Wargame Simulation

Exploiting the Power of a Flow-Based Design

Master Thesis

Drs. H.B. Hovestad

Delft University of Technology

TU Delft

Wargame Simulation

Exploiting the Power of a Flow-Based Design

by

Drs. H.B. Hovestad

Student number: 9369609

Master thesis submitted to Delft University of Technology
in partial fulfilment of the requirements for the degree of

MASTER OF SCIENCE

in
Complex Systems Engineering and Management
Faculty of Technology, Policy and Management

To be defended in public on October 27th, 2023

Graduation committee

Chairperson:	Prof. dr. ir. A. (Alexander) Verbraeck
First Supervisor:	Dr. Y. (Yilin) Huang
Advisor:	Ir. O. (Omar) Quispel (TU Delft)
External Advisor:	Ir. T.W. (Tom) van den Berg (TNO)

Cover image:	*Black pieces versus white team* retrieved from https://www.freepik.com.
Military Symbology:	The unit symbols that are used within the various figures within this research are generated by Military symbology picker.

An electronic version of this thesis is available at http://repository.tudelft.nl.

TU Delft

Preface

Dear reader,

This thesis concludes my time as a student at the Faculty of Technology, Policy and Management at the Delft University of Technology. The courses I attended in the past two years have offered new and inspiring knowledge about the complex domain of system engineering and management. After these two insightful study years, I would like to express my utmost appreciation for my dear wife, Hildegard, for supporting and motivating me to accomplish a Master's Degree in Complex System Engineering and Management. I would also like to express my love and gratitude to my children Maurits and Eveline, because my studies did not always allow me to give them the attention they deserved.

This thesis could not have been completed without the support and supervision of my graduation committee. I want to thank them for their inspiring ideas and positive feedback while researching and writing this thesis. Alexander thanks for your enthusiastic discourse on Modeling & Simulation this not only started a whole new academic journey, it also resulted in my current position within the Dutch Armed Forces; Yilin I am very grateful for your feedback how to improve readability of my thesis for the non-military readers and your questions about my system decomposition; and I would like to thank Omar for his overall support during this research and I wish him the best of luck in further completing his PhD.

I would like to thank Tom from TNO, for his supervision and especially for his insights into military simulation. Furthermore, I would like to thank Desmond and Hans from the Dutch Ministry of Defence for their insights on military simulation within the Dutch Armed Forces and of course for their invitation to join their department, which I gracefully accepted. Lastly, I would like to express my utmost gratitude to Ivo, in his role as CoSEM program director, for his enthusiasm, support, and confidence, which allowed me to pursue this master's degree without having a relevant bachelor degree.

I realize that I have been very privileged as a military officer to be able to follow this master's program. I look forward to return this personal and professional investment by applying the acquired knowledge in my current and future positions to create value for the Dutch Armed Forces.

I hope you enjoy reading my thesis.

Harmen Hovestad

Delft University of Technology, October 2023

Executive Summary

Military decision-making involves art and science and aims to translate strategic goals into tactical actions for achieving specific outcomes. It relies on a commander's capacity to navigate complex and sometimes ambiguous situations to develop detailed plans and actionable orders. At the military tactical level, decisions focus on executing tactical mission tasks that contribute to achieving operational and strategic objectives. Tactical decision-making within the Dutch Army involves a step-by-step rational decision-making process, mainly focused on analytical preparations for battle. An operational analysis or wargame of the developed plans is crucial in military decision-making. Such an analysis offers valuable insights into the most favorable courses of action compared with the presumed plans of other actors. The execution of a wargame empowers a commander to determine the optimal course of action.

Operational analysis, or wargaming, is a formalized military process that technology can support. Modern wargaming originated from the Prussian Army in the 1820s and has since spread to many other armed forces. Wargaming assesses the forces' positions, strengths/weaknesses, opportunities, and environmental factors using doctrine, experience, and intuition. It is an iterative process highlighting critical tasks and tactical possibilities, often aided by modern information technology to increase efficiency.

Armed forces are using information technology to support decision-making, whereby Modeling and Simulation systems offer valuable tools for better situational understanding and assessing interventions' effects. These simulations can provide the forces with crucial data to develop and assess possible courses of action. Military simulation spans various domains, from large-scale field exercises to abstract computational models. While simulations can accelerate the development of military knowledge and experience, it is essential to ensure that increased computing power aligns with a comprehensive understanding of warfare.

Within contemporary military simulation, there is a move from comprehensive monolithic systems to component-based simulation systems to facilitate innovation, flexibility, and cost-effectiveness. While reusability and composability are crucial for military simulation, they have been challenging. Reuse offers various technical, business, and economic benefits. Composability is vital for the system of systems that form contemporary military simulation environments, where linkage is challenging due to differing resolutions and levels of abstraction. High-resolution models help military staff to understand phenomena, while low-resolution models aid analytical understanding qualitatively. In these simulation environments, aggregate-level simulations represent military units with multiple entities, while entity-level simulations depict individuals. However, composing both types of simulation within military wargaming to support decision-making remains challenging. Based on this analysis of contemporary military simulation, the following main research question is formulated:

What are design principles for composable elements that can be used in a simulation system for military wargames?

Because military decision-making processes are not intentionally designed to be represented by simulation systems, the wargaming elements derived from relevant perspectives on military wargaming provide insight into which underlying principles may be usable when designing a wargaming simulation. The following principles have been deduced from military wargaming: a representative wargaming experience (principle of recognizability); a user interface that is designed for the users' intended purpose for the system (principle of utilizability); traceable simulation results that create insightful knowledge (principle of traceability); and adaptable systems that can integrate changes and developments within military doctrine (principle of adaptability).

The challenges arising from Modeling & Simulation research provide rigorous knowledge for the design of military wargaming simulation systems. The following principles are derived from the selected scientific literature: consistency in the models and the attribute states in the wargame simulation system to maintain its validity (principle of consistency); a designer's awareness and understanding of the implications of design choices for performance (principle of coherence); an ability to combine, recombine, configure, and reconfigure sets of components (principle of composability); and a possibility to represent interactions between different military units and echelons (principle of interoperability).

A system requirements structure links requirements to the above principles to assess the extent to which the identified design principles are usable, ultimately aligning with the defined objectives. Evaluation of the requirements reveals that a wargame simulation system must have a utilizable design that represents real-world wargaming, traceable results that create insightful knowledge, adaptable models to assimilate to evolving military doctrine, combinable, configurable, and interoperable components; it should have a recognizable user interface and consistent model behavior; while a such a simulation system could have a coherent design that incorporates the used Run-Time Infrastructure.

The wargame simulation system prototype within this research is designed to support a specific activity within a military decision-making process, the operational analysis, rather than replacing the entire process. While promising, it is still a prototype with limited functionalities. The prototype is tailored to evaluate design principles by a concise military scenario; however, it can be extended to encompass additional tactical tasks or units, showcasing the flexibility of the flow-based design for creating a Course of Action. While the prototype is intended for evaluating the identified design principles, its current utility for practical military operational analysis remains limited.

This research provides both military practitioners and Modeling & Simulation scientists the opportunity for further field experiments and theoretically explore the identified design principles for composable simulation elements that can be used in a flow-based wargame simulation system. Future scientific research should further improve the envisioned simulation concept into a usable wargame simulation system so military commanders and their staffs can exploit the power of simulation while conducting a wargame.

Contents

Preface i

1 Introduction 1
- 1.1 Military Decision-Making 1
- 1.2 A Brief History of Wargaming 2
- 1.3 The Value of Military Simulation 2
- 1.4 The Future of Military Modeling & Simulation 3
- 1.5 Research Questions 4
- 1.6 Research Approach 5
- 1.7 Structure 6

2 Understanding Military Wargaming 8
- 2.1 The Nature of Wargaming 8
- 2.2 Miltary Simulation Standards 12
- 2.3 Modern Conflict Ontology 17
- 2.4 Identified Elements and Principles 19

3 Military Simulation Challenges 22
- 3.1 Composability 23
- 3.2 Multi-Resolution 24
- 3.3 Consistency 28
- 3.4 Run-time Infrastructure 29
- 3.5 Identified Challenges and Principles 29

4 Design Methodology 31
- 4.1 Selection of the Design Process 31
- 4.2 Distributed Simulation Engineering and Execution Process 32
- 4.3 Application of DSEEP 32

5 Prototype Design 35
- 5.1 Wargame Simulation Objective 35
- 5.2 Conceptual Analysis of a Wargame Simulation 36
- 5.3 Wargame Simulation Environment Design 40
- 5.4 Simulation Environment Development 42
- 5.5 Prototype Proposal for the Wargame Simulation System 44

6 Proof of Concept 46
- 6.1 Real-World Wargame Scenario 47
- 6.2 From a SyncMatrix to a Flow of Events 49
- 6.3 Experiment Setup and Simulation Results 52
- 6.4 Analyze Data and Evaluate Requirements 59
- 6.5 Evaluation of the Principles 63

7 Conclusion 65
- 7.1 Answers to Sub-Questions 65
- 7.2 Answer to the Main Research Question 68
- 7.3 Military and Scientific Relevance 69
- 7.4 Recommendations 69

References 71

A Tactical Decision-Making 76

B Wargame Scenario **78**
B.1 Stage 1: Understand the Mission 78
B.2 Stage 2: Assess the Mission 79

C Design Implementation **84**
C.1 Scenario Tasks Linked to Components 84
C.2 Implementation 85

D Proof of Concept Results **90**
D.1 Computations Experiment 1. 90
D.2 Computations Fighting power and Protective Measures 90
D.3 Computations Experiment 2. 91

List of Figures

1.1 The nature of military decision-making. 1
1.2 The spectrum of military simulations. 3
1.3 Design Science Research. 5
1.4 The six dimensions that form the core of the design of the wargame simulation system. 6

2.1 Graphic depicting the relationships between the definition and the general purposes of wargames. 9
2.2 Royal Netherlands Army Tactical Decision-Making Model. 10
2.3 Graphic depicting the relationships between combat components. 11
2.4 Process diagram depicting the conceptual employment of the C-BML standard. 14
2.5 Process diagram depicting the C-BML XML schema file dependencies. 15
2.6 Basic context diagram depicting the Modern Conflict Ontolology. 18

3.1 Notional example of composability. 23
3.2 High Level Architecture. 24
3.3 Disaggregated and aggregated units. 25
3.4 Graphic depicting the multi-resolution challenge for military simulation. 26

4.1 Distributed Simulation Engineering and Execution Process steps. 32

5.1 Military officers engaged in a real-world wargaming. 37
5.2 Conceptual model for a wargame simulation. 38
5.3 System Requirement Structure. 39
5.4 Design for the Wargame Simulation Prototype. 40
5.5 Generic design for the Event-Component. 42
5.6 A mock-up for the user interface of the wargame simulation system. 43

6.1 SyncMatrix showing the concept of operations sub-unit tasking for a possible COA. . . . 49
6.2 The selection of components that form the flow for the component integrity experiment. 53
6.3 Engagement Simulation by connecting Event-Components. 55

B.1 Red Team ORBAT and TOE. 78
B.2 Most likely and most dangerous Red Team Courses of Action. 78
B.3 Blue Team ORBAT. 79
B.4 Blue Team TOE. 79
B.5 Blue Team COA ”SWIFT” - Phase 1. 80
B.6 Blue Team COA ”SWIFT” - Phase 2. 80
B.7 Blue Team COA ”SWIFT” - Phase 3 - Alternative A. 81
B.8 Blue Team COA ”SWIFT” - Phase 3 - Alternative B. 81
B.9 Blue Team COA ”SWIFT” - Phase 4. 82
B.10 Blue Team COA ”SWIFT” - Phase 5. 82

List of Tables

2.1 Basic Military Problem Solving Questions. 10
2.2 C-BML XML Schema Files. 16
2.3 The Five Core Concepts of JC3IEDM. 16
2.4 Simulation concepts for wargaming. 20
2.5 Overview of identified wargaming elements. 20

6.1 Overview of identified Components and tactical tasks. 51
6.2 Experiment setup for the validation the validation of the simulation system. 52
6.3 Input and output of the Event-Component for the on-road-movement tasks. 57
6.4 Input and output of the Event-Component for the off-road-movement tasks. 58
6.5 Input and output of the Event-Components for the inter-operation tests. 58
6.6 Comparison of the simulation and scenario times in minutes for test 1.1. 60
6.7 Comparison of the simulation and scenario times in minutes for test 1.2. 61
6.8 Comparison of the simulation and scenario times in minutes for experiment 2. 61
6.9 Tracing the results back to the User Needs or Simulation Requirements. 63
6.10 Overview of the identified principles for a composable wargame simulation system. . . . 64

A.1 Components within the Royal Netherlands Army Tactical Decision-Making Model. 76

C.1 Sub-unit tasking coupled to simulation components. 84
C.2 Example for the Capability Database. 85
C.3 Data Sources for the movement model within an Event-Component. 87
C.4 Data Sources for the engagement model within an Event-Component. 88
C.5 Unit configuration and initial attribute states that is inserted in the Unit-Component. . . . 88
C.6 Model configuration of Event-Component for the on-road-movement tasks. 89
C.7 Model configuration of Event-Component for the off-road-movement tasks. 89

D.1 Computation of the Event-Component for the on-road-movement tasks. 90
D.2 Computation of the Event-Component for the off-road-movement tasks. 90
D.3 Computation of the fighting power for Most Likely Red Team COAs. 90
D.4 Computation of the protective measures for Most Likely Red Team COAs. 90
D.5 Computation of the fighting power for Most Dangerous Red Team COAs. 91
D.6 Computation of the protective measures for Most Dangerous Red Team COAs. 91
D.7 Computation of engagement results between A(BT)InfCoy and Red Team forces. 91
D.8 Computation of engagement results between C(BT)TkSqdn and Red Team forces. . . . 91
D.9 Computation of engagement results between A(BT)InfCoy and Red Team forces. 91
D.10 Computation of engagement results between A(BT)InfCoy and Red Team forces. 92
D.11 Computation of engagement results between C(BT)TkSqdn and Red Team forces. . . . 92
D.12 Computation of engagement results between A(BT)InfCoy and Red Team forces. 93
D.13 Computation of engagement results between C(BT)TkSqdn and Red Team forces. . . . 93

Abbreviations

AA Assembly Area.

BT Blue Team.

C-BML Coalition Battle Management Language.
C2 Command and Control.
C2SIM Command and Control Systems – Simulation Systems Interoperation.
CCIR Commander's Critical Information Requirements.
COA Course of Action.
CONOPS Concept of Operations.

DIS Distributed Interactive Simulation.
DSEEP Distributed Simulation Engineering and Execution Process.
DSR Design Science Research.

HLA High Level Architecture.

IFV Infantry Fighting Vehicle.
INF Infantry.

JC3IEDM Joint Consultation, Command and Control Information Exchange Data Model.

LVC Live, Virtual and Constructive.

M&S Modeling & Simulation.
MBT Main Battle Tank.
MCO Modern Conflict Ontolology.
MOOTW Military Operations Other Than War.
MSaaS Modeling & Simulation as a Service.
MSDL Military Scenario Definition Language.
MSR Main Supply Route.

NATO North Atlantic Treaty Organization.

ORBAT Order of Battle.

PL Phase Line.

RT Red Team.
RTI Run-Time Infrastructure.

SISO Simulation Interoperability Standards Organization.
SL Start Line.
SOA service-oriented architecture.
SoS System of Systems.
STS Socio-Technical System.

TAA Tactical Assembly Area.
TBM Tactical Decision-Making Model.
TOE Table of Equipment.

UN United Nations.
US United States.

XML Extensible Markup Language.

1

Introduction

> *"No plan survives first contact with the enemy."*
> Helmuth Graf von Moltke - the Elder

Military decision-making is an art as well as a science. Operational art is orchestrating a campaign or major operation, in close cooperation with other (non-)governmental organizations, to convert strategic objectives into tactical activities to achieve a desired outcome. It embraces a commander's ability to take complex and often unstructured problems and provide sufficient clarity and logic (some intuitive) to enable detailed planning and executable orders. It is realized by combining a commander's skill and the staff-assisted processes within military decision-making.

1.1. Military Decision-Making

At the military tactical level, decisions are made about the execution of actual activities that contribute to the operational and strategic objectives. Although the NATO Operational Planning Process can also be helpful in the decision-making process at the tactical level, for that level are more fitting process models that can be used (Ministry of Defense, 2012). Examples of domain-specific models for the tactical level are the Dutch Army's Tactical Decision-Making Model (TBM)[1], the British seven-questions model and the US Army's Military Decision-Making Process. The Dutch Army Tactical Decision-Making Model, in which the analytical preparations for battle are made, is a step-by-step rational decision-making process.

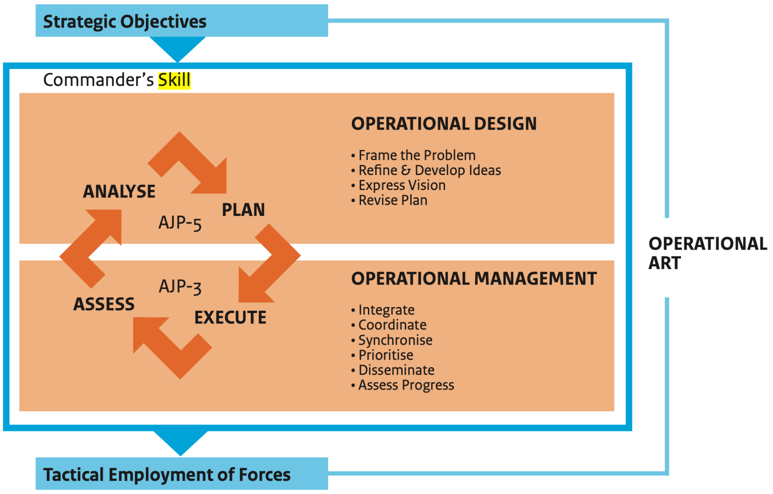


Figure 1.1: The nature of military decision-making (Ministry of Defense, 2012).

The Dutch Army TBM, in which the commander, together with his staff, weighs up various possible solutions and ultimately takes a decision, is interoperable with the corresponding NATO processes (Royal Netherlands Army, 2014). This process is represented by the following steps: analysis, planning, execution, and evaluation. Figure 1.1 depicts the military decision-making process in the broader context of military operational art.

---

[1] TBM is the Dutch Armed Forces acronym for 'Tactical Decision-Making Model'.

While the planning process is finalized by communicating the plan to subordinate commanders, military decision-making remains continuous. Since the nature of war is a battle of human will, the actions of the warring parties remain unpredictable, and a plan is rarely carried out as intended. The Prussian strategist von Moltke (1880) famously asserted that „Kein Operationsplan reicht mit einiger Sicherheit über das erste Zusammentreffen mit der feindlichen Hauptmacht hinaus[2].” Subsequently, after a plan is issued, progress is monitored. If necessary, the plan is adjusted, contingency plans are developed and executed, or even the plan may be completely revised, making military decision-making a continuous process. An operational analysis to compare one's plan to the expected behavior of other relevant actors is an essential step in the decision-making process, whereby it provides insight into the most promising courses of action compared with the assumed plans of the other actors, allowing a commander to make his decide what is the best option.

1.2. A Brief History of Wargaming

Wargames originated from the rulers of early civilizations, and these rulers felt the same need to outwit other rulers. While games like ”Go” and ”chess” are abstract representations of war, they teach players to anticipate the consequences of one's possible moves and the opponent's possible reactions, an essential skill in the deadly game of war (Caffrey, 2000).

Operational analysis, also known as wargaming, is carried out by military staff according to formalized rules and procedures. It is supported by various technical means, making it a Socio-Technical System (Bots et al., 2017). Wargaming in its modern form originated in Prussia in the 1820s. Two officers (von Reiswitz and his son) developed a set of Instructions for the representation of tactical maneuvers in the form of a Kriegsspiel (Caffrey, 2000; McHugh, 1966). In 1824, the Kriegsspiel was demonstrated to General von Muffling, the Chief of the Prussian General Staff, who, in turn, introduced the concept to the Army. While the Prussians were the first to embrace wargaming, other nations soon copied the technique. Over the next two centuries, the armed forces of most nations employed various forms of wargaming for training and planning purposes, and wargaming was generally accepted across the military by the mid-twentieth century (Perla, 1990; Turnitsa et al., 2022; UK MoD, 2017).

During the wargame in support of military decision-making, the following aspects are considered: own positions, strengths and weaknesses of the threat(s), opportunities, developed own possibilities, and the environmental (f)actors. The wargame is often based on knowledge of doctrine, tactical insight, experience, and intuition. The wargame consists of an iterative action, reaction, and counter-action process. It emphasizes critical tasks and provides insight into tactical possibilities that are otherwise difficult to highlight. Carrying out a wargame is the most valuable step in analyzing one's capabilities, and sufficient time should be set aside for this. Support with modern simulation systems can reduce the duration of the analysis and bring in more objectivity (Royal Netherlands Army, 2012).

1.3. The Value of Military Simulation

Using information technology to support military decision-making has been ongoing for years. The Modeling & Simulation (M&S)domain has numerous solutions and systems for decision-making as simulations enable better situational understanding and the possibility of simulating the effects of possible interventions. In addition, M&S can provide a commander with a broad range of relevant information, such as risks and opportunities of possible Courses of Action (COAs) (TNO, 2021).

The military simulation domain covers various activities, ranging from full-scale field exercises to abstract computational models with little or no human involvement (see Figure 1.2). At present, simulation is an indispensable technology for the modern armed forces. Any form of military training can be considered a ”simulation” in the strict sense of the word. Many, if not most, of the simulated exercises aimed at teaching participants the required military skills and drills to perform their duties. To put it forthrightly, simulations create an operational environment in which the training is executed.

On the other hand, computer simulation and analytical models as means for wargaming or concept development remain largely unexplored territory (Bruvoll et al., 2015; Ministry of Defence, 2021). Military simulations are divided into Live, Virtual and Constructive (LVC) according to three criteria: people, systems, and the operation. A simulation in which real people operate the real-world system is called live simulation (e.g., a pilot flying a fighter). If real people use simulated systems in a simulated oper-

---

[2] No plan survives first contact with the enemy.

ation, it is called virtual simulation (e.g., a pilot flying a flight simulator). If humans are not involved in a simulated operation, it is a constructive simulation (e.g., computer-generated forces) (Page & Smith, 1998; Park, 2015; Tolk, 2012). Within this classification of LVC, constructive simulation is the primary tool to support wargaming (Turnitsa et al., 2022).

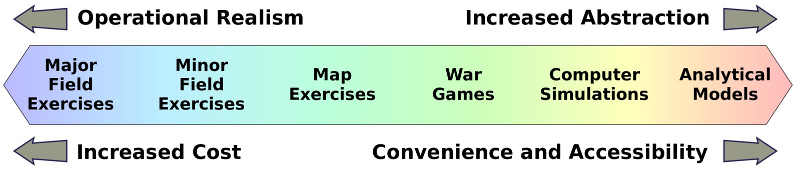


Figure 1.2: Graphic depicting the spectrum of military simulations based on "Modeling and Simulation of Land Combat" by J. G. Taylor (1983), illustration available from Wikimedia Commons (2008).

The time it takes to become skilled at specific tasks is an inherent problem for complex tasks, such as tactical decision-making. Regular field exercises are usually insufficient for tactical decision-making because they are not aimed at developing decision-making skills. These skills are developed by wargaming, where a wargame is "a representation of conflict or competition in which people make decisions and respond to the consequences of those decisions" (U.S. Department of Defense, 2020). Furthermore, wargaming is based upon human judgment, while (computational) analysis is based on mathematical processes. Both are powerful and compatible, yet they are not different expressions of the same phenomenon. Wargames explore the interlocking coherence of the whole, while computational analysis produces precision in isolation (Turnitsa et al., 2022). Simulations can speed up this wargaming process for skill development by enabling extensive experience through large amounts of training and conscious experiences (Hannay et al., 2015).

There is a shift from monolithic simulation systems into component-based architectures to facilitate innovation, flexibility, and cost-effectiveness. This shift will provide opportunities for reusing these components across different simulation systems (Ford, 2021). Furthermore, achieving interoperability between requisite simulation systems and ensuring the credibility and consistency of results still requires large expenditures concerning time, personnel, and budget, whereby it should be noted that more computing power without a more comprehensive understanding of war would produce the wrong answer faster with more persuasive graphics (Caffrey, 2000). Which is beautifully noted by Roman (2005): "Garbage in, Hollywood out."

1.4. The Future of Military Modeling & Simulation

There is a transition from extensive monolithic simulation systems into component-based architectures to facilitate innovation, flexibility, and cost-effectiveness (Van Den Berg, 2021a). Modeling & Simulation as a Service (MSaaS) is developed and maintained under the umbrella of the NATO Modelling & Simulation Group and defines the architecture building blocks that should be considered for the realization of an MSaaS capability (Van Den Berg, 2021b). MSaaS is an Enterprise-level architecture that promotes modularity, loose coupling, agility, and reusability of Modeling and Simulation components from different suppliers by making them available on-demand to a large number of disparate users in order to reduce the cost and time for implementing M&S capabilities to improve operational effectiveness (Ford, 2021; Patel, 2021).

While reusability and composability are essential for military simulation, it has been challenging to achieve in the M&S field. Reuse provides many technical, business, and economic benefits (Patel, 2021). Composability has been increasingly crucial for M&S of a system of systems in which dissimilar systems are coupled (Balci et al., 2011). However, horizontal and vertical coupling of components is complex as models may have different resolutions and levels of abstraction, making its composability fundamentally challenging (Page & Opper, 1999). High-resolution models can be used to understand phenomena qualitatively. Obtaining such an understanding, it may be desirable to use a low-resolution model consistent with the high-resolution understanding to comprehend what is going on analytically, where resolution is a relative matter. Thus, just as theater-level analysis may require dipping into corps-level analysis selectively, so also corps-level analysis may require dipping into division-level analysis selectively, and so on. The problem is inherently hierarchical, and one person's high resolution is another one's low resolution. However, at every level, there are lower and higher resolution views (P. K. Davis, 1993).

The current solution to simulate these different levels of abstraction is the creation of aggregate-level simulations that use models designed to represent military units that consist of multiple entities, such as platoons, companies, or brigades. On the other hand, entity-level simulations use models that represent singular entities, such as a soldier, vehicle, or weapon platform. Where both types of simulation have their purposes, although progress is being made, it remains challenging to compose both types of simulations within military simulation systems (Lee et al., 2020; Rabelo et al., 2015).

1.5. Research Questions

The identified challenges in the above section provide the knowledge gap that is the starting point for the main research question this research will answer.

Main Research Question

What are design principles for composable elements that can be used in a simulation system for military wargames?

The identified design principles are essential as they guide a designer when creating a wargame simulation system. Principles have a generic nature and provide insight into the system they apply. The principles are not only limited to the specific design made in this research; this general nature makes it possible to apply them to the design of wargame simulation systems. The main research question emphasizes the importance of composable simulation elements for wargaming simulation systems. The main research question is divided into four sub-questions to structure the research and generate more tangible answers.

Sub-Question 1

What design principles for composable simulation elements can be derived from the foundations of military wargaming?

The *goal* of the first sub-question is to determine which elements are required for military wargaming by a review of relevant literature and documents. The answer to this question helps to understand the relevant user needs that shape the design principles derived from the nature of military wargaming, existing military simulation standards, and conflict ontologies. The *deliverable* of this sub-question is a first overview of possible design principles that are derived from military perspectives on wargaming.

Sub-Question 2

What design principles for composable simulation elements can be derived from contemporary challenges within military wargaming simulation?

The *goal* of the second sub-question follows the first sub-question and analyzes scientific literature on Modeling & Simulation, which concepts are fundamentally challenging for military simulation. The answer to this question helps to understand the design principles derived from the identified challenges. This sub-question's *deliverable* adds principles from an M&S perspective to the overview of design principles already identified by the first sub-question.

Sub-Question 3

Based on the identified design principles, how can a wargame simulation system using composable simulation elements be designed?

The *goal* of the third sub-question builds upon the two previous sub-questions. It applies the identified design principles to create a military wargaming system that uses composable simulation elements. The answer to this question helps to understand how a military wargame system can be designed and how design principles subsequently shape the envisioned design. The *deliverable* of this sub-question is a conceptual design for a military wargaming system.

Sub-Question 4

To what extent is it possible to use the design principles to create a prototype of a composable wargame simulation system?

The *goal* of the fourth sub-question is to verify, validate, and analyze the designed prototype for the wargame simulation system. The answer to this question uses a real-world wargame scenario as input for a proof of concept that evaluates the envisioned wargame simulation system. The *deliverable* of this sub-question is an evaluation of the design principles for composable simulation elements based on the proof of concept where the outlined real-world scenario is used to discuss the created wargame simulation system.

The following section will outline the research approach, which explains which research and analysis methods are used to answer the above research questions. The research approach, in addition to that, provides direction to the execution of this research.

1.6. Research Approach

The military application of wargames is a comprehensive system of military personnel that takes part in the form of a game, which is driven by organizational processes - the military decision-making process - and all sorts of technical systems that support the decision-making process. In this complex environment, the envisioned wargame simulation system has to meet the defined requirements, which are 'proved' in this research by demonstrating the conceptual design. The proof of concept results may lead to improvements in the prototype design of the wargame simulation system. The goal of Design Science Research (DSR) (see Figure 1.3) is to generate prescriptive knowledge about the design of Information System artifacts like software, methods, models, and concepts (Hevner et al., 2004). Subsequently, the DSR approach provides an appropriate approach to answer the main research question.

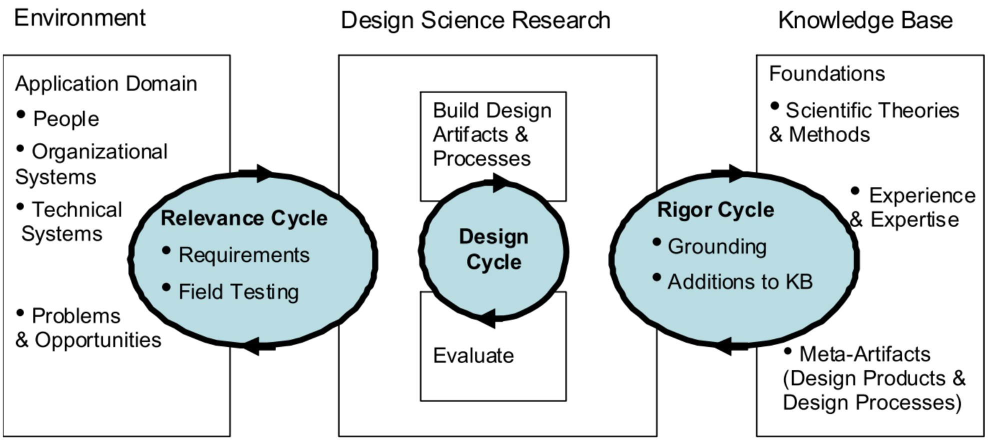
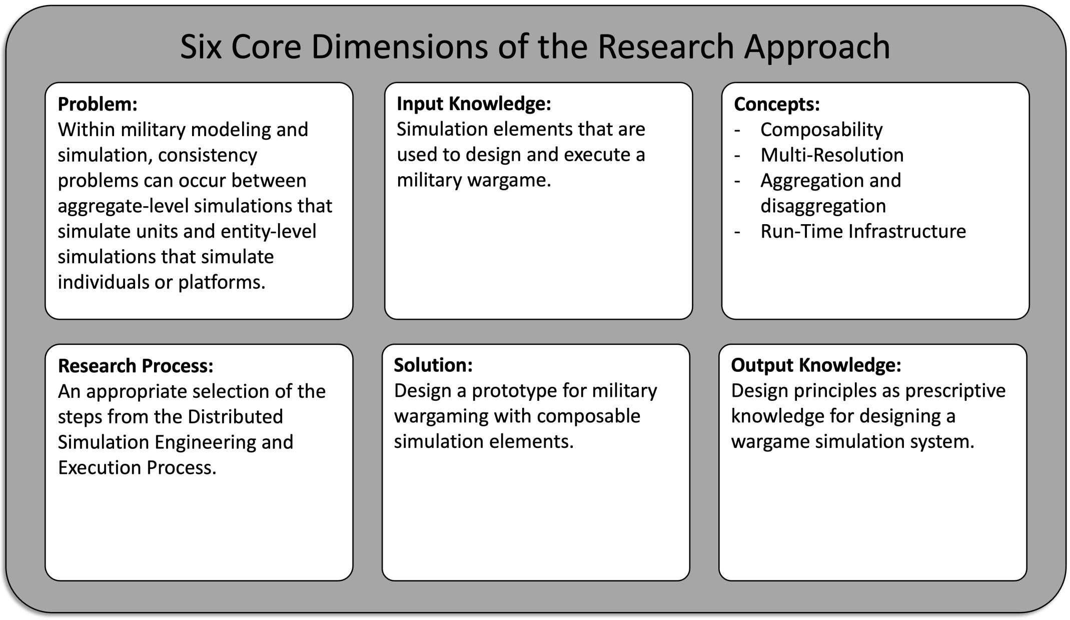


Figure 1.3: Design Science Research (Hevner, 2014).

In the broader sense, DSR intends to generate design knowledge, that is, knowledge about innovative solutions to real-world problems (Brocke & Maedche, 2019). The DSR Cycles (see Figure 1.3) display a more detailed overview of the research method for this research. The research method links the sub-questions to the collected and analyzed data and determines which deliverable(s) are required for the theoretical fundament and systematization within the research approach.

The fundamental principle of design-science research is that knowledge and understanding of a design problem and its solution are acquired by building and applying an IT artifact. Whereby artifacts constructed in DSR are rarely full-grown information systems used in practice. Instead, artifacts are innovations that define the ideas, practices, technical capabilities, and products through which information system analysis, design, implementation, and use can be effectively and efficiently accomplished (Hevner et al., 2004). DSR as a framework for applied sciences provides an applicable approach that fits with the complex nature of the problem addressed in this research. The design approach not only focuses on the technical artifact, but the application domain identifies problems as well as opportunities within the people, organizations, and technical systems, often addressed as Socio-Technical Systems (Bots et al., 2017).

The DSR approach (see Figure 1.3) focuses on three inherent research cycles (Hevner, 2014). The Relevance Cycle bridges the contextual wargame environment of the research project with the design science activities that design and evaluate the envisioned wargame simulation system. The Rigor Cycle connects the design science activities with the knowledge base of scientific foundations, experience, and expertise on military simulation that informs the research project. The central Design Cycle iterates between the core activities of building and evaluating the wargame simulation system and the Distributed Simulation Engineering and Execution Process (DSEEP) research process used in this research. Brocke and Maedche identifies six core dimensions of a DSR project that form a single overview of the framework that describes the project. Figure 1.4 provides an overview of the six core dimensions identified for this research.



Six Core Dimensions of the Research Approach

Problem:
Within military modeling and simulation, consistency problems can occur between aggregate-level simulations that simulate units and entity-level simulations that simulate individuals or platforms.

Input Knowledge:
Simulation elements that are used to design and execute a military wargame.

Concepts:
- Composability
- Multi-Resolution
- Aggregation and disaggregation
- Run-Time Infrastructure

Research Process:
An appropriate selection of the steps from the Distributed Simulation Engineering and Execution Process.

Solution:
Design a prototype for military wargaming with composable simulation elements.

Output Knowledge:
Design principles as prescriptive knowledge for designing a wargame simulation system.

Figure 1.4: The six dimensions that form the core of the design of the wargame simulation system.

This focus of the DSR approach on innovative design makes it an appropriate approach for this research, as the main premesis underlying the main research question of this research is to design from a *greenfield* situation a new concept for a simulation system for military wargaming. This new conceptual design proposes a possible solution to overcome the limitations of contemporary monolithic military wargaming simulation environments.

1.7. Structure

The research method links the formulated sub-questions to the six dimensions. The main research question is associated with the problem dimension: what is the problem for which the design must identify a possible solution? The first sub-question is related to the input knowledge dimension. This question, discussed in the second chapter, formulates an answer for which prior knowledge about wargaming will be used for the design. The second question is connected to the concept dimension. This question, discussed in the third chapter, formulates an answer which modeling and simulation concepts are fundamentally challenging for the design of a wargame simulation system. The process dimension describes the essential activities planned (or conducted) to create the design for the simulation system. The Distributed Simulation Engineering and Execution Process (DSEEP) that is selected for the design of the decision-making support system is described in the fourth chapter. The third sub-question, discussed in the fifth chapter, is directed at the solution dimension for the prototype design of the wargame simulation system. The answer to the third question consists of a use case that provides direction for the solution space for the envisioned design of the wargame simulation system. The fourth sub-question, answered by the sixth chapter, is related to the dimension of output knowledge. The output not only consists of the details of the designed attribute but also creates knowledge for the military environment wherein the simulation system can be used. The final chapter closes with the conclusion, reflection, and recommendations for further developing the envisioned wargame simulation system.

2

Understanding Military Wargaming

”All models are wrong, but some are useful.”
George Box

Wargaming is a distinct and historically significant tool that warriors have used over the centuries to help them understand war in general and the nature of specific upcoming operations (Rubel, 2006). Unlike most other professionals, military personnel can only practice their profession in times of war. The armed forces have therefore supported the development of methods and techniques that will enable military personnel to train their profession in times of peace (see Figure 1.2). A commonly used technique to provide military personnel with decision-making experience and information based on the simulation of war is known as wargaming (McHugh, 1966).

Suppose one accepts that wargaming is inherently a research tool and generates potentially valuable knowledge and experiences. In that case, the question is, ”Under what conditions, or for what problems, can it have validity?” Hanley (1991) describes war gaming as a weakly structured tool appropriate to weakly structured problems. Soldiers play wargames because they must. There are specific warfare problems that only gaming will illuminate. That knowledge is not in the form of a solution to an engineering problem. It is commonly said that wargames produce insights, not proofs (Rubel, 2006). In its most basic form, wargames are essentially simulations of reality and simplified representations of a potential future conflict situation. Simulation of war is less closely coupled to its parent phenomenon because of the high degree of structural indeterminacy involved. In a broad sense, simulation is the attempt to represent reality to the degree necessary to explore the warfare phenomena in which the decision-makers are interested (Perla & McGrady, 2011; Rubel, 2006).

This chapter uses different approaches and perspectives to identify and describe the wargaming elements of interest. To gain insights from the different perspectives but to prevent the creation of a comprehensive ontology about wargaming, the following overview is limited to three perspectives or approaches, all of which have a different purpose and are from different domains. The first section uses a military perspective to introduce the nature of combat modeling and military simulation. The second section identifies the main elements used in military simulation standards based on modeling languages used in systems engineering. The third section uses an ontological approach to look for elements in wargaming and how these elements are used to form the basic structure of wargaming. The last section synthesizes the theoretical insights from the previous sections into an overview of identified elements and their definitions, usable for military wargaming.

