – a roadmap supported by A8 and sequenced over time with Ar.
Process Part	Described in the process description and in MAP, see Figure 8.
Linked Requirements (R)	R9.1 – R9.11, R4.1 – R4.16. These requirements are reported in [35].

Table 9. NAF Views and Representations activated by MC5

NAF Viewpoint	Viewpoints and Representations in NAF [7]	Purpose in MC5
Concept	C1, Capability Taxonomy, (NAF, pp.74–75) C3, Capability Dependencies (NAF, pp.78–79)	C1 and C3 define hierarchical capability structure and dependencies. They are to be used to analyze risks and opportunities across organizational levels.
Logical	L2 - Logical Scenario (NAF, pp.102–103) L5 - Logical States (NAF, p.106)	Models tactical/operational scenarios (L2) and state transitions (L5) to analyze how the system behaves under uncertainty, identifying conditions activating risks or enabling opportunities.
Physical	P4 - Resource Functions (NAF, pp.121–122) P8 - Resource Constraints (NAF, p.126)	Describes functional performance by means of a collaboration or activity diagram (P4) and resource constraints by means of natural language text or the OCL (P8) that affect the system. Supports assessments of trade-offs (mitigation costs vs. potential opportunity gains).
Architecture Foundation	A8 – Standards (NAF, pp.139–140) Ar - Architecture Roadmap (NAF, p.141)	Describes mitigation and exploitation strategies with relevant standards and aligns them with long-term evolution and governance cycles (A8). Creates a timeline clarified cross-domain capability development (Ar)

The process (Figure 8) begins with the intention to *identify risks and opportunities*, focusing on vulnerabilities and strengths within the C2-system. This intention is realized through three main strategies. First, by dependency-driven identification, using the C3 viewpoint to trace how risks and opportunities multiply across capability and system boundaries. Second, stakeholder workshops collect insights from commanders, operators, and designers, identifying risks and opportunities rooted in tactical/operational practices or organizational structure. Third, capability mapping, by applying the C1 viewpoint, classifies risks and opportunities across hierarchical capability levels. These strategies create a foundation to identify and categorize risks and opportunities.

The next step is to a*ssess the significance of risks and opportunities*, which examines system performance in degraded or adversarial conditions. Strategies include: (1) scenario-consequence analysis [36], using the L2 viewpoint to visualize the dynamic tactical/operational context, (2) state-transition modeling [37], employing the L5 viewpoint to describe how systems move between normal and degraded states, and (3) variation testing, which modifies critical parameters to expose resilience and adaptability shortcomings. These strategies help reveal the C2-system's stability and adaptability under uncertain conditions.

The third intention is to *analyze functional and resource constraints*, determining how risks and opportunities impact C2-system performance. The first strategy, resource impact analysis, uses the P4 viewpoint to assess how critical system components are influenced by risk factors. The second, constraint modeling, applies the P8 viewpoint to define limitations on resource availability and performance. The third, trade-off evaluation, compares mitigation costs with potential opportunity gains, prioritization, and balanced decision-making. Collectively, these strategies help identify the C2-system's functional and resource constraints and support designers in making informed design decisions.

Finally, the intention to *document risk-opportunity conclusions* transforms these insights into usable plans. Strategies include: (1) compliance verification, applying the A8 viewpoint to ensure that all recommendations align with doctrinal and interoperability requirements, and (2) roadmap development & synchronization, using the Ar viewpoint to sequence mitigation and exploitation activities over time. This step ensures that risks and opportunities are not only identified, but also integrated into a coherent, time-bound development plan.

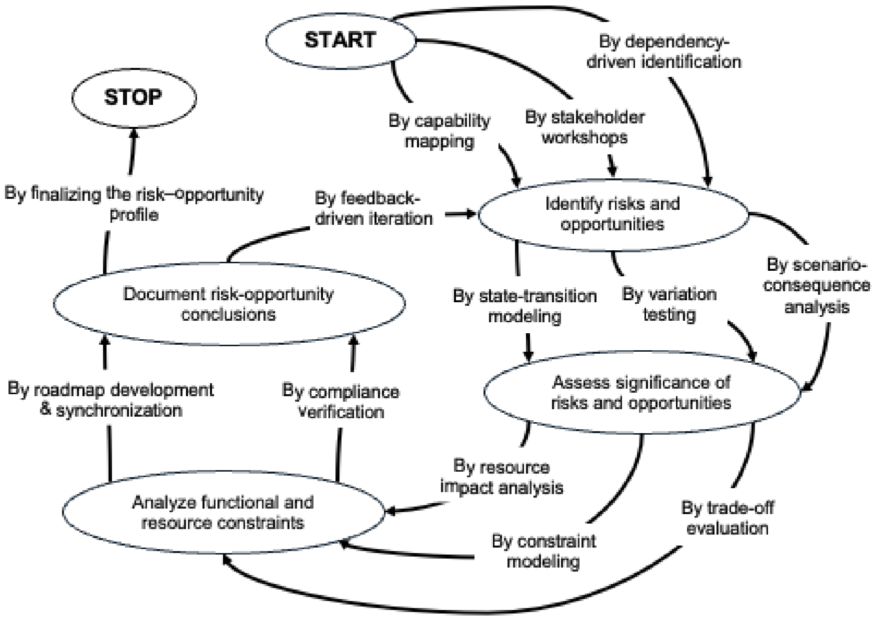

Figure 8. MC5, Risk and Opportunity Assessment

4.7 Method Chunk 6 (MC6) – Integration Strategy

MC6 (Table 10, Table 11, and Figure 9) provides guidance for selecting and formulating integration strategies that align with the technological, organizational, and tactical/doctrinal dimensions of a C2-system. This MC operationalizes the high-level goal to *be able to handle both incremental and disruptive development* (G6), and supports designers in deciding whether integration should follow an incremental, adaptive path or a disruptive, transformational approach.

MC6 ensures that integration decisions are grounded in stakeholders' needs and traceable across architecture frameworks such as NAF. MC6 therefore converts outputs from the previous MCs into informed decisions regarding implementation pathways. It can be applied independently, or preferably, following MC5 when designers need to transform insights into architectural strategies, i.e., select an integration strategy.

Table 10. MC6 – Integration Strategy

Id and Name	**MC6 – Integration Strategy**
Type	Aggregated (consolidate results from MC1–MC5 into a logical integration approach).
Situation	Applied after MC1–MC5 to determine the most feasible integration strategy.
Intention	To identify, assess, and select the most suitable integration strategy, balancing tactical/operational benefits and implementation risk. MC6 supports the high-level goal to *be able to handle both incremental and disruptive development* (G6).
Reuse Situation	MC6 can be reused in similar design decision-making contexts where emerging technologies (or capabilities) require integration assessment before implementation.
Product Part (including linked NAF Representations)	- Integration strategy mapping - analysis of strategies, by using C1 and C3. - Option assessment – a decision-support model comparing incremental and disruptive options using L4 and P4. - Alignment report – assessment of strategy compliance with tactical/operational procedures and technical standards, represented through A8. - Implementation roadmap – a sequenced integration roadmap, using Ar.
Process Part	Described in the process description and in MAP, see Figure 9.
Linked Requirements	R6.1–R6.16, R4.1–R4.16. These requirements are reported in [35].

Table 11. NAF Views and Representations activated by MC6

NAF Viewpoint	Viewpoints and Representations in NAF [7]	Purpose in MC6
Concept	C1, Capability Taxonomy (NAF, p.74–75) C3, Capability Dependencies (NAF, p.78–79)	Defines hierarchical capability structure (C1) and dependencies (C3) to analyze which integration strategy will have the greatest impact.
Logical	L4 - Logical Activities (NAF, p.105)	Models activities and workflows (L4, activity diagrams or collaboration diagrams) affected by alternative integration strategies.
Physical	P4 - Resource Functions (NAF, p.121–122)	Describes C2-system functions and interfaces impacted by the chosen strategy, clarifying technical implications. Model used is a collaboration or activity diagram (P4).
Architecture Foundation	A2 – Architecture Products (NAF, pp.133) A8 – Standards (NAF, pp.139–140) Ar - Architecture Roadmap (NAF, p.141)	Assesses the integration strategy against compliance standards (A8), plans their phased implementation over time, including a timeline for integration (Ar), and describes the products (models) used to report the Architecture (A2).

The process (Figure 9) starts with the intention to *identify feasible integration paths*, which aims to establish an overview of potential strategies for integrating new technologies. Three complementary strategies guide this step. The first is option generation, where designers define potential strategies, typically distinguishing between incremental and disruptive approaches, based on insights from previous MCs. The second strategy, dependency scanning, uses the C3 viewpoint to uncover how each integration alternative impacts capabilities. The third strategy, contextual framing, engages the C1 viewpoint to position the integration approaches within the broader capability hierarchy, clarifying which strategy has the most obvious effect on the C2-system. Together, these strategies ensure that the integration paths are identified and ready for detailed assessment.

The next step is to *assess the tactical and operational impact* of each integration strategy. This step focuses on how each integration approach influences workflows, interoperability, and overall performance. The first strategy, activity-impact mapping, uses the L4 viewpoint to model how or if command processes and decision cycles must adjust or be modified under different strategies. The second, achievability analysis, pairs tactical/operational with technical factors to filter out impractical or low-impact alternatives. These strategies combined support designers to assess the consequences of the integration strategy and to identify the most beneficial option.

The third intention is to *align preferred strategy with doctrinal and technical requirements*, ensuring compliance with standards, e.g., NATO AJPs, while preserving agility. The first strategy, tactical/operational/doctrinal alignment analysis, assesses how well each integration strategy coheres with AJPs. The second, interoperability assessment, uses the A8 viewpoint to assess procedural and technical compatibility with NATO standards. The third, risk-informed selection, integrates outputs from MC5 to balance opportunities and risks in the integration strategy selection process. These strategies ensure that the chosen integration strategy is technically, tactically, operationally, and doctrinally coherent and sound.

