SISO Simulation Interoperability Standards Organization
www.sisostds.org

SISO-GUIDE-006-2018

Guideline on Scenario Development for Simulation Environments

10 May 2018

SAC Approved: 05/02/2018 EXCOM Approved: 05/10/2018

Prepared by
Guideline for Scenario Development (GSD)
Product Development Group

SISO Inc. Board of Directors
P.O. Box 781238
Orlando, FL 32878-1238, USA

Revision History

Version	Section	Date (MM/DD/YYYY)	Description
1.0	All	05/10/2018	Initial Document

Participants

At the time this product was submitted to the Standards Activity Committee (SAC) for approval, the "Guideline on Scenario Development for Simulation Environments" Product Development Group had the following membership and was assigned the following SAC Technical Area Director:

Product Development Group

Stefan Vrieler (Chair)
Lionel Khimeche (Vice-Chair)
Scott Roach (Vice-Chair)
Stefan Vrieler (Secretary)

Jean-Louis Igarza (SAC Technical Area Director)

Jeff Abbott
Uwe Dobrindt
Kevin Galvin
Arno Gerretsen
Jean-Baptiste Guillerit
Axel Lehmann
Robert Lutz
Craig Marsden
Mark McCall
Patrick Merlet
Michael Montgomery
Laurent Prignac
Andreas Reif
Nelson Reynolds
David Ronnfeldt
Roy Scrudder
Tommy Shook
Robert Siegfried

The Product Development Group would like to especially acknowledge those individuals that significantly contributed to the preparation of this product as follows:

PDG Drafting Group

Robert Siegfried (Editor)

Mike Montgomery
Scott Roach
Colin Timmons

The following individuals comprised the ballot group for this product.

Ballot Group

Curtis Blais
Anthony Cramp
Arno Gerretsen
Jean-Louis Gougeat
Frank Hill
Robert Lutz
Lance Marrou
Mark McCall
Mike Montgomery
Nelson Reynolds
David Ronnfeldt
Roy Scrudder
Robert Siegfried
Tom van den Berg
Steven Weiss

When the Standards Activity Committee approved this product on 18 April 2018, it had the following membership:

Standards Activity Committee

Katherine L. Morse (Chair)
Jean-Louis Igarza (Vice Chair)
Lance Marrou (Secretary)

Grant Bailey
Brad Dillman
Kevin Gupton
Michael Heaphy
Aerial Kreiner
Patrice Le Leydour
Chris McGroarty
Thom McLean
John Stevens
Paul Gustavson

When the Executive Committee approved this product on 10 May 2018, it had the following membership:

Executive Committee

Michael O’Connor (Chair)
Robert Lutz (Vice Chair)
Jeff Abbott (Secretary)

Paul Gustavson
Kenneth Konwin
Lana McGlynn
Katherine L. Morse
William Oates
Roy Scrudder
Robert Siegfried
Eric Whittington

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Guideline on Scenario Development for Simulation Environments

1. Introduction

1.1. Motivation

Regardless of the application domain – e.g., training, analysis or decision support – scenarios are used to describe situations and conditions to be represented in a simulation environment for the intended purpose of a simulation application. Typically a scenario description contains information about the geographic location, involved entities, the initial situation and pre-planned courses of action and major events that should occur during the simulation execution. Therefore, scenarios defined by the user of a simulation environment are paramount sources of requirements for the engineers faced with planning and setting up a simulation environment. As such, well-specified scenarios are of utmost importance. If such scenarios are missing, this may lead to various problems due to misunderstandings when talking about the objectives and scope of a simulation environment. As a result, the conceptual model (and the subsequently developed simulation environment) may not reflect what the user originally wanted. In other words, the simulation environment does not fulfil the original requirements or does not answer the questions originally posed. Therefore, missing or incompletely specified scenarios will lead to simulation results that do not reflect what the user was expecting.

Ideally, a scenario description should be

- complete,
- consistent and
- understandable.

Completeness means that a scenario description has to contain sufficient information to enable persons in the subsequent process (especially during development of the conceptual model) to use the scenario in a meaningful way and to extract all information required for their activities. Consistency refers to the internal correctness of a scenario description (e.g., no unit belongs to more than one party, initial positions of all units are within the specified geographic area). Understandability requires that a scenario description has to be written and structured in a way that it is easily accessible by future users.

1.2. Purpose

The purpose of this guideline is to provide detailed information regarding the development of scenarios for (distributed) simulation environments and the relationship of the scenario development process with the overarching simulation environment engineering process. This guideline is based on the Distributed Simulation Engineering and Execution Process (DSEEP) [8] and augments the DSEEP with additional information specific to scenario development.

This guideline gives an overview of existing standards and tools that may be used for scenario development. Whenever possible this guideline prefers open standards and recommends their use.