2.1. The Nature of Wargaming

According to Perla (1990), a war game is ”a warfare model or simulation whose operation does not involve the activities of actual military forces, and whose sequence of events affects and is, in turn, affected by the decisions made by players representing the opposing sides. The essence of war gaming is the examination of selected aspects of a conflict in an artificial environment (McHugh, 1966). Such artificial environments include, amongst others, a sandbox COA analysis, a tabletop exercise, or a digital simulation. Combat simulations can be categorized by the level[1] of war, using probabilistic processes, and the amount of human interaction required by the simulation represents.

---

[1] The three Levels of War, strategic, operational, and tactical, link tactical actions to the achievement of nation-states’ objectives (Turnitsa et al., 2022). There are no finite limits or boundaries between these levels, but they help commanders design and synchronize operations, allocate resources, and assign tasks to the appropriate command. Employment’s strategic, operational, or tactical purpose depends on the nature of the objective, mission, or task (U.S. Department of Defense, 2017). Additionally, in the Dutch doctrine, the Royal Netherlands Army (2014) identifies the technical level as the lowest level of warfare. This level is particularly about operating military systems and the execution of Tactics, Techniques, and Procedures.

In addition to the above categorization by the levels of war, a war game can have different purposes. Wargames are not played for pleasure; they are conducted for military purposes. McHugh formulates two general purposes for wargaming. When used in the educational domain, the insights from the wargame can provide decision-making experience for military personnel. When applied in the analytical domain, the insights from the wargame can provide information to support decision-making by military commanders (McHugh, 1966). As with every game, a wargame needs predetermined rules on how the game needs to be played. It needs data, for example, about the available ammunition or the probability that a specific weapon type can destroy a target.

Furthermore, the game requires adjudication to determine whether an attacked entity is damaged (or wounded in the case of a human ”system”) or destroyed. This adjudication can be probabilistic, like a dice roll in a Risk boardgame, rule-based like in chess, or like in the traditional ”Kriegspiel” based on the interpretation and experience of military personnel in the role of evaluator or umpire (Turnitsa et al., 2022). The relation between a wargame's nature, rules, and purpose is depicted in Figure 2.1.

The following two paragraphs will further elaborate on the nature of wargaming by exploring military decision-making and combat, which are overarching concepts. This section concludes with an overview of the components identified by military decision-making and combat.

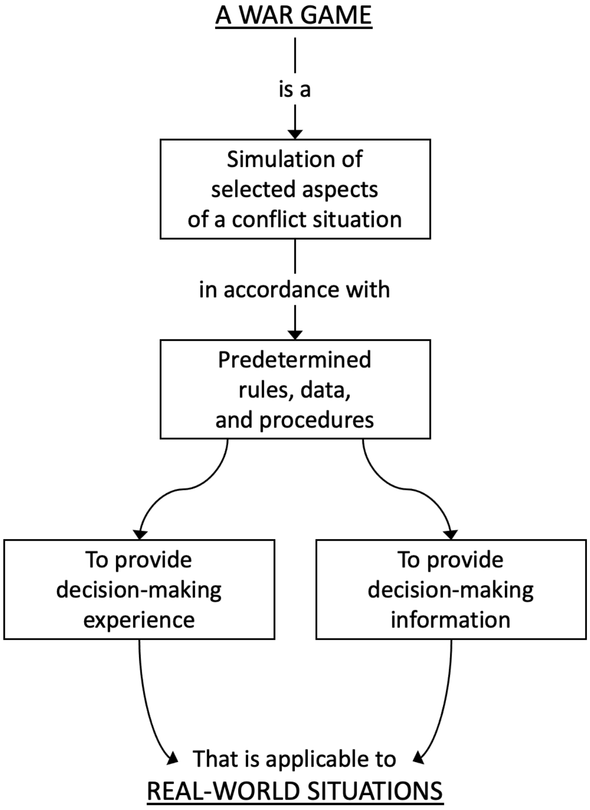


Figure 2.1: Graphic depicting the relationships between the definition and the general purposes of wargames derived from *Fundamentals of war gaming* by McHugh (1966).

2.1.1. Military Decision-Making in the Royal Netherlands Army

Military decision-making combines conceptual and detailed aspects of planning. It is used to create plans and orders for large, long-term, and short-term operations within the framework of an operational plan. Furthermore, it supports a commander in understanding the operational environment, identifying and understanding the problem and determining the desired outcome, and developing and choosing an appropriate COA. Military decision-making results in a comprehensive plan for the execution of the operation (Royal Netherlands Army, 2014).

Table 2.1: Basic Military Problem Solving Questions (5W paradigm).

Element	Description
Who	Who is an information component identifying the battle-space object, that is directed to perform an action (as defined in an order), that has been requested to perform an action (as identified in a request), that has been observed or has performed an action (as described in a report), or on which an action is to be performed (e.g., target).
What	What is an information component identifying action(s) to be performed, effects to be delivered (order or request), or action(s) and effect(s) that have occurred (report).
When	When is an information component describing the time frame in which action(s), effect(s) is to occur (order or request), or when action(s), effect(s), or event(s) has occurred (report).
Where	Where is an information component providing the location of an object in the battle-space, the location where action(s) or effect(s) is to occur (order or request), or the location where action(s), effect(s) or event(s) has occurred (report).
Why	Why is an information component describing the rationale or purpose of action(s) to be performed or effect(s) to be delivered (order or request), or the desired end state of such action(s) or effect(s).

The 5W paradigm (see Table 2.1) is used for decision-making at the lowest service levels or when time is short. On the other hand, more extensive processes are used by higher echelons or in more demanding environments. Within the Royal Netherlands Army, the Tactical Decision-Making Model (TBM) (see Figure 2.2) is used for tactical decision-making. This model consists of three stages: understand, assess, and decide. Each stage consists of several steps by the commander and his staff to analyze and decide about the operation (a more elaborate explanation of the process is included in Annex A). This Dutch decision-making process is based on corresponding processes within NATO. Other countries have corresponding processes for tactical decision-making, for example, the US Army's Military Decision-Making Process.

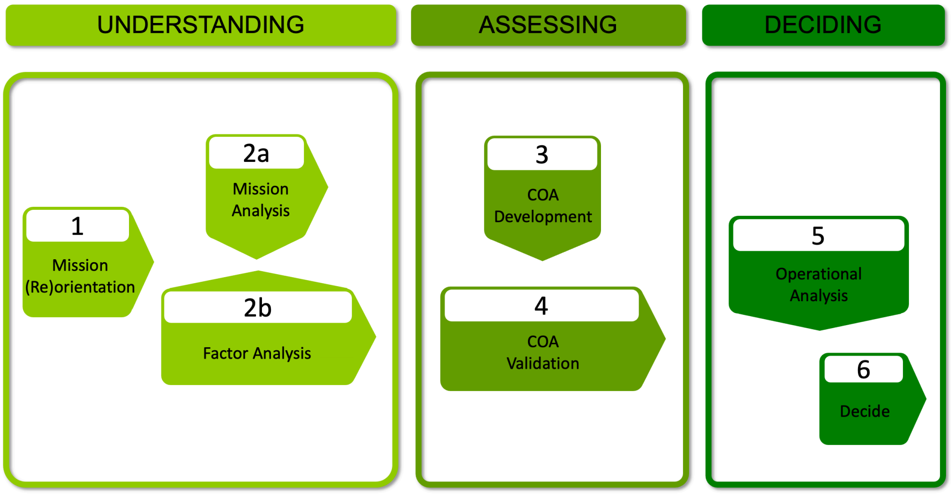


Figure 2.2: Royal Netherlands Army Army Tactical Decision-Making Model (IJntema & van de Haar, 2010).

Decision-making does not begin until an order is received. Nor does it end when a decision is formalized. A military organization constantly monitors the environment in which it operates. This monitoring builds and maintains so-called 'general awareness.' Based on this awareness, the purpose of the mission is determined or adjusted. Usually, the ends are determined by the next higher military level, and orders are given to the levels below. In some cases, however, the mission can also be self-formulated.

To this end, a commander and staff must understand the specific aspects of the environment in which the mission must be executed. The commander and military staff must then convert general awareness into situational awareness. This stage is called 'understanding.' Subsequently, based on the current operational picture, the commander and staff have examined how - COAs - the mission could be carried out. This stage is called ’assessing.’ Only then can a well-considered decision be made about the final choice with the highest chance of success. This stage is called ’deciding.’

The staff still has to carry out many additional planning activities in many cases. Answering the still open questions (’request for information’) and alternative plans (’contingencies’) is, in any case, part of this. Furthermore, the commander will, of course, also have to communicate the decision to his subordinate commanders. While issuing the orders, the commander explains his decision and conveys his intention for the mission to the subordinate commanders (IJntema & van de Haar, 2010). Table A.1 provides an overview of the components resulting from the Royal Netherlands Army Tactical Decision-Making Model.

2.1.2. Military Combat Components

Components of military combat are defined by DuBois et al. (1997) as the set of all things, in the most fundamental sense, that can be said to make up the phenomenon of combat: the constituents that exist in all combat actions. Their top-down approach examines components from the point of view of any political entity set out to provide the means to wage combat. The authors use a top-down to examine components from the (strategical) point of view of a political entity setting out means to wage combat. On the other hand, the micro-derived (tactical) components from the bottom-up approach must mesh with the macro-derived components from the top-down approach. The combination of the top-down and bottom-up approaches is depicted by DuBois et al. (1997) in Figure 2.3 to show the relationship between the identified combat components.

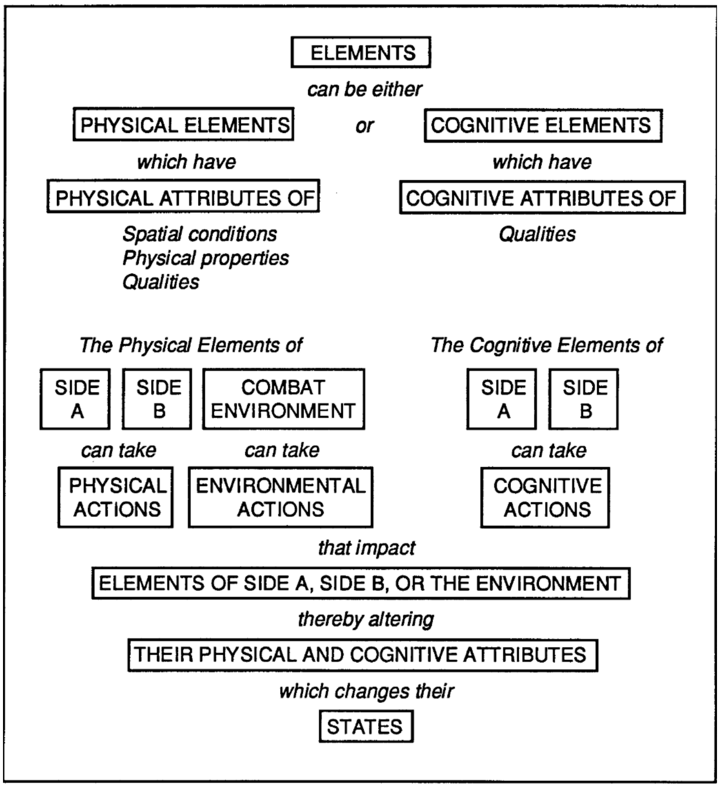


Figure 2.3: Graphic depicting the relationships between combat components derived from “A concise theory of combat” by DuBois et al. (1997).

An important observation in addition to the identified elements in the previous paragraph is that the identified elements by DuBois et al. can exist in the physical domain, for example, a tank or a group of soldiers, or it can be a cognitive element, like doctrine for land forces or the mental health of the soldiers. The attributes of physical elements are generally not very difficult to establish and use, while the attributes of cognitive elements are much harder to determine and conceive. Another critical observation of the identified elements is that the impartial combat environment elements complemented two (or more) sides within the conflict environment (only in the physical domain). The following important observation that can be made from the model is that the elements can perform both physical and cognitive actions. Subsequently, these actions may impact the attributes of the elements, resulting in a change in the state of the elements.

2.1.3. Overview of Wargaming Elements

The above overviews with elements that are part of the 5W model (see Table 2.1), the outcomes of the TBM decision-making model (see Annex A), and the military combat components (see Figure 2.3) show that there are both substantive and process-related components. The active components are represented by the question of who plays a role or which actors are essential. Furthermore, the models clearly show that actors can execute tasks or activities (actions) to change their environment to achieve a particular goal by performing activities (what must be achieved at a specific time and place). These substantive components - actors, actions, and the environment - that emerge from these models provide a first glance at the fundamental elements of a wargame. On the other hand, the process-related components from the TBM, such as Guidance, Commander's Critical Information Requirements, the Decision Support Overlay, or the Operation Order, provide insight into what is conditional for and which products are created and used by military personnel during decision-making processes. Before an overview of wargame elements can be created based on the above-identified components from the decision-making models, the following paragraph first explores the elements and activities of organized combat. An important notion that can be concluded from the model by DuBois et al. is that the physical and cognitive domains should be modeled. The following elements can be deduced from the identified decision-making and combat components.

- **Objective**: An element describing the rationale or purpose of an action to be performed or effects to be delivered, or the desired end state of such action or effects.
- **Unit**: An element identifying a military individual, group, or entity that is directed to act, that has been requested to act, that has acted, or on which an action is to be performed (e.g., target).
- **Task**: (A composite of related) activities a unit performs for an immediate purpose.
- **Environment**: All factors (including climate, weather, and terrain) that can influence the operation and the activities of all other relevant actors.
- **Attributes**: Qualitative or quantitative modifiers of simulation elements.
- **Adjudication**: Rules and procedures describe how the wargame should be executed, and they shape the decision-making process. Furthermore, they dictate the boundaries for activities within an operation.

The primary audience for military decision-making models are military commanders and their staffs. While wargaming and simulation may be part of these models, decision-making models were not designed for the modeling and simulation community or intended to be used by software developers. Another notion is that wargaming has two different, but not mutually exclusive, objectives. On the one hand, to educate and train military personnel in decision-making processes and, on the other hand, provide military commanders with decision-making information in training or during actual deployments of military forces. The following section analyzes military simulation standards to identify possible elements used to simulate wargames.

2.2. Miltary Simulation Standards

The Command and Control Systems – Simulation Systems Interoperation (C2SIM) standard is an approved SISO international standard for the exchange of plans, orders, and reports across Command and Control (C2) systems, simulation systems, and Robotic and Autonomous Systems (SISO, 2023). The C2SIM standard contains among others:

- The Military Scenario Definition Language (MSDL)[2] (SISO, 2015); and
- The Coalition Battle Management Language (C-BML)[3] (SISO, 2013).

---

[2] SISO-STD-007-2008

[3] SISO-STD-011-2014

The goal of C2SIM [standard] is to develop a mechanism by which the connection between C2 and simulation systems becomes (semi-)automated, unambiguous, easy to sustain with low cost, applications independent, expandable, and usable in theater by different levels of command (NATO STO, n.d.). This section reviews in the subsequent paragraphs the MSDL standard and the C-BML standard. The last paragraph of this section concludes with an overview of the components identified from military simulation standards.

2.2.1. Military Scenario Definition Language

The MSDL is a standard for specifying force structures, environment, and other information to initialize simulation systems. The standard provides a common mechanism for validating and loading military scenarios, to promote sharing of scenario files across simulation and C2 systems, and to improve scenario consistency among federated simulations (Blais, 2008). The MSDL is an XML based language designed to support military scenario development that provides the modeling and simulation community with (SISO, 2015):

- A common mechanism for verifying and loading military scenarios;
- The ability to create a military scenario that can be shared between simulations and C4I[4] devices;
- A way to improve scenario consistency between federated simulations;
- The ability to reuse military scenarios as scenario descriptions are standardized throughout the Army, Joint, and international communities and across simulation domains, e.g., training exercise, analysis.

MSDL is defined using an XML schema that allows for format and content verification. Compliance with the MSDL XML schema defined in this specification will permit simulations to generate military scenarios where each scenario consists of either the initial state or a state snapshot. This standard aims to provide the mechanism that permits simulations to utilize the MSDL schema to develop and reuse military scenarios across MSDL compliant simulations and scenario generation tools. The top-level structure of MSDL XML contains the following elements (Blais, 2008; SISO, 2015):

- **ScenarioID**: Provides identification of the scenario and its purpose.
- **Options**: Provides global parameters about the scenario and its content.
- **environment**: Describes the simulated physical environment in which the execution is to occur (e.g., area of interest, weather, time).
- **ForceSides**: Describes the structure of the forces and sides involved in the execution.
- **Organizations**: Describes the structure of the units and equipment involved in the execution.
- **Overlays**: Describes the logical overlays used to group the intelligence elements/instances in the scenario. Ownership of a specific overlay is determined through the intelligence elements/instances contained in that overlay.
- **Installations**: Describes the detected installations as determined by the intelligence-gathering process of each force, side, or as determined by individual units.
- **TacticalGraphics**: Describes the tactical information as known by a particular force, side, or unit individually.
- **MOOTWGraphics**: Describes the detected MOOTWGraphics[5][6] instances are determined by the intelligence-gathering process by each force, side, or unit.

The MSDL is an approved standard for specifying force structures, environment, and other information to initialize simulation systems. The MSDL description of the scenario is expressed as an Extensible Markup Language (XML) file conforming to an XML schema described and provided in the SISO specification. The MSDL XML schema defines one global element, the MilitaryScenario root element. All other constructs in the language are defined as global types, either complex or simple,

---

[4] Command, Control, Communications, Computers, Intelligence

[5] Military Operations Other Than War (MOOTW), also referred to as Stability, Security, Transition, and Reconstruction Operations (Haight, 2007). Examples are Humanitarian Assistance and Disaster Relief or Peace Operations.

[6] The MSDL specification dates from a period in which NATO forces were mainly deployed for Stability, Security, Transition, and Reconstruction operations (NATO, 2022). Due to the war between Ukraine and Russia, contemporary international security requires a renewed focus on conventional combat. This type of operation is covered by the TacticalGraphics element.

to maximize the reuse of the definitions in creating other XML languages (Blais, 2008). The following paragraph will address the Coalition Battle Management Language, the specification within the C2SIM that represents a standard for digitizing a commander's intent.

2.2.2. Coalition Battle Management Language

The C-BML is a standard language for expressing and exchanging plans, orders, requests, and reports across Command and Control (C2) systems, live, virtual, and constructive modeling and simulation systems, and autonomous systems participating in Coalition operations. C-BML is closely related to MSDL, which is used to identify forces and scenario settings that can be used in C-BML expressions of plans, orders, requests, and reports. JC3IEDM or Joint Consultation, Command and Control Information Exchange Data Model is the central reference model for the initial specification of C-BML.

In military operations, the format and content of the information to be exchanged are determined by the doctrine governing the exchange. Forces conducting designated operations are required by doctrine to provide certain information. The purpose of this standardization is to specify requirements for the generation and transfer of this information exchange. The C-BML standard comprises two elements: the C-BML data model and the information exchange structure and content specification. Together, these describe the lexical building blocks for the construction of C-BML expressions (SISO, 2013). The conceptual employment of the C-BML standard is displayed in Figure 2.4. The C-BML standard contains the language's fundamental concepts and building blocks and is intended to be used by software developers.

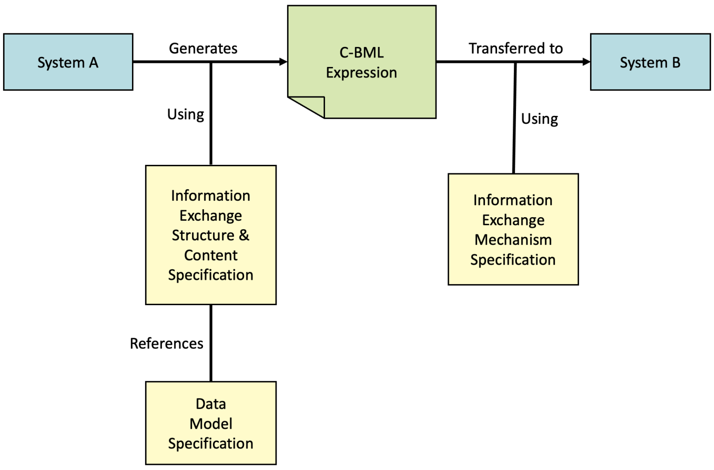


Figure 2.4: Process diagram depicting the conceptual employment of the C-BML standard derived from "Standard for Coalition Battle Management Language (C-BML) Phase 1" by SISO (2013).

Three perspectives formalize the C-BML standard by the initial data model (representation view), the initial information exchange content (protocols view), and an initial structure (an abstraction of the doctrine view):

- The **representation view** structures and relates the terms defined in the doctrine in a way that can result in the description of the information contained in missions, tasks, requests, and reports. Relevant representations include conceptual, logical, or physical data models or fully formalized ontologies.
- **Protocols** standardize the rules by which information is transported from the source system to the target system. A formalism for describing the content and structure of C-BML expressions is the XML Schema language. This language provides a precise description of the information structure and content that can be used to validate (in the XML sense) XML documents containing C-BML expressions encoded in XML.
- Every term within the language must be unambiguously defined and rooted in military **doctrine**. This definition is conveyed in C-BML by generic information elements from which doctrine-specific expressions can be constructed. The 5Ws (see Table 2.1) are the principal information compo-

nents of C-BML. These information components are fundamental to expressing orders, requests, and reports and are an abstraction of the doctrine view.

Basic information in the representation view includes OBJECT-TYPE, OBJECT-ITEM, CAPABILITY, and ACTION. Entities within the operational environments are classified by their OBJECT-TYPE rather than being classified by individually identifiable items. The use of OBJECT-ITEM identifies specific instances of an object type. General attributes, e.g., an element's CAPABILITY, are registered on the OBJECT-TYPE side, while the elements-specific values are registered on the OBJECT-ITEM side. For example, the weapon's characteristics are specified within the OBJECT-TYPE, while the actual ammunition state and location of a specific weapon are specified within the OBJECT-ITEM. An entity in the operational environment represents a discrete object in the structure of the data model. Entities may have semantic links or RELATIONSHIPS to other entities in order to represent a more complex idea than can be represented by a single entity. Entities and relationships can have ATTRIBUTES, data elements that describe physical characteristics for OBJECT-TYPE entities, or index values for relationships.

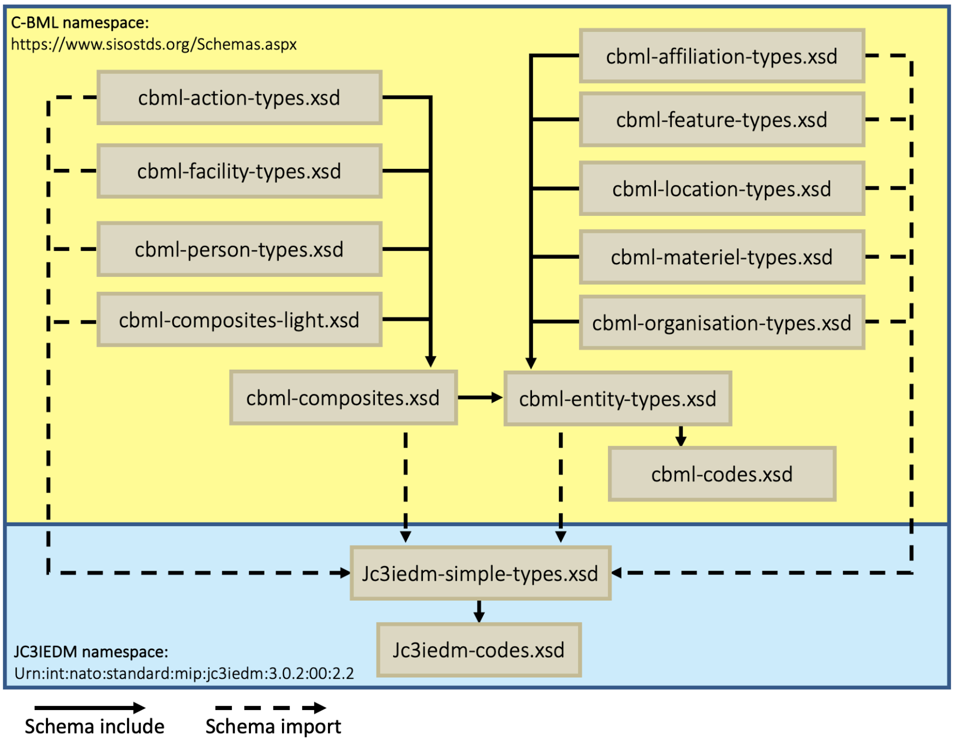


Figure 2.5: Process diagram depicting the C-BML XML schema file dependencies derived from "Standard for Coalition Battle Management Language (C-BML) Phase 1" by SISO (2013).

These fundamental language components construct C-BML expressions. A C-BML expression is an order, request, or report constructed following the C-BML information exchange content and structure specification or constructed by a content and structure specification that uses constructs from the C-BML information exchange content and structure specification. In the expression of orders, requests, and reports, each "W" has a usage particular to that expression. For example, in an order, a "Who" may identify the authority giving the order, while another "Who" identifies the organization that will carry out the order.

Data types and structures in XML schemas can reference data types and structures declared in other XML schemas. In this way, new schemas can reuse data types and structures defined in other schemas. Similarly, schemas for a particular application, such as C-BML, can be constructed modularly, organizing data types and structures to facilitate schema maintenance and use within or by other applications. Figure 2.5 summarizes the XML schema dependency relationships and identifies the used include and import statements in the XML schema files. Furthermore, the purpose of each XML schema file is described in Table 2.2[7][8].

Table 2.2: C-BML XML Schema Files (SISO, 2013).

XML schema file	Purpose
jc3iedm-codes.xsd	For referential convenience, this schema file declares enumeration data types extracted from the JC3IEDM XML schemas and provided in the C-BML distribution files.
jc3iedm-simple-types.xsd	For referential convenience, this schema file declares simple types (string, numeric, etc.) extracted from the JC3IEDM XML schemas and provided in the C-BML distribution files.
cbml-codes.xsd	This schema file declares enumeration data types used in declarations in other C-BML schema files.
cbml-entity-types.xsd	This schema file declares data types relating to fundamental entities (in JC3IEDM parlance) and relationships across entities used in declarations in other C-BML schema files.
cbml-composites.xsd	This schema file declares data types and named information elements (XML structures) relating to the C-BML 5Ws (who, what, when, where, why) and their variations in usage in expression of orders, requests, and reports.
cbml-composites-light.xsd	This schema file declares named information elements (XML structures) for constructing simplified C-BML expressions relating to description of tasks used in orders, requests, and reports. The “light” constructs declared in this schema file and in the cbml-composites.xsd schema file are intended to provide a simple set of information elements, considered adequate for many purposes, that limits the use of XML abstract types in favor of XML “choice” structures
cbml-action-types.xsd	This schema file declares data types and structures relating to actions and events.
cbml-affiliation-types.xsd	This schema file declares data types and structures relating to affiliation information (e.g., geopolitical, ethnic, functional, religion, other).
cbml-facility-types.xsd	This schema file declares data types relating to objects such as airfields, bridges, military obstacles, and roads.
cbml-feature-types.xsd	This schema file declares data types relating to objects such as geographic features (terrain characteristics to which military significance is attached), meteorological features (e.g., cloud cover, precipitation), and control features (e.g., route, airspace control means).
cbml-location-types.xsd	This schema file declares data types relating to positional information, including use of relative positioning and geometric shapes.
cbml-materiel-types.xsd	This schema file declares data types relating to items of materiel, such as aircraft, ammunition, vehicles, and weapons.
cbml-organisation-types.xsd	This schema file declares data types relating to organizations, such as military units, convoys, military posts, and task formations.
cbml-person-types.xsd	This schema file declares data types relating to individual persons or classes of persons.

2.2.3. Overview of Standardized Simulation Elements

MSDL as well C-BML are based on the Joint Consultation, Command and Control Information Exchange Data Model (JC3IEDM), which specifies the minimum set of data that needs to be exchanged in coalition or multinational operations. It is intended to represent the core of the data identified for exchange across multiple functional areas and views of the requirements. JC3IEDM describes all objects of interest on the battlefield, e.g., organizations, persons, equipment, facilities, geographic features, weather phenomena, and military control measures using a standard and extensible data modeling approach. It consists of five key information concepts (see Table 2.3) that are of fundamental importance in generating the structure of the data model (Tolk, 2012).

Table 2.3: The Five Core Concepts of JC3IEDM (Tolk, 2012).

Concept	Definition
OBJECT-ITEM	An individually identified object that has military significance. Examples are a specific person or unit, a specific military system, a specific geographic feature or a specific coordination measure.

Table continues on next page.

---

[7] The file extension “.xsd” is a convention used to identify XML schema files.

[8] All schema files in the C-BML namespace import the jc3iedm-simple-types.xsd schema file.

Table 2.3 Continued: The Five Core Concepts of JC3IEDM.

Concept	Definition
OBJECT-TYPE	An individually identified class of objects that has military significance. Examples are a type of person (e.g. by rank), a type of military system (e.g. a self-propelled Howitzer), a type of facility (e.g. an airfield) or a type of organization (e.g. an armored division).
CAPABILITY	The potential ability to do work, perform a function or mission, achieve an objective, or provide an service.
LOCATION	A specification of position and geometry with respect to a specified horizontal frame of reference and a vertical distance measured from a specified datum. Examples are point, sequence of points, lines, circles, rectangles, block etc. It specifies both location and dimensionality.
ACTION	An activity, or the occurence of an activity that may utitlize recourses and may be focused against an objective. Examples are operation order, operation plan, movement order, fire order, close air support mission, logistic request, or incident (e.g. enemy attack).

JC3IEDM is designed to support the unambiguous definition of information exchange requirements in the operational domain. The contributions of data modeling and operational experts ensure technical maturity and operational applicability based on agreements and multilateral consensus (Tolk, 2012). The relationship between entities and actions is the same as between nouns and verbs. In the grammar of combat, elements (nouns) carry out actions (verbs) against other elements or themselves (DuBois et al., 1997). For example, the OBJECT-TYPE *weather* has an OBJECT-ITEM the *sun* which has the CAPABILITY *to heat* other objects. Subsequently, the weather can perform an ACTION-TASK by making the sun *shine*, which has an ACTION-EFFECT of *warming up* the OBJECT-ITEM of armored vehicles. Because the *warm up time* ATTRIBUTE for vehicles differs from that for the terrain, the infrared signature of the vehicles contrasts with that of the background. With the right equipment, this difference can be observed.

The Modeling and Simulation community is the primary audience for the MSDL specification. In contrast, software developers intend to use the C-BML specification in the C2, modeling and simulation, and autonomous system domains. The most prominent issue with XML is that it is verbose. Indeed, people will not read the raw XML; still, one has to read what they wrote as a writer. The following section will address this issue with raw XML by introducing an ontology that entails [all] required elements to describe modern conflicts.

2.3. Modern Conflict Ontology

Wargames, as well as simulations, require models, which have to be consistent throughout the system. For example, in a COA analysis, this commonality is complex to achieve when the model only resides in the minds of its creators. According to Hartley in *Simulation and wargaming*, an ontology provides a significant starting point for a well-defined model by identifying and defining all of the potentially needed components of the model and their relationships. The initial work for an ontology was formulated by the US Army Training and Doctrine Command in the Irregular Warfare Ontology; this ontology has been further developed by Hartley into the Unconventional Conflict Ontology (Hartley, 2017). Subsequently, the ontology has been extended by Hartley to include conventional as well as unconventional conflicts in the Modern Conflict Ontolology (MCO) (Hartley, 2021). An ontology permits the expression of more robust semantics, offering benefits in automated reasoning, strong validation, query, and information-linking (Blais et al., 2019).

Figure 2.6 illustrates the basic[9] context diagram fundamental to the operational environment from which the MCO domain is derived. The active elements of the domain, which do things, are called Actor classes. These classes are human beings, groups of people, and non-human elements that can act. As shown in the figure, Actors can perform Actions; these actions can cause changes in the Actor or Object classes. In turn, these actions may affect the entire Operational Environment. State Variables describe all operational environment elements. Actors perceive the State Variables. State variables are the only way that Actors know anything, including about themselves. The Objects consist

---

[9] The basic diagram, as used here, is limited to the concrete elements of the MCO. The more abstract sub-ontologies are part of the full context diagram.

of human-made or natural environment elements and conceptual elements. Contrary to the Actors, Objects perceive nothing; they just exist (Hartley, 2021).

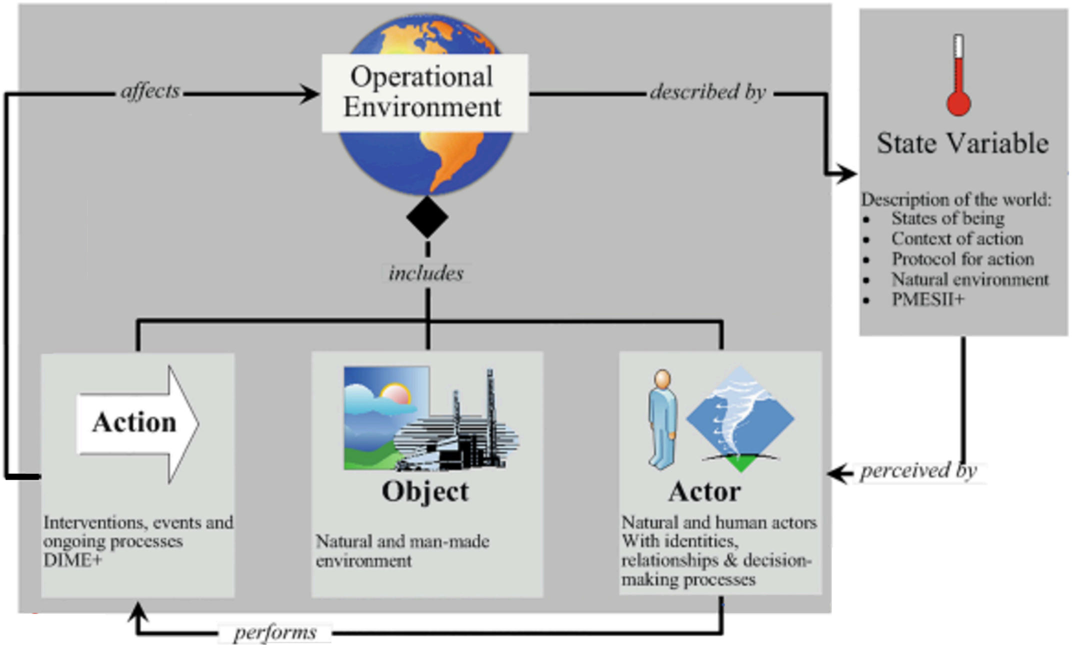


Figure 2.6: Basic context diagram depicting the Modern Conflict Ontolology derived from *An Ontology of Modern Conflict* by Hartley (2021).

Based on the book *An Ontology of Modern Conflict* by Hartley, the following paragraphs provide a brief overview of the sub-ontologies that are part of the basic context diagram. The first paragraph describes the Actor Ontology, where Actors are the motivating elements in the MCO. The second paragraph discusses the Action classes, representing the actions that Actors can initiate. The third paragraph overviews the Object classes, representing passive things. After that, the fourth paragraph outlines the State Variables in the MCO, representing the conditions of the other elements. The final paragraph reviews the applicability of MCO within military wargaming.

2.3.1. Actors

Generally, actors are people or groups of people, yet in the MCO structure, non-human actors can also cause things to happen. Each *Actor class*, often shortened to *Actors*, is a thing that defines the instantiations of the class to be a particular subset of all the things that cause actions. The Actor classes and subcategories aid this definition process; each narrows the subset meaningfully.

Actors are the *nouns* in MCO that represent the active elements, whereas Objects represent passive elements in the Ontology. In this ontology, the Actor's top level is divided into five classes of actors: some Actors represent individuals, some represent groups, some represent demographic populations, and some represent inanimate things. However, all Actor classes are active and capable of initiating Actions, as discussed in the next section.

2.3.2. Actions

Action are the interventions, events, and ongoing processes (by convention addressed as occurring in zero time) in the MCO. Actors initiate them and are the direct causes of changes. Each *Action class*, often shortened to *Action*, is a thing that defines the instantiations of the class to be a particular subset of all of the things that are the direct causes of changes. The Action classes and subcategories aid this definition process; each narrows the subset meaningfully.

Actions are the *verbs* in MCO that represent the activities of Actors. In the ontology, the action top level is divided into nine classes of Actions: Some have diplomatic leverage, some have informational leverage, some have military leverage, some have economic leverage, and some have other leverage. However, despite the (best) intentions of the Actors, there is no assurance that the action will succeed.

An Action does not change any other Action directly. The Actions initiated by the Actors affect the Actors, described in the previous paragraph, and Objects, provided in the next paragraph.

2.3.3. Objects

Objects are the recipients of action in the MCO; they do not cause things to happen; they are passive and acted upon by the Actors. Generally, human beings are not Objects, but they can be manufactured, and in some instances, they may contain people. Each *Object class*, often referred to as *Objects*, is a thing that defines the instantiations of the class to be a particular subset of all of the things that are passive and acted upon. The Object classes and subcategories aid this definition process; each narrows the subset meaningfully.

Objects are the inactive *nouns* in MCO; the Object top level is divided into seven classes of Objects: some of the Objects are tangible things, sometimes literally made of concrete; some are more abstract like concepts and ideas; some have concrete as well as conceptual characteristics and include 'needed thing elements' or natural objects. The status of Actors, Actions, and Objects is described by Metrics or state variables outlined in the next paragraph.

2.3.4. State Variables

State Variables or Metrics describe the status of Actors, Actions, and Objects. The Metric instances hold the Actor, Action, and Object property values. Each of the Actor, Action, and Object classes has a defined set of *Metric Types*, which is uniform for all of the classes within the particular subcategory of the class.

Very roughly, the Metrics correspond to *adjectives* and *adverbs*, modifying the *nouns* (Actors and Objects) and *verbs* (Actions). However, Metrics provide more specific information than common adjectives and adverbs by providing current state variable values.

2.3.5. Overview of MCO elements

Hartley uses an ontology to represent the elements of modern conflict(s). Actors are defined as the main elements of the conceptualization, as they can be direct or indirect stakeholders in modern conflicts; they can take actions and be subject to the actions of other actors while influencing or being influenced by the objects of modern conflicts. Actions are defined as interventions, events, or processes that can be classified as diplomatic, information, military, or economic actions. Objects, conversely, are defined as elements of the natural environment, manufactured elements of the environment, or conceptual elements of their own.