Finally, the process concludes with the intention to *document the implementation roadmap*, transforming outcomes into actionable plans. The first strategy, implementation phasing, develops a sequenced set of milestones using the Ar viewpoint to visualize the integration steps over time. The second, governance mapping, links decision gateways to responsible stakeholders, clarifying relevant guidance and accountability in the process. The third, feedback integration, establishes mechanisms for monitoring and iterative refinement to ensure adaptability throughout the C2-system lifecycle.

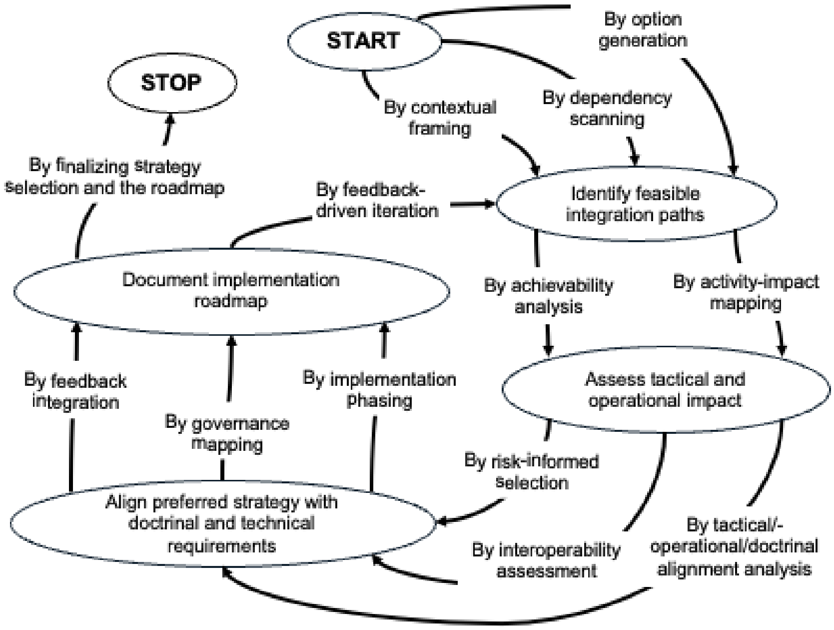

Figure 9. MC6, Integration Strategy

4.8 Method Chunk 7 (MC7) – Documentation and Reporting

MC7 (Table 12, Table 13, and Figure 10) provides guidance in documenting, structuring, and communicating the results produced through MC1 to MC6. Unlike the previous chunks, MC7 represents a governance component rather than a distinct process. Its purpose is to formalize results, ensure traceability, and prepare design architectural deliverables for decision-making or evaluation.

MC7 partially operationalizes high-level goal G11 - *Be able to provide continual feedback and improvements*, by ensuring that results are documented in a NAF-standardized format. It supports designers in producing transparent, traceable outputs aligned with NAF, enabling communication between technical, operational, and decision-making stakeholders.

Table 12. MC7 – Documentation and Reporting

Id and Name	MC7 – Documentation and Reporting
Type	Atomic (focuses on structuring and formalizing outputs generated by other MCs).
Situation	Applied after completion of MC1-MC6 (or a combination of these) to consolidate and document results before decision-making.
Intention	To standardize documentation, ensure traceability, and communicate architectural results in alignment with NAF rules. MC7 represents the governance logic derived from high-level goal G11 – *Be able to provide continual feedback and improvements*.
Reuse Situation	Can be reused as a standalone documentation and traceability procedure across other C2-system development processes.
Product Part (including linked NAF Representations)	- Documentation package – consolidated results and architectural artifacts using A8. - Traceability matrix – cross-links MC outputs and NAF representations, using Ar to visualize relations. - Feedback and method report – documents applied methodologies and improvement recommendations aligned with A4.
Process Part	Described in the process description and in MAP, see Figure 10.
Linked Requirements (R)	R11.1–R11.6. These requirements are reported in [35].

Table 13. NAF Views and Representations activated by MC7

NAF Viewpoint	Viewpoints and Representations in NAF [7]	Purpose in MC7
Architecture Foundation	A2 – Architecture Products (NAF, p.133) A4 – Methodology used (NAF, p.135) A8 – Standards (NAF, pp.139–140) Ar – Architecture Roadmap (NAF, p.141)	Defines the documentation standards used for reporting and traceability, ensuring consistency across all MC outputs (in compliance with NAF), including documentation and visualization of architectural deliverables over time. This is documented in a NAF grid representation (A2), tabular in text documents (A4), and standards compliance in textual or diagram form (A8). A timeline improvement recommendation is also specified (Ar).

The process (Figure 10) begins with the intention to *consolidate design results*, confirming that results from MC1 to MC6 are presented coherently. Two strategies guide this step. The first, standardized reporting templates, gathers and categorizes outputs from each MC, including diagrams, matrices, and textual descriptions, ensuring consistency with NAF representation conventions. The second strategy, traceability-matrix construction, applies the A8 viewpoint to verify and visualize that each representation has links to its originating MC and related viewpoints. Together, these strategies ensure that the architectural process is documented and traceable.

The next intention is to *ensure traceability across MC outputs*, formalizing outcomes for decision-making and review. The first strategy, template-based reporting, applies A2, A4, and A8 to ensure methodological and formatting consistency, documenting not only the results as such, but also the processes and procedures used to generate these reports. The second strategy, NAF-grid consolidation, uses the Ar viewpoint to position the results on a timeline, showing how architectural elements evolve over time. These strategies ensure that design results are communicated in a standardized way that is suitable for both technical and operational stakeholders.

Finally, the process concludes with the intention to *communicate architectural results*, transforming insights from the design process into useful lessons learned. The first strategy, documentation-package assembly, engages stakeholders, i.e., designers and decision-makers, to summarize insights from the employment of all MCs. The second, report preparation, ensures that results are documented and implemented. Together, these activities ensure that architectural outputs remain transparent, standardized, and ready for reuse.

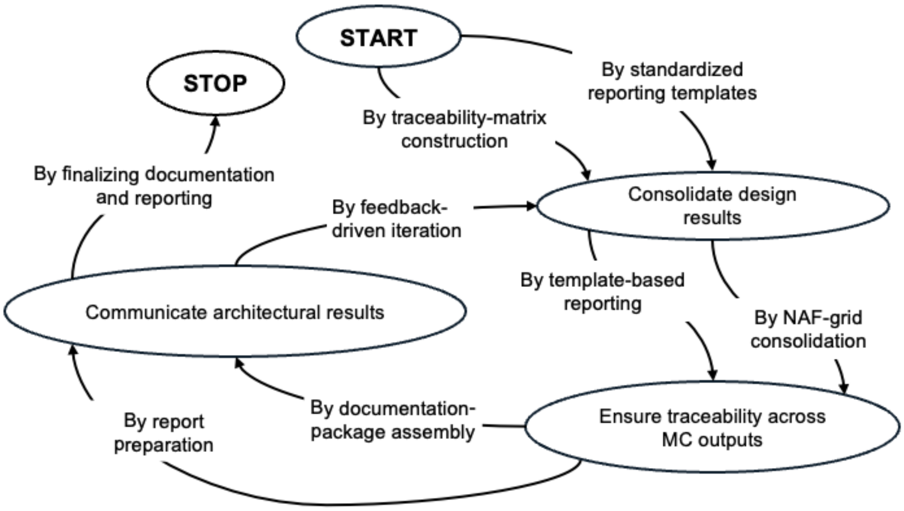

Figure 10. MC7, Documentation and Reporting

4.9 Envisioned Framework

The MCs outlined in this section, combined with the navigation guidance (see Figure 3) and the concepts and goal models validated in [28], form an initial version of the envisioned framework in the overarching project.

The framework ensures vertical alignment across command levels and horizontal consistency across the military domain, through its integration with the NAF. By activating relevant viewpoints and representations, the MCs ensure traceability and architectural coherence. As a result, we argue that the proposed set of MCs not only strengthens the consistency and traceability of C2-system design, but also provides a structured way of handling the evolution of complex, socio-technical systems, such as military C2-systems.

5 Demonstration Case – Intelligence, Surveillance, and Reconnaissance (ISR) Integration in a C2-System Context

5.1 The Demonstration Setting

To validate the internal consistency and practical applicability, this section demonstrates how the proposed Navigation Guidance (Figure 3) and MCs can guide architectural design activities in a realistic context. In DSR [38], a demonstration serves to show feasibility, internal coherence, and applicability of the artifact in a relevant context. In this case, the artifact is the initial version of the envisioned framework, outlined in Section 4.

The demonstration case is designed as a structured process walkthrough based on a scaled-down yet representative Intelligence, Surveillance, and Reconnaissance (ISR) integration scenario. The case was selected because it captures the specific socio-technical complexity that the envisioned framework is intended to address, for instance, cross-domain capability and information dependencies, vertical alignment between tactical and operational levels, and the coexistence of legacy and emerging technologies. This enables a transparent demonstration of the proposed framework's logic, showing how the navigation guidance and associated MCs can be applied in a controlled setting to guide design reasoning and support integration decisions.

From an SME [10] perspective, the demonstration serves as a proof-of-concept instantiation of the proposed initial framework. It exemplifies how the MCs and Navigation Guidance can be tailored to a specific integration challenge. The objective is therefore to confirm the coherence, procedural completeness, and practical usability of the framework. This distinction aligns with the DSR principles our demonstration focuses on showing the artifact's feasibility and internal coherence, whereas a later evaluation will be needed to assess its real-world effectiveness in operational C2-system development contexts.