1.3. Scope

The primary scope of this guideline is the development of scenarios in context of simulation environments. The majority of this guideline is domain neutral and may be applied to a wide variety of application areas. However, parts of the guideline are currently still focused on military scenarios.

Although similar in nature, this guideline does not explicitly address scenarios used in live exercises or computer-assisted exercises (CAX). Nevertheless, many considerations laid out in this guideline are also applicable to these application domains.

1.4. Intended audience

The intended audience of this guideline includes (but is not limited to):

- Primarily, project managers and simulation engineers who need to set up a simulation environment.
- Users and subject matter experts who define and specify requirements for a simulation environment.
- Operator personnel that operate simulation systems and other member applications of a simulation environment.

This guideline is not restricted to a specific application domain (e.g., military scenarios), but may be applied in various different application domains (e.g., crisis management scenarios).

1.5. Document outline

- As many problems arise out of different interpretations of the term “scenario”, Chapter 4 defines the term “scenario” and provides related terminology. Afterwards, different types of scenarios within different steps of the DSEEP are identified and described in detail.
- Chapter 5 presents necessary scenario documentation. Based on both practical experiences as well as literature research, a set of documentation items for describing scenarios is presented.
- Chapter 6 discusses maturity levels for scenario descriptions and highlights the importance of formal scenario descriptions (e.g., using MSDL).
- Chapter 7 provides an overview of currently available standards and tools for scenario description. Features currently missing in available standards are identified.
- Appendices B and C describe how the NATO Architecture Framework and Base Object Models are related to scenario descriptions.
- Appendices D and E present two example scenario descriptions.

2. References (Normative)

2.1. SISO References

#	Title
1.	SISO: *Military Scenario Definition Language (MSDL)*. 14.10.2008. – SISO-STD-007-2008.
2.	SISO: *Base Object Model (BOM) Template Specification*. 31 March 2006. – SISO-STD-003-2006.
3.	SISO: *Standard for Coalition Battle Management Language (C-BML) Phase 1*, Version 1.0, 14 April 2014, SISO-STD-011-2014.

2.2. Other References

#	Title
4.	Andy Bowers, Charles L. Budde, Brian C. Gregg, Dan Winkowski: *JLVC Data Initialization*. SISO 2011 Fall SIW, Orlando, USA, 11F-SIW-034.
5.	E. Cayirci: *Tutorial on Joint Exercise Management Module (JEMM)*. 5th NATO CAX Forum 2010, Ottobrunn, Germany. August 2010.
6.	Department of Defense: *Modeling and Simulation (M&S) Glossary*. Published by Modeling and Simulation Coordination Office. October 1, 2011.
7.	Department of Defense Modeling & Simulation Coordination Office: *Modeling and Simulation (M&S) Community of Interest (COI) Discovery Metadata Specification (MSC-DMS)*. Version 1.4, 24 December 2010.
8.	IEEE: *IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP)*. 2011. – IEEE Std 1730™-2010.
9.	Removed
10.	Joint Publication 3-09.3: *Close Air Support*. 8 July 2009, http://www.fas.org/irp/doddir/dod/jp3_09_3.pdf
11.	Michel Keuning, Arno Gerretsen: *Effective Realism*. I/ITSEC 2008, paper 8187
12.	V. Mojtahed, M. Garcia Lozano, P. Svan, B. Anderson, V. Kabilan: *DCMF-Defence Conceptual Modelling Framework*. FOI-R--1754--SE, ISSN 1650-1942, November 2005
13.	V. Mojtahed, E. Tjörnhammer, J. Zdravkovic, A.Hanzala Khan: *The Knowledge Use in DCMF*. FOI-R--2606--SE, ISSN 1650-1942, November 2008
14.	Multilateral Interoperability Programme (MIP): *Joint Consultation, Command and Control Information Exchange Data Model (JC3IEDM)*. Version 3.1.4, 14 Feb 2012.
15.	NATO: *Code of Best Practice for C2 Assessment*. October 2002.
16.	NATO: *NATO Comprehensive Operations Planning Directive*. 17 December 2010.
17.	NATO: *Final Report of MSG-024 "M&S Support to Non-Article 5 Operations"*. RTO-TR-MSG-024 AC/323(MSG-024)TP/27. January 2008.
18.	NATO: *Final Report of MSG-053 "Rapid Scenario Generation for Simulation Applications".* RTO Technical Report TR-MSG-053 AC/323(MSG-053)TP/302. May 2010.
19.	NATO: *Final Report of MSG-068 "NATO Education and Training Network".* RTO Technical Report RTO-TR-MSG-068 AC/323(MSG-068)TP/407. February 2012.
20.	NATO: *Final Report of MSG-086 "Simulation Interoperability".* STO Technical Report TR-MSG-086. AC/323(MSG-086)TP/562. January 2015.
21.	NATO Consultation, Command and Control Board (NC3B): *NATO Architecture Framework*. Version 3, AC/322-D(2007)0048-AS1. 2007.
22.	NATO Standardization Agency (NSA): *NATO Glossary of Terms and Definitions*. Edition 2013.
23.	ADatP-3, Part 1: *NATO Message Text Formatting System (FORMETS) - System Concept Description and Management*. October 1987. Brussels: NATO Headquarters.
24.	NATO: *NATO Modelling and Simulation Standards Profile*. AMSP-01, Edition (B), Version 1. June 2011.