The above overview and the developed sub-ontologies provide an extensive and in-depth analysis of the elements of modern conflict. Despite the stated focus by Hartley of MCO on conventional as well as unconventional conflicts, the ontology still requires further development in the domain of conventional warfare, especially tactical operations and activities for military units have not been elaborated in great detail. That said, the ontology demonstrates the extensiveness of military doctrine, which is constantly subject to change. The following section provides an overview and classification of the identified elements from the three perspectives to give direction to essential principles in the design.

2.4. Identified Elements and Principles

This chapter has provided an overview of the wargaming elements that emerge from the different perspectives on wargaming. The components included in the Tactical Decision-Making Model are specifically designed for military decision-making and are adopted by Dutch military personnel for training and operation. While these components are not designed for simulation, they shape the simulations that may be used to train military personnel or provide decision-making information. Military simulation standards are designed to meet simulation as well as operational requirements. Nevertheless, as these standards are written in XML, they are not easy to read, let alone to be understood by regular military personnel. Modern Conflict Ontolology (tries to) codify what is known about or what should be known of modern conflicts. It describes and structures the elements and provides insight into the mutual relationships between the elements. The strength of an ontology is also its greatest weakness.

It is never complete[10].

Table 2.4: Simulation concepts for wargaming by Andreas Tolk in *Engineering Principles of Combat Modeling and Distributed Simulation*.

Concept	Simulation Purpose
Entity	Entities are the simulated things that represent weapons systems or military units that represent no further separable conceptualizations. They are characterized by attributes. Entities can active as well as passive. Active entities can initiate events, whereas passive entities cannot initiate events.
Event	Events connect entities, which can be individual entities or groups of entities. When events happen, entities are affected.
State	States of entities are equivalent to specific attribute value constellations of these entities. For example, a system may be defined to be in the state of ”standing” if the attribute velocity has the value zero.

This overview results in a myriad web with a large number of elements. The three simulation concepts for wargaming (see Table 2.4) defined by Tolk in *Engineering Principles of Combat Modeling and Distributed Simulation* provide the framework to classify the identified elements that are required to formulate a simulation model from the three perspectives. The *entity* element is divided into two concepts to make it possible to distinguish between active and inactive elements that are present within the three perspectives. Although the perspective has different interpretations of passive and active[11], this distinction makes it possible to create models that represent the active elements that significantly shape a military operation.

Furthermore, the *other* category is added; this category entails enabling or procedural decision-making elements that are not directly related to the three simulation concepts. These other concepts may be part of the simulator that runs the simulation model(s) or are specific elements of the military decision-making process. Table 2.5 provides an overview and classification of the identified wargaming elements from wargaming practices, military simulation standards, and conflict ontology. These wargame elements provide the building blocks or functions relevant for designing a wargame simulation system. Designing a simulation system consists of transforming the listed wargame elements into relevant components that shape a simulation system that can meet the objectives of a wargame.

Table 2.5: Overview of identified wargaming elements.

Concept	Miltary Decision-Making	Military Simulation Standards	Modern Conflict Ontology
Entity (active)	Actors Factors Targets (individuals or groups) Order of Battle (defined as military units)	OBJECT-PERSON OBJECT-ORGANIZATION OBJECT-MATERIEL OBJECT-FACILITY	Actor Categories
Entity (passive)	Targets (that are objects) Environment Table of Equipment (defined as military systems)	OBJECT-FEATURE	Object Categories
Event	Effects Tasks Threat (Scenario’s) COAs (or the Planning of Activities) Contingencies	ACTION	Action Categories
State	Commander’s Critical Information Requirements	CAPABILITY	State Variables

Table continues on next page.

---

[10] the Modern Conflict Ontology by Hartley (2021) identifies over 43,000 individual elements, and still not all concepts of modern warfare are included.

[11] For example, in *An Ontology of Modern Conflict*, a military platform that is not equipped with personnel is considered an object. In contrast, a human-operated platform is considered an actor.

Table 2.5 Continued: Overview of identified wargaming elements.

Concept	Miltary Decision-Making	Military Simulation Standards	Modern Conflict Ontology
	Table of Equipment (defined by their attributes)	LOCATION	
Other	Mission Analysis Objective & End state Requirements Guidance Concept of Operations Decision Support Overlay Synchronization Matrix Operation Order	Information Exchange Structure Content Specification Data Model Specification Protocols	Context Diagram

Given this brief overview of the wargame elements that can be identified from the different perspectives, these elements provide a relevant base for military wargaming. In doing so, they also shape the requirements or user needs later formulated for designing the wargame simulation system (see Paragraph 5.1.1). The nature of wargaming shows that not only the information from - the results but also the wargaming experience is relevant for the military decision-maker(s), making insight into the process and the outcomes relevant. The international standards of SISO provide the data structures for specifying the information exchange that the simulation system must comply with, together with military doctrine. Furthermore, the extensiveness of the MCO, which is also its most significant limitation as it is never complete, shows that the system must be adaptable and extensible because military doctrine is not a dead language; it evolves and adapts to changing circumstances.

These concluding observations on wargaming elements provide insight into which underlying principles are relevant for designing composable simulation elements. As processes, such as the TBM, are intended for decision-making by military commanders, these processes and their elements are not intentionally designed to be represented by simulation systems. While the military staff is extensively trained in these decision-making processes, they are generally not experienced developers proficient in creating and using complicated simulation systems. The system must provide representative simulation elements that resemble doctrine; the operational analysis steps that are part of the decision-making process must be recognizable and accessible by the military users; the system should be adaptable to incorporate the changes and developments within military doctrine; the and the generation of simulation results should be traceable to provide insightful knowledge to the military staff. These provisions mean that recognizability (principle 1.1.), utilizability (principle 1.2.), traceability (principle 1.3.), and adaptability (principle 3.1.) are relevant principles[12] for the envisioned wargame simulation system.

A last observation derived from the three perspectives is that a military organization consists of multiple levels or military *echelons*. Whereby the operation of each echelon can be represented within a simulation environment. Given the identified elements and derived principles that originate from military wargaming, why is it still difficult to implement these elements in contemporary wargame simulation systems? The next chapter explores contemporary challenges with military simulation to identify design principles from rigorous scientific M&S research to further guide the design of the envisioned wargame simulation system.

---

[12] Numbering of the principles is derived from system requirements structure developed in paragraph 5.2.3.

3

Military Simulation Challenges

> *”We cannot solve our problems with the same thinking we used when we created them.”*
> Albert Einstein

There is a transition from extensive monolithic simulations into component-based architectures, for example, High Level Architecture (HLA), to facilitate innovation, flexibility, and cost-effectiveness (Van Den Berg, 2021a). Recent technical development in cloud computing technology and service-oriented architecture (SOA) offers better opportunities to utilize M&S capabilities. This new service orientation and the provision of M&S applications via the ’as-a-service’ computing model may enable composable simulation environments that can be deployed rapidly and on demand. This new concept is known as Modeling & Simulation as a Service (MSaaS) (Van Den Berg, 2021b).

NATO and the armed forces of its member states predominantly have been using distributed simulation systems for various LVC simulation purposes, such as wargaming, mission rehearsal, strategy analysis, or flight- and platform-simulators. Consequently, M&S has become a critical technology for the coalition and its nations. In contrast, military simulation has a long history; combining these contemporary military simulation systems designed for different purposes remains challenging (Lee et al., 2020; Rabelo et al., 2015).

Military services are generally classified according to the domain in which their troops operate. The Army represents the land component; it consists of Combat[1], Combat Support[2], and Combat Service Support[3] units. The Air Force represents the air component; it uses fighter, transport, and bomber aircraft to conduct its operations. The Navy represents the sea component; it uses surface ships and submarines for sea and littoral operations[4].

The smallest unit in an army usually is a squad or group of soldiers. A squad consists of about five to ten soldiers. Three to five squads are combined into a platoon. Analog to the number of squads in a platoon, a company comprises of platoons, which continue up to the highest echelon. Up to battalion echelon, not deployed Dutch Army units are structured homogeneous, meaning soldiers within a specific unit belong to the same service branch. When deployed, a combination of different units, infantry, cavalry, artillery, or engineers, are constructed into company, battalion, or brigade task forces tailored for specific operations. For example, a combined arms team can consist of two infantry platoons, a tank platoon, and an anti-tank platoon[5].

The Air Force structure is centered around the individual aircraft. Contrary to land forces deployed to occupy land, the air component is designed to destroy air or ground targets and move troops and equipment fast and over greater distances. Similarly to the army structures, although applied less hierarchically, an air force is built up with flights, squadrons, groups, and wings as the primary organizational structures. The Navy structures its forces around the types and number of vessels. A naval structure ranges from a task element or a single vessel to a task force or battle group, with a fleet as the most significant unit size.

The land, air, and sea components are deployed in the same operational area or theater in a joint operation. Analog to the formation of combined arms teams, in a joint operation, air, land, and maritime

---

[1] e.g., Infantry, Cavalry, or Artillery

[2] e.g., Engineer, or Intelligence

[3] e.g., Logistics, or Maintenance

[4] The allocation of service capabilities varies country by country. For example, the 101st US Airborne Division has its dedicated Combat Aviation Brigade, while the Dutch 11 Air Assault Brigade does not include dedicated airborne assets.

[5] The organizational structure of a unit, the Order of Battle (ORBAT), and the lists of all equipment assigned to a unit, the Table of Equipment (TOE), reflects how a unit is organized and what resources it has.

units are combined into aggregated units to fight more effectively. If nations join forces for a particular operation, such as a United Nations (UN) peacekeeping mission, it is called a combined operation.

After this rather elaborate introduction to how the Armed Forces can be organized, the question is how to compose these complicated structures that can execute complex missions, which are subjected to change, in a simulation environment. This composability of simulation elements for wargaming is further complicated by adding terrain, weather, and missions or tasks to the simulation environment. A wargame becomes complex when the actor's actions are added to the equation. If a simulation models single weapon system platforms like tanks, soldiers, or single aircraft, it is called entity-level simulation. If several of these entities are aggregated into a higher object, like a company with ten tanks or an airstrike of twelve aircraft, it is called aggregate-level simulations (Page & Smith, 1998).

To untangle this knot created by composing different military simulation environments subject to changing and different military organizational structures, this chapter analyses scientific M&S research to identify the challenges within contemporary constructive military simulation. The first section addresses the challenge of composability as applied within military simulation. The second section discusses the challenges that arise from the multi-resolution, aggregation, and disaggregation of entities within a constructive military simulation system. The third section analyzes the challenge of maintaining consistency when using multi-resolution combat models in simulation environments for wargaming. The fourth section discusses the challenges when entities representing different military echelons interact within the same simulation system. The fifth section addresses challenges from contemporary military simulation designs that can lead to increased system complexity or degraded system performances. Based on the identified challenges in the preceding sections, the last section synthesizes the identified challenges into design principles for composable simulation elements that can be applied to the envisioned wargame simulation environment.

3.1. Composability

The concepts of interoperability and composability used within military simulation are subject to discussion amongst scientists (Van Den Berg, 2021b). Petty and Weisel (2003) defines interoperability as: “the ability of different simulations, connected in a distributed simulation, to collaborate to simulate a common scenario or virtual world meaningfully.” While Petty and Weisel defines composability as: “the capability to select and assemble simulation components in various combinations into simulation systems to satisfy specific user requirements.” According to Balci et al. (2011), composability is the degree to which an artifact can be constituted by combining things, parts, or elements. Many societal, technical, and organizational domains require the composition of disparate capabilities, where modeling the capability interactions prior to employment is both efficient and risk-averse. Component-based development of M&S applications is an unsolved problem. Components that must be assembled are coded in different programming languages intended to run under different Operating Systems on different hardware platforms. The level of granularity and fidelity of components may not be compatible when deployed in different M&S environments. Furthermore, a component providing much more functionality than needed can degrade the execution efficiency (Balci et al., 2011).

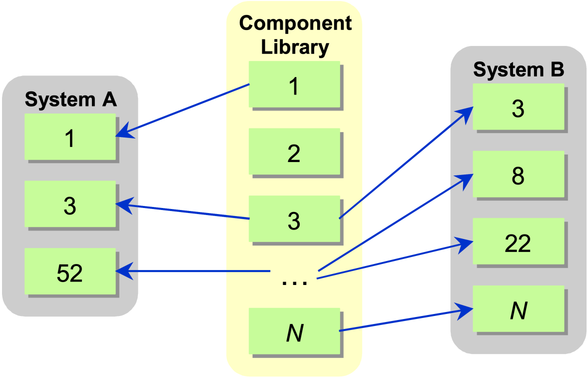


Figure 3.1: Notional example of composability (Weisel et al., 2003).

The interoperability of simulation components is necessary, but more is needed for composability.

Composability requires interoperability, but interoperability is possible without composability (Petty & Weisel, 2003). Weisel et al. (2003) defines the main characteristic of composability as a different simulation system that can be composed at configuration time in various ways, each suited to some distinct purpose, and the different possible compositions will be valid simulation systems. Making composability more than just the ability to put simulations together from components; it is the ability to combine and recombine, to configure and reconfigure, sets of components from those available into different simulation systems to meet different needs. The process of composing different simulation systems from components is illustrated notionally in Figure 3.1.

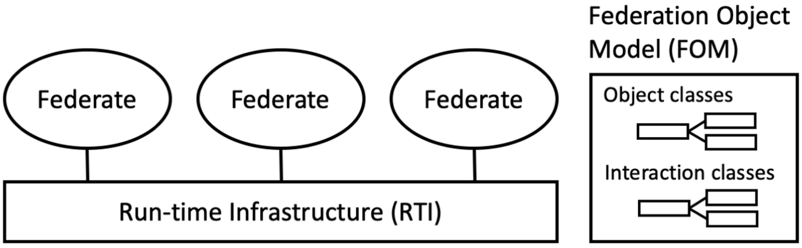


Figure 3.2: High Level Architecture derived from "NATO M&S High Level Architecture" by NATO (n.d.).

High Level Architecture considers the act of coupling components to facilitate their inter-operation (see Figure 3.2). The presence of multiple models and multiple levels of abstraction (see Figure 3.4), represented in the federates and connected by the Run-Time Infrastructure (RTI), increases complexity. This problem resulting from this complexity has been called the multi-resolution modeling problem. This problem is referred to as the multi-resolution modeling problem. Theoretical work in this area also indicates that multi-resolution modeling is fundamentally complex to do correctly (P. K. Davis & Bigelow, 1998; P. Davis & Bigelow, 1998; Reynolds et al., 1997). The vertical dimension of composability involves the act of coupling two components for the sake of aggregation. Commonly, (dis)aggregation forms the basis for resolving resolution differences in inter-operating defense simulations. However, it is easily seen that abstraction through aggregation may not provide the best (or even a valid) solution. In the vertical dimension in Figure 3.4, composability encourages abstraction through (dis)aggregation, which may hinder the application of other, more suitable methods of abstraction (Page & Opper, 1999). In addition to integration challenges M&S applications share with other component-based applications, the grand challenge of creating and recognizing composable services remains open. It needs to be solved (S. Taylor et al., 2015).

3.2. Multi-Resolution

There is a transition from extensive monolithic simulations into component-based environments within federated architectures to facilitate innovation, flexibility, and cost-effectiveness (Van Den Berg, 2021a). Despite their benefits, these federations also create new challenges for composing the various simulation elements. The first paragraph discusses the challenges of multi-resolution and the (dis)aggregation of entities. The second paragraph analyzes the subsequent challenges that arise from inter-level interactions.

3.2.1. Aggregation and Disaggregation

Figure 3.3 shows blue team 'A Infantry Company' being disaggregated and 'B Infantry Company' being aggregated; this is further specified by Figure 3.4 that subsequently depicts how the resolution of the military echelon can change during a simulation or between federates. The challenge in military simulation systems is how to decompose the orders of the higher echelon into tasks for sub(-ordinate) units and how to use lower level information to support reasoning at the higher echelon (Alstad et al., 2013; Bruvoll et al., 2015). There are several possible ways for triggering such resolution change operations, especially disaggregation and aggregation:

1. Designated area: Disaggregation occurs when a unit is located within a portion of the terrain area

designated in advance as a disaggregation area. There may be multiple designated disaggregation areas. Aggregation occurs when the center of mass of the disaggregated unit, as determined by the entities in the unit, moves out of the disaggregation area. In order to reduce the likelihood of a rapid sequence of resolution change operations for a unit moving along the perimeter of a disaggregation area, the boundary of the disaggregation area can be defined to be slightly larger for triggering aggregation than for disaggregation.

2. Proximity: Disaggregation occurs when a unit is within a critical range of one or more hostile units or entities. The range may be the maximum detection range of the sensor systems in the hostile unit or entities. This criterion can lead to spreading disaggregation, which is described in the following paragraph.
3. Intent to interact: The aggregate unit intends to interact (e.g., employ sensors or conduct direct fire) with another disaggregated unit.
4. Operator initiative: Disaggregation or aggregation occurs when triggered by a human operator, who may do so at will. This criterion allows human intelligence to decide when a resolution change is appropriate. For example, when military personnel want to develop or view a particular operation at a specific military level. Another valuable application of operator initiative is during the testing and development of a new multi-resolution combat model to validate the simulation results at different levels of granularity.
5. Commander's focus of interest: A commander may wish to view different parts of the battlefield at different levels of granularity. As the commander's focus of interest shifts around the battlefield, units may be disaggregated or aggregated in response (Downes-Martin, 1991). The precise mechanism may be manual or automatic depending on whether the commander's focus of interest is available on the system.

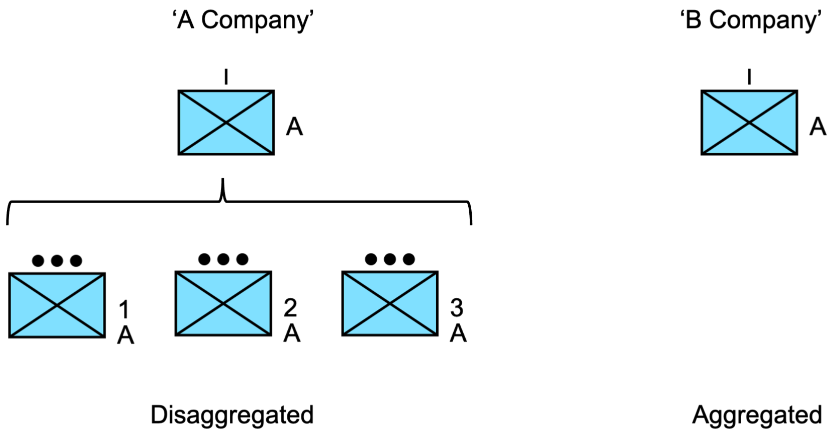


Figure 3.3: Disaggregated and aggregated units.

Because many data items can be represented at different detail levels in a multi-resolution combat model, the user must be given sophisticated tools to select a resolution level for individual data items efficiently and effectively. Many automated triggers suffer from the spreading disaggregation problem (also known as chain disaggregation), in which the disaggregation of one entity causes the disaggregation of other entities and spreads across the simulation. In the worst case, the advantages of multi-resolution combat modeling are defeated because all entities will be represented at the disaggregate level. Research in this area has developed and compared different disaggregation triggers, including operator selection, fixed geographic areas, and proximity thresholds between units (Williams et al., 1998). More work is needed on devising semi-automated or automated approaches that provide users with proper ways of managing resolution changes without introducing effects like spreading disaggregation.

Most existing multi-resolution combat models exploit the military echelons hierarchy when defining resolution levels that pervade the simulation lie so the object, including the synthetic natural environment, is represented at the ”company level” or the ”entity level”). Increasing resolution flexibility aims

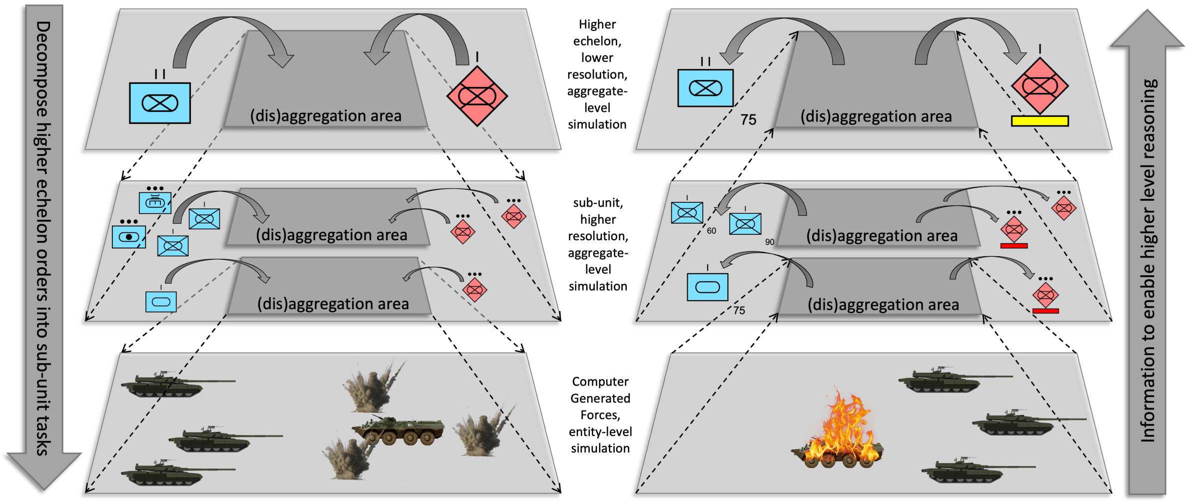


Figure 3.4: Graphic depicting the multi-resolution challenge for military simulation derived from “A Dynamic Multi-Resolution Model Based on HLA Interconnection of Commercial-Off-The-Shelf (COTS) Simulation Tools” by Raue and Gallois (2011).

at creating multi-resolution combat models that can be built and controlled on a per-data-item basis. For example, the resolution of the synthetic natural environment does not need to be directly tied to the resolution of military units maneuvering in; it may be helpful in various environment resolutions to be available to a single military unit model. This availability is a more complex problem than exploiting the military echelon hierarchy for resolution levels. However, there would be a big payoff in flexibility if such systems were built.

In a multi-resolution combat model, the number of entities instantiated as a result of disaggregation operations can become too large for some parts of the system; perhaps the computational or storage capacity of the entity-level model’s host computer or bandwidth of the network connecting the unit level and entity level models’ host computers (Williams et al., 1998); the condition has been referred to as disaggregation overload (Trinker, 1994). Disaggregation overload may result from spreading disaggregation, which can occur when disaggregation of units is triggered by geographic proximity to disaggregate entities, one of the disaggregation triggers listed earlier (Petty, 1995). Spreading disaggregation, which is a chain reaction of disaggregations, may occur when each disaggregation operation creates new entities close enough to additional units to trigger their disaggregation as well[6].

Suppose a high-resolution entity (e.g., a friendly tank) begins interacting with a low-resolution entity (e.g., an enemy infantry platoon). Typically, the low-resolution entity is disaggregated (e.g., the four infantry fighting vehicles that are part of the infantry platoon) so that the tank and the infantry fighting vehicles can interact at the disaggregate level. However, other entities that may have been interacting with the infantry platoon would now also have to disaggregate. Extending this reaction to other entities forced to disaggregate is easy because a low-level entity they were interacting with disaggregated. The domino effect caused by the initial disaggregation is called chain disaggregation, also known as spreading disaggregation in the literature. Chain disaggregation causes the number of simulated entities to increase rapidly. This chain disaggregation increases the load on processors and the network.

3.2.2. Inter-Level Interactions

As previously stated, one design intent or a multi-resolution combat model is That simulated events occurring in one of the models’ simulations may affect events in the other. The usual means by which this may occur is the resolution change operation, e.g., a unit is disaggregated into entities, which then participate in the entity-level simulation. However, an alternative to the resolution change operations exists. Inter-level interactions are interactions (as previously defined) between a unit and an entity directly across the resolution boundary without first disaggregating the unit into entities or aggregating the entities into a unit. Inter-level interactions can avoid spreading disaggregation and the resulting disaggregation overload mentioned earlier and allow the system to include unit-level or entity-level

---

[6] Instead of *spreading disaggregation*, some sources i.e. Reynolds et al. (1997) use the term chain disaggregation.

models that cannot support resolution change operations.

Inter-level interactions were introduced in Franceschini and Petty (1995) and Powell (1997), who explained the need for them and gave example scenarios to show how they might be used. Some types of possible inter-level interactions and summaries of how they might be implemented are listed; for an extended discussion of these types of inter-level interactions and others, see Petty and Franceschini (1998).

1. Indirect fire: Indirect fire is generally aimed at a geographical area where hostile entities are expected to be. Artillery volleys at the unit level can be automatically translated into individual artillery detonations at the entity level and applied against the entities there. Similarly, but with more difficulty, individual artillery detonations at the entity level can be aggregated into volleys and resolved at the unit level. Implementations in both directions exist (Karr et al., 1993; Karr & Root, 1994).
2. Direct fire: Direct fire, aimed at a specific entity, is more problematic as an inter-level interaction than indirect fire, primarily because of the effect of intervisibility on direct fire and the differences between inter-visibility handling at the unit and entity levels. Inter-level direct fire can be converted to intra-level direct fire, either at the entity level by using pseudo-disaggregation or partial disaggregation of the unit to produce entities or at the unit level by first using pseudo-aggregation or partial aggregation to produce a unit and then resolving the direct fire at that level.
3. Operations orders: Military forces are typically operating under a set of orders that define the mission and provide direction for their actions. Suppose the entity-level model has automated behaviors available to control the simulated entities, as is typically the case with SAF systems. In that case, the operations orders controlling the actions of a unit in the unit-level model can be translated automatically or by the operator of the SAF system into entity-level behaviors when the unit is disaggregated.
4. Communications: Communications, in reality, are always from an entity to an entity, i.e., there is no ”unit” in any physical sense. To resolve inter-level point-to-point communications in a multi-resolution combat model, determine which entity in the communicating unit would be communicating (e.g., the command vehicle) and a location for that communicating entity (by pseudo disaggregation, partial disaggregation, subset disaggregation, or by default, e.g., the center of mass of the unit) and resolve the communications in the entity level model using that location. For inter-level broadcast communications, the process is more complicated, as locations for all possible recipients of the communications may be needed. Deliberate jamming of communications and detection of non-communication emissions are similarly complicated in the inter-level context.
5. Terrain updates: During a simulation, units or entities may modify the terrain at their level in ways that are tactically significant at the other level, e.g., an engineering unit demolishes a bridge that should be impassable to entities for the remainder of the execution. Such modifications are best first resolved at the initial level, i.e., as an intra-level interaction, then translated into terrain updates suitable for the terrain representation at the other level, conveyed to that level, and applied there (Petty & Franceschini, 1998).

Existing multi-resolution combat models have some limited inter-level interactions between entities. The most common example is indirect fire from an aggregate artillery battery against a disaggregated company. However, other more complex inter-level interactions could be implemented helpfully. For example, would it make sense to allow a tank (at the disaggregated entity level) to engage in direct-fire combat with an aggregate company? However, neither implementing inter-level interactions nor understanding their impact on results correlation is simple (Reynolds et al., 1997). Unlike the familiar and well-understood mathematical models of combat at the unit level and the natural resolution and data-supported combat models at the entity level, there is no theoretical or experiential basis for direct inter-level interactions. The most focused study of inter-level interactions found that the best available mechanism for implementing such interactions was generally some form of pseudo-disaggregation (Petty & Franceschini, 1998).

3.3. Consistency

In a multi-resolution combat model, the outcome or results of a scenario are considered desirable to be independent of the resolution level at which that scenario is simulated. In other words, the results of executing a scenario should be the same or similar within a desired tolerance, regardless of which units were disaggregated and aggregated throughout the simulation and when those operations occurred; here this similarity is termed results correlation[7]. If the results do not correlate this way, the difference between the unit-level model's and the entity-level model's representations of the same events introduces discrepancies.

Mapping inconsistency occurs when an entity undergoes a sequence of transitions across levels of resolution, resulting in a state it could not have achieved in the simulated time spanned by that sequence. Any scheme in which entities transition across resolution levels (e.g., aggregation-disaggregation) must consistently map attributes across levels. Specifically, the translation should enable switching levels without changing attributes, provided no other interactions occur. Poor translation strategies cause "jumps" in the state of entities. A jump in visual perception is caused when the perceived changed position of an entity violates simulation semantics due to rapid translations between states. The aggregate-level information may be insufficient in providing disaggregate-level consistency. In other words, a disaggregated-to-aggregated transition may lose some information about the high-resolution entities, say position.

Consequently, a second transition, this time aggregated-to-disaggregated, may result in a disaggregated state inconsistent with the first disaggregated state because a standard algorithm or doctrine has been applied to position the entities (Clark & Brewer, 1994; P. K. Davis, 1993; Franceschini, 1993). While perfectly state-maintaining translation strategies are desirable, often, these may not be found readily. In such cases, the potential perceptual inconsistencies arising from translations from one state to another must be addressed differently.

For example, suppose that in a multi-resolution combat model, a tank squadron is ordered to perform a road march from its current location to a destination location. That movement may be simulated as a unit moving in the unit-level model or as the individual entities of the company moving in the entity-level model. In each case, the terrain database in the model at the level where the movement is simulated will be accessed to get the information needed to simulate the movement. It is reasonable to expect that the amount of simulated time required for the unit to complete the move in the unit-level model will be consistent with the time required for the corresponding formation of entities to complete the move in the entity-level model, but it may not be. Differences in terrain representation or movement model may introduce discrepancies, and those discrepancies could have a significant effect on the scenario outcome.

The aggregation and disaggregation operations introduce fundamental validity questions regarding the correlation between levels of resolution (P. K. Davis, 1995; Franceschini, 1999; Franceschini & Mukherjee, 1999; Reynolds et al., 1997). Research has been ongoing in this area; a variety of approaches have been investigated, including disaggregation methods designed to avoid introducing correlation issues (Franceschini et al., 2000) and predictive aggregation/disaggregation algorithms (Chua & Low, 2009). A systematic comparison of the results produced by multiple models at different levels of resolution on a set of identical combat scenarios revealed that differences in spatial representation, force aggregation, and time step can lead to different outcomes; this suggests potential sources of results correlation error in multi-resolution models (Hillestad et al., 1995). A detailed investigation into the fundamental issues associated with maintaining consistency between multiple levels of resolution led to several important observations about multi-resolution modeling, including the need to handle concurrent interactions at multiple levels of resolution and the potential for time step differences between levels to introduce inconsistencies (Natrajan, 2000; Reynolds et al., 1997). Multi-resolution entities, which can exist and interact at multiple levels of resolution concurrently and maintain an internal state consistent across those levels, were proposed as a solution. The useful notion of mapping functions, which map object attributes from one level of resolution to another, was introduced in the context of this research.

---

[7] Instead of results correlation, some source use the terms consistency of prediction, or simply consistency, e.g. P. K. Davis (1995), Powell (1997), Reynolds et al. (1997) and Franceschini et al. (2000).

3.4. Run-time Infrastructure

During disaggregation, a translation must be made from the state of the aggregate unit to the state of its disaggregate constituents. This transition involves a set-up time, populating dis-aggregate attributes from the aggregate, and initiating protocols to place entities. A similar protocol may exist from the disaggregate to the aggregate level. The time taken to effect an aggregation or disaggregation is known as the transition latency. Transition latency can be significantly high depending on the complexity of the protocol. For example, a protocol in Robkin (1992) takes ten seconds to complete the aggregation process. High transition latencies are incompatible with real-time constraints, for example, in human-in-the-loop simulations, because they may cause perceptual or conceptual inconsistencies. An entity that does not change position during a transition period and then suddenly undergoes a large displacement at the end causes a perceptual inconsistency or "jump."

When an entity undergoes rapid and repeated transitions from one level to the other, it thrashes (Reynolds et al., 1997). For example, a red team mechanized infantry platoon (one of the red diamonds in the middle left area of Figure 3.4), a low-resolution entity, may disaggregate into vehicles on commencing interactions with the blue team tank company (one of the blue rectangles the middle left area of Figure 3.4), a higher resolution entity. When the tank unit is out of range, the infantry fighting vehicles may revert to the aggregate level of the infantry platoon. However, the tank unit's movement may cause the infantry platoon to change levels within a short time repeatedly. This movement would cause the infantry platoon to "flip-flop" levels, each time incurring the overhead associated with a level change. While *thrashing* depends primarily on the triggering policy that causes a change of level, it is nevertheless an issue that must be addressed in the design of multilevel simulations. High transition latency compounds the problems due to thrashing because it causes some entities to spend considerable time just changing levels.

Depending on the scheme used, network resources may be strained by aggregation and disaggregation. Disaggregation creates new entities, each of which could be a sender and receiver of messages. Even if only the entity state messages generated by all the entities are considered, this increases network traffic. Also, aggregation and disaggregation protocols typically require the exchange of many control messages, an overhead that must be incurred every time a level change occurs. This overhead can reduce the effective throughput of the network. Frequent changes in level and many entities may put an unacceptable burden on the network (Reynolds et al., 1997).

A conceptual inconsistency may be caused when it takes so long for an entity to disaggregate in order to comply with a request made by another entity that the request becomes obsolete. Because an entity is implemented using distributed simulation interoperability protocols, such as Distributed Interactive Simulation or High Level Architecture, to link the unit level and entity level models, an extra layer of software complexity is introduced (Franceschini & Petty, 1995). Because they are linkages of two different models that separate organizations may maintain, configuration management effort is potentially doubled.

The challenge for the simulation designer is to be aware of the consequences for the Run-Time Infrastructure (RTI) by the design choices made when developing a military simulation system. Ill-considered choices or added flexibility can increase software complexity or degrade (reduce) system performance.

3.5. Identified Challenges and Principles

This chapter has provided an overview of the challenges with contemporary military simulation identified by scientific research on simulation. Composability is an essential characteristic for military operations (and simulation) that has been difficult to achieve. *Composability* has been increasingly crucial for M&S of a System of Systems, in which dissimilar systems are coupled. However, horizontal and vertical coupling of components is complex as parts may have different resolutions and levels of abstraction (corresponding with the military echelon they represent), making its composability fundamentally challenging (Page & Opper, 1999). As different existing (legacy) systems or components may be coupled, orders must be translated to a lower sub-unit level, or information must support higher echelon reasoning, making it challenging to ensure *consistency* throughout the used systems. The challenge of *coherency* is not about the simulator's design or the used models within the system; it concerns the effects of design choices on the RTI that supports the simulation system. The last identified challenge of *interoperability* is also related to the different systems or components representing entities at various military echelons; the question remains how to couple these systems or components when inter-level interactions are required.

Given the above overview of the challenges derived from scientific M&S research, these challenges provide rigorous knowledge for the design of simulation systems for military wargaming. Four overarching design principles[8] are linked to the identified challenges. Aggregation and disaggregation operations within multi-resolution simulation introduce fundamental validity questions regarding the correlation between levels of resolution. To maintain validity, it is essential to pursue consistency (principle 2.1.) throughout the system. Coherency (principle 2.2.) is not about the actual design of the wargame simulation system; it is about the simulation designer being aware of and understanding the envisioned design's consequences on the simulation's performance RTI. Although creating composable simulation services remains challenging, composability (principle 3.2.), whereby it is possible to combine, recombine, configure, and reconfigure sets of components, is an essential principle for military simulation systems. The last principle that needs to be addressed is that multi-resolution simulation models must be able to represent different military echelons, and as interaction can take place between these various echelons, interoperability (principle 3.3.) must be applied for multi-resolution wargame systems.

As eight design principles have been identified from the overview of relevant wargame elements and a rigorous analysis of military simulation challenges, the next question appears: which process can be used to design a wargame simulation system? The next chapter will outline the selection and application of the design methodology that will be used for creating the envisioned wargame simulation system.

---

[8] Numbering of the principles is derived from system requirements structure developed in paragraph 5.2.3.

4
Design Methodology

> *”Design can be art.*
> *Design can be aesthetics.*
> *Design is so simple; that is why it is so complicated.”*
> Paul Rand

The Design Science Research (DSR) approach used in this research requires the creation of an innovative, purposeful artifact for a specified problem domain. Because the artifact is purposeful, it must yield utility for the specified problem, and a thorough evaluation of it is crucial. Novelty is similarly crucial since the artifact must be innovative, solving an unsolved problem or solving a known problem more effectively or efficiently. In this way, DSR is differentiated from the practice of design. The artifact must be rigorously defined, formally represented, coherent, and internally consistent. The process by which it is created, and often the artifact itself, incorporates or enables a search process whereby a problem space is constructed and a mechanism posed or enacted to find an effective solution (Hevner et al., 2004).

The first section of this chapter discusses selecting an appropriate design process to guide the design of the envisioned wargame simulation system. The second section provides a concise background on Distributed Simulation Engineering and Execution Process (DSEEP), which is the selected design process for this research. The last section describes how the DSEEP steps and activities have been applied to fit this research’s context.

4.1. Selection of the Design Process

This section will briefly analyze different design processes that can be used to design technical artifacts. When designing an artifact, both requirement qualifications and object descriptions are used. Requirement qualifications are qualitative expressions of the extent to which (individual or groups of) requirements must be met, individually or about each other. A design object description is a specification of the object to be realized. A design process is described by a series of design decisions (and their rationale) regarding changes in sets of requirements and their qualifications and in (partial) design object descriptions (Brazier et al., 1996).

Technical, institutional, and process artifacts are design subjects in complex systems. In the military environment where wargaming is executed, the institutions and the processes have already been well established. As there is ample room for changing or improving these institutions and processes, this research focuses on the design of the technical artifact, a wargame simulation system. For the design cycle within the DSR approach, it is conducive to select a comprehensive design process that is suitable for the identifying requirements that underlay the design principles while it also provides the necessary steps to design the envisioned simulation system from a *greenfield* perspective; has a focus on the design of the technical artifact; and has its foundations within the simulation domain.

The waterfall model is a linear, sequential software development approach popular in software engineering and product development. This model by Royce (1970) perceives the system’s scope and environment as stable. Furthermore, it assumes comparable systems have been developed multiple times. The V-model for systems development is applied when the system’s scope and environment are stable, as are the understanding and expectations of targeted stakeholders (Fairley et al., 2023). Both approaches are less suited to the design process in this research as they are not intended for complex or unstable environments and do not originate from the simulation domain.

The Incremental Commitment Spiral Model by Boehm (1988) and the contemporary *Agile* development methods can be used when stakeholders’ expectations are not fully known and are expected to change as experience grows. These iterative and incremental methods better fit the *greenfield* perspective for the envisioned system that simulates the complex wargame system designed within the context of this research. Despite their suitability, such methods are not simulation domain-specific.

DSEEP is a process aimed at the design of a simulation environment, a technical artifact. It is intended as a higher-level framework into which low-level management and systems engineering practices can be integrated and tailored for specific uses. Furthermore, it is the standard used for the design of military simulation systems. Subsequently, to formalize the envisioned design for a wargame simulation system, this research relies on DSEEP for the design process.