The demonstration focuses on a current and practically relevant case: the integration of a new ISR platform into an existing C2-system. The purpose is twofold. First, it illustrates how the proposed method can support system architects by guiding them through the sequencing and selection of MCs in response to specific design concerns. Second, it exemplifies how each MC contributes to architectural artifacts aligned with NAF.

The architecture produced in this demonstration is not intended as a full system design. Rather, it serves as an example that captures key design decisions and helps designers address the following focused design questions:

Q1: How will the ISR platform affect current mission scenarios, operational flows, and responsibilities?
Q2: What tactical/operational capability motivates the integration of the ISR platform?
Q3: What activities and resources must be synchronized to enable the integration?
Q4: How are architectural choices traceable to stakeholder concerns and outputs?

To answer these questions, the demonstration is scoped and structured to produce five NAF viewpoints:

- *Producing L2-L3 – Logical concept*, to describe mission-level scenario logic and key interactions addressing question 1.
- *Producing L4 – Logical activities*, to illustrate domain-spanning activities and where synchronization is required, addressing question 3.
- *Producing C1 – Capability Taxonomy*, to position the new ISR capability in a capability hierarchy, addressing question 2.
- *Producing C3 – Capability dependencies*, to reveal cross-level capability interactions and potential gaps, addressing question 2.
- *Producing A2 – Architecture products*, to document architectural traceability and integration reasoning, addressing question 4.

This demonstration applies four of the seven MCs (MC2, MC1, MC3, and MC7) in sequence. Other MCs (i.e., MC4, MC5, and MC6) are excluded to keep the demonstration focused on the core design questions. The Navigation Guidance (Figure 3) is used to choose an appropriate entry point and then to guide the choice of successive steps. Each step corresponds to one or more intentions defined in the navigation guidance, with the MCs providing practical strategies and model representations for addressing them.

The demonstration case concerns the integration of a next-generation ISR capability into a joint targeting sequence. This refers to the process of selecting and prioritizing military targets and matching them with appropriate effects, based on tactical or operational intent and available capabilities. The new ISR capability introduces an AI-supported target recognition and sensor fusion module that processes data across multiple sensors, platforms, and domains (e.g., air, land, sea, space, and cyber). While this functionality promises improved target acquisition, speed, and accuracy, it also introduces architectural challenges.

The integration affects the entire *kill chain* [39], [40], [41], i.e., a methodical process used to plan and execute attacks on enemy targets, from sensor tasking and multi-source data fusion to command-level decision-making and effector coordination. This requires maintaining interoperability across technical systems with differing levels of crypto protection, information assurance, and resilience.

The operational environment includes degraded electromagnetic conditions, contested communication channels, and the presence of legacy platforms, all of which introduce friction into orchestration. However, the challenge is not merely technical. It also demands the adaptation of workflows, responsibilities, and coordination mechanisms across organizational and domain boundaries.

To make the application of the proposed method explicit, the demonstration case is next presented as an instantiated route through the navigation guidance.

5.2 Instantiation of the Navigation Guidance

Figure 11 presents the route instantiated in the demonstration case. Rather than traversing the full set of alternatives available in the navigation guidance, the case followed a focused path through the intentions most relevant to the MC enacted in the demonstration (MC1, MC2, MC3, and MC7). More specifically, the route began with assessing the necessity of integration, proceeded to establishing a holistic evaluation basis, continued through simulation of operational impact, and then advanced to ensuring cross-level alignment before concluding with documented recommendations and validation of the integration recommendation.

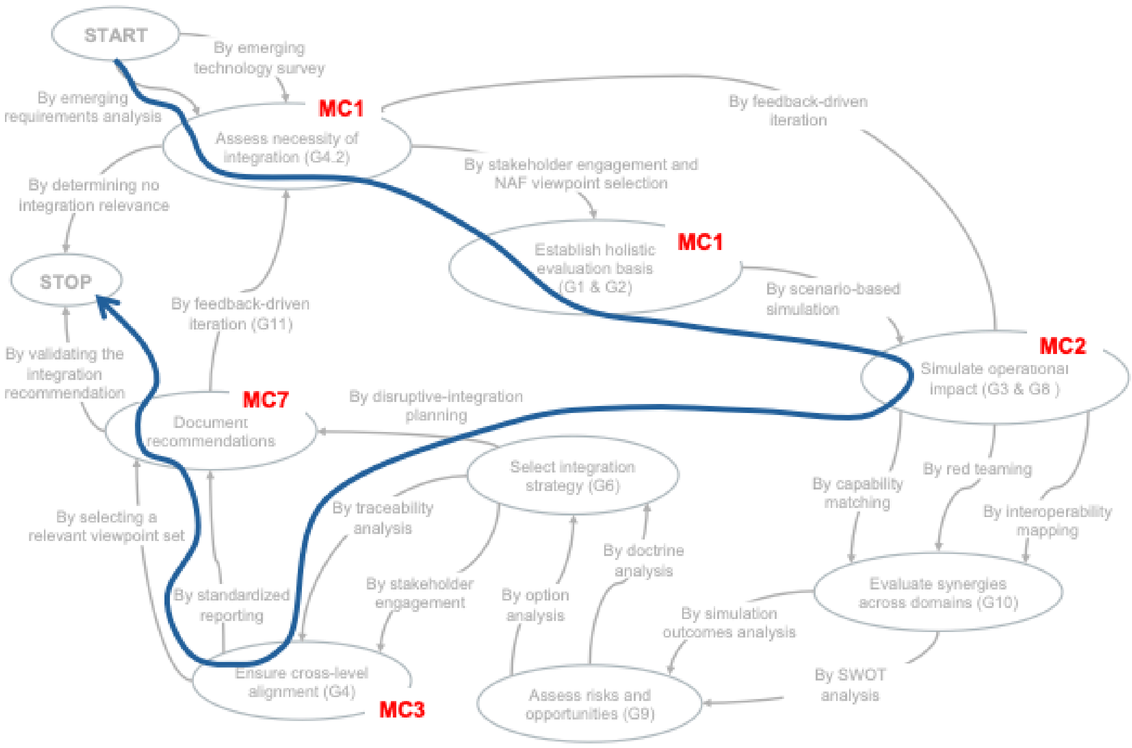

Figure 11. Instantiation of the Navigation Guidance

MC1 supported the early framing of the integration problem and the establishment of an initial evaluation basis grounded in stakeholder engagement and relevant NAF viewpoints. MC2 was then used to explore the operational impact through scenario-based simulation. Thereafter, MC3 supported the assessment of cross-level coherence, including the relation between architectural elements, information flows, and procedural arrangements. Finally, MC7 supported the consolidation, traceability, and communication of the resulting recommendations.

In this way, the instantiated route clarifies how the navigation guidance informed the selection and sequencing of MCs in this case and how the methodological proposal was enacted in practice rather than only described conceptually.

5.3 Producing L2-L3 – Logical Concept

At this stage, the Navigation Guidance (Figure 3) is engaged to determine an appropriate entry point for the architectural design activities. Given the nature of the integration challenge, introducing a new ISR capability into an existing C2-system, the Navigation Guidance indicates that an early understanding of operational impact is required before conducting a structured capability assessment.

Although MC1 (Integration Assessment) is a foundational part of the overall method, the demonstration does not begin with MC1. Instead, the process starts with MC2 (Scenario-Based Simulation). This choice reflects the need to first establish a shared, operationally grounded understanding of how the new ISR capability affects mission execution. Without such scenario-level insight, subsequent capability analysis would risk being detached from the operational realities it is intended to support.

Within MC2, the design activity is used to turn the ISR-integration idea into a grounded mission narrative before moving to capability and dependency modeling. As illustrated in Figure 5, MC2 structures this process by (1) defining the simulation objective, (2) configuring a representative simulation environment, including relevant terrain/context and explicitly modeled degraded conditions (e.g., electronic warfare (EW) influence, bandwidth/latency constraints), and (3) instantiating the key kill-chain elements, such as the ISR platform, external ISR feeds, C2-element, effectors, and EW threats.

The next step in MC2 is to perform multiple scenario runs where specific variables, e.g., sensor coverage, availability of external data feeds, and levels of data-link degradation, are varied. The results are analyzed for impact on mission outcomes and consolidated into a coherent set of operational insights rather than treated as isolated simulation outputs.

These insights are captured and communicated via a *Concept Diagram*, which in this demonstration serves as the architectural output of MC2. The *Concept Diagram* merges the scenario-based reasoning performed in MC2 into an executive-level architectural representation, providing an operational narrative that can be used to engage stakeholders and guide subsequent design activities. The *Concept Diagram* addresses the first design question: "*How will the ISR platform affect current mission scenarios, operational flows, and responsibilities?*"

As outlined in Figure 12, the *Concept Diagram* describes the kill chain sequence as a mission-oriented scenario, illustrating how sensor assets, C2-elements, and effectors interact following the integration of the new ISR capability. It highlights key elements within the kill chain, such as the ISR platform, external sensor contributions, Processing, Exploitation, and Dissemination (PED) functionality, and C2-nodes, and exemplifies how information flows between them across tactical and operational levels. PED [42] is a critical intelligence process comprising: *Processing*, i.e., transforming collected data into usable information; *Exploitation*, i.e., interpreting and analyzing information to derive meaning; and Dissemination, i.e., distributing outputs to *stakeholders* at the right time and in the right form. Relevant interactions with elements outside the immediate scope, e.g., external sensor data sources or coalition effectors, are included to reflect the broad-ranging operational context.

By focusing on purpose, scope, and key interactions rather than detailed system design, the *Concept Diagram* supports high-level communication of the architecture and enables early stakeholder engagement. It places the ISR integration in its operational context, allowing designers to reason about mission-level consequences and facilitating dialogue between system architects and commanders.