4.2. Distributed Simulation Engineering and Execution Process

DSEEP is maintained by SISO, and the standard is published as IEEE Std 1730-2022, which is a revision of IEEE Std 1730-2010 (IEEE, 2022). The DSEEP is independent of a particular simulation environment architecture and provides a consistent approach for objectives definition, conceptual analysis, design and development, integration and test, simulation execution, and finally, data analysis. DSEEP is a recommended practice for the development and execution of a simulation environment.

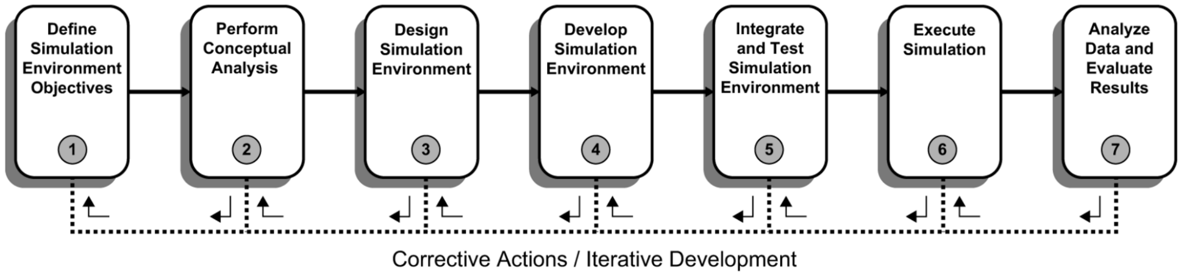


Figure 4.1: Distributed Simulation Engineering and Execution Process steps (IEEE, 2022).

Furthermore, DSEEP is a recommended systems engineering process in the *NATO Modelling and Simulation Standards Profile AMSP-01* (NATO, 2018). DSEEP identifies a sequence of seven basic steps as illustrated in Figure 4.1; note that by no means these steps are intended to be performed strictly sequentially. Each DSEEP step contains several activities that can be used to implement the step.

4.3. Application of DSEEP

Although the DSEEP standard is created to design federated simulation environments, the premesis that it is a domain-specific process for simulation environments makes it suitable for designing the wargame simulation system in this research. As DSEEP Steps 5, 6, and 7 are aimed at the actual implementation and evaluation of the simulation environment, these steps are adapted to better fit the applied proof of concept used in this research. Below is a brief description of each DSEEP step, how the subsequent activities for each step are applied within the context of this research, and which activities are not further included as they are less relevant.

- Step 1 - **Define Simulation Environment Objectives:** This step aims to identify the relevant user needs that must be fulfilled to develop and implement the envisioned wargame simulation system successfully. The user needs that have been identified in the first activity (Identify user/sponsor needs) are then used to formulate the system's objective(s) in the second activity (Develop objectives). These objective(s) are later used to test whether the designed system meets the identified user needs. The design requirements based on the user must provide measurable criteria linked to the more generally defined design principles as identified in Chapter 2. The first step's third activity (Execute initial planning) is not further included in this definition of the wargame scenario due to its project management nature.
- Step 2 - **Perform Conceptual Analysis:** This step aims to reconstruct a real-world wargame performed by soldiers during exercises or operations into a scenario that provides the foundation for the simulation system design. A wargame simulation system cannot be perceived in isolation; it is a tool used within a decision-making process. In such a process, a military commander and

his staff go through several steps to determine a possible concept of operation to execute the assigned mission. The first activity of this second step conceptualizes the wargame process. This conceptualization provides input for the wargame simulation conceptual model. The second activity within this step is to develop the conceptual model for the simulation environment. For this purpose, wargaming, as formulated within the Dutch Armed Forces Doctrine, is analyzed to create a conceptual model. The resulting model then serves as a basis for further design of the wargame simulation system. This second step concludes with an outline of the simulation system requirements. Similarly to the design requirements based on the user needs, the simulation system requirements are based on the identified design principles. These simulation system requirements are linked to the design principles uncovered in Chapter 3. These simulation system requirements provide, together with the defined test criteria, tangible measures for evaluating the performance of the designed system.

- Step 3 - **Design Simulation Environment:** This step aims to design the simulation environment implemented in the 4th step. The first activity is to identify the required support services and applications for creating the wargame simulation system. Within the envisioned design, for example, the choice can be made to obtain terrain or weather data from external services. The second activity of this step is to design the wargame simulation system. This activity uses the previously developed conceptual model and identified design principles to design a simulation environment. The design activity starts with the choice of which underlying architecture to use (e.g., HLA or DIS). In some cases, the design requirements may require a specific architecture, or the user (or a sponsor) prefers a specific architecture[1]. Subsequently, each component or member application of the envisioned wargame simulation system is constructed. The third activity (Prepare a detailed plan) is not further included in the design of the wargame simulation system due to its project management nature.
- Step 4 - **Develop Simulation Environment:** The purpose of this step is to define the information exchanged at run-time during the execution of the simulation system. The scenario that has already been introduced in step two serves as the basis for this information exchange in the provided simulation system. This contextual information is supplemented with configuration data to ensure users can set up the provided simulator and components. The contextual information from various TBM steps provides the necessary input to the simulation system and correct operation of the envisioned simulation system and included components. Implementing the various components and infrastructure ensures that their operation corresponds to the operation described in the conceptual design. Where necessary, the design of the wargame simulation system and its components are adapted for a better fit to the conceptual design.
- Step 5 - **Integrate and Test Simulation Environment:** This step aims to test the coherence of the various simulation components and the operation of the entire system. To verify, validate, and analyze the wargame simulation design, a concise real-world TBM-process is required, making it necessary to execute the TBM steps for a fictional military mission to create the information - *the scenario*. This scenario provides the benchmark for the planned proof of concept of the envisioned wargame simulation system. This scenario includes information about Blue and Red Team capabilities, which terrain features are essential, the weather's influence, and what is possible COAs for both teams. The simulation design is verified to see whether it complies with the conceptual design.
- Step 6 - **Execute Simulation:** This step aims to execute one or more simulation experiments to validate the models used within the wargame simulation system. The outcomes of the system are then compared with the expected outcomes of the formulated scenario.
- Step 7 - **Analyze Data and Evaluate Results:** This last step concludes the DSEEP process by evaluating whether the provided system fulfills the previously identified requirements and objectives. Naturally, the comparison of the (interim) results of the simulation experiments with the results of the formulated COAs analysis from the wargame scenario is part of this evaluation. This requirements evaluation shows to what extent it is possible to use the design principles to create the envisioned wargame simulation system.

---

[1] Although in the Dutch Armed Forces, many existing simulations are based on the DIS architecture, newly developed simulations for the Dutch Armed Forces have to comply with the HLA standard.

The above DSEEP steps describe the recommended practice for distributed simulation engineering and execution processes. The purpose of this standard is to provide a generalized process for building and executing distributed simulation environments. The overview of the seven DSEEP steps and description of how the associated activities are applied to create the envisioned wargame simulation system shows that this process is, at its core, suitable for a proof of concept but that it is still essential to consider the added value of the various activities and adjust them where necessary.

DSEEP seems mainly designed to compose simulation environments consisting of existing or modified simulation systems (member applications). The envisioned design for a wargame simulation system in this research does use different member applications, but at its core, the design of the prototype is about the design of one simulation system. Although DSEEP does not seem primarily designed for this, its steps provide sufficient guidance for the design of a simulation system. Various DSEEP activities or sub-products of the process seem or are less suitable. Using DSEEP requires a critical eye from a designer who uses the process for designing a simulation system as to whether or not an activity should be performed or included in the specific design process.

Chapter 5 will outline DSEEP steps 1, 2, 3, and 4, as these four steps are focused on the actual design of the system. Chapter 6 will subsequently discuss DSEEP steps 5, 6, and 7, as these steps focus on verifying, validating, and evaluating the designed system.

5

Prototype Design

> *”The best way to have a good idea is to have a lot of ideas.”*
> Linus Pauling

This chapter uses the first four steps from the DSEEP to prototype a wargame simulation system design. The first section in this chapter uses the activities outlined in the first step to define the objectives for the envisioned wargame simulation system. The second section applies a conceptual analysis to conceptualize the real-world usage of wargaming. The third section designs the wargame simulation system and the required member applications. The chapter concludes with the last section, which implements the simulation system through the activities that form the fourth step. The remaining three steps of DSEEP are discussed in the next chapter, which outlines the proof-of-concept of the prototype for the wargame simulation system.

5.1. Wargame Simulation Objective

The purpose of Step 1 of the DSEEP is to define and document a set of needs to be addressed through the development and execution of a simulation environment and to transform these needs into a more detailed and specific objective for that environment. To define the simulation environment objective, this first step of DSEEP starts by outlining the needs of a military commander and staff - the users - for a wargame simulation system. Subsequently, the objective for a wargame simulation system is formulated.

5.1.1. User Needs for the Wargame Simulation system

The primary purpose of defining the user needs is to develop a clear understanding of the problem to be addressed by the simulation environment. These users’ needs are based on the input knowledge that describes the nature of wargaming (see Chapter 2). For further reference, the *User Needs* are numbered according to ’System Requirement Structure’ that is presented by Figure 5.3 in Paragraph 5.2.3.

- *Requirement 1.1.1.* - Multiple levels of analysis based on military echelons. Commanders and their staffs execute missions on different military levels or *echelons*. The system should be able to simulate the planned operation on the different operational and tactical planning levels. Furthermore, an operational mission analysis assesses whether the sub-unit tasking is realizable; therefore, the system should be able to be configured to run the simulator on the sub-unit level. For example, when designing a brigade COA, the planning officer also validates whether the assigned sub-unit tasks are realizable given the available time and space and do not conflict with the tasks of other sub-units.
- *Requirement 1.1.2.* - A representational interface ensures that the layout of the envisioned design for the wargame simulation system resembles the real-world wargaming processes.
- *Requirement 1.2.1.* - The system can be used by ’*non-simulation-savvy*’ users. Contemporary military simulation systems with a comprehensive and monolithic nature require extensive knowledge and training to configure and operate. Military units will benefit from a simulation system that is easy to use without extensive additional education and training.
- *Requirement 1.2.2.* - The wargame simulation system should be able to run the simulation as fast as possible and faster than in real-time. Running as fast as possible enables the iterative COA analysis process, whereby a user can simulate a COA, based on the insights from the simulation, make adjustments to the COA and rerun the simulation. High *transition latencies* are incompatible with this requirement for the simulation to run as fast as possible.

- *Requirement 1.3.1.* - It should be possible to trace the simulation results to intermediate steps to provide insightful knowledge. Not only will the outcome of the simulation be relevant for analysis and subsequent decision about the best Course of Action, but also the intermediate results that give the commander and staff the insight into strengths and weaknesses, risks of an COA and all other factors of the wargame that contribute to the decision.
- *Requirement 3.1.2.* - Possibility to add new or change existing tactical mission tasks. Military doctrine evolves, or military forces may be deployed to an operational environment requiring other or additional tactical tasks. A peacekeeping mission may require other tactical tasks as a large-scale combat operation. Subsequently, it should be possible to ’load’ the appropriate combination of tactical mission tasks within the simulation environment.

5.1.2. Objectives

The objectives statement is intended as a foundation for generating explicit simulation requirements, whereby high-level user/sponsor expectations can be translated into more concrete, measurable goals for the simulation environment. Based on the above user needs, the following purpose for the wargame simulation environment has been formulated:

> *The purpose of the wargame simulation system is to provide decision-making support to a commander and staff by creating a compliant, consistent, and scalable simulation environment that can shorten the duration of a wargame and deliver objective adjudication.*

This objective distinguishes three objectives for the simulation system; these objectives link the defined purpose for the simulation system to the design principles:

- *Objective 1.* - **Compliant**, the system is designed according to rules or standards used during the execution of a real-world wargame.
- *Objective 2.* - **Consistent**, the simulation elements used within the wargame simulation system behave logically.
- *Objective 3.* - **Scalable**, new elements that represent, not yet included, tactical military tasks can be added to the wargame simulation system.

5.2. Conceptual Analysis of a Wargame Simulation

This step of the DSEEP aims to develop an appropriate representation of the real-world domain that applies to the defined problem space and to develop the appropriate use case. In this step, the objectives for the simulation system are transformed into a set of precise requirements that will be used during design and subsequent steps of the DSEEP. The conceptual analysis starts with the use-case describing the intended problem space for the simulation by a functional specification of real-world wargaming. The subsequent paragraph develops the conceptual representation of the intended problem space based on their interpretation of the use case, the user needs, and the defined objective. This section concludes by defining detailed requirements for the wargame simulation system.

5.2.1. Real-World Wargaming

This activity aims to develop a functional use case specification based on a real-world wargame scenario. Tactical decision-making combines conceptual and detailed aspects of planning. It is used to create plans and orders for both significant or long-term operations and short-term operations within the framework of an operational plan. Tactical decision-making can be based on tactical design, a higher-level plan, or direction. Tactical decision-making supports a commander in understanding the operational environment, identifying and understanding the problem, determining the desired outcome, and developing and choosing an appropriate COA. Tactical decision-making results in a comprehensive plan for the execution of the operation (IJntema & van de Haar, 2010; Royal Netherlands Army, 2014).

Usually, the ends and means for a military operation are determined by the next higher military level, and orders are given to the subordinate echelons. To this end, a commander must understand the specific aspects of the environment in which the mission must be executed. The commander and staff must then, based on the current operational picture that has been built up, examine in which ways - by which concept of operations - the mission could be carried out. Only then can a well-considered

Figure 5.1: Military officers engaged in a real-world wargaming (Work & Selva, 2015).

decision be made about the final choice with the highest chance of success. A wargame (see Figure 5.1) is a commonly used tool to support this stage in the decision-making process and can be executed in the following way. Wargaming aims to identify the vulnerable or critical elements by ’testing’ your own COAs against the expected behavior of the other actors deemed relevant (in any case, the opponent). A wargame consists of an iterative process of action, reaction, and counter-action that attempts to formulate answers to the following questions. In order to be successful, how can or should the various formulated COAs be adapted based on the interactive behavior of the other actors from the operating environment? What opportunities and threats exist, and which approach has the best chance of success? In summary, the effects of a particular COA are evaluated based on how they perform against the other COAs and the stated objective of the mission. The purpose is not only to get the ’best’ performing COAs but also to get in-depth insight into and refine the concept of operations.

A synchronization matrix (see Figure 6.1 in Paragraph 6.1.2) is a commonly used tool during the decision-making process to formalize a specific COA. A synchronization matrix provides a workspace that can visualize the cohesion between different activities in time and space and coordinated deployment of the different (types of) sub-units and capabilities against the threat(s) or effects to be achieved. Such a matrix is not intended to provide a high-resolution model of the operational concept; it is a highly abstract visualization of an operation. By considering the individual activities in their reciprocal relationship, the synchronization matrix helps to clarify how the deployment of the various capacities must be coordinated; this supports the military staff in developing an integrated plan for the operation (Royal Netherlands Army, 2011). For example, analyzing the performance of a COA across the various axis of advance is a way to reveal differences in the pace of each axis. Subsequently, control measures can be added to the operation to synchronize the activities on the various avenues of approach. When evaluating the identified COAs, a synchronization matrix can be used as the script whereby the wargame is executed.

Wargaming aims to identify the vulnerable or critical effects by ’testing’ one’s plan against the expected behavior of the other actors deemed relevant (in any case, the opponent) and, when necessary, to refine the plan further. A wargame consists of an iterative action, reaction, and counter-action process. It emphasizes critical tasks and provides insight into tactical possibilities that are otherwise difficult to highlight. Wargaming is the most valuable part of the decision-making process, and sufficient time should be set aside. For example, to ’test’ two COAs against two enemies COAs, four wargames are needed. Within the limited time of a decision-making process, more time and staff must be available to ’test’ all possible combinations of friendly and enemy COAs. Based on the above use case, the following paragraph describes the conceptual wargaming model that defines the simulation design problem space.

5.2.2. Conceptual Model

In a real-world situation (see Figure 5.1), the wargame can be under the supervision of the chief of staff, whereby the various staff officers look at the possible COAs from their specific point of view and, whenever necessary or relevant, contribute a point of view from their expertise. In any case, a designated person, usually a staff officer from the intelligence branch, will always be responsible for inputting the threat scenarios in the wargame. Similarly, staff officers can contribute to the possible behavior of the other relevant actors or point out relevant environmental factors. A conceptual representation (see Figure 5.2) of the intended problem space for a wargame simulation is developed to understand the wargaming process better. This conceptual model for wargaming is based on the user needs, simulation objective, and the described use case.

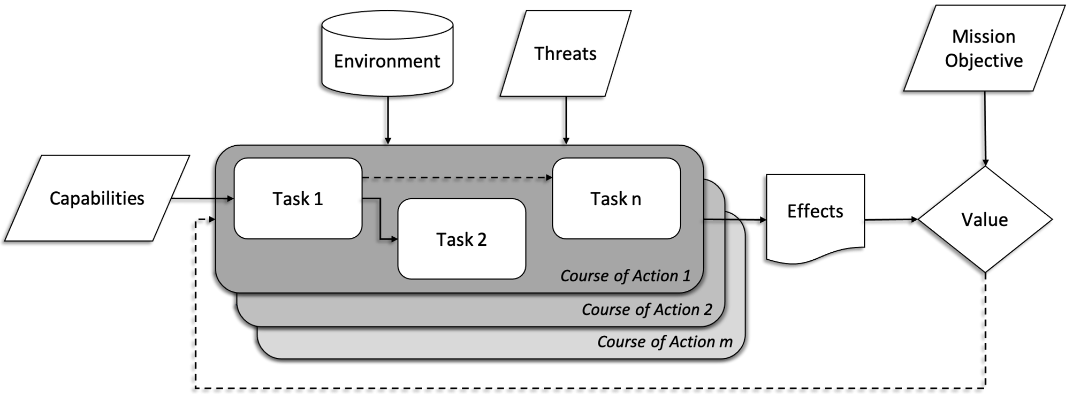


Figure 5.2: Conceptual model for a wargame.

The elements within the wargame conceptualization that contribute to the simulation design are based on the wargaming elements in Paragraph 2.1.3 identified prior to this conceptualization. The basic principle for the conceptualization (see Figure 5.2) is, in order to be effective, a commander has his sub(ordinate)-units (the *capabilities*) execute a coherent set of tactical mission tasks (*task 1, 2, n*) in a coordinated action. Coordinated action aggregates parallel or sequential tactical mission tasks and describes how an operation can be executed. Multiple coordinated actions (*COAs 1, 2, m*) can be formulated, as there may be different ways to achieve the identified objective. The tactical mission tasks used for sub-units and the order in which these tasks are used differ for each specified COA. During a wargame, the effectiveness of each formulated COA is valued by the given objective. Whereby each possible COA is influenced by both *environmental factors* and *threats from the adversary*. As wargaming has an iterative nature, the *valuation* of COA may change the initially identified COAs. Subsequently, evaluating the improved COAs can start over again, assuming time is still available.

During a wargame, a sub-unit's attributes change when a unit is engaged in an assigned tactical mission task. These attributes include information about the unit's capability, e.g., location, ammunition and fuel levels, equipment damage, or personnel employability. For example, when the blue team is fighting with its adversary, the red team, fuel and ammunition are consumed, equipment may get damaged or destroyed, and people may get injured or even killed. The tasks in the conceptualization represent tactical mission tasks that military units can execute. Military units conduct these tasks in a specific manner guided by military doctrine. A convoy movement, a tactical movement, an attack by fire, follow and assume[1] or a logistical resupply are examples of tactical mission tasks that military units can execute.

5.2.3. Simulation System Requirements

The above-developed conceptual model leads to the definition of detailed requirements for the simulation system. Based on the original objectives statement, the following requirements provide implementation guidance to design and develop the simulation environment. For further reference, similar to the User Needs, the *Simulation Requirements* are numbered according to the 'system requirement struc-

---

[1] A tactical mission task 'to follow a force conducting an offensive operation and to continue the mission if the lead force is disabled.'

ture' that is presented by Figure 5.3. The following requirement that addresses the identified challenges in Paragraph 3.5 provides the theoretical background for the simulation system requirements:

- *Requirement 2.1.1.* - The behavior of the components in the environment simulating a particular tactical mission task should resemble military doctrine and procedures. This resemblance ensures *consistent attribute mapping* as a mission task simulation component may change their states. The used models should be able to represent different military echelons.
- *Requirement 2.1.2.* - The simulation system should be able to execute a constructive aggregate-level simulation of military operations. Whereby the individual troop or equipment level is the lowest aggregate level. User-controlled *aggregation and disaggregation* is required to prevent *thrashing* of simulation entities and to limit uncontrolled usage of network resources.
- *Requirement 2.2.1.* - The impact of the design on the performance of the Run-Time Infrastructure should be considered by the designer of the simulation system.
- *Requirement 3.1.1.* - As the *resolution* of the echelons is *changed* within the simulation, the behavior of tactical mission task components must be consistent across echelons.
- *Requirement 3.2.1.* - Combining, recombining, configuring, and reconfiguring military units to tailor them to specific operations should be possible.
- *Requirement 3.3.1.* - Interaction between the various echelons should be possible.

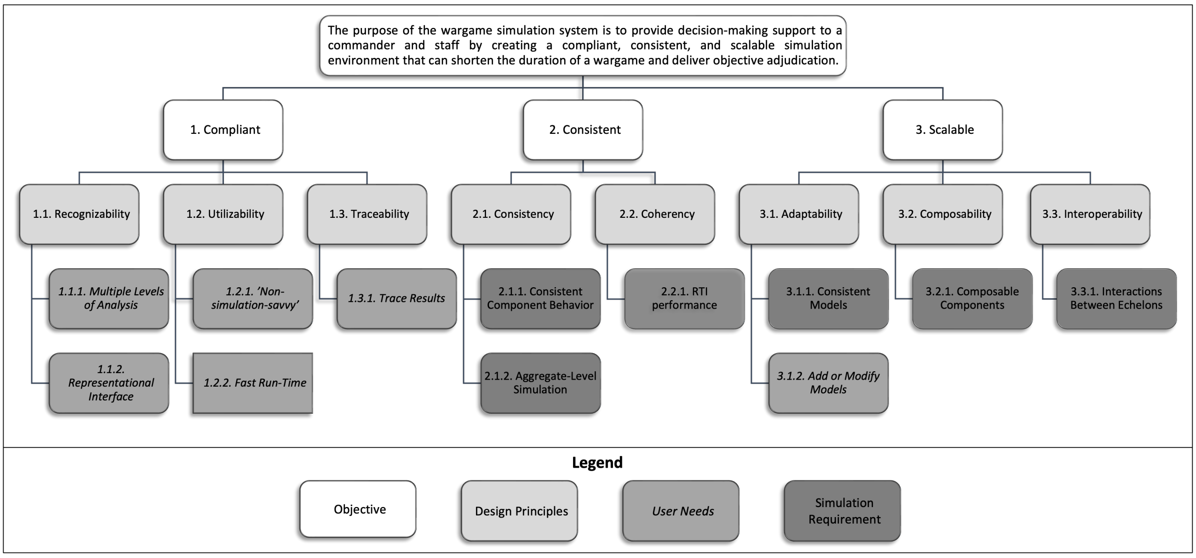


Figure 5.3: System Requirement Structure.

The *User Needs* outlined in Paragraph 5.1.1 and the *Simulation Requirements* explained in Paragraph 5.2.3 are linked to eight overarching design principles that were previously identified in Chapters 2 and 3. In turn, these design principles are associated with the three main functional requirements: (1) Compliant, (2) Consistent, and (3) Scalable, which are part of the defined objective for the wargame simulation system in Paragraph 5.1.2. The linkage of the requirements, through design principles, to the objective is presented in Figure 5.3.

5.2.4. Simulation System Test Criteria

In addition to the above user-defined and theoretical-based requirements, three levels of testing, which are defined by IEEE (2022) for simulation applications, give direction to the development of the experiments that are used within the proof of concept in Chapter 6:

1. **Component testing**, each component is tested to confirm that it can correctly model its real-world representation in the simulation system.
2. **Integration testing**, components can interact correctly according to the designed simulation data execution model, and components can interact correctly with other components.

3. **Interoperability testing**, components can interact according to the defined use case and to the required level of fidelity.

5.3. Wargame Simulation Environment Design

The purpose of step 5 of the DSEEP is to produce the design of the simulation environment that will be implemented in the next section. The first paragraph identifies the applications that will assume some defined role in the simulation system (member applications) that are suitable for reuse and, if required, create new member applications. The second paragraph describes the preparation of the wargame simulation system design. The third paragraph concludes the design of the wargame simulation system by designing individual member applications, where existing member applications cannot fully address all defined requirements for the wargame simulation system.

5.3.1. Select Member Applications

This activity aims to determine the suitability of individual simulation systems to become member applications of the simulation environment. This activity is typically driven by the perceived ability of potential member applications to represent entities and events according to the conceptual model. As the main premesis of this wargame simulation prototype is a design to propose a new simulation concept for military wargaming, the existing member applications are limited to terrain and capability databases.

5.3.2. Design Simulation System

This activity aims to prepare the simulation environment design and allocate the responsibility to represent the entities and actions in the conceptual model to the components. The design for the wargame simulation system is based on Flow-Based Programming. This form of programming is a paradigm created in the late ’60s that defines applications as networks of ’black box’ processes, which communicate via data information packets traveling across predefined connections. The connections between the processes are specified externally to these processes. These black-box processes can be reconnected endlessly to form different networked structures without changing internally (Morrison, n.d.).

The wargame simulation system includes different components to create a flow representing a planned COA. The *Unit-Component* is the start node of each planned task flow. When the simulation is started, this component inserts the ’ORBAT’ in the flow. The *Control-Component* enables the user to include a specific military control measure in the flow of Event-Components. Each flow is closed by the *Report-Component* that enables the user to specify which data is required for further analysis of the COA. The *control-component* enables the user to control the flow of events. The *Event-Component* forms the basis by which the flows can be created. These components represent a specific tactical military task modeled by military doctrine. Whereby they are classified according to the function they represent. The classification for the Event-Components includes i.e. offensive, defensive, movement, and logistical components. Depending on the operation type, event-component classes can be added or removed from the component overview. The design of the wargame simulation prototype is displayed in Figure 5.4.

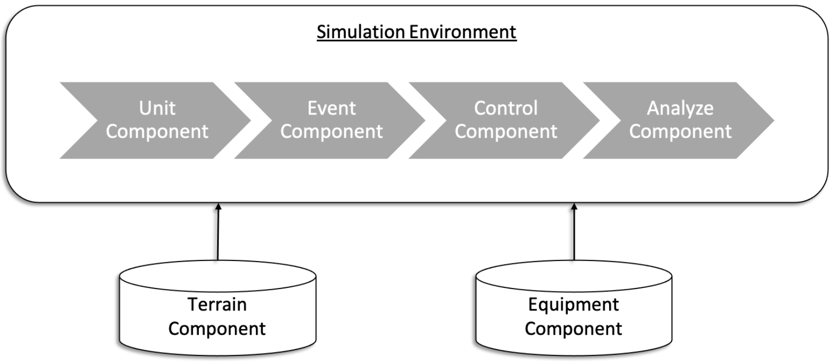


Figure 5.4: Design for the Wargame Simulation Prototype.

The above description of how a combination of different components can create a flow representing a specific COA can be used to create the possible COAs for Blue Team forces and Red Team forces. The question remains: when one or more COAs are created for both sides, how can the simulator within the system simulate a battle between the two warring teams? The answer to this question is provided by the presumption that to start some sort of engagement, both sides want to create some effect in a specific area within the same time frame. For example, the Blue Team wants to neutralize the Red Team forces in a specific area, while at the same time, the Red Team wants to block the movement of the Blue Team through that area. In order to achieve their objectives, the teams have to fight one and the other. An artillery barrage is another example; the Blue Team moves through a specific area. The Red Team can use this movement to conduct an artillery barrage on that location within the time frame to reduce the Blue Team's fighting power.

Subsequently, the simulator must know which Blue Team Event-Components must be connected to Red Team Event-Components within the created flows for each side. By adding engagement areas to the wargame simulation system, Blue Team Event-Components can be connected to Red Team Event-Components. These engagement areas result from the terrain analysis that is part of the decision-making process, and together, these engagement areas form the battle space, an abstract view of the area of operations. These engagement areas are defined in the simulation system configuration to enable the simulator to 'test' the Blue Team COAs against the Red Team COAs. The simulator has to determine whether the flows representing each team COA 'cross' each other at a predetermined engagement area. In the above example, where the Blue Team wants to neutralize the Red Team forces, if the Red Team Event-Component that models the defense of the area is not activated before or at the same time the Blue Team attack Event-Component is activated, the Blue Team flow will continue to the next component, while they do not incur any damage. In the example of the artillery barrage, if the Blue Team is not within the engagement area when the artillery barrage is planned, the Red Team Event-Component for the barrage does not inflict any damage on Blue Team forces. The engagement areas have to be predetermined within the simulation system. When an Event-Component is added to a flow, this component must be linked to a specific engagement area.

5.3.3. Component Design

This activity aims to transform the top-level design for the simulation environment into a set of detailed designs for the components. From the envisioned wargame simulation system, the following components are derived:

- In the **Unit-Component**, the user compiles the unit's ORBAT and specifies the initial attribute states for the unit. This component has four functionalities: the first functionality enables the user to insert an ORBAT, the second enables the user to add a table of equipment, the fourth enables the user to add the personnel composition within the organization, and the fourth functionality enables the user to specify the initial attribute state for the units, equipment, and personnel within the inserted ORBAT.
- Each **Event-Component** in the simulation system that represents a tactical mission task, as executed by military units, can modify the unit's attribute states that may be affected by the activity model. When creating the flow of military activity to plan a specific COA, the user has to configure each Event-Component to input the information that shapes the modeled behavior. For example, to compute the duration of a tactical movement, the user needs to set the travel distance for the unit in the component. The specific component configuration is part of the design of that component, depends on the military tasks being modeled, and can differ per Event-Component. A battle model used in the Event-Component must be able to process the configured unit settings. In addition, the component must be able to simulate the execution of the modeled task, whereby the simulation may change the state of the attributes that reflect the unit's state. Furthermore, the used model in the component may require additional terrain or specific unit data. The component must be able to import this data from external data sources. The unit configuration, initial, and modified attribute states are transferred to the next planned activity when the component has finished its computation. Figure 5.5 displays the generic design for an Event-Component.
- Each **Control-Component** enables the user to include a specific military control measure in the flow of event-components. The military uses control measures to synchronize and control their activities during operations. For example, a checkpoint is a predetermined point on the ground used to control movement, whereby a unit can be ordered to report in, making the unit's location

immediately clear. A Start Line (SL) is another example of a control measure; the line units must cross simultaneously when starting an attack on the opposing forces.

- The **Report-Component** allows the user to store the simulation results for further analysis or comparison with the results of the other simulated COAs. For example, the decrease in the unit's combat effectiveness over time can be the subject of the analysis.

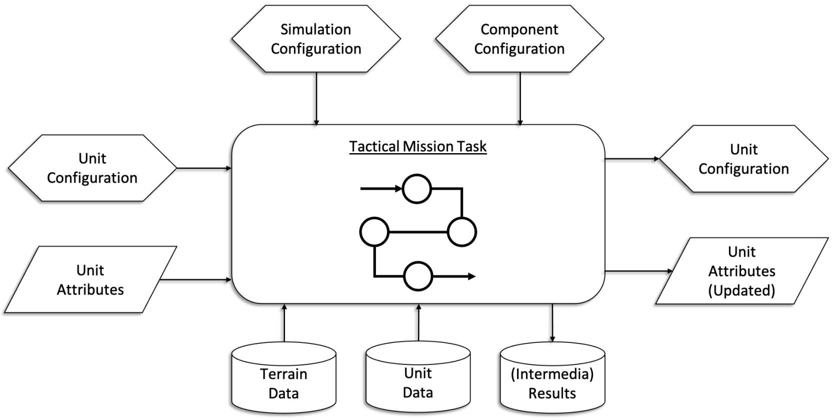


Figure 5.5: Generic design for the Event-Component.

5.4. Simulation Environment Development

The purpose of this step of the DSEEP is to define the information that will be exchanged at run-time during the execution of a simulation system, modify member applications if necessary, and prepare the wargame simulation system for integration and testing. This section starts with a description of the simulation data exchange model. The paragraph section briefly introduces the implementation of the external member applications. The last paragraph presents the mock-up that was created as a way to implement the simulation system.

5.4.1. Simulation Data Exchange Model

For a simulation environment to operate correctly, there needs to be some means for member applications to interact. At a minimum, this implies the need for run-time data exchange. The basic idea of flow-based networks is that every node has a well-defined purpose; it is given some data, does something with that data, and then passes that data on. The network is responsible for the flow of data between the nodes. Within the wargame simulation system, a user can create a *flow* of connected components that act as the nodes in the network. In order to simulate a specific action or control measure, these components require data to configure the model within the component. They may require external data influencing the unit's behavior within the component (the component does not modify this data). Finally, data is inserted into the component that the model may modify before it is transferred to the next component.

By using the SISO C2SIM standard that specifies i.e. force structures for the definition of the ORBAT, it enables sharing of ORBAT-data across simulation and C2 systems, whereby it enables the user to import saved ORBATs or force structures from other applications. Furthermore, using a generic scripting language and standardized data formats, the system ensures that users can easily share, store, and import scenario information with other users. Besides the already mentioned ORBATs, examples of things to share are newly created simulation components, flows that are developed to represent a certain COA, or simulator and component configuration data.

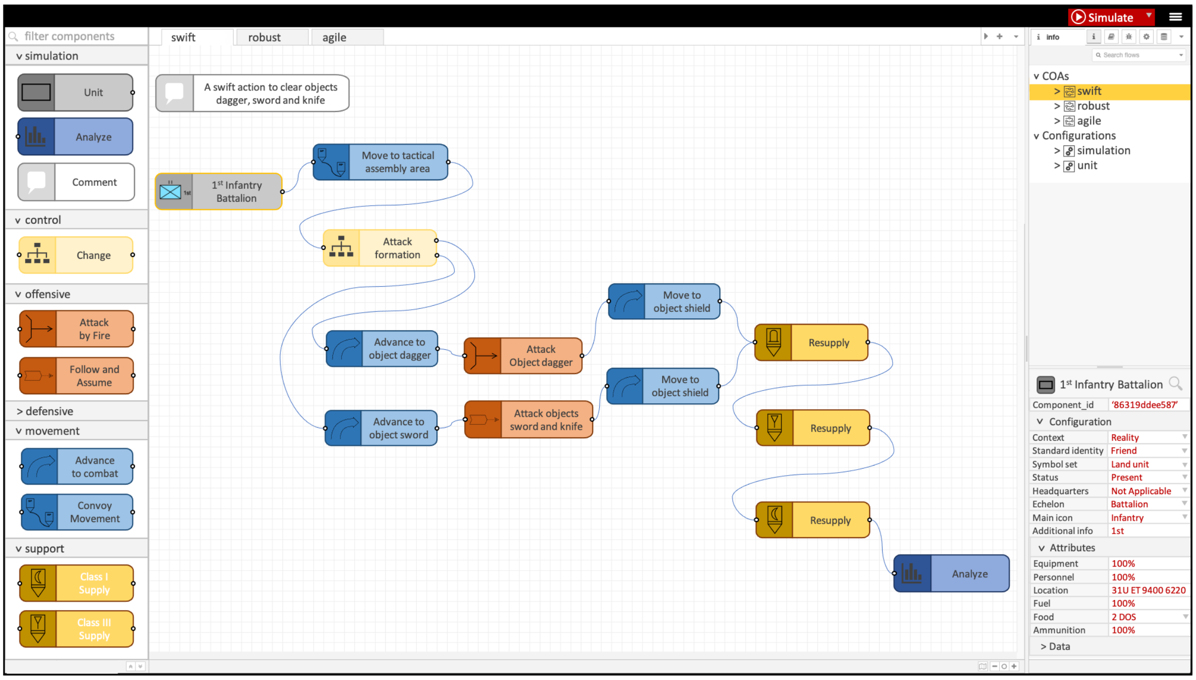
5.4.2. Member Application Implementation

This activity aims to implement modifications necessary to the member applications to represent assigned objects and associated behaviors described in the conceptual model (Step 2) and produce and exchange data with other member applications. The usage of existing member applications is limited to existing terrain databases as it goes beyond the scope of the research to develop a new terrain database service. Additionally, the design of a database with unit and equipment data is beyond this research's scope. Data about military equipment can be classified, whereas adding this data is not allowed as this research is unclassified. For the design of the wargame simulation system, it assumed there are appropriate terrain and equipment databases that can be connected (by middleware or HLA) to the simulation system. The implementation of newly created components is discussed in the next paragraph, together with the implementation of the simulation system.

5.4.3. Implementation of the Wargame Simulation System with Node-RED

This activity aims to implement, configure, and initialize the infrastructure necessary to support a simulation environment and verify that it can support the execution and intercommunication of all member applications. Figure 5.6 shows a possible mock-up for the user interface of the wargame simulation system. This mock-up is based on Node-RED, a low-code programming tool for event-driven applications (OpenJS Foundation & Contributors, n.d.).

The Node-RED programming tool provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its run-time in a single click. A COA can be developed by dragging the 'ingredients' for a COA from the palette on the left side into the workspace. The components can be linked together by wiring their output and input ports. This linkage allows the user to structure the planned tasks for the unit to be built up in the COA. Within the simulation system, the user can create multiple tabs. Users can create a flow on each tab by adding and wiring the components required to simulate a given COA.



A swift action to clear objects dagger, sword and knife

1st Infantry Battalion

Component_id	'86319ddee587'
v Configuration	
Context	Reality
Standard identity	Friend
Symbol set	Land unit
Status	Present
Headquarters	Not Applicable
Echelon	Battalion
Main icon	Infantry
Additional info	1st
v Attributes	
Equipment	100%
Personnel	100%
Location	31U ET 9400 6220
Fuel	100%
Food	2 DOS
Ammunition	100%
> Data	

Figure 5.6: A mock-up for the user interface of the wargame simulation system.

The lightweight RTI of Node-RED is built on Node.js, taking full advantage of its event-driven, non-blocking model. This lightweight RTI makes Node-RED executable on hardware with limited processing power and in the cloud. With over 225,000 modules in Node's package repository, it is easy to extend the range of nodes to add new capabilities (OpenJS Foundation & Contributors, n.d.). Users can use the nodes, the basic building blocks of a flow, to create flows representing a certain COA. Experienced users or system administrators can create nodes that, as Event-Component, represent a specific tactical military task that has not yet been included in the system. These Node-RED features fit well with user needs, so the system should enable fast run times and be adaptable and extensible.

Nodes are triggered by either receiving a message from the previous node in a flow or waiting for some external event, such as an incoming HTTP request, a timer, or a hardware change. They process that message or event and then may send a message to the following nodes in the flow. Furthermore, Node-RED provides a way to store ‘context’ information that can be shared between different nodes[2] without using the messages that pass through a flow. There are three levels of context information. Node information that is only visible to the node that sets the value, flow information that is visible to all nodes on the same flow, and global information that is visible to all nodes. These ’context’ levels in Node-RED are usable to configure the simulator and components, set up the ORBAT that starts a flow, and store the (intermediate) results produced by the created flow of components. The way Node-RED can be configured to process information complies with the Simulation Data Exchange Model developed for the simulation system.

The global configuration of the simulation system enables the user (or the unit’s system administrator) to configure the ’global’ configuration parameters that apply to the whole system. These parameters include the environmental databases used, specific information about unit personnel and equipment, the application of the environment, e.g., for an exercise or an actual operation, or the system’s security level. The configuration required for a specific COA must be inserted by the user when a new ’flow’ for a COA is created, whereby the user can change this configuration. For example, when a user creates a flow for specific COA, the user has to specify which echelon the simulator will use to execute the simulation; when this is set to ’platoon,’ the simulator will configure all Event-Component to simulate their associated tactical mission tasks at the ’platoon level.’

Another example of a configurable parameter for a certain COA is whether the simulator should log all the changes for attribute states that are triggered by the Event-Components in order to enable the user to see what is happening by looking at the intermediate attribute states for each event in the flow. On the other hand, when a user is only interested in the final results, only logging the last attribute state is the appropriate configuration. Specific parameters used by a model used by a component (or node) to alter units’ attributes are stored within the component.

Messages contain the information that flows through the components. In Node-RED, these messages are plain JavaScript objects that can have any set of properties. Thus, it can be modified to represent the required information for a military war game. The units’ attribute states are defined within these JavaScript objects, whereby the message that contains these objects is handed from component to component. The models that are used within the component can modify the JavaScript objects and thereby modify attribute states. For example, if an Event-Component calculates movement time and fuel consumption, the component can change the time and fuel level objects. While Node-RED uses JavaScript for its internal messages, the standard node-set contains nodes to parse JavaScript objects to other data structures (and vice versa), such as the XML file format that is used within C-BML and MSDL.

In Node-RED, messages are the vehicles for information, represented as plain JavaScript objects. These messages can be easily tailored to represent military wargame data, and Node-RED provides the tools for parsing data to different formats, such as XML and CSV. Furthermore, its open-source nature fosters an environment where users can easily create, modify, import, and export nodes and flows, thus facilitating collaboration and sharing among users.

Using Node-RED as the core RTI for this wargame simulation system ensures a composable, flexible, and user-friendly system for *’Non-simulation Savvy’* users. It enables fast configuration, versatile models, and seamless information sharing, all while adhering to the Simulation Data Exchange Model. This low-level coding system aligns well with users’ needs that require fast run times and adaptability, ultimately contributing to the system’s effectiveness. This simulation prototype shows that it is possible to exploit a flow-based architecture to implement a wargame simulation system.

5.5. Prototype Proposal for the Wargame Simulation System

The wargame simulation system prototype will be designed for ease of use, even by users who may not have extensive experience with simulation systems. Whereby the simulation system adheres to established military doctrine and standards. Based on these user needs, the objective for the simulation system is defined as ”The purpose of the wargame simulation system is to provide decision-making support to a commander and staff by creating an accessible, consistent, and scalable simulation system that can shorten the duration of a wargame and deliver objective adjudication.”

The developed conceptual model for wargaming is based on user needs, the simulation objective,

---

[2] The ’nodes’ that are used in the context of this research are the components that are designed as part of the wargame simulation system.

and the described use case. The conceptual model leads to specific requirements for the simulation system's design and development, which include consistent component behavior, aggregate-level simulation, RTI performance, consistent models, composable components, and interaction between echelons. Three levels of testing (component testing, integration testing, and interoperability testing) guide the test criteria for the wargame simulation system's design.

Existing member applications in the conceptual design are limited to terrain and capability databases, as this system aims to propose a new concept for military wargaming. The components' design involves refining the simulation environment's top-level design into detailed designs for member applications. The Unit-Component lets users specify unit organization, equipment, personnel composition, and initial attribute states. Each Event-Component represents a tactical mission task and allows users to configure it for specific behavior. The Control-Component facilitates the inclusion of military control measures in the flow of Event-Components, aiding in synchronization. The Report-Component lets users store simulation results for analysis or comparison, such as tracking changes in unit combat effectiveness over time.

The mock-up of the user interface for the wargame simulation system is used to visualize the infrastructure supporting the simulation system. Based on Node-RED, a low-code programming tool, users can create flows representing COAs by dragging components from a palette and wiring them together. Furthermore, Node-RED provides an appropriate RTI that facilitates data exchange among member applications and the components within a flow.

The proposed design for the wargame simulation system is limited to a prototype or conceptual design, comparable to the high level of abstraction used when formulating an operational concept for a military mission. Nevertheless, many factors are meticulously examined and considered in developing the operational concept in a real-world decision-making process, such as the TBM. However, the current prototype only considers a minimal part of the identified relevant elements crucial for military operations. Although the prototype is designed to evaluate the identified design principles, the possibility of using it for actual military operational analysis is still extremely limited, if not undesirable.

Because this research adopts a 'greenfield' approach for the design of the prototype, the prototype only serves to demonstrate the core features of the wargame simulation system required to evaluate the identified design principles. Integrating a fully functional user interface, which enables military users to experience navigation, data entry, and interaction with the wargame simulation system, is not part of the prototype. This limitation of the prototype does not make it possible yet to assess the current user-friendliness for the military user.

The designed prototype aims to provide solutions to all identified challenges. Nevertheless, it is essential to note that not all design principles related to the challenges are likely to be met, as the requirements are not mutually exclusive. Fulfilling one requirement can negatively impact the fulfillment of another, resulting in a trade-off between those requirements. The proof of concept in chapter 6 will further evaluate the design principles to determine how much the prototype addresses the rigorously identified challenges.

The above proposal concludes the first four DSEEP steps that were used to define the user needs, perform the conceptual analysis, design, and develop the prototype simulation system. The next chapter will use DSEEP steps 5, 6, and 7 to guide the proof of concept of the envisioned wargame system.

6
Proof of Concept

”To kill an error is as good a service as,
and sometimes even better than,
the establishing of a new truth or fact.”
Charles Darwin

A Proof of Concept or Proof of Principle is a conceptual demonstration of an artifact focused on determining whether an idea can be turned into a reality. In the previous chapter, the first four DSEEP steps were used as guidance for designing the wargame simulation system, so it makes sense to use the remaining three steps to prove the conceptual design for the simulation system. However, before putting additional effort into the untested hypothesis underlying the design by applying the three remaining DSEEP steps to evaluate the designed system, this Proof of Concept checks whether the conceptual design can be successful and practically implemented. The goal of this research is not to plan the execution of the simulation (DSEEP step 5), establish all required inter-connectivity between member applications (DSEEP step 6), and test the simulation prototype prior to execution (DSEEP step 7), the goal is to demonstrate whether the idea for the simulation system is feasible and viable.

The subsequent question is, as fulfilling DSEEP steps 5, 6, and 7 is not yet advantageous, which method is suitable for this Proof of Concept? Hevner et al. suggests descriptive evaluation as a method for innovative artifacts for which other evaluation forms may not be feasible. This evaluation method uses information from the knowledge base to build a convincing argument by constructing detailed scenarios around the artifact to demonstrate its utility. This descriptive evaluation of the simulation system is augmented by a quantitative evaluation of the models used within the components.

The first step to prove the concept is to implement the scenario within the wargame simulation system to determine whether the design for the simulation system is correctly implemented in the conceptual model. For this verification of the simulation system prototype, the formulated scenario evaluates the implementation to unveil and remove inconsistencies with the conceptual model. In order to do so, the verification uses the detailed courses of action to provide the input for connecting the components to create the flow of events within the system, whereby the sub-unit tasking defines the information needed to configure the required components to create the flow of events. The second step in this proof of concept is the validation to check whether the accuracy of the simulation system design represents the real wargame system outlined in the scenario. The elements that are required to conduct a wargame, i.e., the ORBATs, Blue Team (BT) COAs and Red Team (RT) COAs within the formulated scenario, are used as the testing data set for the validation. Although the goal of this research is not to use DSEEP - Step 5 to plan the execution of the simulation, the testing levels that are defined by DSEEP Step 5 form the test framework to analyze whether the designed simulation system functions as required.

This chapter starts by constructing a concise scenario for a military operation in the first section. The second section will discuss implementing the components needed to create a COA within the simulation system. The scenario and components are used in the third section to verify whether the designed system can be used for a wargame simulation in support of military decision-making. The fourth section validates the designed components and the design of the wargame simulation data exchange model. Subsequently, the final section analyzes whether the designed system meets the formulated design requirements and identifies the limitations of this new conceptual design for a wargame simulation system.

6.1. Real-World Wargame Scenario

This section explains the first step in Proof of Concept, a description of the scenario that is used to demonstrate the feasibility and viability of the designed wargame simulation system. As part of developing a simulation environment, a conceptualization of the real-world situation wherein the wargame simulation is situated was introduced in the previous chapter. While this conceptualization intends to provide background about the military organization and processes that are relevant for conducting a military wargame, a specific military mission that can be used to develop the COAs that can be analyzed within the designed simulation system has yet not been introduced in Paragraph 5.2.2.

As with any other military decision-making process, several activities must be executed before a conclusive decision based on the insights out of the operational analysis or the *wargame* can be formulated. This scenario uses three stages of the Dutch Army military decision-making process (see Figure 2.2) described by IJntema and van de Haar (2010), as guidance to formulate a concise and fictive scenario that contains the required elements to prove the designed wargame simulation concept. The first paragraph is focused on describing the elements that are required to ’understand’ the mission environment. This understanding of the situation provides the input for the second paragraph describing the elements required to ’asses’ the mission environment. The third stage of the decision-making process, ’decide,’ is not included in the scenario outline, as this stage is about the operational analysis and the selection of the preferred COA. These two activities within the decision stage are not part of the scenario outline formulated for the Proof of Concept, as the decision stage starts with the wargame that is being simulated by the designed system. The purpose of the information in Paragraphs 6.1.1 and 6.1.2 is not to create a representative scenario for a military decision-making process or go through all its steps in detail, but it serves as input to create the necessary components for the Proof of Concept of the wargame simulation system and make it understandable for non-military professionals, while military professionals still recognize it.

6.1.1. Understanding the Mission

After receiving an order, a commander formulates an answer to the following questions in this step: What is the actual problem that needs to be resolved? What is the cause or origin of the problem, and which (f)actors are relevant and contribute to a possible solution? The following information can be used to understand the mission:

- **Mission:** To allow for the 3rd Blue Team Air Assault Battalion (3(AASLT)AMBn) to neutralize Red Team units in object shield, 1st Blue Team Combined Arms Battalion (1(BT)CABn) attacks objects DAGGER, SWORD, and KNIFE.
- **Opposing Forces:** Red Team can deploy an Infantry Company within the Blue Team area of operations. According to Red Team doctrine, an infantry company is equipped with four platoons. The Red Team organization is specified in Figure B.1 that is included in Annex B.1.
- **Threats:** The exact locations of Red Team forces are unknown; presumably, the platoons are deployed to prepare defenses in objects DAGGER, SWORD, and KNIFE. Two possible COAs are formulated for the Red Team (both COAs are visualized in Annex B.1):

    1. In the most likely enemy-COA, the Red Team uses an area defense to block Blue Team Forces. Whereby one Red Team platoon is deployed to retain the fortified hilltop (object KNIFE), two platoons are deployed to battle positions at objects DAGGER and SWORD to block the Blue Team advance; the fourth platoon is withdrawn to support Red Team units defending object SHIELD.
    2. In the most dangerous enemy-COA, the 4 Red Team platoons have occupied battle positions at objects DAGGER and SWORD to canalize Blue Team Forces to the lower grounds within the area of operations. Subsequently, the Red Team uses a mobile defense with prepared battle positions to slow down and wear out the Blue Team forces.

- **Terrain:** 1(BT)CABn area of operations consists of rolling hills creating two avenues of approach to the objects DAGGER and SWORD, whereby trees and bushes cover these objects. Object KNIFE is a key terrain feature located on a prominent hilltop, whereas object SHIELD is within its Fields of Fire. The area of operations for the Blue Team Combined Arms Battalion is roughly 6 kilometers wide and 30 kilometers deep, whereby the distance from Phase Line (PL)-Z(ulu) to the SL is roughly 8 kilometers. The distance from the SL to PL-Y is also roughly 8 kilometers, making the rear area, east of PL-Z, about 15 kilometers deep. Route Pie is 15 kilometers long, and Route Apple is 25 kilometers long.

- **Weather Forecast:** no precipitation expected, partly cloudy, with a low around 15°C.

6.1.2. Assessing the Mission

The next stage in the decision-making process is to formulate an answer to the question: What solutions are possible, given the influence of the operating environment and the capacities available to the mission and therein identified tasks? Based on the intended effects and the Blue Team organization, possible COAs are developed and validated for the Blue and Red Team.

Blue Team Organization

For the execution phase of the operation, the 1st Blue Team Infantry Battalion has detached an Infantry Company from the 2nd Blue Tank Battalion. It has attached a Tank Squadron from the 2nd Blue Tank Battalion. Attachment and detachment transform the battalion into the 1st Blue Team Combat Arms Battalion. The Blue Team organization is specified in Annex B.2 whereby Figure B.3 displays an overview of the Blue Team organizational structure (the ORBAT and the TOE within the Blue Team battalion are shown in Figure B.4.

Blue Team Course of Action

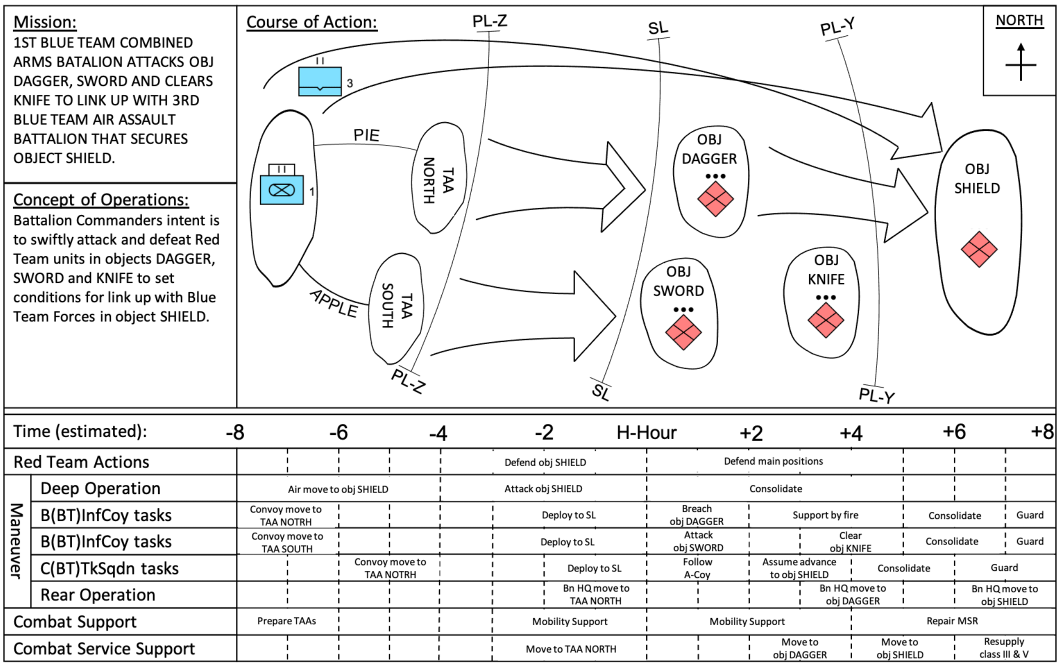
The following COA has been developed for the Blue Team to achieve the determined mission goals. This COA intends to swiftly create a link up with the 3(BT)AASLTBn. Five phases are identified to execute the operation; for phase 3, two alternatives are proposed:

1. The two Infantry companies and the Tank squadron move by route PIE (60 km) and APPLE (40 km) from the battalion Assembly Area (AA) to the tactical assembly areas, Tactical Assembly Area (TAA) North & South (see Figure B.5 in Annex B.2).
2. Within the northern and southern avenues of approach, the infantry companies deploy towards the SL (see Figure B.6 in Annex B.2).
3. The Southern infantry company suppresses Red Team Forces in objective SWORD. The Battalion Headquarters (HQ) moves to TAA NORTH. For the two units in the northern avenue of approach, two variants are developed to advance through object DAGGER:
   (a) The infantry company neutralizes Red Team forces to create a passage through object DAGGER for the tank squadron. The tank squadron follows and assumes the infantry company to continue the link up with Blue Team Air Assault Battalion (see Figure B.7 in Annex B.2).
   (b) The tank squadron attacks by fire Red Team forces in order to break through object DAGGER. The infantry company follows and supports the tank squadron and neutralizes remaining Red Team forces in object DAGGER (see Figure B.8 in Annex B.2).
4. The infantry in the southern avenue of advance company clears objects SWORD and KNIFE from Red Team forces while supported by fire from the northern infantry company (see Figure B.9 in Annex B.2).
5. The Tank squadron and two Infantry companies deploy a screen to the eastern, southern, and northern flanks of the area of operations to protect the Blue Team forces during the resupply. The Battalion HQ moves to object SHIELD whereby the Main Supply Route (MSR) BLUE completes the link-up between the AA and object SHIELD (see Figure B.10 in Annex B.2).

Figure 6.1 summarizes the operational concept and the associated sub-unit tasking, derived from the above COA description. Annex B.2 provides the detailed courses of action for each operation phase and gives an overview of the tasks assigned to the sub-units. It should be noted that the courses of action outlined in the Sync(hronization)Matrix and the detailed courses of action are not to scale and therefore do not have an underlying map.

6.1.3. Decision-Making

The COA and the identified sub-units tasks, displayed by the SyncMatrix, provide the operational analysis or wargame input to determine how the Red Team reacts to Blue Team actions. During the analysis, the Blue Team's actions may be adjusted to better fit the Red Team's actions. Ideally, the analysis



Mission:
1ST BLUE TEAM COMBINED ARMS BATALION ATTACKS OBJ DAGGER, SWORD AND CLEARS KNIFE TO LINK UP WITH 3RD BLUE TEAM AIR ASSAULT BATTALION THAT SECURES OBJECT SHIELD.

Concept of Operations:
Battalion Commanders intent is to swiftly attack and defeat Red Team units in objects DAGGER, SWORD and KNIFE to set conditions for link up with Blue Team Forces in object SHIELD.

Time (estimated):		-8		-6		-4		-2		H-Hour		+2		+4		+6		+8
Red Team Actions						Defend obj SHIELD				Defend main positions								
Maneuver	Deep Operation	Air move to obj SHIELD				Attack obj SHIELD				Consolidate								
	B(BT)InfCoy tasks	Convoy move to TAA NOTRH					Deploy to SL			Breach obj DAGGER		Support by fire			Consolidate		Guard	
	B(BT)InfCoy tasks	Convoy move to TAA SOUTH					Deploy to SL			Attack obj SWORD			Clear obj KNIFE		Consolidate		Guard	
	C(BT)TkSqdn tasks			Convoy move to TAA NOTRH				Deploy to SL		Follow A-Coy		Assume advance to obj SHIELD		Consolidate		Guard		
	Rear Operation							Bn HQ move to TAA NORTH					Bn HQ move to obj DAGGER			Bn HQ move to obj SHIELD		
Combat Support		Prepare TAAs					Mobility Support			Mobility Support				Repair MSR				
Combat Service Support							Move to TAA NORTH					Move to obj DAGGER		Move to obj SHIELD		Resupply class III & V		

Figure 6.1: SyncMatrix showing the concept of operations and sub-unit tasking for a possible COA, the template for this SyncMatrix is derived from Dutch Army Doctrine (Royal Netherlands Army, 2011).

should answer the following questions: What advantages, disadvantages, opportunities, and risks are associated with the COA? A wargame is conducted to answer these questions; the commander determines that the timings of the deployment to the SL and the sequence of the units in the northern avenue of approach are the two critical parts of the operation that the wargame must address. Finally, based on the previous steps, experience, intuition, and the higher commander's intention, a commander decides which COA delivers the most significant opportunity to fulfill the mission. The wargame simulation system provides additional digital means to prepare and execute such an operational analysis before making the final decision. The following section uses the outlined scenario to implement the components required to create an COA within the simulation system.

6.2. From a SyncMatrix to a Flow of Events

The Blue Team organization outlined in Annex B.2.1 needs to be inserted into the Unit-Component to compile the unit's ORBAT and specify the initial attribute states for the unit. Several tasks are identified and assigned to the sub-units to execute the COA developed within the scenario. These tasks, listed in Annex B.2.3, are linked in Table C.1 to the corresponding Event-Components representing the required tactical mission tasks. Table 6.1 provides an overview of Event-Components and related tactical tasks for each Event-Component category required to formulate the wargame scenario. When the Blue Team organization changes during the operation, the changes have to be included in the flow of events by the Control-Component, which makes it possible to include specific military control measures in the flow of Event-Components.

Furthermore, to be able to compare with the results of the other simulated wargames between the Blue COA and Red Team COAs, the Report-Component needs to be tested whether stored simulation results are usable for the decision-making in the third stage of TBM. As the implementation of the member applications is already discussed in Paragraph 5.4.2, the subsequent paragraphs address the first step to proof the conceptual design of the components by explaining how their design is based on the associated wargame elements, how they can be implemented and what their purpose is within the overarching simulation system design.

6.2.1. Unit-Component

The Blue Team organization that is created for the scenario (see Annex B.2.1) shows the ORBAT for the operation. This ORBAT consists of the structure of the Blue Team, the main combat platforms assigned to the units, and the distribution of personnel over the various units that form the Combined Arms Battalion. As outlined in the scenario, the battalion has detached one of its infantry companies from another battalion and has attached a Tank Squadron for this operation. As the capabilities for a mission are usually established by the higher level prior to the start of the operation, it is up to the commander and his staff to compose the available capabilities in such a way that the 'order of battle' fits the operational concept for a specific COA. Subsequently, each flow representing an COA should have a dedicated Unit-Component specifying the required ORBAT.

The implementation of an ORBAT editor within the Unit-Component that uses military map symbols, as defined in “NATO Standard APP-6 - NATO Joint Military Symbology”, to visualize the ORBAT enables users to create or import the organizational structure, table of equipment and personnel composition that have been developed for an COA. Additionally, the user can specify the initial unit, personnel, and equipment states, e.g., the amount of food (supply class I), fuel (supply class III), or ammunition (supply class V) that are available for the operation. Furthermore, within the Unit-Component, the user can specify the initial unit's combat readiness, e.g., when a unit has already suffered damage in a previous operation that has reduced its combat effectiveness.

The Unit-Component is designed to represent the way military staff creates Orders of Battle (ORBATs), whereby the component uses organizational structure, table of equipment, and personnel composition to generate the configuration data and the defined unit, personnel, and equipment states to generate the initial attribute states. Static attributes for units, personnel, and equipment that do not change during the simulation, e.g., a platform's ammunition storage capacity or fuel consumption, are stored and can be changed in a separate database or databases outside the wargame simulation system.

6.2.2. Event-Component

The tactical mission tasks overview in Table 6.1 that results from the scenario shows four categories that will use the conceptual design for the Event-Component. It should be noted that the Event-Component categories are not limited to the four identified categories; defensive, amongst other activities, is not included in the formulated scenario due to its offensive nature. How a tactical mission task should be executed is part of military doctrine and is internalized by military training and exercises. Subsequently, military personnel uses military doctrine and experience to create the mental models to compose the tactical tasks into a COA. Subsequently, when implementing an Event-Component, the foremost step is to select an existing model or design a new model that resembles that tactical task.

Lanchester (1956)[1] Laws are commonly used for modeling combat; although they excel in simplicity, it is also their main limitation. As these laws model how a fight between two parties might go and what the outcome might be, they are suited for offensive and defensive combat, as ”that is just a matter of perspective .” These laws can be enhanced by including additional factors, i.e., to model the terrain effects, a platform's weapons effects, and protection level and factors to include the entity's posture. Subsequently, implementing Lanchester's Laws within an offensive or defensive Event-Component is commonly applied within existing simulation systems. However, examination of existing wargame simulation(s) and ontologies shows only a limited set of tactical tasks that can be used or are included. They are, subsequently, limiting the user when using such environments.

The Event-Component is designed to decouple the model representing a specific military task from the system's components to simulate the COA. As the models can be developed independently from a simulation environment, developing and adding models for new or not yet included tactical tasks can be done without modifying the whole wargame simulation system. While in the defined scenario, only five different offensive tactical tasks are used, it is possible to use these activities by replacing them with one 'generic combat' Event-Component, based on Lanchester's Laws. The various factors that play a role in the application of the laws already provide variances in the outcomes of the different simulated engagements[2]. Although the outcomes of the simulated engagement may be valid, it becomes easier

---

[1] In order to strengthen his military thesis, Lanchester was forced to analyze relations between the combatants in different kinds of combat and published many of his ideas in 1914 in the journal Engineering (Wrigge et al., 1995).

[2] The outcomes of different engagements may vary, yet if a particular single engagement is simulated again, the outcome will not vary due to the deterministic nature of Lanchester's Laws.

to verify the outcomes as more specific models describing the required tactical activities are used to create additional Event-Components. Paragraphs 6.3 show how a ’generic combat’ Event-Component can be implemented by showing the required tactical behavior and how it complies with the formulated simulation data exchange model.

Table 6.1: Overview of identified Components and tactical tasks.

Event-Component	Tactical Task	Description (mental model) (NATO, 2023)
Movement	Convoy	A movement of a group of vehicles organized for the purpose of control and orderly movement with or without escort protection.
	Tactical	The movement of personnel and/or materiel to or from nodes and within an assigned area of operations.
Offensive	Breach	To force a passage through an obstacle or fortification.
	Attack by fire	To engage a target with direct and/or indirect fire without closing with that target.
	Neutralize	To render a hostile entity or its materiel temporarily incapable of interfering with friendly forces.
	Clear	To ensure an area is free of enemy troops and their obstacles.
	Follow and assume	To follow a force and be prepared to continue its assigned mission.
Support	Supply Class III	Resupply of used fuel.
	Supply Class V	Resupply of used ammunition.
	Clear route/area	The total elimination or neutralization of an obstacle, usually performed by follow-on engineers and not done under fire.
	Repair road	Prioritized restoration of the minimum operating capability of essential road infrastructure.
Control	Consolidate	Organizing and strengthening a newly captured position so that it can be used against the enemy.
	Screen	A security element whose primary task is to observe, identify and report information, and which only fights in self-protection.
	Link up	The point where two infiltrating elements in the same or different infiltration lanes are scheduled to meet to consolidate before proceeding on with their missions.

6.2.3. Control-Component

Control measures are directives given graphically or orally by a commander to sub-units to assign responsibilities, coordinate fires and maneuvers, and control combat operations. The planned attack on the Red Team forces in the objects DAGGER and SWORD, whereby Blue Team forces (the two infantry companies) launch a coordinated attack over two avenues of advancement and the crossing of the SL at H-Hour are two examples of such control measures. Control measures can be points, lines, or areas. Subsequently, when implementing a Control-Component, it has a geographical attribute, and it triggers a predetermined behavior by the unit that flows through the component.

The control measures consist of various military symbols that require an understanding of the underlying military doctrine. Together with tactical mission tasks, the control measures enable the military staff to create and understand concepts of operations. They provide the graphical language to sketch a military operation. Some are used as visual support, for example, to point out a specific object, e.g., objects DAGGER and SWORD, or to visualize the direction of attack. In other situations, the control measure affects the wargame simulation. In the first situation, where the control measure is only used as a visual support and does not directly affect the simulation, the actual event has to be handled by an Event-Component, e.g., neutralize the enemy in object DAGGER. In the latter situation, where the control measure affects the simulation, a Control-Component has to be inserted, e.g., wait until H-Hour before crossing the SL and pause simulation until the starting time for the next event in the flow.

The Control-Component is designed to enable the user to control the flow of events. In the wargame simulation system, these components are, by definition, unable to change the units’ attribute states. However, in a real-world scenario, a platform that has to wait before it crosses the SL does consume some fuel, subsequently changing the fuel state of the platform. In the simulation system, these attribute state changes due to control measures are not modeled because they only trigger marginal changes. If the control measure results in an actual activity or task, an Event-Component can still simulate the unit attribute state changes. Subsequently, control measures that can be constructed by (a combination of) tactical mission tasks and are used to support the concept of operations graphically are not implemented in the prototype for the simulation system.

6.2.4. Report-Component

A designated staff member notes the operational analysis's (intermediate) details during a wargame. These notes help the commander to recall the wargame when selecting the preferred COA; staff members can improve or address issues within the preferred COA and use these notes to finalize or change the order that will be given to the sub-units. Subsequently, the Report-Component allows the user to store the simulation results for further analysis or comparison with the results of the other simulated COAs.

The Report-Component enables the user to export the required data in the appropriate format for further processing or analysis. The user can select which file format the selected state attributes must be exported. A CSV table with the state changes for a specific attribute at each component in the flow is an example of the export file a certain Report-Component can deliver. A PNG image containing a graph showing the state changes over time is another example of an export file a certain Report-Component can deliver.

6.3. Experiment Setup and Simulation Results

DSEEP step 5 provides a usable perspective to test that all of the designed components can interoperate to the degree required to achieve core objectives. The formulated scenario in Section 6.1 provides the real system used to validate the wargame simulation system. The first paragraph tests the Event-Component to validate whether the component output corresponds with the output of the modeled real-life tactical activity. However, before the output by the Event-Component can be validated, the system integration test has also had to be started; due to the flow-based design, an Event-Component can not function without input from the Unit-Component and the external databases. Therefore, the individual component and system integration tests are combined in one experiment, further outlined in the first paragraph. Combining a Blue Team Event-Component and a Red Team Event-Component to simulate a battle in a predetermined engagement area adds a layer of complexity to the simulation system, which is tested by the system interoperability experiment described in the second paragraph. The second paragraph uses this inter-operation test to validate whether the linkage between a Blue Team Event-Component and a Red Team Event-Component for a combat simulation corresponds with the real war game for that specific combat situation. The total flow of connected components created within the simulation system workspace must resemble the planned sub-unit tasks presented by the SyncMatrix in Figure 6.1. Following the real-world scenario, the two identified critical parts of the operation, (1) the deployment to the Start Line and (2) the sequence of the units in the northern avenue of approach, deliver the input for the executed experiments. The setup of these experiments, where the first experiment consists of two tests and the second consists of four tests, is shown in Table 6.2. The output of the experiments is presented in the last paragraph.

Table 6.2: Experiment setup for the validation the validation of the simulation system.

Experiment	Test	Setup
1.	1.	Blue Team on-road movement from AA to TAAs
	2.	Blue Team off-road movement from TAA to SL
2.	1.	Blue Team SWIFT, Alternative A x Red Team Most Likely COA
	2.	Blue Team SWIFT, Alternative B x Red Team Most Likely COA
	3.	Blue Team SWIFT, Alternative A x Red Team Most Dangerous COA
	4.	Blue Team SWIFT, Alternative B x Red Team Most Dangerous COA

6.3.1. Experiment 1. - Simulation Component Integrity

In this activity, each type component is tested to confirm that any flow of components is correctly implemented, whereby the Event-Component calculates attribute state changes that correspond with the expected outcomes of the tactical tasks that form a specific COA. The Event-Component categories selected for this experiment are convoy movement and tactical movement, as these components do not require any input from a component created within a Red Team flow. The experiment evaluates whether the movement times generated within the Event-Component correspond with the movement times for Blue Team forces from the Assembly Area via the Tactical Assembly Areas to the SL as determined by the SyncMatrix. As the tasks within the SyncMatrix are assigned for the Blue Team at the company echelon, the simulator is configured to run the models within the Event-Components at the company echelon for the Blue Team COAs.

Based on the Blue Team COA, a flow with a selection of components is created to test the individual components and their integration. Figure 6.2 show this flow of components that starts with a Unit-Component to create the Blue Team unit organization and its initial state; the second and fourth component model the unit's subsequent convoy and tactical movement towards the SL, the Control-Component changes the time to correspond with the time that is planned for the units to cross PL-Z, the last component exports a selection of the data in the determined file format.

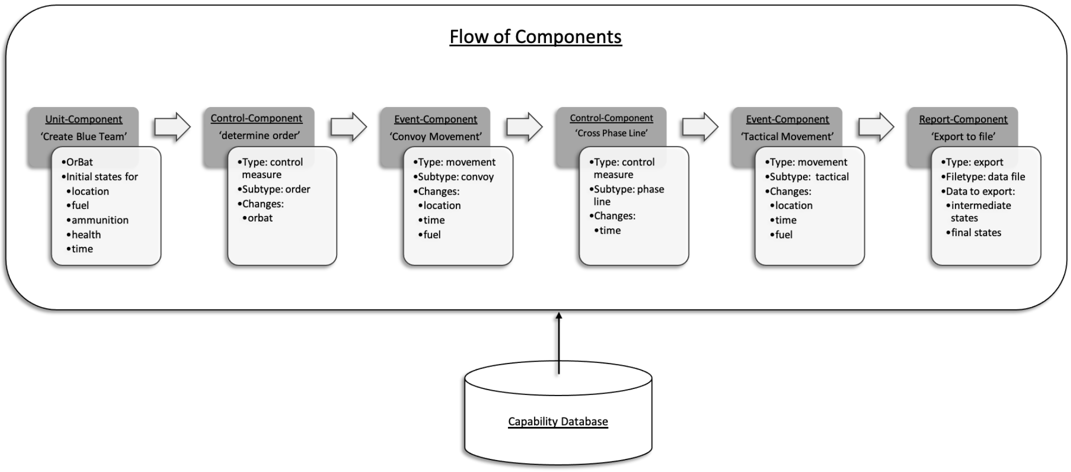


Figure 6.2: The selection of components based on the Blue Team COA that form the flow for the component integrity experiment.

The 'test' flow of components starts with the Unit-Component. The first step to set up the Unit-Component is to create the organizational structure; the ORBAT that is included in Annex B.2.1 provides the blueprint for the organization that has to be inserted into the component. The next step is configuring each (sub-)unit with the additional information required for the simulation. Table C.5 shows an example configuration, as well as the attribute, states that will be used to test the elements within the flow, whereby it should be noted that the majority of the information is generated automatically based on the ORBAT, and could change to the state the user want to use for the simulation. The Unit-Components generate this data for the top-level echelon and each sub-unit within the defined ORBAT.

In the real-world scenario modeled by the SyncMatrix, a convoy movement is a road movement where the unit's vehicle moves in a column formation with a predetermined distance or waiting time (V) between the vehicles. Furthermore, in case of an incident, extra time (S) between the sub-units is included in the convoy movement to create some reaction time. For the movement, the model determines the vehicle time (T) by calculating the time for the move from the starting point to the end point of the route. Furthermore, it will cost some preparation time (P) to move the vehicle to the proper formation at the starting locations, and it will take some completion time (C) to park the vehicles at the destination. The above tactical model for a convoy movement is used to model the total movement time (M) for each convoy movement of the two infantry companies and tank squadron within the Convoy-Movement Event-Component and based on the calculated movement time of each vehicle and the time the vehicle has to wait, multiplied by the vehicle's fuel consumption (f) and the number of vehicles (n) the total fuel consumption (F) is calculated for the movement of a unit.

$$M = P + V + S + T + C \tag{6.1}$$

$$F = \frac{f(n(P + T + C) + V + S)}{60} \tag{6.2}$$

where:

M = total movement time (min)
P = movement preparation time (min)
V = total waiting time between vehicles (min)
S = total waiting time between sub-units (min)
T = vehicle travel time (min)
C = movement completion time (min)
F = total fuel consumption (l)
f = vehicle fuel consumption (l/h)
n = number of vehicles (-)

After the simulator has executed the component, it changes with a new timestamp, the attribute states for the unit location, and the remaining fuel supply. The Event-Component uses the unit configuration and capability data to calculate the changes for appropriate attribute states. In this conceptual design, the user has to insert the travel distance into the component as a component that delivers automatic routing based on the position of the units, vehicle type, and environmental conditions that are not yet anticipated in the system. Equation 6.1 provides the abstracted movement model that is used by the component to calculate the *total movement time (T)* and Equation 6.2 provides the model used to calculate the *total fuel consumption (F)* that is consumed during the engagement.

Test 1.1. - Blue Team on-road movement from AA to TAAs
The tactics for road movement prescribe for vehicles and units to move one by one in a column formation over the given route. Only with multiple routes can units move parallel. Therefore, it is required to add a control measure into the part of the flow that models the road movement in the northern avenue of approach. In this control measure, the sequence for the road movement is set to A(BT)InfCoy as the first unit and C(BT)TkSqdn as the second unit. Subsequently, an Event-Component has to be able to execute the simulation of the road movement model for each unit that flows through the component.

Test 1.2. - Blue Team off-road movement from TAAs to SL
The tactics for a tactical movement describe different formations for vehicle movement, whereby vehicles are not limited to following roads. Tolk (2012) identifies three primary forms: line, column, and wedge-shaped formations. Vehicles can cover the determined distance without having to wait on each other.

6.3.2. Experiment 2. - Simulation Environment Interconnection

In this experiment, the ability of a simulation environment to inter-operate is tested. This interoperability includes observing the ability of components to interact according to the defined scenario and the level of fidelity required for the application. The results from this testing may contribute to verifying and validating the wargame simulation system as required. For this experiment, an Event-Component of the Blue Team is linked to an Event-Component of the Red Team to confirm that that the engagement between the two teams is correctly simulated, whereby an Event-Component calculates attribute state changes that correspond with the expected outcomes of the tactical tasks that form a specific COA. To evaluate if the system can simulate different engagements between warring teams, the tactical tasks of the Blue Team to neutralize object DAGGER and the tasks of the BLUE TEAM to clear object KNIFE are selected as part of the experiment. The experiment evaluates whether the reduction of the health and the outcome of the battle calculated within the Event-Component correspond with the the expected outcome of the engagements in the scenario. As the tasks within the SyncMatrix are assigned for the Blue Team at the company echelon, the simulator is configured to run the models within the Event-Components at the company echelon for the Blue Team COAs.