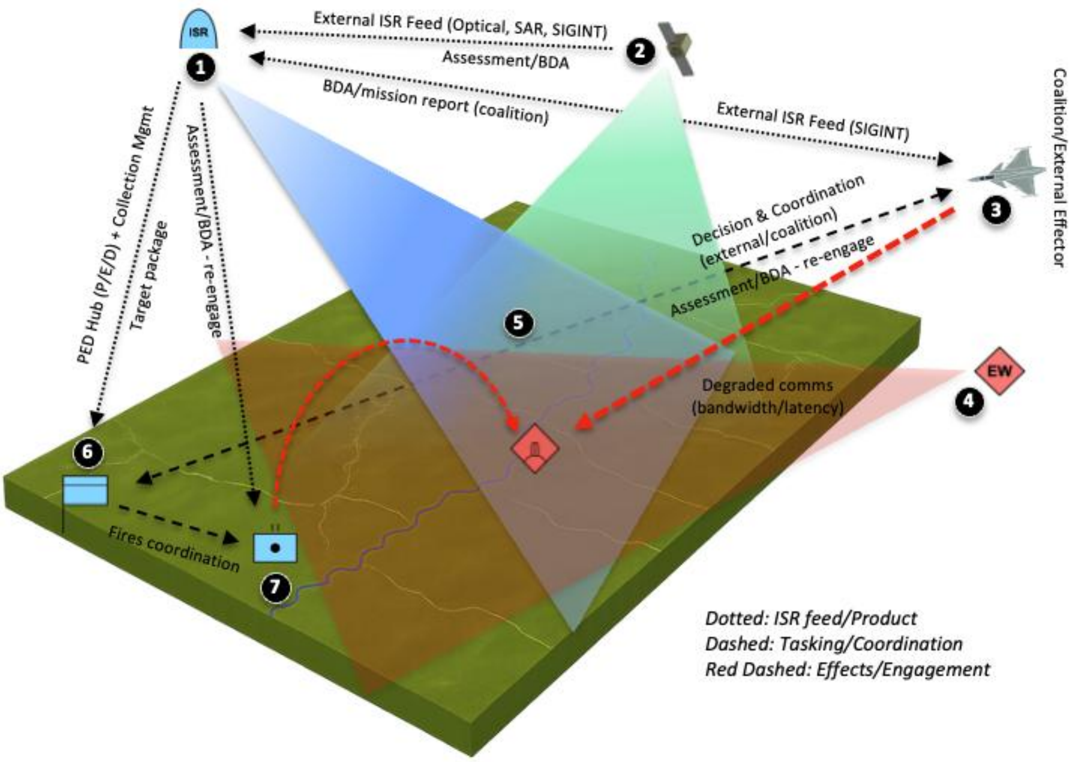

Figure 12. The resulting L2-L3 (Logical Concept) – Concept Diagram

In contrast to traditional ISR architectures, where sensors primarily forward sensor/target data to C2-elements, the new ISR platform should be understood as a PED-hub whose primary function is to refine multi-source sensor data into decision-ready products. This product-centric role is highlighted by the dotted information flows (from 2 and 3 to 1) and the target-package, e.g., decision-ready ISR products supporting planning and engagement, towards the C2-element (1 to 6). Within the targeting cycle, the C2-element provides intent, constraints, and prioritization, while coordinating fires with the artillery battalion (7) and/or coalition/external effectors (3). The core output from the C2-element is the fires coordination and engagement tasking, building on the target package produced by the ISR platform, delivered to effectors (3 and 7), i.e., engagement direction based on target context and constraints, tailored for decentralized planning and engagement.

The external Synthetic Aperture Radar (SAR) [42] sensor provides all-weather detection and supports the assessment phase through Battle Damage Assessment [43] , e.g., the evaluation of the damage caused by the application of military force, lethal or non-lethal. In parallel, additional external ISR feeds from other domains or coalition partners, i.e., Signals Intelligence [44], provide capabilities that can strengthen identification precision and improve confidence in the target package. Rather than exposing the C2-element to multiple disparate feeds, the new ISR capability exploits different sources internally and disseminates a consistent target package that supports a decentralized C2 approach. Importantly, external capabilities (2 and 3) are also reflected in the closure of the targeting cycle, by supporting BDA after engagement, enabling rapid re-tasking and re-engagement decisions. Overall, the *Concept Diagram* underlines that the next-generation ISR asset is not simply a sensor, but an operational capability that enhances the full kill chain, from target development to assessment after engagement.

5.4 Producing L4 – Logical Activities

Building on the operational narrative captured in the *Concept Diagram*, the demonstration continues, still within MC2, by translating the scenario-level kill-chain reasoning into an activity flow. The L2-L3 viewpoints and the *Concept Diagram* will together explain: *"who interacts with whom* and *what is exchanged?"*, the L4 viewpoint will explain: *"what must be done*, *in what logical order*, and *where is synchronization required?",* should the new ISR platform be introduced.

In this demonstration, the activity flow is deliberately aligned with the established NATO doctrine for Joint Targeting [45], i.e., a workflow that links planning/tasking, target development, engagement, and assessment into a coherent cycle. The L4 view therefore uses the *Concept Diagram* to map the specific roles and information exchanges onto a targeting workflow, making explicit how the new ISR capability affects the internal logic of that workflow. Concretely, the L4 *Operational Activity Diagram*, Figure 13*,* expresses how planning and tasking initiate collection, how collection and PED (processing and exploitation) on the ISR platform convert multi-sensor inputs into decision-ready ISR products (e.g., target packages), how these products enable decision support, and how assessment/BDA closes the loop by informing re-tasking and potential re-engagement.

This answers design question 3 by explaining what activities must be carried out and coordinated to enable ISR integration. By making these activity steps explicit, the *Operational Activity Diagram* complements the *Concept Diagram* as the second MC2 product and provides a basis for later coherence checks and traceability in subsequent method components.

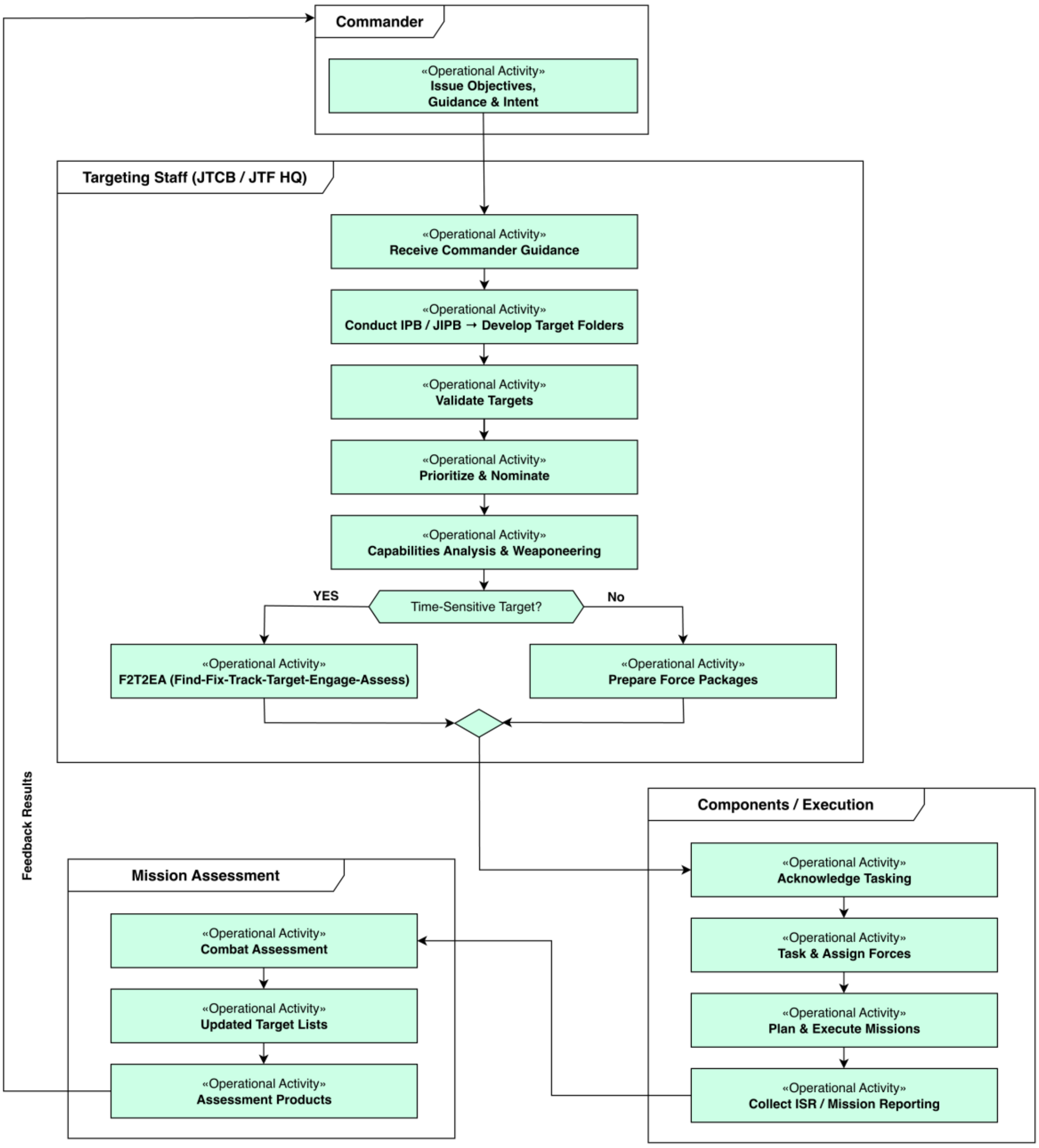

Figure 13. The resulting L4 (Logical Activities) – Operational Activity Diagram

5.5 Producing C1 – Capability Taxonomy

After constructing the *Operational Activity Diagram* and the *Concept Diagram*, the Navigation Guidance (Figure 3) is engaged again to determine the next modeling step required to address design question 2: "*What tactical/operational capability motivates the integration of the ISR platform?"* The Navigation Guidance points to MC1 (Integration Assessment), which supports the transition from scenario insight (L2-L3) to capability logic. In this demonstration, MC1 is not executed as a full-scale assessment; instead, it is applied to assess the ISR platform's integration potential and to generate the capability framing (Capability Taxonomy represented in Figure 14) required for subsequent steps.

MC1 is instantiated by following the logic outlined in Figure 4 and begins by selecting the capability-driven entry strategy (rather than the opportunity-driven), since the issue is defined as integrating an ISR capability into an existing C2-system.

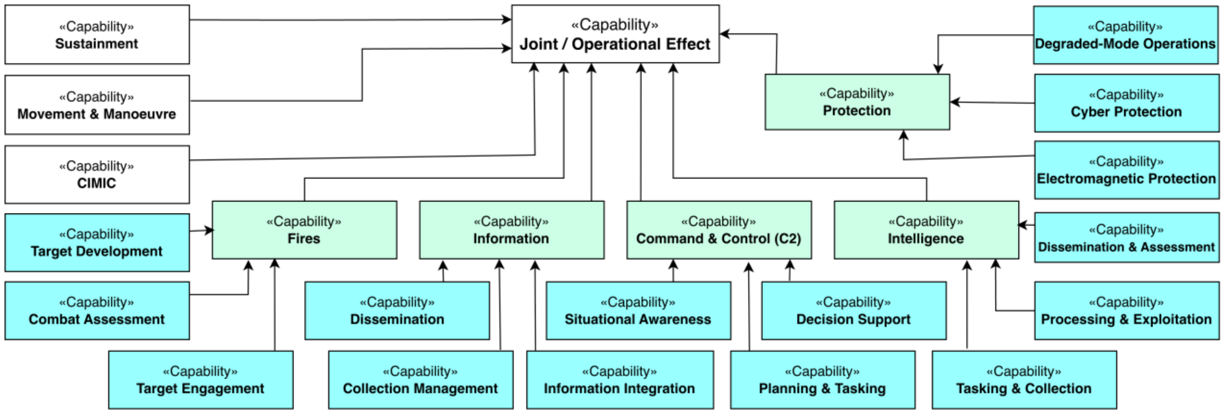

Figure 14. The resulting C1 (Capability Taxonomy**)** – Capability Hierarchical Diagram

The first objective, *Identify integration candidates*, is operationalized by defining the next-generation ISR platform, grounded in the L2-L3 view, which highlights the core contribution, i.e., to process multi-sensor data into ISR products, such as target packages, and to support BDA.

MC1 then guides the process into *Assess integration potential*, *Score candidates,* and *Assess Feasibility*. This can be performed as a qualitative assessment, e.g., by performing an *Interoperability limitations assessment* highlighting that benefits depend on alignment with the existing C2-system, and on the ISR platform's ability to exploit external/coalition data contributions without introducing connectivity bottlenecks. A second example is to perform an *operational/tactical necessity assessment*, by linking the integration of the ISR platform to the operational need for tempo, such as timely PED outputs, credible target packages, and rapid BDA/re-tasking decisions, all under contested conditions. A third example is to conduct a *Resource sources assessment*, capturing the expected capability improvements in relation to the integration effort required to implement and sustain the solution. Following this reasoning, in this demonstration it is assumed that the integration of the ISR platform improves mission performance by means of (1) faster and more reliable dissemination of decision-ready ISR products, (2) strengthened decision support for targeting and fires coordination, and (3) improved assessment/BDA, enabling rapid re-tasking and re-engagement.

Together, these assessments provide a capability-based motivation for the integration and are consolidated into a preliminary integration recommendation: *Proceed with the integration of the ISR platform as a PED-centric capability as long as (1) interoperability (technology, procedures, organization) is ensured, and (2) the platform can operate under degraded conditions. If these conditions cannot be met, limit the integration to a reduced scope (e.g., PED product generation for selected targets) and reassess against conditions (1) and (2).*

The principal architectural product used to communicate MC1 in this demonstration is the *Capability Hierarchical Diagram*, shown in Figure 14. Following MC1's intent to support a structured assessment of the integration of the ISR platform into a C2-systems, the *Capability Hierarchical Diagram* provides a taxonomy from Joint/Operational Effect, through NATO Joint Functions [46], and into sub-capabilities, using the reasoning and definitions in [47]. The decomposition depth is use-case–driven in this demonstration, where only the capability areas most directly affected by the ISR platform integration (highlighted in green in Figure 13) are decomposed beyond the Joint Function level. The other Joint Functions remain visible only at a higher level to preserve context without decomposing them.

Decomposed here means that the selected Joint Functions are expanded into the specific sub-capabilities (highlighted in blue in Figure 14) needed to reason about the integration in subsequent modeling steps. In this way, the *Capability Hierarchical Diagram* visualizes MC1's outcome by defining the capability boundary and identifying capability areas carried forward to subsequent modeling.

To summarize, in this demonstration, MC1 is applied up to a preliminary recommendation and capability scoping in order to enable the transition to MC3. This step produces (1) a motivated integration recommendation (answering design question 2), and (2) a capability hierarchy that serves as the direct input to MC3 (Cross-Level Integration Coherence).

5.6 Producing C3 – Capability Dependencies

The insights from L2-L3 – Concept Diagram and C1-Capability Hierarchical Diagram serve as input to the next step of the demonstration. Following the previous step, the Navigation Guidance (Figure 3) points to MC3 (Cross-Level Integration Coherence) as the next suitable step. This is because there is a need to assess how the changes introduced by the new ISR platform could be translated into a pattern of capability interaction across domains and levels of command. More specifically, MC3 is used to transform the operational context and the capability mapping into capability dependencies, making explicit which capabilities depend on one another in the kill chain, and thereby to uncover integration-critical areas or redundancies before moving on to more detailed modeling. In this step, a *Capability Dependency Diagram* is used to capture the logical dynamics between capabilities, e.g., how ISR capabilities support decision-making and fires execution across the kill chain. Figure 15 presents the resulting C3 view, focusing on the capability interactions affected by integrating the new ISR platform and illustrating how these interactions may change the overall mission performance.

In this demonstration, MC3 is instantiated by following the intentions outlined in Section 4.4 and visualized in Figure 6: (1) Identify Dependencies, (2) Map Interfaces, (3) Validate Coherence, and (4) Document outcomes and support iteration. Each intention is enacted through specific strategies, consistent with the MC3 description in Section 4.4, and uses the outlined capabilities from the *Capability Hierarchical Diagram* as starting points.

The C3 (Capability Dependencies) viewpoint focuses on how capabilities depend on and support one another in the kill chain, rather than how they are decomposed in a hierarchy. In Figure 15, each connection represents a *Capability Dependency* [7] and is read as a logical statement such as "*Capability A tasks/feeds/enables Capability B"*.

Following the internal logic of MC3, the first intention is to *Identify Dependencies* (Figure 6), where the analysis starts from the capability taxonomy defined in the *Capability Hierarchical Diagram*, i.e., the capability areas considered relevant to the ISR integration. Dependencies are then extracted by translating the MC2 outcome, the *Concept Diagram,* into capability relationships. Concretely, the demonstration elicits three types of dependency patterns from the *Concept Diagram.* First, tasking dependencies, where execution, planning, and prioritization drive ISR collection and fires activities; second, information/product dependencies, where ISR processing and dissemination feed situational awareness, decision support, and the targeting process; and finally, enable dependencies, where protection capabilities support continuity of the kill chain under degraded conditions.

To elicit dependencies, two complementary strategies are applied: (1) the identified dependencies are cross-checked against the scoped capability taxonomy in the *Capability Hierarchical Diagram* (C1) to ensure they are in-scope and consistently defined (model-based consistency check), and (2) the dependencies are cross-checked against the mission logic captured in the *Concept Diagram* to ensure they are scenario-grounded and reflect cross-domain and cross-echelon interactions that become integration-critical when the new ISR platform is introduced.

The *Capability Dependency Diagram*, outlined in Figure 15, is organized into capability clusters reflecting the Joint Functions: Intelligence (Intel), Command and Control (C2), Fires, Protection, and Information.