Based on the Blue Team and Red Team COAs, two flows with a selection of components are created, whereby the two flows are interconnected through the determined engagement areas. Figure 6.3 shows both flows of components that start with a Unit-Component to create the Blue and Red Team units organization and its initial state; the second component in each flow models the tactical mission tasks for the Blue and the Red Team. In order to simulate an engagement, the model in the Event-Component needs information about both sides. With its fighting power and protective capabilities, the Blue Team Event-Component information is transferred to the Red Team Event-Component, and vice versa, when both Event-Components are simultaneously active within the same engagement area. As with the first experiment, the last component exports a selection of the data in the determined file format.

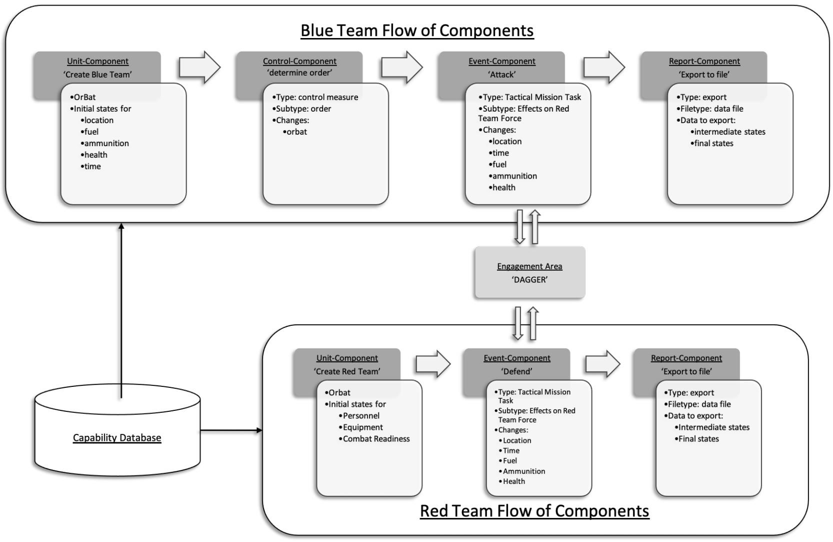


Figure 6.3: Engagement Simulation by connecting the Blue Team Event-Component to the Red Team Event-Component through engagement area 'DAGGER'.

In the real-world scenario outlined in the SyncMatrix, an engagement between the teams is a battle where the fighting power of the teams determines the outcome. While the military doctrine is a vibrant language that includes in-depth knowledge of executing tactical mission tasks, Lanchester's Laws provide a generalized model that can be used to adjudicate combat engagements. Lanchester's square law calculates the number of soldiers lost on each side using the following equations. Here, dR/dt in equation 6.4 represents the rate at which the Red Team size is changing at a particular instant, and dB/dt in equation 6.3 represents the rate of the Blue Team size is changing at a particular instant. Their overall state is expressed by health, fuel, and ammunition. In the wargame simulation system, the military units have combat power and combat vulnerability, which are abstract values that represent the relative strengths and protection of different units. When opposing teams come into contact, they damage each other until one can no longer inflict damage on the other team. A unit is considered *hors de combat* when its health, ammunition, or fuel reaches zero. Combat power is the ability of a unit to cause casualties and damage. It is a function of how many weapons it has, the size (destructive power) of its munitions, the rate of fire, and the weapons' accuracy. Combat vulnerability is the ability of a unit to take on enemy fire. It is a function of its protection by armor, its posture, and the impact angle of the enemy fire. The factors $alpha$ and $beta$ in both equations determine the relative fighting strength. This relative fighting strength is calculated for $alpha$ by dividing the fighting strength of the Red Team by the protective measures of the Blue Team and for the $beta$ by dividing the fighting strength of the Blue Team by the protective measures of the Red Team (the detailed computational models for the calculation of $alpha$ and $beta$ are included in Annex C.2.2). The fighting strength and protective measures are determined by a combination of user configuration of the Event-Component, data acquired from the capability database, and model parameters. Based on the calculated engagement time (T), multiplied by the vehicle's fuel consumption (f) and the number of vehicles (n), the total fuel consumption (F) is calculated by equation 6.5 that is consumed during the engagement.

$$\frac{dB}{dt} = -alpha * R \tag{6.3}$$

$$\frac{dR}{dt} = -beta * B \tag{6.4}$$

$$F = \frac{fnT}{60} \tag{6.5}$$

where:

B = Blue Team size
R = Red Team size
t = combat round (-)
$alpha$ = relative fighting strength Red Team (min^{-1})
$beta$ = relative fighting strength Blue Team (min^{-1})
F = total fuel consumption (l)
f = vehicle fuel consumption (l/h)
n = number of vehicles (-)
T = engagement time (min)

As the simulator executes the component, it changes with a new timestamp the attribute states for the unit's remaining health, ammunition, and fuel supply. Equations 6.3 and 6.4 provide the abstracted Lanchester's model that is used by the component to calculate the outcome of the engagement, and Equation 6.5 provides the model used to calculate the *total fuel consumption (F)* that is consumed during the engagement. The following paragraph will discuss the outcomes of this experiment. Two alternatives within the Blue Team COA are juxtaposed to the two Red Team COAs, creating four tests within the second experiment.

Evaluation of the sequence of tasks for the Blue Team in the SyncMatrix shows that in order to create a flow for the *follow and assume* tactical mission task, the order of the units has to be configured before the flow starts the unit in front's tactical activity. Therefore, an additional Control-Event has to be inserted to set the sequence of the sub-units. For Alternative A of the Blue Team COA within the Control-Component, the A(BT)InfCoy is set as the first unit, and C(BT)TkSqdn is set as the second unit, while for Alternative B C(BT)TkSqdn is set as the first unit and A(BT)InfCoy is set as the second unit. Furthermore, it should be noted that the Event-Component is designed only to model an engagement; no unit movements are included in this component. In all four tests, the time calculated by the Event-Component for the duration of the engagement consists only of the actual time the teams are engaged in combat. The same applies to the fuel consumption, which only represents the fuel consumed during the engagement. To simulate the movement of the Blue Team units through object DAGGER to PL-Z, an additional Event-Component that simulates the tactical movement has to be added to the flow of components.

Test 2.1. - Blue Team SWIFT, Alternative A against Red Team Most Likely COA
The engagement between the A(BT)InfCoy and the Red Team forces in object DAGGER is simulated in this first test of the second experiment. When the Event-Component has finished the simulation of the planned neutralization of the Red Team forces, the flow proceeds to the next second Event-Component. The second Event-Component in the flow determines whether there are any remaining Red Team forces in object DAGGER; if so, it simulates the required neutralization of Red Team forces by the C(BT)TkSqdn. If no Red Team forces are present at the start of the second Blue Team Event-Component that flows through object DAGGER, in other words, the Red Team Event-Component for defending object DAGGER has reached 0% health, whereby the Event-Component cannot be active anymore in Engagement area DAGGER subsequently not triggering another engagement. Hence, the Blue Team flow proceeds to the next component.

Test 2.2. - Blue Team SWIFT, Alternative B against Red Team Most Likely COA
In this second test of the second experiment, the engagement between the C(BT)TkSqdn and the Red Team forces in object DAGGER is simulated. The Event-Component finishes the simulation of this engagement when the Red Team forces are below 50%, whereby the C(BT)TkSqdn breaks through the Red Team defenses at object DAGGER; subsequently, the flow proceeds to the next second Event-Component. The second Event-Component in the flow determines whether any remaining Red Team forces are in object DAGGER; if so, it simulates an engagement between A(BT)InfCoy and the remaining Red Team forces. The flow proceeds to the next component if no Red Team forces are present at object DAGGER.

Test 2.3. - Blue Team SWIFT, Alternative A against Red Team Most Dangerous COA
The engagement between the A(BT)InfCoy and the Red Team forces in object DAGGER is simulated in this third test of the second experiment. Two successive Event-Components simulate the canalization where the Blue Team A(BT)InfCoy and successively C(BT)TkSqdn engage the Red Team forces. When the Event-Component has finished the simulation of this engagement, the flow proceeds to the next second Event-Component. The second Event-Component in the flow determines whether there

are any remaining Red Team forces in object DAGGER; if so, it simulates an engagement between C(BT)TkSqdn and the remaining Red Team forces. The flow proceeds to the next component if no Red Team forces are present at object DAGGER.

Test 2.4. - Blue Team SWIFT, Alternative B against Red Team Most Dangerous COA
The engagement between the A(BT)InfCoy and the Red Team forces in object DAGGER is simulated in this third test of the second experiment. Two successive Event-Components simulate the canalization where the Blue Team C(BT)TkSqdn and successively A(BT)InfCoy engage the Red Team forces. When the Event-Component has finished the simulation of this engagement, the flow proceeds to the next second Event-Component. The second Event-Component in the flow determines whether there are any remaining Red Team forces in object DAGGER; if so, it simulates an engagement between C(BT)TkSqdn and the remaining Red Team forces. The flow proceeds to the next component if no Red Team forces are present at object DAGGER.

6.3.3. Results of the Experiments

The results for the convoy movement from the Assembly Area to the Tactical Assembly Areas show that the Event-Component for the convoy movement can, based on the movement model, the determined model parameters, and the connected capability database, calculate the time and associated attribute state changes for the tactical tasks it simulates. Table 6.3 provides an overview of the input and output of the component that simulates the convoy movement. The timestamp for the C(BT)TkSqdn deviates from the other two units, as the unit has to wait on another unit, the A(BT)InfCoy, which moves first to the TAA by route PIE.

Table 6.3: Input and output of the Event-Component for the on-road-movement tasks.

I/O	Unit	Attribute	Time	State
Input	A(BT)InfCoy	Location Health Ammunition Fuel	-480	AA 100% 100% 100%
	B(BT)InfCoy	Location Health Ammunition Fuel	-480	AA 100% 100% 100%
	C(BT)TkSqdn	Location Health Ammunition Fuel	-480	AA 100% 100% 100%
Output	A(BT)InfCoy	Location Health Ammunition Fuel	-388	TAA-N 100% 100% 90%
	B(BT)InfCoy	Location Health Ammunition Fuel	-388	TAA-S 100% 100% 90%
	C(BT)TkSqdn	Location Health Ammunition Fuel	-297	TAA-N 100% 100% 88%

The starting time, determined within the concept of operations for the tactical movement from the Tactical Assembly Areas to the SL, is set by the Control-Component to the time of -4 hours (-240 minutes). Subsequently, the input time for the Event-Component for the tactical movement is set to -240. The test results for the tactical movement show that the movement model can, based on the movement model, the determined model parameters, and the connected capability database, calculate the time and associated attribute state changes for the tactical tasks it simulates. Table 6.4 provides an overview of the input and output of the component that simulates the tactical movement.

Table 6.4: Input and output of the Event-Component for the off-road-movement tasks.

I/O	Unit	Attribute	Time	State
Input	A(BT)InfCoy	Location	-240	TAA
		Health		100%
		Ammunition		100%
		Fuel		90%
	B(BT)InfCoy	Location	-240	TAA
		Health		100%
		Ammunition		100%
		Fuel		90%
	C(BT)TkSqdn	Location	-240	TAA
		Health		100%
		Ammunition		100%
		Fuel		88%
Output	A(BT)InfCoy	Location	-97	SL
		Health		100%
		Ammunition		100%
		Fuel		70%
	B(BT)InfCoy	Location	-97	SL
		Health		100%
		Ammunition		100%
		Fuel		70%
	C(BT)TkSqdn	Location	-111	SL
		Health		100%
		Fuel		64%
		Ammunition		100%

Table 6.5 provides an overview of the in- and output of the component that simulates the engagement; the attribute states for the B(BT)InfCoy are not included in this table as this unit is not part of this test. The four test runs clearly show the impact on the Blue Team by the Red Team COAs. The Most Dangerous Red Team COA is indeed the most dangerous, which means that the simulation corresponds with the interpretation of both COAs in the outlined scenario. The test further shows that *Alternative B* of the Blue Team COA performs better when the Blue Team engages with the Red Team forces. The Blue Team ammunition consumption during the engagement with the Red Team in the Most Dangerous COA also reveals an issue the Blue Team commander and staff must address, as the engagement requires more ammunition than the units have available.

Table 6.5: Input and output of the Event-Components for the inter-operation tests.

		Test	**2.1** SWIFT A x ML		**2.2** SWIFT B x ML		**2.2** SWIFT A x MD		**2.2** SWIFT B x MD	
I/O	**Unit**	**Attribute**	**Time**	**State**	**Time**	**State**	**Time**	**State**	**Time**	**State**
Input	A(BT)InfCoy	Location	0	SL	0	SL	0	SL	0	SL
		Health		100%		100%		100%		100%
		Ammunition		100%		100%		100%		100%
		Fuel		85%		85%		85%		85%
	C(BT)TkSqdn	Location	0	SL	0	SL	0	SL	0	SL
		Health		100%		100%		100%		100%
		Ammunition		100%		100%		100%		100%
		Fuel		82%		82%		82%		82%
Output	A(BT)InfCoy	Location	8	DAGGER	5	PL-Y	42	DAGGER	19	PL-Y
		Health		0%		97%		60%		80%
		Ammunition		100%		14%		-161%		-13%
		Fuel		85%		84%		78%		80%
	C(BT)TkSqdn	Location	8	DAGGER	2	PL-Y	51	DAGGER	37	PL-Y
		Health		100%		100%		97%		89%
		Ammunition		100%		48%		-114%		-543%
		Fuel		82%		82%		82%		76%

The two experiments' intermediate output of the created Event Components is presented in Annex

D.1.

6.4. Analyze Data and Evaluate Requirements

This section aims to analyze the data obtained during the designed system's implementation, verification, and validation and evaluate whether the results are consistent with the formulated requirements. This confirmation of the results makes it possible to determine whether the designed wargame simulation system meets the formulated objective and helps identify this design concept's limitations. The following paragraphs link the experiment results to the different user needs and simulation requirements presented by Figure 5.3. For each requirement is determined whether it passes or fails.

6.4.1. Requirement 1.1.1. - Multiple Levels of Analysis

As commanders and their staffs execute missions on different military levels or echelons, the system should be able to simulate the planned operation on the different operational and tactical planning levels. Evaluation of the components application shows that it is not easy to configure the used models to the desired analysis level.

The movement models initially calculate the results based on the individual platform, after which this is adjusted to the desired level. Lanchester's laws are more accessible for different echelons because these models assume that each echelon's relative combat strength can be determined. It should be noted that the amount of fighting power is considered constant as a product of the size and strength of a unit. If the relative size decreases at a higher echelon (e.g., three companies forming one battalion), the combat power increases by the same factor.

A control component not only divides the unit over the two avenues but also adjusts the attribute states to the required level of analysis. By setting the analysis level to a certain echelon and, if required, adjusting it by the Control-Component, a user stays in control of the analysis level to get the required insights.

While it is possible to configure multiple levels of analysis that correspond with military echelons within the used models of the Event-Components, the current implementation of the Event-Component requires additional development of the used models to simplify the analysis on multiple echelons. Nevertheless, the Control-Components provide an additional way to configure the level of analysis; this requirement is fulfilled. Therefore, this requirement is classified as '*pass*'.

6.4.2. Requirement 1.1.2. - Representational Interface

This requirement is based on the User Need, whereby the layout of the envisioned design for the wargame simulation system should resemble the real-world wargaming interfaces. The results from both experiments show that it is possible to translate the tasks and concept of operations from the SyncMatrix to a flow of components, whereby a dedicated instance can handle each flow.

Although the wargame simulation system has a visual way of presenting the flow of events, one still has to understand the concept of operations and military doctrine to be able to create a specific COA and test it against an enemy COA. While the interface of the wargame simulation system is not an exact copy, the flow-based design that plans different actions in time and space and then links them together corresponds to the principles of a SyncMatrix. In both systems, sub-unit activities are planned and synchronized in time and space. While the interface is not a factual representation, it still fulfills the requirement and is classified as '*pass*'.

6.4.3. Requirement 1.2.1. - 'Non-simulation-savvy'

The system can be used by' non-simulation-savvy' users. Contemporary military simulation systems with a comprehensive and monolithic nature require extensive knowledge and training to configure and operate. If someone can break down a problem into discrete steps, they can look at a flow and get a sense of what it is doing without understanding the individual lines of code within each node.

The flow-based interface allows connecting different components without extensive instructions, allowing a user to easily create a COA. Because military icons, terms, and definitions are used, understanding what an element does becomes immediately evident to a military 'non-simulation-savvy' user. This user understanding means that the envisioned design for the wargame simulation system, as presented by the mock-up, meets the stated requirement and is therefore classified as '*pass*'.

6.4.4. Requirement 1.2.2. - Fast Run-Time

This requirement is based on the user need that a wargame simulation system should be able to run the simulation as fast as possible and faster than in real-time. A terrain database was included in the conceptual model design for the simulation system. While terrain (and weather) are essential factors for military operations, they also create vast amounts of data that have to be handled by the simulator, which may require lots of processing power and subsequent ample data storage. Terrain and weather data were purposefully abstracted to keep the developed simulation system as lightweight as possible. For example, in the convoy and tactical movement models, a trafficability service did not calculate the actual routes based on actual terrain data. However, the travel distances must be configured within the component by the user or the engagement areas in the simulation system that are used to create team engagements.

As the system is not intended to provide a high-resolution environmental representation of the battlefield, for example, the absence of terrain and weather data limits the real-world representation of the simulation system, subsequently increasing the run-times of the simulation system and fulfilling this requirement, therefore this requirement is classified as '*pass*'.

6.4.5. Requirement 1.3.1. - Trace Results

This requirement is based on the user's need to trace the results to intermediate steps to provide insightful knowledge. Although the basis for modeling the used tactical tasks are Lanchester's Laws, using different model parameters can adjust a component to show different outcomes for different tactical tasks. Both experiments show that the designed simulation data exchange model supports the flow of data through the implemented components, whereby the user can review the intermediate output for each Event-Component and the final output.

Furthermore, the flow-based design and data exchange model where a distinction has been made between data that configures the simulation and its components, attribute data (states) that can be modified by the components, and data that is not modified by any component and subsequently stored in an external database(s), enables users to add additional attributes or configuration settings. The stated requirement is fulfilled by making (intermediate) results visible and enabling the configuration of components. Therefore, this requirement is classified as '*pass*'.

6.4.6. Requirement 2.1.1. - Consistent Component Behavior

The system components' behavior simulating a particular tactical mission task should resemble military doctrine and procedures. This resemblance should ensure consistent attribute mapping as a mission task simulation component may change their states. Although the movement time for each unit has the same magnitude as the planned time within the Sync-Matrix (see Table 6.6 for a comparison of the simulated and planned times for each sub-unit), according to the simulation, the convoy movement will take less time compared with the times planned in the Sync-Matrix. The difference between the simulated and planned timings could originate from the user configuration of the model that differs from the timing used for the planned time within the Sync-Matrix. Benchmarking the assumed simulation times by real-world timings should correct for the identified deviations within the wargame simulation system. The simulated timings for the infantry units are higher than the simulated timings for the tank units; this matches the planned timings from the Sync-Matrix.

Table 6.6: Comparison of the simulation and scenario times in minutes for test 1.1.

Unit	Simulation	Scenario
A(BT)InfCoy	92	120
B(BT)InfCoy	92	120
C(BT)TkSqdn	91	120

Examination of the results in Table 6.7 for the Event-Component that simulates the movement from the Assembly Area to the SL for the infantry and tank units shows that the difference originates from a slower tactical movement speed for the infantry units. This difference partly explains the planned difference in the Sync-Matrix; further analysis of the tactical tasks reveals that the final deployment to the SL costs more time for an infantry unit with dismounted personnel than a tank unit with no dismounted personnel. Subsequently, this can be corrected by making the movement completion time unit-specific. It should be noted that with the above observation for the convoy movement, further benchmarking of the assumed timing may be required.

Table 6.7: Comparison of the simulation and scenario times in minutes for test 1.2.

Unit	Simulation	Scenario
A(BT)InfCoy	143	180
B(BT)InfCoy	143	180
C(BT)TkSqdn	129	120

Examination of the results in table 6.8 for the Event-Components of both teams combined in the simulated engagements shows that the model used only calculates the net battle time. At the same time, the scenario also includes movement times. This difference explains the difference in times between the simulation and SyncMatrix. However, because the scheduled times in the SyncMatrix do not distinguish between battle time and movement time during the battle, it is impossible to determine whether the simulated battle times match the military doctrine used for planning the engagements in the SyncMatrix. Further deepening of the models used in both the simulation and the SyncMatrix are needed to validate the engagement times calculated by the Event-Components.

Table 6.8: Comparison of the simulation and scenario times in minutes for experiment 2.

Test	2.1		2.2		2.3		2.4	
Unit	Simulation	Scenario	Simulation	Scenario	Simulation	Scenario	Simulation	Scenario
A(BT)InfCoy	8	240	5	240	58	240	25	240
B(BT)InfCoy	-	-	-	-	-	-	-	-
C(BT)TkSqdn	8	240	2	240	67	240	18	240

The attribute state changes that result from the movement model in the first experiment and combat models in the second experiment; the wargame simulation system was configured in such a way that the models would calculate the components output on the behavior at the lowest military echelon, the individual soldier or (weapon)platform level. Based on Lanchester's laws, one has to assume that the combat strength has to be constant to generate consistent outcomes of the engagements between the teams. For example, the combat strength of C(BT)TkSqdn was calculated by multiplying the number of entities (12 tanks) with their relative individual entity strength (0.8) and, of course, several other factors specific to the engagement area.

As the synchronization of tactical mission tasks is essential for the development of a specific COA, the Event-Component and Control-Component within the wargame simulation system use time-driven models to compute the actual execution time of a specific event that represents the assigned tasks. For example, the used 'Lanchester's Laws' to determine the outcome of a battle, the real-time it takes for one of the sides to win the battle is not included in the model. Subsequently, real-time firing rates by the weapon platforms are added to model the duration of the team's engagements. This model calculates the time by the number of rounds it will take to reduce the strength of one of the sides below zero; this number of rounds is multiplied by the firing rate of the used weapon system. This method ignores the real world, where combat combines fire and movement, where the time taken by movement is not included in Lanchester's laws. The actual location of entities has to be determined within the system, creating the (unwanted) challenges that are part of contemporary aggregate and entity-level simulations to overcome this limitation. Adding an Event-Component that models movement directly after the component that models the combat is a possible solution to include the movement time, which results from an engagement between the teams.

Therefore, if the analysis level is configured to the platoon echelon level, the number of entities is divided by 4, i.e., a tank platoon consists of 4 tanks; subsequently, the relative entity strength should be multiplied by 4, and this factor is inversely proportional to factor that is used to divide the number of entities to determine the higher echelon level for the analysis. Similarly, changing the entity resolution from individual vehicles to a company in the movement model can be achieved by changing travel time from an individual vehicle to a company and removing the wait time between the vehicles and the platoons. Although the outputs of the used simulation models are roughly equivalent to the SyncMatrix planning, further validation of the models is still required to fulfill the requirement. Therefore, this requirement is still classified as a '*fail*'.

6.4.7. Requirement 2.1.2. - Aggregate-Level Simulation

The wargame simulation system should be able to execute a constructive aggregate-level simulation of military operations. The models used during both experiments focus on aggregated units. The intermediate results, calculated using the configured Event Components, show results for the simulated aggregated units. Therefore, this requirement is classified as '*pass*'.

6.4.8. Requirement 2.2.1. - RTI performance

The impact of the design on the performance of the RTI should be considered by the designer of the wargame simulation system. The lightweight run-time of Node-RED makes it executable on hardware with limited processing power and deployment in the cloud. This Node-RED feature enables fast run times for the simulation system. On the other hand, how the current design abstracts terrain and locations provides a low resolution for both factors that play a significant role in military decision-making. As units flow through the components within the simulation system, their location is perceived conceptually as combat is simulated within the engagement area determined by its point in the simulation flow, not by its actual location. Subsequently, the exact location of each unit and the associated problems that arise from spreading (dis)aggregation and thrashing cannot be present. While the undesired effects of exact geographic locations on RTI performance are incorporated into the designed wargame system, further experiments are required to determine other design choices that negatively impact the system's performance. Therefore, this requirement has not yet been fulfilled and is classified as '*fail*'.

6.4.9. Requirement 3.1.1. - Consistent Models

As the resolution of the echelons is changed within the simulation, the behavior of tactical mission task components must be consistent across echelons. The fidelity of the designed components and subsequent models corresponds with the abstraction of the real world realized through the concept of operations in the SyncMatrix. Subsequently, despite the absence of environmental data, the users can still configure the required data within the component; therefore, the system can still comply with the first and fourth test criteria, correct real-world representation, and the required level of fidelity. Therefore, this requirement is classified as '*pass*'.

6.4.10. Requirement 3.1.2. - Add or Modify Models

This User Need requires adding new or changing tactical mission tasks within the envisioned wargame system. The decoupling between the component and the models enables the creation of new or improved existing models for the various tactical mission tasks and the implementation of them in the system by the generic Event-Component. Furthermore, the simulation component integrity experiment shows that an Event-Component can be accommodated with a movement model suitable for convoys and tactical movements. Therefore, this requirement is classified as '*pass*'.

6.4.11. Requirement 3.2.1. - Composable Components

Combining, recombining, configuring, and reconfiguring military units and tasks should be possible to tailor them to specific operations. The simulation or flow configuration should prescribe to the individual components at which echelon they have to change the attribute states to ensure consistent attribute mapping. If necessary, a Control-Component can override the analysis level. For example, when the 1(BT)CABn crosses PL-Z, the formation is divided over two avenues of approach, whereby it becomes unnecessarily complex to have the attribute state of the entire battalion changed by all the Event-Components in the various avenues.

6.4.12. Requirement 3.3.1. - Interactions Between Echelons

Interaction between the various echelons should be possible. The wargame simulation system interconnection experiment shows that using an Event-Component equipped with the engagement model to simulate different tactical mission tasks is possible. Furthermore, the experiment shows that Blue and Red Team units with different echelons can engage in a battle. While this feature can be related to Lanchester's laws and is not directly related to the design of the simulation system, the question remains whether this requirement is still fulfilled if a different model is used. Therefore, this requirement is currently classified as '*pass*'.

6.5. Evaluation of the Principles

There are two main evaluation tasks. In the first paragraph, the derived results from the previous activity are evaluated to determine if the objective has been met. This evaluation requires retracing results to the measurable set of requirements originally generated during conceptual analysis. This retracing also includes evaluating the results against the test and execution “pass/fail” criteria for the wargame simulation system. The second paragraph assesses the design principles regarding their reuse potential within the domain or broader user community (IEEE, 2022).

6.5.1. Objective

The wargame simulation prototype has the following design limitations. The scenario serves as initial input for the proof of concept aimed at verifying and validating the usability of the wargame simulation system prototype. Only two experiments were conducted during this proof of concept. The findings from these two experiments were then used to evaluate the identified design requirements. Based on this evaluation, it is determined whether each design requirement meets or falls short of the criteria. This evaluation shows that of the twelve requirements, nine ’pass’ and three ’fail’ (see Table 6.9).

Table 6.9: Tracing the results back to the User Needs (UN) or Simulation Requirements (SR).

Objective	Principle	Requirement	Source	Pass / Fail
1. Compliant	1.1. Recognizability	1.1.1. Multiple Levels of Analysis	UN	Fail
		1.1.2. Representational Interface	UN	Pass
	1.2. Utilizability	1.2.1. ’Non-simulation-savvy’	UN	Pass
		1.2.2. Fast Run-Time	UN	Pass
	1.3. Traceability	1.3.1. Trace Results	UN	Pass
2. Consistent	2.1. Consistency	2.1.1. Consistent Component Behavior	SR	Fail
		2.1.2. Aggregate-Level Simulation	SR	Pass
	2.2. Coherency	2.2.1. RTI Performance	SR	Fail
3.1 Scalable	3.1. Adaptability	3.1.1. Consistent Models	SR	Pass
		3.1.2. Add or modify models	UN	Pass
	3.2. Composability	3.2.1. Composable Components	SR	Pass
	3.3. Interoperability	3.3.1. Interactions Between Echelons	SR	Pass

The purpose of the design was formulated as, *the wargame simulation system has to provide decision-making support to a commander and staff by creating an accessible, consistent, and scalable simulation system that can shorten the duration of a wargame and deliver objective adjudication.* The system requirements structure is used to determine the usability of the design principles. This structure connects the requirements to the design principles, aligning with the project objectives. This link shows to what extent the formulated principles can be applied when creating a prototype for a composable simulation system. The revisiting the systems requirements structure by Table 6.9 shows that the *Compliant* (objective 1.) and *Scalable* (objective 3.) objectives are achievable by the designed simulation system in this research. As requirements *Consistent Component Behavior* (2.1.1.) and *RTI Performance* 2.2.1. not yet have been fulfilled, the *Consistent* (objective 2.) objective cannot be achieved by the current design for the simulation system. Another iteration of the DSEEP process can be executed to refine the design further to fulfill all requirements.

6.5.2. Broader applicability of Design Principles

Table 6.10 provides an overview of the identified principles, whereby the prioritization of each principle explains to what extent a principle is usable to create a composable wargame simulation system. The extent to which a principle is useful is based on the MoSCoW prioritization by Clegg and Barker (1994) assigned to the linked requirements. The letters in the MoSCoW acronym represent their relative importance: M - Must Have, all requirements linked to a principle ’pass’; S - Should Have, several requirements linked to a principle ’pass’; C - Could Have, the requirements linked to a principle do not ’pass’ yet; and W - Will not Have, which is not applicable for any of the requirements in this research.

Table 6.10: Overview of the identified principles for a composable wargame simulation system.

Principle		**Prioritization**
1.1.	Recognizability	The design of a wargame simulation system should represent the real-world processes of military decision-making. Furthermore, the way the simulation elements are composed within the system should resemble the way military personnel develops operational concepts, whereby the behavior of the models within the components must comply with the tactical mission tasks described by military doctrine.
1.2.	Utilizability	The system must be designed in such way 'Non-simulation-savvy' understand it. Contemporary military simulation systems that have a comprehensive and monolithic nature require extensive knowledge and training to configure and operate. Military users benefit from a simulation environment that is easy to use, without extensive additional education and training.
1.3.	Traceability	It must be possible to trace the simulation results to intermediate steps to provide insightful knowledge to the military user.
2.1.	Consistency	To maintain the validity of the models and results, consistency in the behavior, models, and attribute states throughout the wargame simulation system must be a priority.
2.2.	Coherency	A simulation designer needs to be aware of and understand the consequences that the envisioned design could have on the performance of the simulation RTI.
3.1.	Adaptability	The system must be adaptable to assimilate the changes and developments that result from the ever evolving military doctrine.
3.2.	Composability	It must be possible to combine, recombine, configure, and reconfigure sets of components, which is important for a military organization that uses a variety of military units tailored to specific operations.
3.3.	Interoperability	The used models within the simulation system must be able represent interactions between different military units and echelons without requiring extensive modifications or adaptations.

The designed prototype decouples the wargame simulation system from the components and then decouples the components from the possible models. The simulation data exchange model is the foundation on which the various elements are designed. By separating configuration data, attributes, and static data, expansion or expansion of the various functionalities is (relatively) easy to achieve. Equipping an Event-Component with a logistics model in the first experiment and a combat model in the second experiment shows that the simulation system has many possible models that can be included in the envisioned system. While the conceptual design of this simulation system is aimed at supporting military decision-making, its design may also make it suitable for other domains where simulations support decision-making. Because the models used are decoupled from the designed components, applying a wide scale of models should be possible.

The next chapter will conclude this research by answering the research questions, reflecting on the scientific value of the designed wargame simulation system that supports military decision-making, and denoting several recommendations to develop decision support systems further.

7
Conclusion

”One is always a long way from solving a problem until one actually has the answer.”
Stephen Hawking

Wargaming has a long worldwide history within the armed forces to support military decision-making. Wargames are analytic games that simulate aspects of warfare at the tactical, operational, or strategic level. They are used to examine warfighting concepts, train and educate commanders and their military staff, explore COAs, and assess whether the planned concept of operations creates the intended effects. Wargames are perceived as a weakly structured tool for weakly structured problems, such as military conflict. There are specific warfare problems that only gaming can illuminate, so the armed forces play these games because they must.

Simulation systems can reduce the duration of a war game, increase its efficiency, and introduce more objectivity. Military simulation systems that support contemporary wargaming can be complex systems or large monolithic systems that model complicated military constructs of units and complex tactical mission tasks guided by evolving military doctrine. Composing the required simulation elements for military wargaming remains challenging in these complex, constructed simulation systems. This research takes a first step to gain insight into which design principles are relevant when designing a simulation system for wargaming that consists of composable elements.

This final chapter concludes this research. First, the answers to the sub-research questions are outlined, and then, the answer to the main research question is presented. Next, the military and scientific contributions of this research are discussed. Lastly, several recommendations for future research are provided.

7.1. Answers to Sub-Questions

This section addresses the four sub-questions formulated in the first chapter of this research. These sub-questions have been formulated to provide tangible answers that provide direction to the answer to the main research question; accordingly, the first two sub-questions concern military simulation’s domain-specific and scientific background. The answers to these sub-questions provide relevant and rigorous design principles for the design of a wargame simulation system. The third and fourth sub-questions then discuss how the design of the simulation system is envisioned. The third sub-question focuses on selecting and applying a suitable design method to design the system. The last sub-question focuses on a proof of concept to evaluate the envisioned wargame simulation system.

7.1.1. Answer to Sub-Question 1

The first sub-question is intended to determine which elements are necessary for military wargaming by making an overview based on relevant literature and documents. The answer to this question helps to understand the relevant user needs that shape the design principles. The first sub-question is formulated as:

What design principles for composable simulation elements can be derived from the foundations of military wargaming?

The second chapter provides a relevant background on wargaming by identifying and describing the wargaming elements of interest from three perspectives on military wargaming. A wargame is ”a warfare model or simulation whose operation does not involve the activities of actual military forces, and whose sequence of events affects and is, in turn, affected by the decisions made by players representing the opposing sides. The essence of war gaming is the examination of selected aspects of a conflict in an artificial environment. Military decision-making, the overarching concept of wargaming, combines conceptual ideas and detailed processes into elements used to plan military operations. Military simulation standards aim to develop mechanisms by which the connection between command and control and simulation systems becomes (semi-)automated, unambiguous, easy to sustain with low cost, applications independent, expandable, and usable in theater by different levels of command. The most prominent issue with the standards is that its primary audience is software developers, whose verbose structure makes it less suited for usage by military staff. The Modern Conflict Ontolology provides extensive and in-depth analysis of key elements of modern conflict. Despite the stated focus on conventional and unconventional conflicts, the ontology still requires further development in conventional warfare, especially tactical mission tasks for conventional conflicts, which have not been covered in great detail. That said, the ontology demonstrates the extensiveness of military doctrine, which is constantly subject to change.

Military decision-making processes are not intentionally designed to be represented by simulation systems. Military staff, while experienced in decision-making processes, may lack expertise in developing complex simulation systems. From a military user perspective, the envisioned simulation system should resemble military doctrine, resemble operational analysis steps, be recognizable and accessible to users, adapt to evolving military doctrine, and provide traceable simulation results. These military user needs to deliver the first four design principles, *recognizability*, *utilizability*, *traceability*, and *adaptability*, which are deemed relevant to the design of the envisioned wargame simulation system.

7.1.2. Answer to Sub-Question 2

The second sub-question follows the first sub-question and analyzes scientific literature on Modeling & Simulation, in which modeling and simulation concepts are fundamentally challenging for military simulation. The answer to this question helps to understand the design principles derived from the identified challenges. The second sub-question is formulated as:

What design principles for composable simulation elements can be derived from contemporary challenges within military wargaming simulation?

The third chapter analyses scientific M&S research to identify the challenges within contemporary military simulation. It highlights the difficulty of composing military wargame elements in a simulation environment, especially when considering factors like terrain, weather, and the actions of other actors. Composability is more than just the ability to combine simulations from components; it is the ability to combine, recombine, configure, and reconfigure sets of components from those available into different simulation systems to meet different needs. There is a transition from extensive monolithic simulation systems into component-based environments and federated architectures to facilitate innovation, flexibility, and cost-effectiveness. Despite their benefits, these environments create new challenges by the (dis)aggregation of entities and their inter-level interactions. The results of executing a scenario should be the same or similar within a desired tolerance, regardless of the selected (aggregated) entities or the interactions between these entities. A simulation environment cannot be considered consistent if differences between the models' behaviors introduce discrepancies. Subsequently, the challenge to overcome is maintaining an internal state consistent with the used models. The challenge for the simulation designer is to be aware of the consequences for the RTI by the design choices made when developing a military simulation system. Ill-considered choices or added flexibility can increase software complexity or degrade (reduce) system performance.

The analysis of the Modeling & Simulation literature on contemporary challenges for military simulation shows that it remains challenging to create a simulation system consisting of different (existing) simulation systems. It is, therefore, essential to consider these challenges when designing the wargame simulation system. Therefore, these challenges for the design of military simulation systems are translated into the second four design principles, *consistency*, *coherence*, *composability*, and *interoperability*, which are deemed relevant to the design of the envisioned wargame simulation system.

7.1.3. Answer to Sub-Question 3

The third sub-question builds on the two previous sub-questions and applies the identified design principles to create a military war game system that uses composable simulation elements. The answer to this question helps to understand how a military wargaming system can be designed and how design principles shape the envisioned design. The third sub-question is formulated as:

Based on the identified design principles, how can a wargame simulation system using composable simulation elements be designed?

The fourth chapter selects and applies an appropriate design process to design the envisioned wargame simulation system. Subsequently, the fifth chapter uses the first four steps from the Distributed Simulation Engineering and Execution Process, selected in the fourth chapter as the appropriate design process, to design a simulation system. The first step in the design process is to define the user needs and objectives derived from the previously identified wargame elements. The next step in the design process entails a conceptual analysis describing the simulation's intended problem space by a functional specification of real-world wargaming. Subsequently, this functional specification is used to develop a conceptual model that can be used to design the components of the simulation system. This second step shows that a flow-based simulation system fits the conceptual model. The system is designed by selecting the required member applications, i.e., external databases that contain terrain and unit data. The next step is to design the four main components, the Unit-Component, Event-Component, Control-Component, and Result-Component, that can be used to create a flow that represents a certain COA. The designed simulation data exchange model ensures a consistent data flow through the system, a description of the member applications, and a mock-up that shows how the envisioned wargame system could be implemented give substance to the fourth step of the design.

The conceptual design of the wargame simulation system shows that it is possible to create composable components within a flow-based environment with which different Course of Action can be set up. Using Node-RED, a low-level coding system, for the Run-Time Infrastructure ensures a composable, flexible, and easy-to-use system for *'Non-simulation Savvy'* users; if someone can break down a problem into discrete steps, they can create a flow of events in the envisioned simulation system. Nevertheless, although the outputs of the used models are roughly equivalent to the created real-world scenario, further validation of the models is still required. The design created using the Distributed Simulation Engineering and Execution Process, in which the identified design principles from the first two sub-questions have been applied, delivers the answer to this third sub-question.