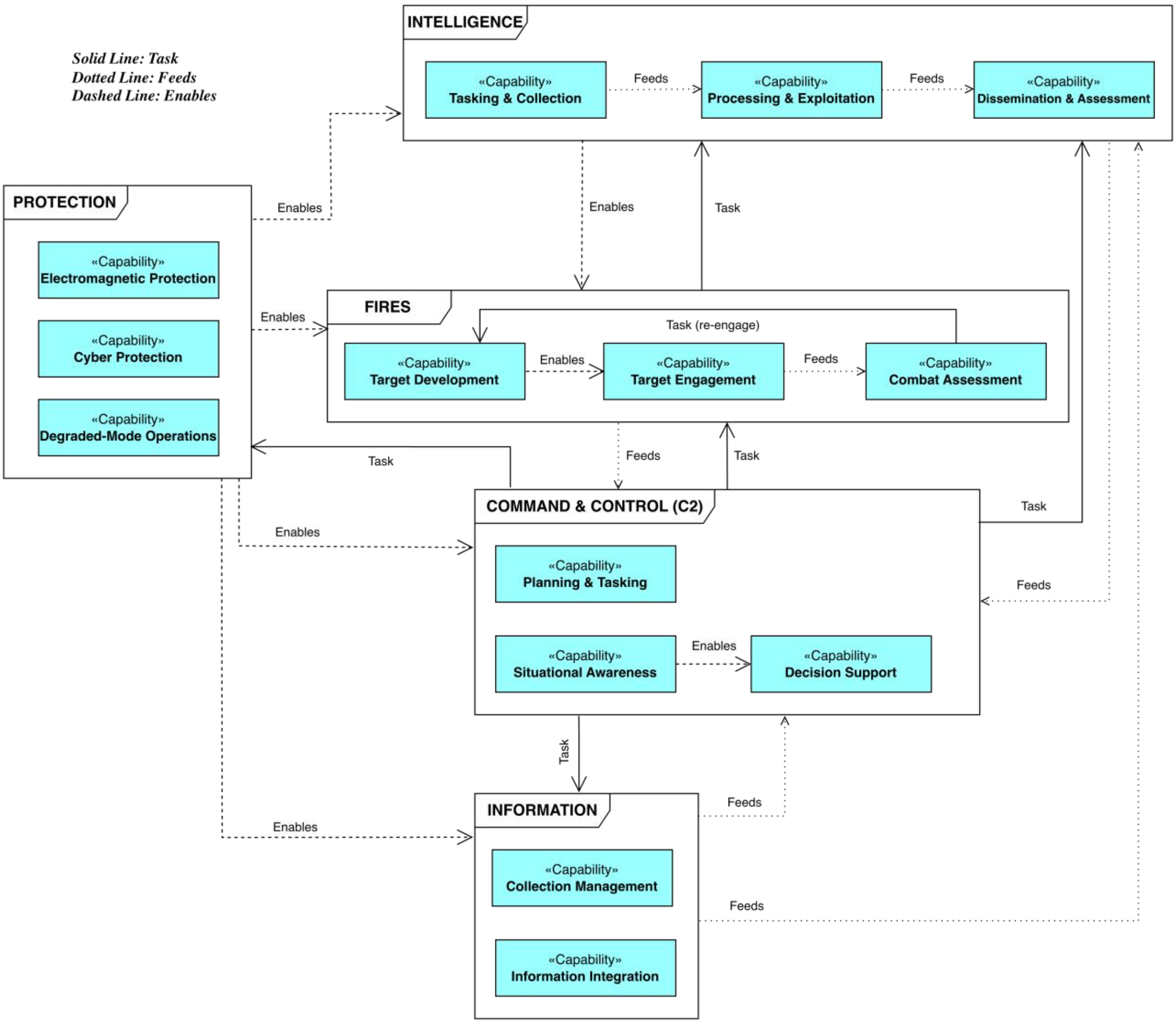

Figure 15. C3 (Capability Dependencies) – Capability Dependency Diagram

The Intel-cluster (Tasking & Collection, Processing & Exploitation, and Dissemination & Assessment) forms a production chain in which collected data is processed and exploited and subsequently disseminated as decision-ready ISR products. These products feed the C2-function, most notably Situational Awareness, which in turn enables Decision Support. The ISR products also support the Fires-cluster by enabling the targeting process. The C2-cluster (Planning & Tasking, Situational Awareness and Decision Support) issues tasks to ISR, Protection, and Fires, reflecting how operational intent and prioritization drive both collection, engagement, and protective activities. Within the Fires-cluster, Target Development enables Target Engagement, which in turn, using ISR products, feeds Combat Assessment (BDA). This assessment provides feedback that informs potential re-tasking and re-engagement.

The Protection-cluster (Electromagnetic Protection, Cyber Protection, and Degraded-Mode Operations) are represented as enabling dependencies that support continuity and resilience of the tasking and information flows under degraded conditions. Finally, the Information-cluster, Collection Management and Information Integration, supports the overall coherence of the cycle by feeding collection-related needs and integrating information to strengthen the C2-cluster. In the *Map Interfaces* intention (Figure 6), these dependencies are interpreted as interface-relevant relationships (tasking, information exchange, enabling services) without committing to a technical implementation. The strategies applied here include procedural workflow mapping (to align C2 processes across functions) and technical interface modeling at a logical level (to clarify system-to-system exchange needs), as well as interoperability considerations under degraded conditions (e.g., bandwidth/latency constraints).

In the *Validate Coherence* intention, the diagram is checked for consistency of the kill-chain loop (task-feed-assess-retask) and for resilience under degraded conditions (i.e., whether enabling protection is represented where tasking and information flows depend on it). The strategies applied here include alignment assessments across viewpoints, model-based verification of information-flow consistency, and doctrinal alignment analysis to ensure the dependency pattern remains compatible with existing operational procedures. Finally, consistent with *Document outcomes and support iteration* (Figure 6), the resulting C3 view is recorded as a dependency baseline that informs subsequent modeling and supports iterative refinement (G11). The strategies applied here include documenting the dependency diagram and producing a concise alignment-oriented summary to support iteration and traceability into later steps.

All connectors in Figure 15 represent the NAF *Capability Dependency* relation [10]. The line styles and labels (task, feeds, enables) are provided only to clarify the intended interpretation of each dependency in this demonstration. The approach demonstrated maintains focus on logical dependencies while providing additional clarity for interpreting how the new ISR platform changes capability interaction patterns across the kill chain.

5.7 Producing A2 – Architecture Products

After the scenario and capability-oriented steps, the demonstration reaches a point where the remaining design question must be addressed: *"How are architectural choices traceable to stakeholder concerns and outputs?"* To focus on this, the Navigation Guidance (Figure 3) is engaged once more, now prioritizing *governance and traceability* rather than additional operational or structural modeling. Following this logic, the Navigation Guidance points to MC7 (Documentation and Reporting) as the appropriate next step, when the purpose is to consolidate what has been produced so far, make the modeling path transparent, and document how each architectural product supports the design questions and the integration reasoning.

In this demonstration, MC7 is instantiated by applying its two core intentions, first, to *consolidate and structure results*, and second, to *produce standardized documentation and traceable reporting*. The consolidation intention is performed through an MC compliance analysis, where outputs from the demonstration steps are organized as a coherent package, e.g., the *Concept Diagram* as the scenario anchor, the *Capability Hierarchical Diagram* as capability scoping, the *Capability Dependency Diagram* as dependency reasoning, and the *Operational Activity Diagram* as the operational activity logic. In parallel, a traceability alignment analysis is applied to ensure that each produced model/representation is linked to (1) the MC that generated it, (2) the viewpoint it conforms to, and (3) the purpose and outcome of the design activity.

The principal architectural viewpoint used to communicate MC7 in this demonstration is the A2 (Architecture Products), because A2 provides a structured way to describe what the architecture contains and how its products correspond to stakeholder concerns.

Concretely, A2 is expressed through two complementary artifacts. First, the A2 viewpoint is complemented by a *NAF grid representation* (Figure 16), which provides an overview of the selected NAF viewpoints used in the demonstration and how they collectively form a coherent architecture description. This grid makes explicit that the demonstration is intentionally scoped to a subset of viewpoints (C1, C3, L2-L3, L4, and A2).

Second, a *Traceability Tabular* (Table 14) documents the modeling path through the demonstration by linking: (1) the applied MCs, (2) the produced models/representations, and (3) the purpose and outcome of the models/representations.

Together with the *NAF grid representation,* the *Traceability Tabular* ensures traceability by showing both: (i) *what was produced* and (ii) *why it was produced, and in what order*, thereby closing design question 4: *"How are architectural choices traceable to stakeholder concerns and* *outputs?"* and providing a documented baseline for iteration and refinement in subsequent design work.

Figure 16. The resulting A2 (Architecture Products) – NAF grid representation

Table 14. The resulting Traceability Tabular – MC used/outputs, viewpoints, and key representations

	Viewpoint(s)	Purpose	Representation	Outcome
MC 2	- L2-L3 (Logical Concept) - L4 (Logical Activity)	Analyze mission execution under representative conditions to understand how the new ISR capability changes operational flow, decision-making, and coordination.	- Concept Diagram (NAF p. 113) - Operational Activity Diagram (NAF p.105)	An operational scenario narrative (Concept Diagram) and an activity workflow (Operational Activity Diagram) describing sequences, information products, and synchronization needs.
MC 1	- C1 (Capability Taxonomy)	Define and assess where the new ISR capability fits within the existing capability taxonomy to scope the integration and frame expected operational benefits.	- Capability Hierarchical Diagram (NAF p.75)	A scoped capability mapping, outlining where the new ISR capability integrates, which capability areas are affected, and which dependencies need further analysis.
MC 3	- C3 (Capability Dependencies)	Evaluate how ISR integration reshapes cross-level capability interactions by mapping tasking, information, and enabling dependencies between capabilities.	- Capability Dependency Diagram (NAF p.79)	A dependency pattern showing how capabilities task, feed, and enable one another across the kill chain.
MC 7	- A2 (Architecture Products)	Consolidate and document the produced architectural products to ensure transparency, traceability, and support stakeholder communication.	- Traceability Tabular (NAF p.135) - NAF grid representation (NAF p.133)	Traceability is achieved via a table linking methods to models, and an overview grid of views is produced.

MC4, MC5, MC6 are not included in this particular demonstration scope

5.8 Summary of the Demonstration Case

The produced architecture does not aim to represent a complete system design. Instead, it functions as a focused method demonstration, constructed to illustrate how the proposed framework, through the Navigation Guidance (Figure 3) and Method Chunks, can guide architectural activities in response to specific design concerns. This specific demonstration is scoped by four guiding design questions, outlined in Section 5.1, and structured to produce representative architectural outputs, while making modeling decisions traceable and dependent on operational context.

The demonstration illustrates a modeling progression anchored in the Navigation Guidance. Starting with MC2 (Scenario-Based Simulation), the process first establishes an operationally grounded understanding of how the new ISR capability affects mission execution. This is expressed through the *Concept Diagram*, Figure 12 (L2-L3 viewpoint) and the *Operational Activity Diagram*, Figure 13 (L4 viewpoint), which together provide an operational narrative and workflow understanding. These two models ensure that subsequent capability-oriented reasoning remains anchored in an operationally relevant context, reflecting real-world mission dynamics.

Building on this, MC1 (Integration Assessment) translates scenario insight into a structured capability framing. This produces a *Capability Hierarchical Diagram,* Figure 14 (C1 viewpoint), that scopes the integration and supports design decisions with a capability-based motivation. The capability scoping and integration recommendation provide a direct input to MC3 (Cross-Level Integration Coherence), which models cross-level dependencies using the *Capability Dependency Diagram*, Figure 15 (C3 viewpoint). This step visualizes how ISR-related capabilities interact across the kill chain and between echelons, uncovering where integration efforts and coordination are required. To ensure that results are transparent and reusable, MC7 (Documentation and Reporting) is applied to consolidate outputs, while maintaining traceability. This is operationalized through a *NAF grid representation*, Figure 16, and a *Traceability Tabular*, Table 14 (A2 viewpoint) capturing the logic of the modeling steps and how each representation links to design questions and stakeholder concerns.

As a whole, the demonstration illustrates how the Navigation Guidance activates the appropriate MCs in response to design needs, and how each MC links intentions and strategies to NAF viewpoints and representations. Presented approach ensures not only modeling coherence, but also situational relevance, traceability, and adaptability. The demonstration produces a coherent architecture, composed of logically connected models that collectively address the four design questions, namely: Q1 (mission impact) by the *Concept Diagram* and the *Operational Activity Diagram,* Q2 (capability framing) by the *Capability Hierarchical Diagram*, Q3 (activity coordination and dependencies) by the *Capability Dependency Diagrams*, and Q4 (traceability) by the *Traceability Tabular* and *NAF Grid representation*.

6 Discussion

This work builds on existing military architectural frameworks, particularly NAF [7], by proposing a complementary, more specific method-oriented layer to support the design process. While NAF provides a structured set of viewpoints and representations for describing design activities, it is less explicit about providing method guidance for how to select an entry point, sequence modeling activities, and iterate in response to specific design concerns. The proposed framework, which includes a Navigation Guidance (Figure 3) and a repository of MCs, is introduced as an attempt to operationalize such process guidance in a way that remains compatible with NAF. Some MCs are included for the purpose of having a complete method framework.

The demonstration case illustrates the plausibility and potential value of this approach by presenting a coherent modeling progression for an ISR-integration situation, thereby addressing the four design questions outlined in Section 5. In our demonstration, the Navigation Guidance suggested a sequence of steps. First, a scenario-based simulation (MC2) established an operational baseline through a *Concept Diagram* and an *Operational Activity Diagram*. This was followed by capability framing (using a *Capability Hierarchical Diagram*) and cross-level dependency analysis (*Capability Dependency Diagram*). Finally, we consolidated the outputs via documentation and traceability artifacts (a *NAF grid representation* and a *Traceability Tabular*).

The demonstration illustrates how operational reasoning can inform capability-level representations without losing context. It also shows how artifacts can be connected so that later models remain traceable to the operational workflow. One implication is that the method layer makes the principles for selecting and sequencing modeling activities explicit, which may help keep the modeling effort aligned with stakeholders' concerns. The demonstration was performed on a hypothetical, yet operationally relevant use case scoped by the authors, thus, the framework's generalizability to other operational scenarios or in live settings has not yet been assessed. Additionally, no external stakeholders were involved in the preparation of this demonstration case.

Because the demonstration is scenario-based and also author-constructed, its main value is feasibility and internal coherence. The next DSR cycle will therefore involve external stakeholders (C2-system architects and practitioners) in structured walkthroughs to assess relevance and appropriateness of MC selection and the usefulness of the generated NAF artifacts for integration design decisions.

The Navigation Guidance and MCs draw on SME principles [11], [12], [13] by defining MCs as modular units with specific intentions and strategies, supported by MAP-based formalization [32], [33]. The MC repository should be understood as a reusable set of context-driven method elements, where each MC activates one or more NAF viewpoints and representations. This clarifies how viewpoints provide the modeling framing while representations become the concrete artifacts produced.

In DSR, a robust artifact, in this work the proposed framework, typically emerges through multiple design-evaluation cycles that include evaluation in realistic settings. In this regard, the contributions of this article provide an initial foundation and a motivation for further evaluation. A subsequent evaluation phase will assess the framework using explicit criteria focusing on utility for integration decisions, usability of navigation/MC selection, and traceability between stakeholder concerns and produced NAF artifacts.

7 Conclusions and Future Work

This study completes the development iteration of a modular and context-informed extension to the NATO Architecture Framework (NAF) for supporting the development of military command and control systems. A framework consisting of a Method Chunk (MC) repository and a Navigation Guidance (Figure 3) is proposed and its utility is demonstrated through an instantiated walkthrough, and in DSR terms, this supports internal coherence and plausibility. The framework is intended to support the selection and sequencing of design activities and strengthen traceability between the operational context, design decisions, and architectural products, i.e., contributes to method guidance, navigation logic, and MCs, that suggests how design activities may proceed in a given situation. Our demonstration of the framework's use starts at the point of integrating a new Intelligence, Surveillance, and Reconnaissance (ISR) capability. In addition, the framework is useful in guiding earlier phases, such as technology scouting [48]. For instance, the MC1 outputs include a prioritized technology list that could support technology scouting efforts.

The demonstration reveals how the framework can be applied to an ISR-integration challenge. It exemplifies how a consistent set of linked models can be produced and used to structure reasoning against the four design questions. This work demonstrates how SME principles can be successfully used to extend architecture frameworks such as NAF and provide stakeholders with a candidate method to systematically integrate new technologies using NAF. The proposed framework produces standard NAF outputs (e.g., NAF models), with the Navigation Guidance and Method Chunk repository serving as a methodological extension. However, the framework does not replace NAF's role as an architecture framework. NAF viewpoints and representations remain the primary means of expressing the architecture, where the framework of this research specifies how to select and sequence these products to support emerging-technology integration decisions.

Furthermore, consistent with DSR logic, this demonstration should be treated as a proof-of-concept instantiation of the framework, not as a full evaluation of its effectiveness. Future work will focus on evaluating the framework with stakeholders using this demonstration case and subsequent iterative refinements. From a methodological perspective, future work will extend the framework in four concrete ways: (1) Navigation Guidance refinements using stakeholder feedback and instantiations to reduce uncertainties in MC selection; (2) execute stakeholder-based walkthroughs and to iteratively improve MC completeness and usability; (3) additional MCs; (4) tailoring our MCs to support other EA frameworks, such as TOGAF [16]. Planned activities include applying the framework to realistic design challenges with multiple stakeholders.

References

[1] M. I. Handel, “Technological Surprise in War,” *Intelligence and National Security*, vol. 2, no. 1, pp. 1–53, 1987. Available: https://doi.org/10.1080/02684528708431875

[2] M. Finkel, *On flexibility: Recovery from Technological and Doctrinal Surprise on the Battlefield*. Stanford Security Studies, 2011. Available: https://www.adlibris.com/se/bok/on-flexibility-9780804774895. Accessed on Jun. 13, 2024.

[3] D. Phillips, “Multi-Domain Operations: Passing the Torch,” in *Hague Centre for Strategic Studies, JSTOR*, 2023. Available: https://www.jstor.org/stable/resrep54863. Accessed on Jan. 04, 2024.

[4] J. Lundberg, “Towards a Conceptual Framework for System of Systems,” in *Proceedings of the Doctoral Consortium Papers Presented at the 35th International Conference on Advanced Information Systems Engineering (CAiSE 2023), CEUR Workshop Proceedings*, vol. 3407, pp. 18–24, 2023. Available: https://ceur-ws.org/Vol-3407/paper3.pdf

[5] J. Lundberg, J. Stirna, and K. Andersson, “Designing Military Command and Control Systems as System of Systems – An Analysis of Stakeholder Needs and Challenges,” in *Advanced Information Systems Engineering, CAiSE 2024. Lecture Notes in Computer Science*, vol. 14663, Springer, pp. 336–351, 2024. Available: https://doi.org/10.1007/978-3-031-61057-8_20

[6] E. Niewood, G. Grant, and T. Lewis, “A new battle command Architecture for Multi-Domain Operations,” 2019. Available: https://www.mitre.org/news-insights/publication/new-battle-command-architecture-multi-domain-operations. Accessed on Aug. 13, 2023.

[7] NATO, “NATO – NATO Architecture Framework, Version 4.1.” Available: https://www.nato.int/content/dam/nato/webready/documents/publications-and-reports/NATO-Architecture-Framework-v4-1-en.pdf. Accessed on Jul. 27, 2026.

[8] A. R. Hevner, S. T. March, J. Park, and S. Ram, “Design Science in Information Systems Research,” *MIS Quarterly*, vol. 28, no. 1, pp. 75–106, 2004. Available: https://doi.org/10.2307/25148625

[9] J. Lundberg, K. Andersson, J. Ralyté, and J. Stirna, “Supporting Future Military Command- and Control (C2) System Design: Defining Method Components to Enhance the Existing Development Framework,” in *ICCRTS25 – International Command and Control Research & Technology Symposium*, 2025.