7.1.4. Answer to Sub-Question 4

The fourth sub-question verifies, validates, and analyzes the designed wargame simulation system. The answer to this question uses a real-world wargame scenario as input for a proof of concept that evaluates the envisioned simulation system. The fourth sub-question is formulated as follows:

To what extent is it possible to use the design principles to create a prototype of a composable wargame simulation system?

The sixth chapter provides the answer to this sub-question by constructing a concise real-world wargame scenario that has been developed to design a prototype wargame simulation system. The subsequent designed system has been verified, validated, and analyzed through a proof of concept, a conceptual demonstration that the envisioned simulation system can be used to wargame the developed real-world scenario. The scenario is subsequently used as input for the proof of concept that verifies and validates whether the designed prototype can be used for a wargame simulation to support military decision-making. As part of the proof of concept, two experiments were conducted. The results of these experiments were then used to analyze the identified design requirements. Based on this analysis, it is determined whether each design requirement passes or fails. This evaluation shows that nine pass and three fail of the twelve requirements. The system requirements structure, which links the requirements to the design principles and then links the design principles to the objectives, is used to determine which design principles are useful. The proof of concept is not only aimed at the practical design of the prototype system; foremost, it is intended to demonstrate the usefulness of the eight uncovered design principles. This linkage between the design requirements and principles that were developed within the system requirements structure shows to what extent the formulated principles are usable for the creation of a prototype of a composable wargame simulation system:

- Utilizability, Traceability, Adaptability, Composability, and Interoperability are usable.
- Recognizability and Consistency are, to a certain extent, usable.
- The usability of Coherency cannot yet be determined.

Subsequently, the proof of concept shows that the identified principles contribute to the *Compliant* and *Scalable* objectives defined for the wargame simulation system. The *Consistent* objective cannot yet be achieved by the current prototype for the wargame simulation system.

7.2. Answer to the Main Research Question

The main research question can be answered based on the theoretical and experimental insights obtained from the conclusions that relate to the four sub-questions. The main research question is formulated as follows:

What are design principles for composable elements that can be used in a simulation system for military wargames?

Based on the executed research, it can be stated that the eight following identified design principles are usable as they guide a designer when creating a wargame simulation system:

1. **Recognizability** - The design of a wargame simulation system should represent the real-world processes of military decision-making. Furthermore, how the simulation elements are composed within the system should resemble how military personnel develop operational concepts, whereby the behavior of the models within the components must comply with the tactical mission tasks described by military doctrine.
2. **Utilizability** - The system must be designed in such way 'Non-simulation-savvy' understand it. Contemporary military simulation systems with a comprehensive and monolithic nature require extensive knowledge and training to configure and operate. Military users benefit from a simulation system that is easy to use without extensive additional education and training.
3. **Traceability** - It must be possible to trace the simulation results to intermediate steps to provide insightful knowledge to the military user.
4. **Consistency** - To maintain the validity of the models and results, consistency in the behavior, models, and attribute states throughout the wargame simulation system should be a priority.
5. **Coherency** - A simulation designer needs to be aware of and understand the consequences that the envisioned design could have on the performance of the simulation RTI.
6. **Adaptability** - The system must be adaptable by the user to assimilate the changes and developments that result from the ever-evolving military doctrine.
7. **Composability** - It must be possible to combine, recombine, configure, and reconfigure sets of components, which is essential for a military organization that uses a variety of military units tailored to specific operations.
8. **Interoperability** - The used models within the wargame simulation system must be able to represent interactions between different military units and echelons without requiring extensive modifications or adaptations.

Because the performed proof of concept relates to one specific prototype design, the question arises whether the conclusions on the sub-questions can be generalized to answer the main research question. Principles have a generic nature and provide insight into the system they apply. This notion about the generic nature not only suggests that the principles are not only limited to the prototype wargame simulation system that is designed in this research; this general applicability of principles within a particular domain implies that the identified principles in this research are also usable for the design of other wargame simulation systems. While the conceptual design of this simulation system is aimed at supporting military decision-making, its design may also make it suitable for other domains where simulations support decision-making. Furthermore, as the models used are decoupled from the designed components, applying a wide scale of models is possible.

The prototype design separates the simulator from its components and the components from potential models. Central to this prototype is the simulation data exchange model, serving as the data structure on which all components are based. This separation of configuration, attributes, and static data makes expanding or enhancing various functionalities relatively straightforward. Notably, decoupling models from the designed components enables the possibility of incorporating a wide range of models to suit diverse decision-making processes that consist of (a flow of) intermediate activities. Although the initial premesis behind this prototype for a wargame simulation system is to support military decision-making, its design also holds promise for application in other domains where simulations are instrumental in decision-making processes.

The designed simulation prototype shows that it is possible to exploit a flow-based system to create a usable wargame simulation system, which military commanders and their staff can utilize when conducting a wargame.

7.3. Military and Scientific Relevance

This research is conducted within the framework of the broader scientific research program by TNO[1] into military simulation. In this research program, TNO and the Delft University of Technology, commissioned by the Dutch Ministry of Defence, research the development and broader application of simulation in various military domains, such as training, operational support (i.e. decision-making), and concept development.

The envisioned wargame simulation system is not intended to replace a military wargame entirely, nor does it cover all stages of the Tactical Decision-Making Model. The system is intended as a technical means to support a specific step - *the operational analysis* - within the military decision-making process executed by military practitioners. As with every other (war)game, the simulation results do predict a possible future as they are always abstractions of the real-world, limited by the models that are used within the simulation system and driven by the mental models that the users use to create the flow of components that represents a certain COA. Furthermore, the design of the prototype in this research is specially designed to evaluate the design principles, whereby a concise military scenario is used for the proof of concept. The prototype only uses the models representing the tactical military tasks and units arising from the developed scenario. Therefore, it will be required to do additional modeling and design to include other tactical military tasks or units that can be required to simulate other operations. While the proof of concept uses a concise (somewhat limited) scenario, the usage of Node-RED's flow-based design and low-coding system shows that users can plan different COA by linking different components corresponds to the principles of a SyncMatrix that military commanders and their staffs use prior to a wargame to plan and visualize an operational concept.

This research has addressed the knowledge gap in which, according to M&S scholars, it remains challenging to construct aggregate- and entity-level simulations within an identical system used for operational analysis or wargaming. Although these challenges are extensively debated within M&S literature, evaluating design principles for such simulation systems is relatively unexplored. Primarily, this research uses a 'greenfield' approach to design the prototype for a wargame simulation system. This research can be seen as one of the first studies to close the literature gap and provide insight into which design principles for composable simulation elements can be used in a military wargame simulation system. This research has, therefore, contributed to the existing literature by identifying these design principles. This research, therefore, goes beyond the studies of, for example, Franceschini and Petty (1995), Reynolds et al. (1997), Petty and Franceschini (1998) Williams et al. (1998), Tolk (2012), and S. Taylor et al. (2015), who identify various challenges with military simulation but do not yet link design principles to the identified challenges.

This research allows military practitioners and M&S scholars to further experiment in practice and theoretically explore the identified design principles for composable simulation elements that can be used in a flow-based wargame simulation system.

7.4. Recommendations

This section provides recommendations based on the research's implications, outcomes, limitations, and scientific relevance. Future research may strengthen the results obtained from this study and increase the feasibility of implementing a flow-based wargame simulation system. As described in chapter 6, this study has several limitations. These are primarily due to the prototype being limited to the conceptual level, the applied proof of concept that consisted of two experiments and a limited set of models, and the chosen design focus on the technical artifact.

As this research was based on a 'greenfield' premesis, only a conceptual model and a mock-up for the design of the prototype were created. Creating a functional prototype further demonstrates the core functionalities of the wargame simulation system. The proof of concept could further validate the used

---

[1] Netherlands Organisation for Applied Scientific Research.

models. A working user interface that enables military users to navigate, input data, and interact with the wargame simulation would enable further analysis of whether the recognizability principle for the system design can be improved. Both components are essential in the further development process of the envisioned wargame simulation system, as they allow for practical testing and validation of the models that are used in the flow-based simulation concept and provide a presumed user-friendly means for military personnel to engage with the system in order to verify military recognizability further.

The design was limited to a prototype or conceptual level. However, it corresponds to the high degree of abstraction applied when designing an operational concept for a military operation. However, many factors are analyzed and included in developing the operational concept during an actual decision-making process. However, in its current form, the prototype only considers the factors essential for a military operation to a very limited extent, partly because a concise scenario is used for the proof of concept. Further research into the optimal fidelity level for, for example, environmental and location data is needed because the current prototype deliberately abstracts these factors. Future researchers can experiment with different or more complex military operations by adding more models to the various components. Furthermore, the current prototype design only provides limited scope for using different levels of analysis. By further adapting the components used, it can be tested whether this design requirement is also feasible. Future scientific research should further improve the envisioned simulation concept into a usable wargame simulation system so military commanders and their staff can further exploit the power of simulation when conducting a wargame.

The consistent behavior of the entities at the different echelons is partly complicated by how military doctrine formulates tasks for the different levels of military action. Tactical decision-making, which uses tactical military tasks, originates from operational decision-making, which involves thinking about effects that must be achieved through the execution of tactical activities. This decomposition of higher-level orders into lower-level tasks is perceived as ’art’ within military decision-making. A military commander and his staff translate these effects into activities at different levels based on mental models based on their knowledge and experience. Within the military simulation domain, it is still a topic of debate how to model this ’military art’ within military constructive simulation, such as the envisioned prototype wargame simulation system in this research. The question that can or should be asked is: ”there are any possibilities to overcome this challenge by including the design of the two other artifacts, the ’institutional’ and ’process’ artifacts, which complex systems engineers usually consider when designing a Socio-Technical System?” Although military organizations (institutions) and doctrine (processes) have a long history, they seem rigid, yet as with many other things, they are also subjected to change. Future research that includes all three domains of complex systems engineering may reveal new models to translate high-level orders to low-level tasks that can be applied within the technical realm of simulation and used by military institutions to develop their processes.

References

Alstad, A., Mevassvik, O. M., Nielsen, M. N., Løvlid, R., Henderson, H., Jansen, R. E. J., & de Reus, N. M. (2013). Low-level battle management language.

Balci, O., Arthur, J., & Ormsby, W. (2011). Achieving reusability and composability with a simulation conceptual model. *Journal of Simulation*, *5*(3), 157–165. https://doi.org/10.1057/jos.2011.7

Blais, C. (2008). Military Scenario Definition Language (MSDL): How Broadly Can It Be Applied. *Proceedings of the Spring 2008 Simulation Interoperability Workshop*.

Blais, C., Reece, D., & Singapogu, S. (2019). From Information Description to Information Understanding: The Role of Ontology in Emerging SISO Standards.

Boehm, B. W. (1988). A spiral model of software development and enhancement. *Computer*, *21*(5), 61–72.

Bots, P., van Daalen, E., Steenhuisen, B., Ruijgh, T., & van der Voort, H. (2017). Designing in socio-technical systems.

Box, G. E. P. (1979). Robustness in the Strategy of Scientific Model Building. In R. Launer & G. Wilkinson (Eds.), *Robustness in statistics* (pp. 201–236). Academic Press. https://doi.org/10.1016/B978-0-12-438150-6.50018-2

Brazier, F., Van Langen, P., & Treur, J. (1996). A logical theory of design. *Advances in Formal Design Methods for CAD: Proceedings of the IFIP WG5. 2 Workshop on Formal Design Methods for Computer-Aided Design, June 1995*, 243–266.

Brocke, J. v., & Maedche, A. (2019). The DSR Grid: Six Core Dimensions for Effectively Planning and Communicating Design Science Research Projects. https://doi.org/10.1007/s12525-019-00358-7

Bruvoll, S., Hannay, J. E., Svendsen, G. K., Asprusten, M. L., Fauske, K. M., Kvernelv, V. B., Løvlid, R. A., & Hyndøy, J. I. (2015). Simulation-Supported Wargaming for Analysis of Plans. *NATO Modelling and Simulation Group Symposium on M&S*. https://www.researchgate.net/publication/282503601_Simulation-Supported_Wargaming_for_Analysis_of_Plans

Caffrey, M. (2000). Toward a History-Based Doctrine for Wargaming. *Aerospace Power Journal*, 33–56. https://apps.dtic.mil/sti/citations/ADA521381

Chua, B. Y. W., & Low, M. Y. H. (2009). Predictive algorithms for aggregation and disaggregation in mixed mode simulation. *Proceedings of the 2009 Winter Simulation Conference (WSC)*, 1356–1365.

Clark, K. J., & Brewer, D. (1994). Bridging the gap between aggregate level and object level exercises. *Proceedings of the 4th Conference on Computer Generated Forces and Behavioral Representation*.

Clegg, D., & Barker, R. (1994). *Case method fast-track: a RAD approach*. Addison-Wesley Longman Publishing Co., Inc.

Davis, P. K. (1993). *An Introduction to Variable-Resolution Modeling and Cross-Resolution Model Connection* (tech. rep.).

Davis, P. K. (1995). An introduction to variable resolution modeling. *Naval Research Logistics (NRL)*, *42*(2), 151–181.

Davis, P. K., & Bigelow, J. H. (1998). Experiments In Multiresolution Modeling (MRM). https://www.rand.org/pubs/monograph_reports/MR1004.html

Davis, P., & Bigelow, J. (1998). Introduction to multi-resolution modeling (MRM) with an example involving precision fires. *Proceedings of SPIE - The International Society for Optical Engineering*, *3369*, 14–27. https://doi.org/10.1117/12.319334

Downes-Martin, S. (1991). The combinatorics of vehicle level wargaming for senior commanders. *2nd Behavioral Representation and Computer Generated Forces Symposium Proceedings*, 6–7.

DuBois, E. L., Hughes Jr, W. P., & Low, L. J. (1997). A concise theory of combat. https://apps.dtic.mil/sti/citations/ADA331950

Fairley, D., Forsberg, K., Madachy, R., & Marbach, P. (2023). System Lifecycle Process Models: Vee. https://sebokwiki.org/wiki/System_Lifecycle_Process_Models:_Vee

Ford, K. (2021). MSG-168 Lecture Series on Modelling and Simulation as a Service (MSaaS) 2. Overview of MSaaS Concept to Ecosystem. *NATO S&T*.

Franceschini, R. W. (1993). Intelligent Placement of Disaggregated Entities.

Franceschini, R. W. (1999). *Correlation error in multiple resolution entity simulations* (Doctoral dissertation). University of Central Florida.

Franceschini, R. W., & Mukherjee, A. (1999). A simple multiple resolution entity simulation. *Proceedings of the Eighth Conference on Computer Generated Forces and Behavioral Representation, Orlando, Fla., May*, 11–13.

Franceschini, R. W., & Petty, M. D. (1995). Linking constructive and virtual simulation in distributed interactive simulation (DIS). *Distributed Interactive Simulation Systems for Simulation and Training in the Aerospace Environment: A Critical Review*, *10280*, 276–293.

Franceschini, R. W., Wu, A. S., & Mukherjee, A. (2000). Computational strategies for disaggregation. *Proc. 9th Conf. on Computer Generated Forces and Behavioral Representation*.

Haight, D. B. (2007). Preparing Military Leaders for Security, Stability, Transition and Reconstruction Operations. https://apps.dtic.mil/sti/citations/ADA469089

Hanley, J. T. J. (1991). On wargaming: a critique of strategic operational gaming.

Hannay, J. E., Brathen, K., & Hyndøy, J. I. (2015). On How Simulations Can Support Adaptive Thinking in Operations Planning. *Proceedings of NATO Modelling and Simulation Group Symposium on M&S*.

Hartley, D. S. (2017). Modeling Unconventional Conflict. In D. S. Hartley III (Ed.), *Unconventional conflict: A modeling perspective* (pp. 171–194). Springer International Publishing. https://doi.org/10.1007/978-3-319-51935-7

Hartley, D. S. (2021). *An Ontology of Modern Conflict*. Springer International Publishing. https://doi.org/10.1007/978-3-030-53214-7

Hevner, A. (2014). A Three Cycle View of Design Science Research. https://www.researchgate.net/publication/254804390

Hevner, A., March, S., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, *28*(1), 75–105. https://doi.org/10.2307/25148625

Hillestad, R., Owen, J., & Blumenthal, D. (1995). Experiments in variable resolution combat modeling. *Naval Research Logistics (NRL)*, *42*(2), 209–232.

IEEE. (2022). IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP). *IEEE Std 1730-2022 (Revision of IEEE Std 1730-2010)*, 1–74. https://doi.org/10.1109/IEEESTD.2022.9919118

IJntema, R., & van de Haar, G. (2010). Het tactisch besluitvormingsmodel (TBM) - Afstemming op het tactische niveau en op hedendaagse ‘operating environments’. *Miltaire Spectator*, *179*(2), 98–107.

Karr, C. R., Franceschini, R. W., Perumalla, K. R. S., & Petty, M. (1993). Integrating aggregate and vehicle level simulations. *proceedings of the Third Conference on Computer Generated Forces and Behavioral Representation*, 231–240.

Karr, C. R., & Root, E. (1994). Integrating aggregate and vehicle level simulations. *Proceedings of the Third Conference on Computer Generated Forces and Behavioral Representation*, 231–239.

Lanchester, F. W. (1956). *Mathematics in warfare*.

Lee, K., Lee, G., & Rabelo, L. (2020). A systematic review of the multi-resolution modeling (MRM) for integration of live, virtual, and constructive systems. https://doi.org/10.3390/info11100480

McHugh, F. J. (1966). *Fundamentals of war gaming*. The United States Naval War College.

Ministry of Defence. (2021). Programmavoorstel: Military Cloud Based Simulation.

Ministry of Defense. (2012). Joint Doctrine Publication 5: Command and Control.

Morrison, J. P. (n.d.). Flow-based Programming. https://jpaulm.github.io/fbp/

NATO. (n.d.). NATO M&S High Level Architecture. https://nmsg.sto.nato.int/amsp/hla

NATO. (2017). NATO Standard APP-6 - NATO Joint Military Symbology.

NATO. (2018). *NATO Modelling and Simulation Standards Profile AMSP-01* (tech. rep.).

NATO. (2022). Operations and missions: past and present. https://www.nato.int/cps/en/natohq/topics_52060.htm

NATO. (2023). NATO Term: The Official NATO Terminology Database. https://nso.nato.int/natoterm/content/nato/pages/home.html?lg=en

NATO STO. (n.d.). NATO C2 & Simulation Interoperability. https://nmsg.sto.nato.int/themes/c2sim

Natrajan, A. (2000). *Consistency maintenance in concurrent representations* (Doctoral dissertation). University of Virginia.

OpenJS Foundation & Contributors. (n.d.). Node-RED. https://nodered.org

Page, E. H., & Opper, J. M. (1999). On the complexity of composable simulation.

Page, E. H., & Smith, R. (1998). Introduction to military training simulation: a guide for discrete event simulationists. *1998 Winter Simulation Conference. Proceedings (Cat. No. 98CH36274)*, *1*, 53–60.

Park, T. W. (2015). An Agile Roadmap for Live, Virtual and Constructive-integrating Training Architecture (LVC-ITA): A Case Study Using a Component Based Integrated Simulation Engine.

Patel, B. (2021). MSG-168 Lecture Series on Modelling and Simulation as a Service (MSaaS) 1. Need and Motivation for MSaaS; 5. MSaaS Benefits and Achievability; 7. MSaaS Business Model Development. *NATO S&T*.

Perla, P. (1990). *The art of wargaming : a guide for professionals and hobbyists*. https://www.worldcat.org/nl/title/1035695854

Perla, P., & McGrady, E. (2011). Why Wargaming Works. *Naval War College Review*, *64*(3), 111–130. https://www.jstor.org/stable/10.2307/26397225

Petty, M. D. (1995). Computer-generated forces in distributed interactive simulation. *Distributed Interactive Simulation Systems for Simulation and Training in the Aerospace Environment: A Critical Review*, *10280*, 102800I. https://doi.org/10.1117/12.204224

Petty, M. D., & Franceschini, R. W. (1998). Special Topics in Multi-Resolution SImulation.

Petty, M. D., & Weisel, E. W. (2003). A composability lexicon. *SISO simulation interoperability workshop, Kissimmee, FL*.

Powell, D. R. (1997). *Control of entity interactions in a hierarchical variable resolution simulation* (tech. rep.). Los Alamos National Lab.(LANL), Los Alamos, NM (United States).

Rabelo, L., Kim, K., Park, T. W., Pastrana, J., Marin, M., Lee, G., Nagadi, K., Ibrahim, B., & Gutierrez, E. (2015). Multi resolution modeling. *2015 Winter Simulation Conference (WSC)*, 2523–2534. https://doi.org/10.1109/WSC.2015.7408362

Raue, E., & Gallois, V. (2011). A Dynamic Multi-Resolution Model Based on HLA Interconnection of Commercial-Off-The-Shelf (COTS) Simulation Tools. *NATO Modeling and Simulation Group (NMSG) Symposium*.

Reynolds, P. F., Natrajan, A., & Srinivasan, S. (1997). Consistency maintenance in multiresolution simulation. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, *7*(3), 368–392.

Robkin, M. (1992). A proposal to modify the distributed interactive simulation aggregate PDU. Hughes Training.

Roman, P. A. (2005). Garbage in, Hollywood out. *SimTecT 2005*.

Royal Netherlands Army. (2011). Doctrine Publicatie 3.2.2: Commandovoering.

Royal Netherlands Army. (2012). Doctrine publication 3.2.2.4: Staftechniek.

Royal Netherlands Army. (2014). Doctrine Publicatie 3.2: Landoperaties.

Royce, W. (1970). Managing the Development of Large Software Systems. *Proceedings of IEEE WESCON*.

Rubel, R. C. (2006). The epistemology of war gaming. *Naval War College Review*, *59*(2), 108–128. https://www.jstor.org/stable/10.2307/26396724

SISO. (2013). Standard for Coalition Battle Management Language (C-BML) Phase 1.

SISO. (2015). Standard for Military Scenario Definition Language.

SISO. (2023). C2SIM PDG/PSG - Command and Control Systems - Simulation Systems Interoperation.

Taylor, J. G. (1983). Modeling and Simulation of Land Combat. *Proceedings of the Workshop on Modeling and Simulation of Land Combat*.

Taylor, S., Khan, A., Morse, K., Tolk, A., Yilmaz, L., Zander, J., & Mosterman, P. (2015). Grand challenges for modeling and simulation: Simulation everywhere—from cyberinfrastructure to clouds to citizens. *SIMULATION*, *91*(7), 648–665. https://doi.org/10.1177/0037549715590594

TNO. (2021). COSMOS Decision Support concepts.

Tolk, A. (2012). *Engineering Principles of Combat Modeling and Distributed Simulation* (A. Tolk, Ed.). Wiley. https://doi.org/10.1002/9781118180310

Trinker, A. (1994). General Architecture for Interfacing Virtual and Constructive Simulations in DIS Environment. *Internal Document, Institute for Simulation and Training, September*, *141994*.

Turnitsa, C., Blais, C., & Tolk, A. (2022). *Simulation and wargaming*. John Wiley & Sons, Incorporated.

UK MoD. (2017). Wargaming Handbook.

U.S. Department of Defense. (2017). Joint Publication 1: Doctrine for the Armed Forces of the United States.

U.S. Department of Defense. (2020). Joint Publication 5-0: Joint Planning.

Van Den Berg, T. (2021a). MSG-168 Lecture Series on Modelling and Simulation as a Service (MSaaS) 12. MSaaS Composition and Technical Approach. *NATO S&T*.

Van Den Berg, T. (2021b). MSG-168 Lecture Series on Modelling and Simulation as a Service (MSaaS) 9. MSaaS Technical Reference Architecture. *NATO S&T*.

von Moltke, H. (1880). Kriegsgeschichtliche Einzelschriften.

Weisel, E. W., Petty, M. D., & Mielke, R. R. (2003). Validity of Models and Classes of Models in Semantic Composability.

Wikimedia Commons. (2008). MilSim Spectrum. https://commons.wikimedia.org/wiki/File:MilSim_Spectrum.svg

Williams, T. M., Sale, N., Smith, M., & Shakoor, A. S. (1998). Automated variable representation. *Proceedings of the Simulation Interoperability Workshop (SISO), Orlando, FL, USA*, 14–18.

Work, R., & Selva, P. (2015). Revitalizing wargaming is necessary to be prepared for future wars. https://warontherocks.com/2015/12/revitalizing-wargaming-is-necessary-to-be-prepared-for-future-wars/

Wrigge, S., Fransén, A., & Wigg, L. (1995). *The Lanchester Theory of Combat and Some Related Subjects: A Bibliography, 1900-1993*. Försvarets Forskningsanstalt, Avedelningen för Försvarsanalys. https://archive.org/stream/DTIC_ADA302237/DTIC_ADA302237_djvu.txt

Annexes

A

Tactical Decision-Making

Tactical decision-making combines conceptual and detailed aspects of planning. It is used to create plans and orders for large, long-term, and short-term operations within the framework of an operational plan. Tactical decision-making can be based on tactical design, a higher-level plan, or direction. Tactical decision-making supports a commander in understanding the operational environment, identifying and understanding the problem, determining the desired outcome, and developing and choosing an appropriate COA. Tactical decision-making results in a comprehensive plan for the execution of the operation (Royal Netherlands Army, 2014). Figure 2.2 displayed in Paragraph 2.1.1 outlines the three stages and underlying steps (see Table A.1) of Dutch TBM. The following explanation of these stages is based on the article by IJntema and van de Haar (2010) on the tactical decision-making model.

Table A.1: Components within the Royal Netherlands Army Tactical Decision-Making Model.

Step	**Component**	**Description** (NATO, 2023; Royal Netherlands Army, 2011)
1	Mission Analysis	A logical process for extracting and deducing from a superior's orders the tasks necessary to fulfil a mission.
	Actors	An individual, group or entity whose actions are affecting the attainment of the end state.
	Planning of Activities	The factor of time plays a major role in the planning and execution of military activities. A planning or schedule is an indispensable tool for the synchronization of all military activities, actions and allocation of resources.
2a	Objectives & End State	Operations are guided by clear and clearly formulated objectives, which contribute to the achievement of the desired End State. The objectives are the essential preconditions that must be achieved before the desired End State can be reached.
	Effects	Effects support the achievement of an objective. Effects can be achieved by performing actions or tasks, which can change the circumstances from unfavorable to favorable for achieving the desired objective(s). Effects can relate to the behavior of an actor, a change in capabilities or a change in a system.
	CCIR	In preparing to conduct his decision making process, the commander will be able to identify, from the outset, what information, relating to both the adversary and friendly forces, he requires in order to be able to reach a decision and make his plan. The questions to which he needs answers, form the Commander's Critical Information Requirements (CCIR).
	Tasks	A composite of related activities (perceptions, decisions, responses) performed for an immediate purpose.
	Requirements	Requirements can restrictive as well constrictive. Restraints are requirements placed on a commander that prohibits an action. Whereas constraints are imposed requirements placed on a commander that dictates an action.
	Guidance	An overview of all kinds of (substantive and procedural) guidelines such as go, no go and abort criteria; instructions for the development of COAs; necessity and/or permission for reconnaissance.
2b	Targets	Targets can be objects or (groups of) people. For objects, the desired effect is primarily aimed at influencing the possibilities of use. Its side effects have an on the combat readiness and behavior of the owners and users of these objects. For (groups of) individuals, the desired effect is primarily aimed at influencing their understanding and perception of the situation. This will also have an impact on the combat readiness and behavior of these individuals.
	Environment	An analysis of all environmental factors (including climate, weather and terrain) that can influence the operation and the activities of all other relevant actors.
	Threat (scenario's)	A specific analysis of (f)actors who (knowingly or unknowingly) want to or can pose a threat to the operation. Threat scenarios are often based on the actions of opposing forces (Enemy COAs). There are also situations conceivable without an opponent, then there is is a different threat, such as a flood (disaster relief operations).

Table continues on next page.

Table A.1 Continued: Components within the Royal Netherlands Army Tactical Decision-Making Model.

Step	Component	Description
	Factors	As concise as possible description how the other actors, the local population, are expected to behave during the execution of the operation, how the opposing forces are likely to act. For each actor, it is important to determine their objectives.
	Capabilities	The ability of an (subordinate) capacity to meet a service demand of given quantitative characteristics under given internal conditions.
3	Concept of Operations	A Concept of Operations is clear and concise statement of the line of action chosen by a commander in order to accomplish his given mission.
	COAs	In the estimate process, an option that will accomplish or contribute to the accomplishment of a mission or task, and from which a detailed plan is developed.
4	Decision Support Overlay	A staff product initially used in the wargaming process which graphically represents the decisive points and projected situations and indicates when, where, and under what conditions a decision is most likely to be required to initiate a specific activity (such as a branch or sequel) or event (such as lifting or shifting of fires.)
	Synchronization Matrix	A synchronization matrix is an overview that can be used to visualize the cohesion between different activities and the cohesive deployment of the different (types of) units and capabilities against the threat(s) and/or effects to be achieved, in time and space. Considering the individual activities in relation to each other makes it clear how the deployment of the various capacities should be coordinated.
	ORBAT	The organizational structure for the mission.
	TOE	Lists all equipment assigned to a unit, reflects how a unit is organized and what resources are available.
5	Contingencies	Circumstances may change during the execution of an operation and the intended COA may be jeopardized. As a result, adjustment of the plan or change in the execution of the operation the may be necessary.
	Operation Order	A directive issued by a commander to subordinate commanders for the purpose of coordinating the execution of an operation.

Decision-making does not begin until an order is received. Nor does it end when a decision is formalized. A military organization constantly monitors the environment in which it operates. This monitoring builds and maintains so-called ’general awareness.’ Based on this awareness, the purpose of the mission is determined or adjusted. Usually, the ends are determined by the next higher military level, and orders are given to the levels below. In some cases, however, the mission can also be self-formulated.

To this end, a unit must understand the specific aspects of the environment in which the mission must be executed. The commander and military staff must then convert general awareness into situational awareness. This stage is called ’understanding’. Subsequently, based on the current operational picture, the commander and staff have examined how - COAs - the mission could be carried out. This stage is called ’assessing’. Only then can a well-considered decision be made about the final choice with the highest chance of success. This stage is called ’deciding’.

The staff still has to carry out many additional planning activities in many cases. Answering the still open questions (’request for information’) and alternative plans (’contingencies’) is, in any case, part of this. Furthermore, the commander will, of course, also have to communicate the decision to his subordinate commanders. While issuing the orders, the commander explains his decision and conveys his intention for the mission to the subordinate commanders.

The following Annexes B.1 and B.2 use the above outline of the Dutch TBM to create the scenario that is used to prove the designed simulation concept.

B

Wargame Scenario

B.1. Stage 1: Understand the Mission

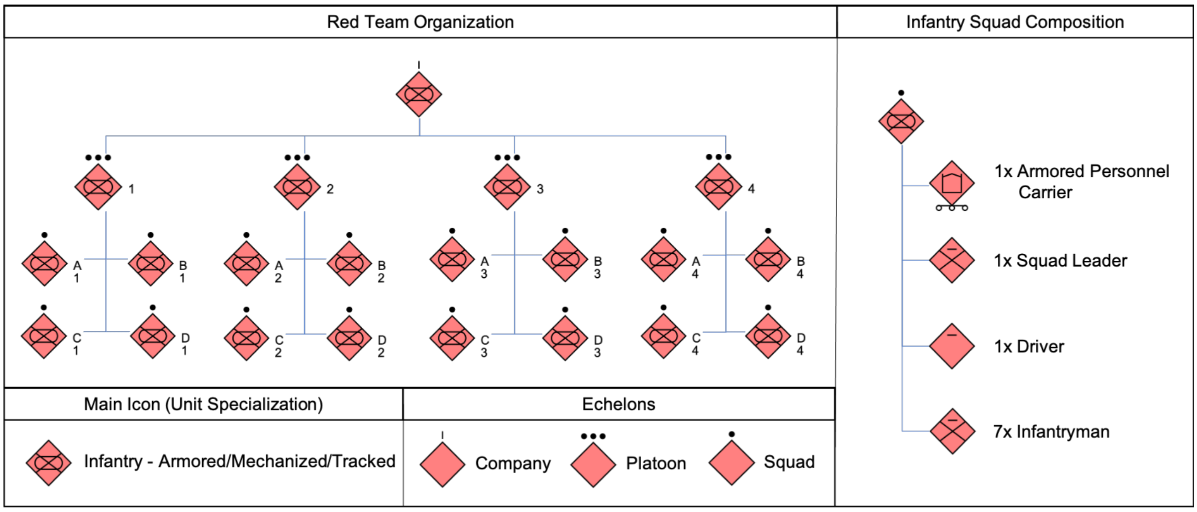


Figure B.1: Red Team ORBAT and TOE.

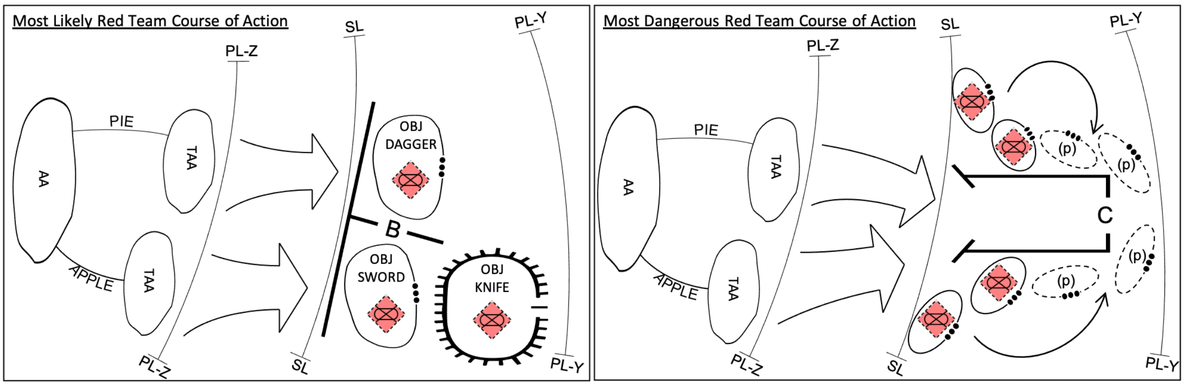


Figure B.2: Most likely and most dangerous Red Team COAs.

B.2. Stage 2: Assess the Mission

B.2.1. Blue Team Organization

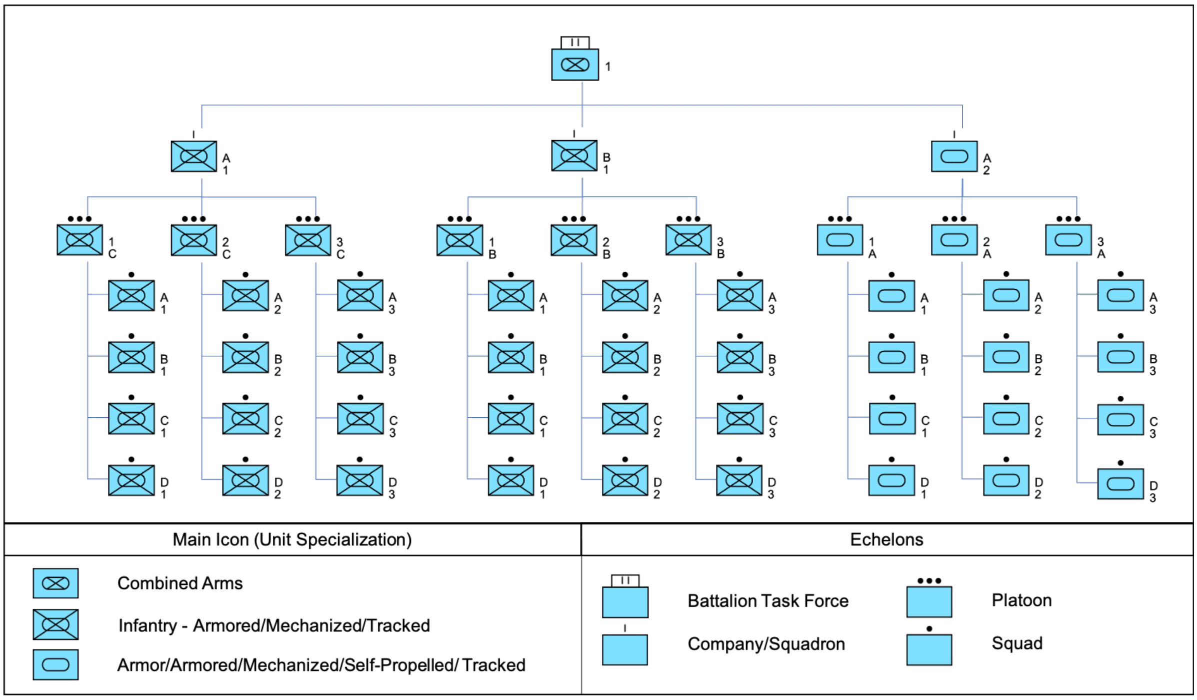


Figure B.3: Blue Team ORBAT.

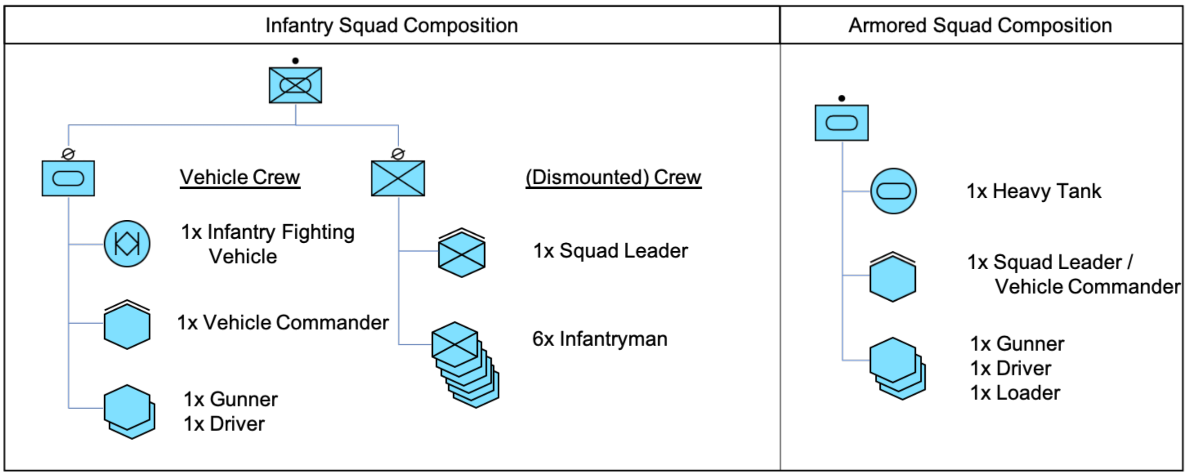


Figure B.4: Blue Team TOE.