[10] B. Henderson-Sellers, J. Ralyté, P. J. Ågerfalk, and M. Rossi, *Situational Method Engineering*. Springer, 2014. Available: https://doi.org/10.1007/978-3-642-41467-1

[11] J. Ralyté, R. Deneckère, and C. Rolland, “Towards a Generic Model for Situational Method Engineering,” in *Advanced Information Systems Engineering. CAiSE 2003. Lecture Notes in Computer Science*, vol. 2681, Springer, pp. 95–110, 2003. Available: https://doi.org/10.1007/3-540-45017-3_9

[12] I. Mirbel and J. Ralyté, “Situational method engineering: Combining assembly-based and roadmap-driven approaches,” *Requirements Engineering*, vol. 11, pp. 58–78, 2006. Available: https://doi.org/10.1007/s00766-005-0019-0

[13] J. Ralyté, P. Backlund, H. Kühn, and M. A. Jeusfeld, “Method chunks for interoperability,” in *Conceptual Modeling – ER 2006. ER 2006. Lecture Notes in Computer Science*, vol. 4215, Springer, pp. 339–353, 2006. Available: https://doi.org/10.1007/11901181_26

[14] B. Henderson-Sellers, C. Gonzalez-Perez, and J. Ralyté, “Comparison of method chunks and method fragments for situational method engineering,” in *Proceedings of the Australian Software Engineering Conference, ASWEC*, pp. 479–488, 2008. Available: https://doi.org/10.1109/ASWEC.2008.4483237

[15] DODAF – DOD Architecture Framework Version 2.02. Available: https://dodcio.defense.gov/library/dod-architecture-framework/. Accessed on Jun. 13, 2024.

[16] The TOGAF® Standard, 10th Edition. Available: https://www.opengroup.org/togaf/10thedition. Accessed on Feb. 25, 2025.

[17] J. Bankauskaite, “Comparative analysis of enterprise architecture frameworks,” in *Proceedings of the International Conference on Information Technologies,* CEUR *Workshop Proceedings*, vol. 2470, pp. 61–64, 2019. Available: https://ceur-ws.org/Vol-2470/p19.pdf

[18] H. A. H. Handley, “A Design Methodology for Fit-for-Purpose Human Views,” *Systems Engineering*, vol. 19, no. 6, pp. 498–509, 2016. Available: https://doi.org/10.1002/sys.21371

[19] S. Buckl, F. Matthes, and C. M. Schweda, “A Method Base for Enterprise Architecture Management,” in *Engineering Methods in the Service-Oriented Context. ME 2011. IFIP Advances in Information and Communication Technology*, vol. 351, Springer, pp. 34–48, 2011. Available: https://doi.org/10.1007/978-3-642-19997-4_5

[20] M. Farwick, C. M. Schweda, R. Breu, and I. Hanschke, “A situational method for semi-automated Enterprise Architecture Documentation,” *Software & Systems Modeling*, vol. 15, pp. 397–426, 2016. Available: https://doi.org/10.1007/s10270-014-0407-3

[21] E. Kornyshova and R. Deneckère, “A Proposal of a Situational Approach for Enterprise Architecture Frameworks: Application to TOGAF,” *Procedia Computer Science*, vol. 207, pp. 3493–3506, 2022. Available: https://doi.org/10.1016/j.procs.2022.09.408

[22] H. D. Jørgensen, T. Liland, and S. Skogvold, “Aligning TOGAF and NAF - Experiences from the Norwegian armed forces,” *Lecture Notes in Business Information Processing*, vol. 92, Springer, pp. 131–146, 2011. Available: https://doi.org/10.1007/978-3-642-24849-8_11

[23] INCOSE, *Systems engineering handbook – A Guide for system life cycle processes and activities*. Fourth edition. Wiley, 2015.

[24] A. Hevner and S. Chatterjee, *Design Research in Information Systems*. Springer, 2010. Available: https://doi.org/10.1007/978-1-4419-5653-8

[25] K. Andersson, J. Lundberg, and J. Stirna, “Emerging technology calls for a systemic view on military capability,” in *Companion Proceedings of the 16th IFIP WG 8.1 Working Conference on the Practice of Enterprise Modeling and the 13th Enterprise Design and Engineering Working Conference: BES, DTE, FACETE, Tools & Demos, Forum, EDEN Doctoral Consortium, co-located with PoEM 2023, CEUR Workshop Proceedings*, vol. 3645, 2023. Available: https://ceur-ws.org/Vol-3645/facete2.pdf

[26] J. Lundberg, J. Stirna, J. Zdravkovic, and K. Andersson, “Beyond Technology: Goal-Oriented Analysis for Integrating Emerging Technologies into Military Command and Control Systems,” in *29th International Command and Control Research and Technology Symposium*, London, Sep. 2024.

[27] J. Lundberg, S. Hacks, and K. Andersson, “Refinement of a Conceptual Model of a military C2-system through Low-Level Goal Decomposition,” in *Companion Proceedings of the 17th IFIP WG 8.1 Working Conference on the Practice of Enterprise Modeling Forum, M4S, FACETE, AEM, Tools and Demos, CEUR Workshop Proceedings*, vol. 3855, 2024. Available: https://ceur-ws.org/Vol-3855/forum9.pdf

[28] J. Lundberg, K. Andersson, and J. Stirna, “Enhancing C2-Systems: Validation of Goal and Concept Models with Stakeholders,” in *Enterprise, Business-Process and Information Systems Modeling. BPMDS EMMSAD 2025. Lecture Notes in Business Information Processing*, vol. 558, Springer, pp. 353–367, 2025. Available: https://doi.org/10.1007/978-3-031-95397-2_22

[29] J. Lundberg, “Operationalizing Evaluation in Design Science Research: A Structured Framework for Artifact Assessment in C2-System Development,” in *Companion Proceedings of the 18th IFIP Working Conference on the Practice of Enterprise Modeling: PoEM Forum, Doctoral Consortium, Business Case and Tool Forum*, *CEUR Workshop Proceedings*, vol. 4171, 2025. Available: https://ceur-ws.org/Vol-4171/paper_60.pdf

[30] S. Brinkkemper, “Method engineering: engineering of information systems development methods and tools,” *Information and Software Technology*, vol. 38, no. 4, pp. 275–280, 1996. Available: https://doi.org/10.1016/0950-5849(95)01059-9

[31] J. Ralyté and C. Rolland, "An approach for method reengineering," in *Conceptual Modeling – ER 2001. ER 2001. Lecture Notes in Computer Science*, vol. 2224, Springer, pp. 471–484, 2001. Available: https://doi.org/10.1007/3-540-45581-7_35

[32] C. Rolland, N. Prakash, and A. Benjamen, "A multi-model view of process modelling," *Requirements Engineering*, vol. 4, pp. 169–187, 1999. Available: https://doi.org/10.1007/s007660050018

[33] C. Rolland, "Capturing System Intentionality with Maps," in *Conceptual Modelling in Information Systems Engineering*, pp. 141–158, 2007. Available: https://doi.org/10.1007/978-3-540-72677-7_9

[34] C. Rolland and N. Prakash, "A Proposal for Context-Specific Method Engineering," in *Method Engineering, ME 1996. IFIP – The International Federation for Information Processing*, pp. 191–208, 1996. Available: https://doi.org/10.1007/978-0-387-35080-6_13

[35] J. Lundberg, J. Stirna, and K. Andersson, "Operationalizing military insights – A second report on C2-System development." Available: https://www.researchgate.net/publication/386328609_Operationalizing_military_insights_-A_second_report_on_C2-System_development. Accessed on Dec. 02, 2024.

[36] M. A. Harwell, T. C. Hutchinson, J. W. P. Cropper, C. C. Harwell, and H. D. Grover, *Environmental Consequences of Nuclear War*. John Wiley and Sons, 1989.

[37] U. Siebert et al., "State-Transition Modeling: A Report of the ISPOR-SMDM Modeling Good Research Practices Task Force-3," *Value in Health*, vol. 15, no. 6, pp. 812–820, 2012. Available: https://doi.org/10.1016/j.jval.2012.06.014

[38] P. Johannesson and E. Perjons, *An Introduction to Design Science*. Springer, 2014. Available: https://doi.org/10.1007/978-3-319-10632-8

[39] C. Brose, *The Kill Chain: Defending America in the Future of High-Tech Warfare*. Hachette Books, 2020.

[40] H. R. Penney, "Scale, Scope, Speed & Survivability: Winning the Kill Chain Competition," *MITCHELL INSTITUTE Policy Papers*, vol. 40, 2023. Available: https://www.mitchellaerospacepower.org/app/uploads/2023/05/Scale_Scope_Speed_Survivability_-KillChain_-Policy_Paper_40-New.pdf. Accessed on Aug. 13, 2023.

[41] A. Cockburn, *Kill Chain: The Rise of the High-Tech Assassins*. Henry Holt and Co., 2015.

[42] P. Berens, "Introduction to Synthetic Aperture Radar (SAR)," in *Advanced Radar Systems, Signal and Data Processing, Educational Notes RTO-EN-SET-086bis*, paper 3, 2007. Available: https://publications.sto.nato.int/publications/STO%20Educational%20Notes/RTO-EN-SET-086bis/EN-SET-086bis-03.pdf. Accessed on Jan. 13, 2026.

[43] J. Andress and S. Winterfeld, "Cyber Doctrine," *Cyber Warfare*, pp. 53–82, 2014. Available: https://doi.org/10.1016/B978-0-12-416672-1.00004-0

[44] A. Dupont, "Intelligence for the Twenty-First Century," *Intelligence and National Security*, vol. 18, no. 4, pp. 15–39, 2006. Available: https://doi.org/10.1080/02684520310001688862

[45] NATO, "Allied Joint Publication (AJP) 3.9 Joint Targeting." Available: https://nllp.jallc.nato.int/cmnt/ciedcoi/CIED%20PUBLICATIONS/Handbooks%20and%20Doctrines/AJP%203.9.%20DOCTRINE%20OF%20JOINT%20TARGETING.pdf. Accessed on Jan. 06, 2026.

[46] NATO, "NATO AJP-3 Joint Doctrine for the Conduct of Operations," 2025.

[47] NATO, "NATO AJP 3.2 Joint Doctrine for Land Operations," 2022.

[48] O. L. de Weck, "Technology Scouting," in *Technology Roadmapping and Development*, pp. 395–424, 2022. Available: https://doi.org/10.1007/978-3-030-88346-1_14