B.2.2. Phases within 'COA SWIFT'

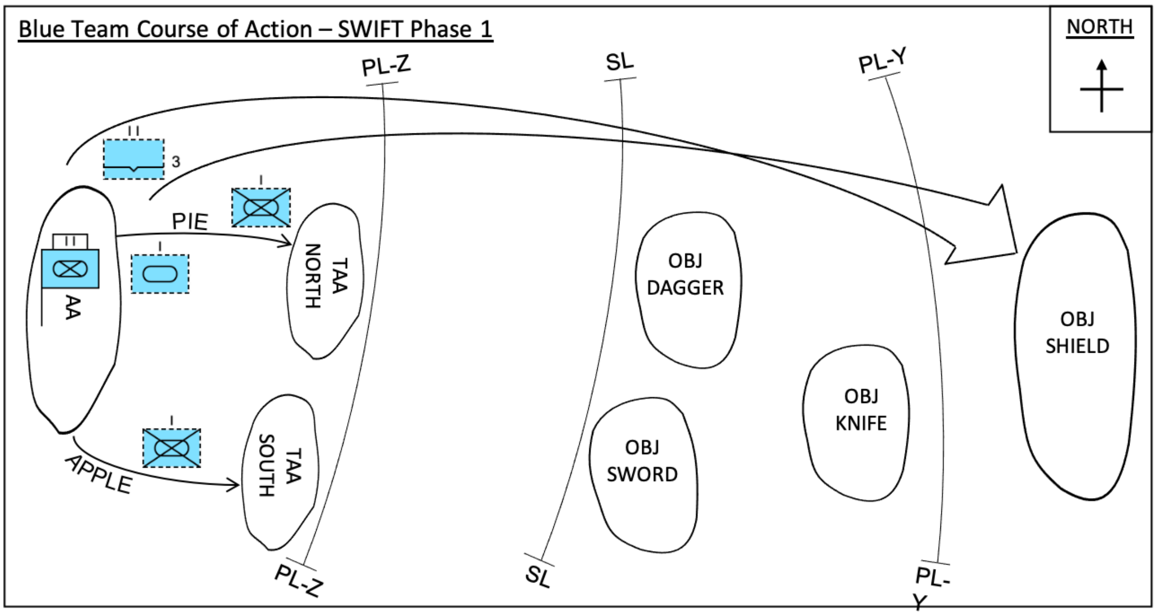


Figure B.5: Blue Team COA "SWIFT" - Phase 1.

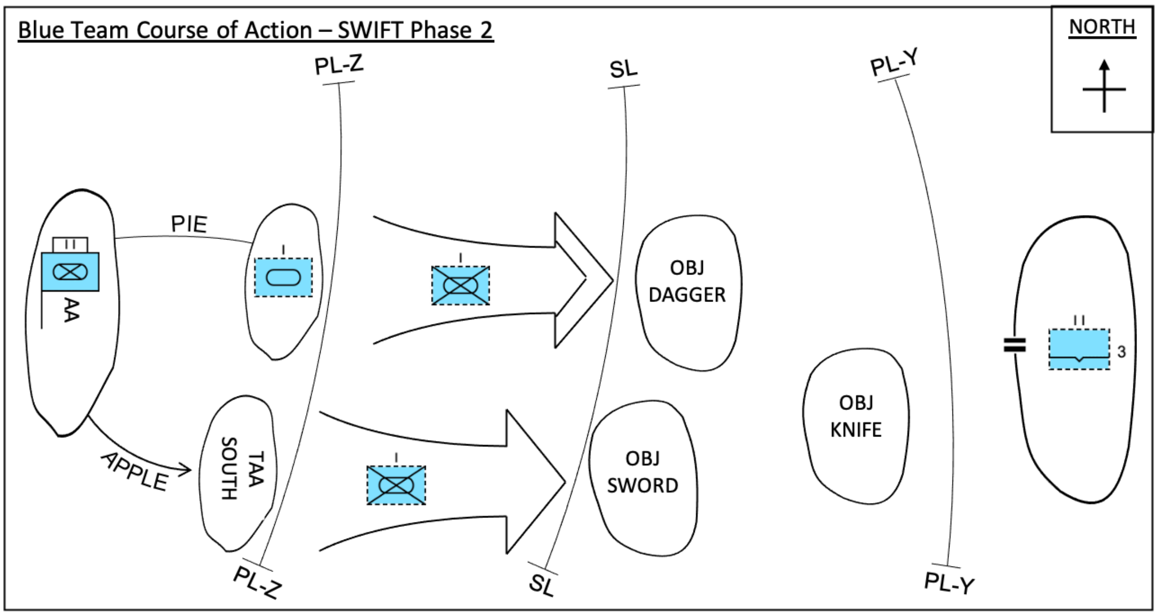


Figure B.6: Blue Team COA "SWIFT" - Phase 2.

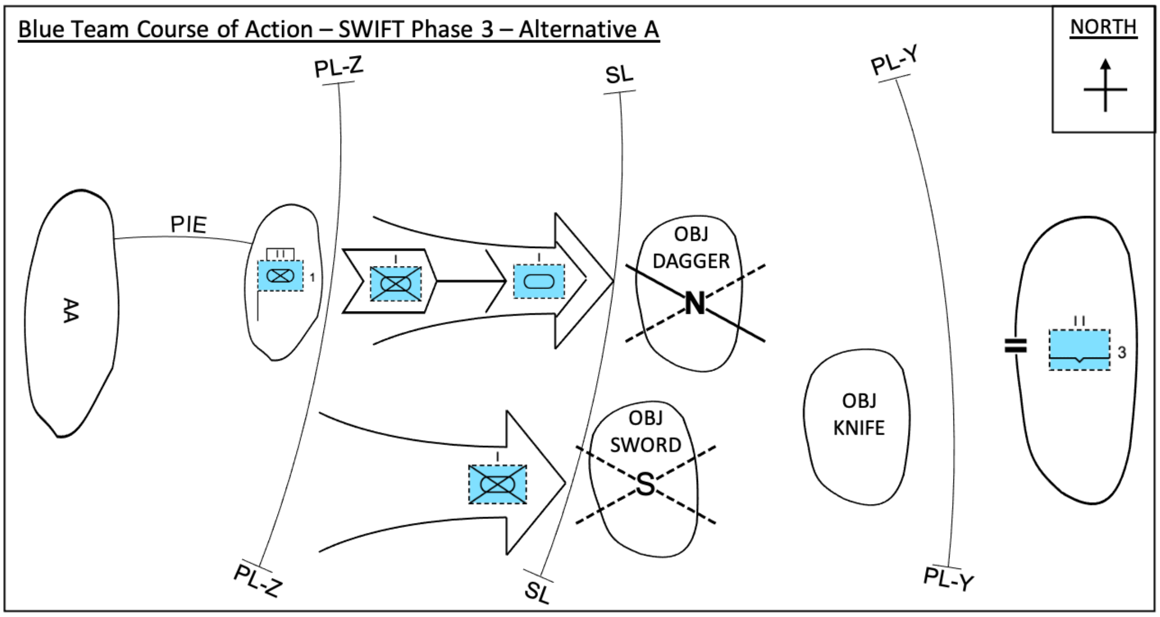


Figure B.7: Blue Team COA ”SWIFT” - Phase 3 - Alternative A.

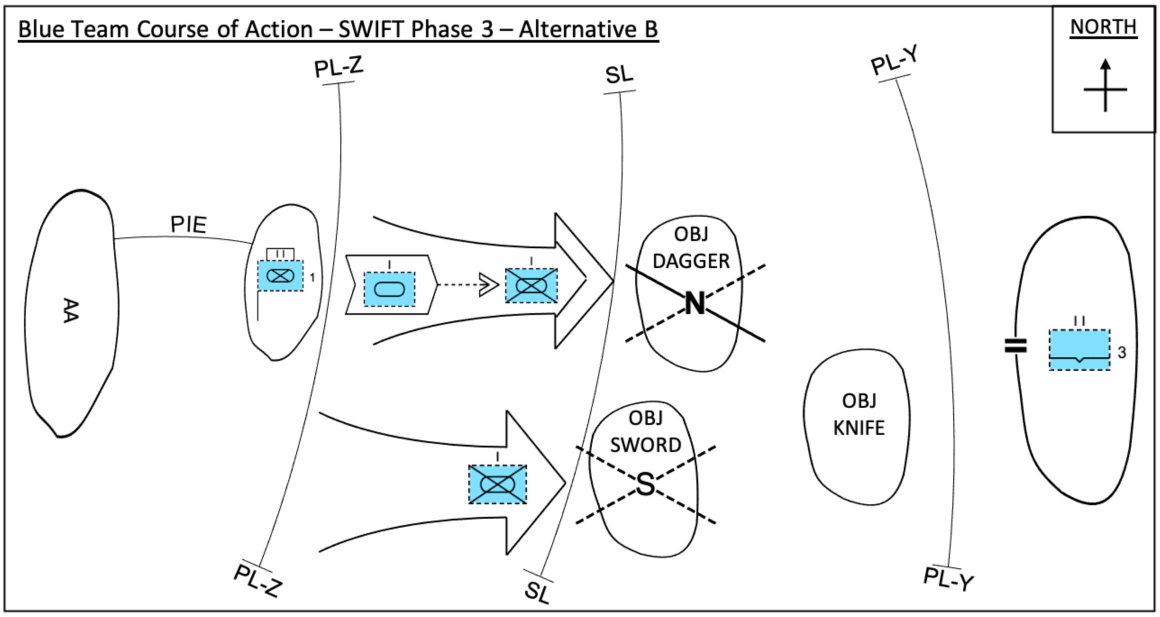


Figure B.8: Blue Team COA ”SWIFT” - Phase 3 - Alternative B.

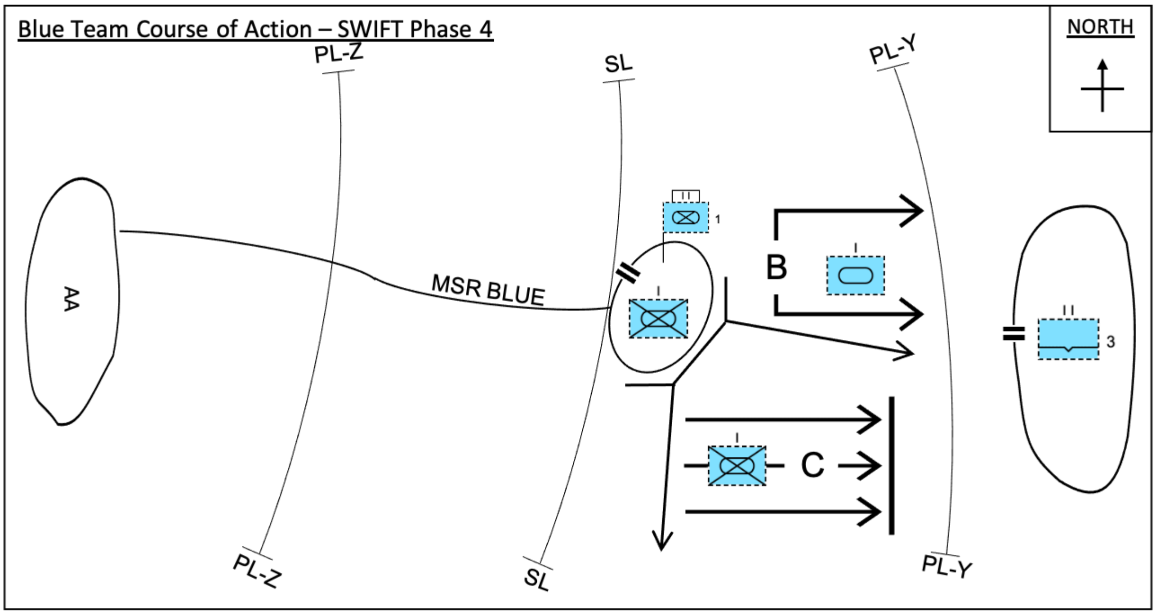


Figure B.9: Blue Team COA "SWIFT" - Phase 4.

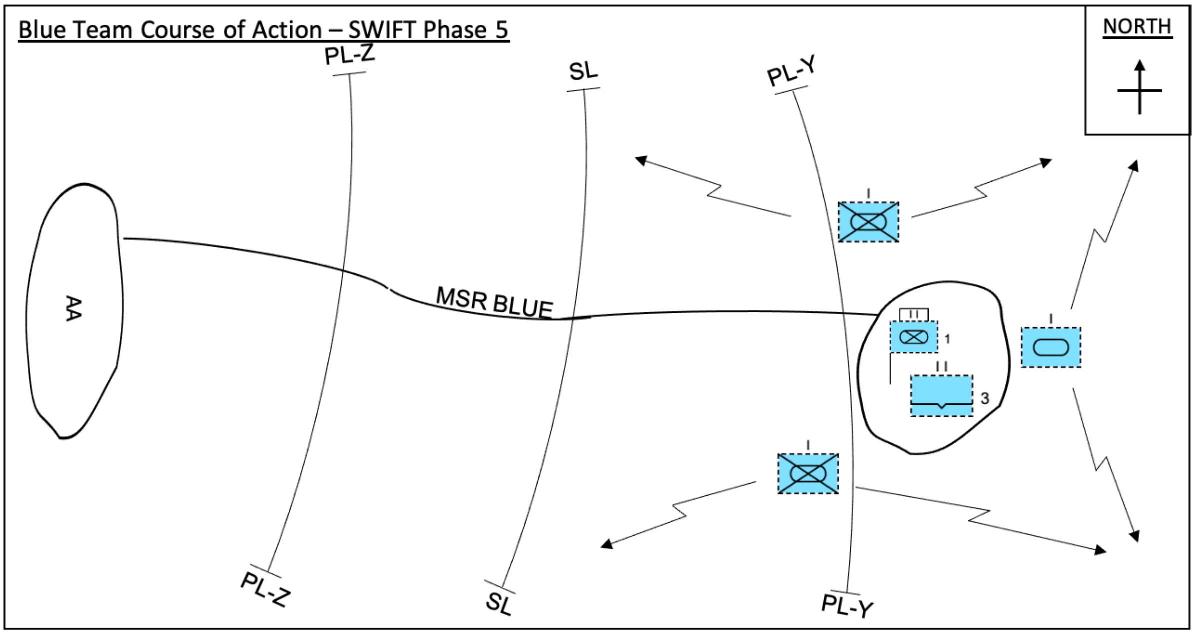


Figure B.10: Blue Team COA "SWIFT" - Phase 5.

B.2.3. Sub-unit Tasks

1. **A-Infantry Company (A-Coy):**
    - 1.1 Convoy move to TAA NORTH
    - 1.2 Deploy to SL
    - 1.3 Attack object DAGGER
    - 1.4 Support by fire B-Coy and TK-Sqdn
    - 1.5 Consolidate
    - 1.6 Deploy a screen to the northern flank
    - 1.7 Resupply with Class III (fuel) and V (ammunition)
2. **B-Infantry Company (B-Coy):**
    - 2.1 Convoy move to TAA SOUTH
    - 2.2 Deploy to SL
    - 2.3 Attack object SWORD
    - 2.4 Clear object KNIFE
    - 2.5 Consolidate
    - 2.6 Deploy a screen to the southern flank
    - 2.7 Resupply with Class III (fuel) and V (ammunition)
3. **Tank Squadron (TK-Sqdn):**
    - 3.1 Convoy move to TAA NORTH
    - 3.2 Deploy to SL
    - 3.3 Follow A-Coy and assume advance to object SHIELD
    - 3.4 Link up with 3(BT)AASLTBn in object SHIELD
    - 3.5 Consolidate
    - 3.6 Deploy a screen to the eastern flank
    - 3.7 Resupply with Class III (fuel) and V (ammunition)
4. **Rear Operations**
    - 4.1 BnHQ move to TAA NORTH
    - 4.2 BnHQ move to object DAGGER
    - 4.3 BnHQ move to object SHIELD
5. **Combat Support:**
    - 5.1 Prepare TAAs
    - 5.2 Mobility Support
    - 5.3 Repair Main Supply Route (MSR)
6. **Combat Service Support:**
    - 6.1 Move to TAA NORTH
    - 6.2 Move to object DAGGER
    - 6.3 Move to object SHIELD
    - 6.4 Resupply Class III (fuel) and V (ammunition)

C
Design Implementation

C.1. Scenario Tasks Linked to Components

Table C.1: Sub-unit tasking coupled to simulation components.

Component		Event			Control
		Movement	Offensive	Support	Regroup
Sub-unit Task	Ref#				
Convoy move to TAA NORTH	1.1	Convoy			
Deploy to SL	1.2	Tactical			
Attack object DAGGER	1.3		Breach		
Support B-Coy and TK-Sqdn	1.4		Attack by fire		
Consolidate	1.5				Consolidate
Deploy a screen to the northern flank	1.6				Screen
Resupply with Class III (fuel)	1.7a			Re-supply	
Resupply with Class V (ammunition)	1.7b			Re-supply	
Convoy move to TAA SOUTH	2.1	Convoy			
Deploy to SL	2.2	Tactical			
Neutralize object SWORD	2.3		Neutralize		
Clear object KNIFE	2.4		Clear		
Consolidate	2.5				Consolidate
Deploy a screen to the southern flank	2.6				Screen
Resupply with Class III (fuel)	2.7a			Re-supply	
Resupply with Class V (ammunition)	2.7b			Re-supply	
Convoy move to TAA NORTH	3.1	Convoy			
Deploy to SL	3.2	Tactical			
Follow A-Coy and assume advance to obj SHIELD	3.3		Follow and assume		
Link up with 3(BT)AASLTBn in obj SHIELD	3.4				Link up
Consolidate	3.5				Consolidate
Deploy a screen to the eastern flank	3.6				Screen
Resupply with Class III (fuel)	3.7a			Re-supply	
Resupply with Class V (ammunition)	3.7b			Re-supply	
BnHQ move to TAA NORTH	4.1	Tactical			
BnHQ move to object DAGGER	4.2	Tactical			
BnHQ move to object SHIELD	4.3	Tactical			
Prepare TAAs	5.1			Clear area	
Mobility Support	5.2			Clear route	
Repair Main Supply Route (MSR)	5.3			Repair road	
Move to TAA NORTH	6.1	Convoy			
Move to object DAGGER	6.2	Convoy			
Move to object SHIELD	6.3	Convoy			
Resupply with Class III (fuel)	6.4			Supply Class III	
Resupply with Class V (ammunition)	6.5			Supply Class V	

C.2. Implementation

C.2.1. Capability Database

Table C.2: Example for the Capability Database.

Capability	Factor	Value	Unit
Infantry Fighting Vehicle (IFV)	operational_range_off-road	320	km
	operational_range_on-road	500	km
	fuel_capacity	940	l
	fuel_consumption	60	l/h
	max_road_speed	70	km/h
	average_tactical_speed	10	km/h
	ammunition_storage	2500	rounds
	fire_rate	100	rpm
	weapon damage	0.05	-
	chance of hitting a target (accuracy)	2,5	%
	armor	0.2	-
Dismounted Infantry (INF)	operational_range_off-road	n.a.	
	operational_range_on-road	n.a.	
	fuel_capacity	n.a.	
	fuel_consumption	n.a.	
	max_road_speed	5	km/h
	average_tactical_speed	2.5	km/h
	ammunition_storage	200	rounds
	fire_rate	15	rpm
	weapon damage	0.01	-
	chance of hitting a target (accuracy)	10	%
	armor	0.01	-
Main Battle Tank (MBT)	operational_range_off-road	320	km
	operational_range_on-road	340	km
	fuel_capacity	1200	l
	fuel_consumption	150	l/h
	max_road_speed	72	km/h
	average_tactical_speed	10	km/h
	ammunition_storage	42	rounds
	fire_rate	10	rpm
	weapon damage	0.20	-
	chance of hitting a target (accuracy)	25	%
	armor	0.8	-

C.2.2. Simulation Models

Equation C.1 describes the applied movement model within the Event-Component, configured to simulate a (tactical) movement, to calculate the movement times. Equation C.2 describes the applied fuel consumption model within the Event-Component, configured to simulate a (tactical) movement, to calculate the amount of fuel that is consumed.

$$M = P + V + S + T + C \tag{C.1}$$

$$F = \frac{f(n(P + T + C) + V + S)}{60} \tag{C.2}$$

where:

M = total movement time (min)
P = movement preparation time (min)
V = total waiting time between vehicles (min)
S = total waiting time between sub-units (min)
T = vehicle travel time (min)
C = movement completion time (min)
F = total fuel consumption (l)
f = vehicle fuel consumption (l/h)
n = number of vehicles (-)

To support the above equations: equation C.3 calculates the net movement time for a individual vehicle or platform; equation C.4 calculates the time a vehicle or platform has to wait on the preceding vehicle before it can start with its movement; and equation C.5 calculates the time a unit has to wait on the preceding unit before it can start with its movement.

$$T = m_p + d_t + m_c \tag{C.3}$$

where:

T = vehicle movement time (min)
m_p = movement preparation time (min)
d_t = time to cover distance
m_c = movement completion time (min)

$$V = m * \frac{(n-1) * m_d}{v_{max} * u} \tag{C.4}$$

where:

V = total vehicle waiting time (min)
m = movement factor
n = number of vehicles (-)
m_d = movement distance between vehicles (m)
u = formation speed correction factor (-)
v_{max} = maximum vehicle speed (m/s)

$$S = m_t(\frac{plt}{sqd} - 1) \tag{C.5}$$

where:

S = total unit waiting time (min)
m_t = movement time between platoons (min)
plt = number of platoons (-)
sqd = number of squads (-)

Equations C.6 and C.7 describe decrease in unit size during an engagement between the teams as a (negative) factor of the other sides fighting strength. Based on the duration of the engagment equation C.8 calculates the amount of fuel that is consumed.

$$\frac{dB}{dt} = -alpha * R \tag{C.6}$$

$$\frac{dR}{dt} = -beta * B \tag{C.7}$$

$$F = \frac{fnT}{60} \tag{C.8}$$

where:

B = Blue Team size
R = Red Team size
t = combat round (-)
$alpha$ = relative fighting strength Red Team (min^{-1})
$beta$ = relative fighting strength Blue Team (min^{-1})
F = total fuel consumption (l)
f = vehicle fuel consumption (l/h)
n = number of vehicles (-)
T = engagement time (min)

To support the above equations: equations C.9 and C.10 calculates the relative fighting strength, the offensive strength divided by the defensive protection, for each team; equation C.11 calculates the offensive strength of a team; and equation C.12 calculates defensive protection of a team.

$$alpha = \frac{S(R)}{P(B)} \tag{C.9}$$

$$beta = \frac{S(B)}{P(R)} \tag{C.10}$$

$$S = S_{platform} * R * rho * A * E \tag{C.11}$$

$$P = P_{platform} * I * E * O \tag{C.12}$$

where:

$alpha$ = relative fighting strength Red Team
$beta$ = relative fighting strength Blue Team
S = strength (1/s)
$S_{platform}$ = initial platform strength (-)
P = protection (-)
$P_{platform}$ = initial platform protection (-)
R = firing rate (1/s)
rho = shooting sector (-)
A = platform accuracy (-)
E = execution (-)
I = impact sector (-)
O = posture (-)

C.2.3. Data Sources

Table C.3: Data Sources for the movement model within an Event-Component.

Data Source	Component Variable
simulator configuration	simulation level
Unit-Component (config)	vehicle type (-) number of vehicles (n) number of platoons (plt) number of squads (sqd)
Unit-Component (attributes states)	capability personnel location fuel ammunition combat readiness (health)
user configuration of component	type of movement (on/off-road) type of formation (column/line/wedge) movement preparation time (P) movement distance (d) movement completion time (C)
capability database	maximum vehicle speed (v_{max}) fuel consumption (f)
model parameters	movement factor (m) formation speed correction factor (u) movement distance between vehicles (m_d) movement time between platoons (m_t)
computation by model	total waiting time between vehicles (V) total waiting time between sub-units (S) vehicle movement time (T) consumed fuel (F)

Table C.4: Data Sources for the engagement model within an Event-Component.

Data Source	Component Variable
simulator configuration	simulation level
Unit-Component (config)	vehicle type (-) number of vehicles (n) number of platoons (plt) number of squads (sqd) number of troops (trp)
Unit-Component (attributes states)	capability personnel location fuel ammunition combat readiness (health)
user configuration of component	type of movement (on/off-road) sector (front/flank/rear) execution (deliberate/hasty) posture (moving/stationary/covered)
capability database	initial platform strength ($S_{platform}$) initial platform protection ($P_{platform}$) firing rate (R) accuracy (A) fuel consumption (f)
model parameters	shooting sector (rho) impact sector (I) offensive/defensive execution (E) posture (O)
computation by model	relative fighting strength (alpha/beta) strength (S) protection (P) consumed fuel (F)

C.2.4. Unit-Component Configuration

Table C.5: Unit configuration and initial attribute states that is inserted in the Unit-Component.

Data Type	Attribute	State
Configuration	Component_id	86319ddee587
	Symbol Identification Code	30-0-3-10-0-4-16-121000-00-00
	Context	Reality
	Standard identity	Friend
	Symbol set	Land unit
	Status	Present
	Headquarters/Task force/Dummy	Task force
	Echelon	Battalion/Squadron
	Main icon	Movement and Manoeuvre - Combined Arms
	Additional information	1(BT)CABn
Attributes	Equipment	100%
	Personnel	100%
	Location	31U ET 9400 6220
	Fuel	100%
	Food	100%
	Ammunition	100%
	Combat Readiness	100%

C.2.5. Event-Component Configuration

Table C.6: Model configuration of Event-Component for the on-road-movement tasks.

Data Source	Variable	Value	Unit
Unit-Component (config)	number of vehicles (n)	12	
	number of platoons (plt)	3	
	number of squads (sqd)	12	
component configuration	type of movement (on/off-road)	on-road	
	type of formation (column/line/wedge)	column	
	movement preparation time (P)	15	min
	movement distance (d)	15000	m
	movement completion time (C)	30	min
model parameters	movement factor (m)	1	
	formation speed correction factor (u)	0.5	
	movement distance between vehicles (m_d)	100	m
	movement time between platoons (m_t)	5	min
equipment database	IFV max speed	60	km/h
	MBT max speed	75	km/h
	IFV fuel consumption	60	l/h
	MBT fuel consumption	100	l/h
	IFV fuel capacity	940	l
	MBT fuel capacity	1200	l

Table C.7: Model configuration of Event-Component for the off-road-movement tasks.

Data Source	Variable	Value	Unit
Unit-Component (config)	number of vehicles (n)	12	
	number of platoons (plt)	3	
	number of squads (sqd)	12	
component configuration	type of movement (on/off-road)	off-road	
	type of formation (column/line/wedge)	wedge	
	movement preparation time (P)	60	min
	movement distance (d)	15000	m
	movement completion time (C)	5	min
model parameters	movement factor (m)	0	
	formation speed correction factor (u)	0.2	
	movement distance between vehicles (m_d)	50	m
	movement time between platoons (m_t)	1	min
equipment database	IFV max speed	60	km/h
	MBT max speed	75	km/h
	IFV fuel consupmtion	60	l/h
	MBT fuel consumption	100	l/h
	IFV fuel capacity	940	l
	MBT fuel capacity	1200	l

D

Proof of Concept Results

D.1. Computations Experiment 1.

D.1.1. Experiment 1.1. Movement AA to TAAs

Table D.1: Computation of the Event-Component for the on-road-movement tasks.

Unit	Vehicle Type	T	F	P	V	S	T	C
A(BT)InfCoy	Infantry Fighting Vehicle	92	1082	15	2.2	15	30	30
B(BT)InfCoy	Infantry Fighting Vehicle	92	1082	15	2.2	15	30	30
C(BT)TkSqdn	Main Battle Tank	91	1784	15	2.2	20	24	30

D.1.2. Experiment 1.2. Movement TAAs to SL

Table D.2: Computation of the Event-Component for the off-road-movement tasks.

Unit	Vehicle Type	T	F	P	V	S	T	C	
A(BT)InfCoy	Infantry Fighting Vehicle	143	1716		60	0.0	3	75	5
B(BT)InfCoy	Infantry Fighting Vehicle	143	1716		60	0.0	3	75	5
C(BT)TkSqdn	Main Battle Tank	129	2580		60	0.0	4	60	5

D.2. Computations Fighting power and Protective Measures

Table D.3: Computation of the fighting power for Most Likely Red Team COAs.

Team	Capability	Damage	Rate	Accuracy	Sector	Execution	Fighting Power
Blue	Main Battle Tank	0.20	0.17	0.25	1.00	1.00	0.00833
	Infantry Fighting Vehicle	0.05	0.17	0.25	1.00	1.00	0.00208
	dismounted Infantry	0.01	0.25	0.10	1.00	1.00	0.00025
Red	dismounted Infantry	0.01	0.25	0.10	1.00	1.00	0.00025

Table D.4: Computation of the protective measures for Most Likely Red Team COAs.

Team	Capability	Armor	Posture	Sector	Execution	Protective Measure
Blue	Main Battle Tank	0.80	0.40	1.00	1.00	0.32000
	Infantry Fighting Vehicle	0.20	0.10	1.00	1.00	0.02000
	dismounted Infantry	0.01	1.00	1.00	1.00	0.0040
Red	dismounted Infantry	0.01	1.00	1.00	1.00	0.01000

Table D.5: Computation of the fighting power for Most Dangerous Red Team COAs.

Team	Capability	Damage	Rate	Accuracy	Sector	Execution	Fighting Power
Blue	Main Battle Tank	0.20	0.17	0.25	0.70	0.50	0.0029
	Infantry Fighting Vehicle	0.05	0.25	0.25	0.70	0.50	0.0011
	dismounted Infantry	0.01	0.25	0.10	0.70	0.50	0.0001
Red	dismounted Infantry	0.01	0.25	0.10	1.00	1.00	0.0003

Table D.6: Computation of the protective measures for Most Dangerous Red Team COAs.

Team	Capability	Armor	Posture	Sector	Execution	Protective Measure
Blue	Main Battle Tank	0.80	0.40	0.50	0.60	0.0960
	Infantry Fighting Vehicle	0.20	0.40	0.50	0.60	0.0240
	dismounted Infantry	0.01	0.40	0.50	0.60	0.0012
Red	dismounted Infantry	0.01	1.00	1.00	1.00	0.0100

D.3. Computations Experiment 2.

D.3.1. Experiment 2.1. BT SWIFT-A x RT ML

Table D.7: Computation of engagement results between A(BT)InfCoy and Red Team Forces.

	Blue Team				Red Team	
	IFV		INF		INF	
Round (t)	Health	Reduction	Health	Reduction	Health	Reduction
0	12.0	0.1	72.0	0.6	28.0	3.7
1	11.9	0.1	71.4	0.6	24.3	3.7
2	11.7	0.1	70.7	0.6	20.6	3.7
3	11.6	0.1	70.1	0.6	16.9	3.6
4	11.5	0.1	69.5	0.6	13.3	3.6
5	11.4	0.1	68.9	0.6	9.7	3.6
6	11.2	0.1	68.2	0.6	6.1	3.5
7	11.1	0.1	67.6	0.6	2.6	3.5
8	11.0	0.1	67.0	0.6	-0.9	3.5

D.3.2. Experiment 2.2. BT SWIFT-B x RT ML

Table D.8: Computation of engagement results between C(BT)TkSqdn and Red Team forces.

	Blue Team		Red Team	
	MBT		INF	
Round (t)	Health	Reduction	Health	Reduction
0	12.0	0.0	28.0	10.0
1	12.0	0.0	18.0	10.0
2	12.0	0.0	8.0	10.0

Table D.9: Computation of engagement results between A(BT)InfCoy and Red Team forces.

	Blue Team				Red Team	
	IFV		INF		INF	
Round (t)	Health	Reduction	Health	Reduction	Health	Reduction
0	12.0	0.1	72.0	0.6	8.0	3.7
1	11.9	0.1	72.0	0.6	4.3	3.7
2	11.7	0.1	72.0	0.6	0.6	3.7
3	11.6	0.1	72.0	0.6	-3.1	3.6

D.3.3. Experiment 2.3. BT SWIFT-A x RT MD

Table D.10: Computation of engagement results between A(BT)InfCoy and Red Team Forces.

	Blue Team				**Red Team**	
	IFV		**INF**		**INF**	
Round (t)	**Health**	**Reduction**	**Health**	**Reduction**	**Health**	**Reduction**
0	12.0	0.1	72.0	0.8	56.0	1.7
1	11.9	0.1	71.3	0.7	54.3	1.6
2	11.8	0.1	70.5	0.7	52.7	1.6
3	11.7	0.1	69.8	0.7	51.1	1.6
4	11.6	0.1	69.0	0.7	49.4	1.6
5	11.5	0.1	68.3	0.7	47.8	1.6
6	11.4	0.1	67.6	0.7	46.3	1.6
7	11.3	0.1	66.9	0.7	44.7	1.5
8	11.2	0.1	66.2	0.7	43.2	1.5
9	11.1	0.1	65.5	0.7	41.6	1.5
10	11.0	0.1	64.8	0.7	40.1	1.5
11	10.9	0.1	64.1	0.7	38.6	1.5
12	10.8	0.1	63.4	0.7	37.1	1.5
13	10.7	0.1	62.8	0.7	35.7	1.5
14	10.6	0.1	62.1	0.7	34.2	1.4
15	10.5	0.1	61.4	0.7	32.8	1.4
16	10.4	0.1	60.8	0.7	31.4	1.4
17	10.3	0.1	60.1	0.6	30.0	1.4
18	10.2	0.1	59.5	0.6	28.6	1.4
19	10.1	0.1	58.8	0.6	27.2	1.4
20	10.1	0.1	58.2	0.6	25.8	1.3
21	10.0	0.1	57.6	0.6	24.5	1.3
22	9.9	0.1	56.9	0.6	23.1	1.3
23	9.8	0.1	56.3	0.6	21.8	1.3
24	9.7	0.1	55.7	0.6	20.5	1.3
25	9.6	0.1	55.1	0.6	19.2	1.3
26	9.6	0.1	54.5	0.6	18.0	1.3
27	9.5	0.1	53.9	0.6	16.7	1.2
28	9.4	0.1	53.3	0.6	15.5	1.2
29	9.3	0.1	52.7	0.6	14.2	1.2
30	9.2	0.1	52.1	0.6	13.0	1.2
31	9.2	0.1	51.6	0.6	11.8	1.2
32	9.1	0.1	51.0	0.6	10.6	1.2
33	9.0	0.1	50.4	0.6	9.4	1.2
34	8.9	0.1	49.9	0.6	8.3	1.2
35	8.9	0.1	49.3	0.6	7.1	1.1
36	8.8	0.1	48.8	0.5	6.0	1.1
37	8.7	0.1	48.2	0.5	4.8	1.1
38	8.6	0.1	47.7	0.5	3.7	1.1
39	8.6	0.1	47.1	0.5	2.6	1.1
40	8.5	0.1	46.6	0.5	1.5	1.1
41	8.4	0.1	46.1	0.5	0.5	1.1
42	8.4	0.1	45.5	0.5	-0.6	1.1

Table D.11: Computation of engagement results between C(BT)TkSqdn and Red Team forces.

	Blue Team		**Red Team**	
	MBT		**INF**	
Round (t)	**Health**	**Reduction**	**Health**	**Reduction**
0	12.0	0.1	28.0	3.5
1	11.9	0.1	24.5	3.5
2	11.9	0.1	21.0	3.5
3	11.8	0.0	17.6	3.4
4	11.8	0.0	14.1	3.4
5	11.7	0.0	10.7	3.4
6	11.7	0.0	7.3	3.4
7	11.7	0.0	3.9	3.4
8	11.7	0.0	0.4	3.4
9	11.7	0.0	-3.0	3.4

D.3.4. Experiment 2.4. BT SWIFT-B x RT MD

Table D.12: Computation of engagement results between A(BT)InfCoy and Red Team forces.

	Blue Team				**Red Team**	
	IFV		**INF**		**INF**	
Round (t)	**Health**	**Reduction**	**Health**	**Reduction**	**Health**	**Reduction**
0	12.0	0.1	72.0	0.8	28.0	1.7
1	11.9	0.1	71.3	0.7	26.3	1.6
2	11.8	0.1	70.5	0.7	24.7	1.6
3	11.7	0.1	69.8	0.7	23.1	1.6
4	11.6	0.1	69.0	0.7	21.4	1.6
5	11.5	0.1	68.3	0.7	19.8	1.6
6	11.4	0.1	67.6	0.7	18.3	1.6
7	11.3	0.1	66.9	0.7	16.7	1.5
8	11.2	0.1	66.2	0.7	15.2	1.5
9	11.1	0.1	65.5	0.7	13.6	1.5
10	11.0	0.1	64.8	0.7	12.1	1.5
11	10.9	0.1	64.1	0.7	10.6	1.5
12	10.8	0.1	63.4	0.7	9.1	1.5
13	10.7	0.1	62.8	0.7	7.7	1.5
14	10.6	0.1	62.1	0.7	6.2	1.4
15	10.5	0.1	61.4	0.7	4.8	1.4
16	10.4	0.1	60.8	0.7	3.4	1.4
17	10.3	0.1	60.1	0.6	2.0	1.4
18	10.2	0.1	59.5	0.6	0.6	1.4
19	10.1	0.1	58.8	0.6	-0.8	1.4

Table D.13: Computation of engagement results between C(BT)TkSqdn and Red Team forces.

	Blue Team		**Red Team**	
	MBT		**INF**	
Round (t)	**Health**	**Reduction**	**Health**	**Reduction**
0	12.0	0.1	56.0	3.5
1	11.9	0.1	52.5	3.5
2	11.7	0.1	49.0	3.4
3	11.6	0.1	45.6	3.4
4	11.5	0.1	42.2	3.3
5	11.4	0.1	38.9	3.3
6	11.3	0.1	35.6	3.3
7	11.2	0.1	32.3	3.3
8	11.1	0.1	29.0	3.2
9	11.0	0.1	25.8	3.2
10	10.9	0.1	22.6	3.2
11	10.9	0.1	19.4	3.2
12	10.8	0.0	16.2	3.2
13	10.8	0.0	13.1	3.1
14	10.8	0.0	9.9	3.1
15	10.7	0.0	6.8	3.1
16	10.7	0.0	3.7	3.1
17	10.7	0.0	0.5	3.1
18	10.7	0.0	-2.6	3.1