#	Title
25.	The Organization for the Advancement of Structured Information Standards (OASIS): *RELAX NG Specification*. Committee Specification, http://relaxng.org/spec-20011203.html, 3 December 2001. (See also “ISO/IEC 19757-2:2003 Document Schema Definition Language (DSDL) – Part 2: Regular-grammar-based validation – RELAX NG” for an ISO version of RELAX NG Specification).
26.	Object Management Group (OMG): *“Unified Modeling Language”*. Version 2.4.1, http://www.omg.org/spec/UML/2.4.1/, August 2011.
27.	Saab Systems & Combitech: *Core Technical Framework for distributed simulation*. Part of EU Core Technical Framework Study. EDA reference number 08-R&T-004. July 2009.
28.	Robert Siegfried, Günter Herrmann, Johannes Lüthi, Matthias Hahn: *“VEVA as German approach towards operationalizing the DSEEP: Overview and first experiences”.* SISO 2011 Spring SIW, Boston, USA, Paper 11S-SIW-044.
29.	Robert Siegfried, Günter Herrmann, Johannes Lüthi, Matthias Hahn: *“A comparison of DSEEP with the German approach VEVA”*. SISO 2011 Fall SIW, Orlando, USA, Paper 11F-SIW-023.
30.	Robert Siegfried, Alexander Laux, Martin Rother, Dieter Steinkamp, Günter Herrmann, Johannes Lüthi, Matthias Hahn: *“Scenarios in military (distributed) simulation environments”.* SISO 2012 Spring SIW, Paper 12S-SIW-014
31.	Robert Siegfried, Alexander Laux, Martin Rother, Dieter Steinkamp, Günter Herrmann, Johannes Lüthi, Matthias Hahn: *“Components and reuse in scenario development processes for distributed simulation environments”*. SISO 2012 Fall SIW, Orlando, USA, Paper 12F-SIW-046
32.	Robert Siegfried, Halit Oguztüzün, Umut Durak, Aylin Hatip, Günter Herrmann, Paul Gustavson, Matthias Hahn: *“Specification and documentation of conceptual scenarios using Base Object Models (BOMs)”.* 2013 SISO Spring SIW, San Diego, USA, Paper 13S-SIW-019.
33.	Robert Siegfried, Alexander Laux, Günter Herrmann, Johannes Lüthi: *“VEVA – A Procedure Model for Distributed Simulation Experiments”.* NATO MSG Symposium 2010, Utrecht, Netherlands, paper 010.
34.	Software Engineering Institute: *CMMI for Development.* Version 1.3, Technical report CMU/SEI-2010-TR-033, November 2010.
35.	Jan Tegner, Robert Suzic: *“Core Technical Framework Stud: DSEEP Adaption and MODAF mapping”*. SISO 2010 Spring SIW, Orlando, USA, 10S-SIW-027.
36.	Jeroen Voogd, Arjan Lemmers, Arno Gerretsen, Manfred Roza, Rombout Karelse: *“Effective Use of Simulation Means in Collective Mission Simulation”*. I/ITSEC 2009, paper 9198
37.	W3C: *“OWL 2 Web Ontology Language”*. W3C Recommendation, http://www.w3.org/TR/owl2-overview/, 27 October 2009.
38.	W3C: *“XML Schema”.* W3C Recommendation, http://www.w3.org/XML/Schema, 28 October 2004.
39.	Wenguang Wang, Andreas Tolk, Weiping Wang: *“The Levels of Conceptual Interoperability Model: Applying Systems Engineering Principles to M&S”*. SCS Spring Simulation Multiconference, San Diego, USA, 23-27 March 2009.
40.	Tom van den Berg, Robert Lutz: “ Simulation environment architecture development using the DoDAF”. SISO 2015 Fall SIW, Orlando, USA, Paper 15F-SIW-019.
41.	Robert H. Kewley, Marc Wood: “Federated Simulation for System of Systems Engineering”. In: Andreas Tolk, Engineering Principles of Combat Modeling and Distributed Simulation, 2012, pages 765-810. 10.1002/9781118180310.ch30.

3. Acronyms and Abbreviations

English words are used in accordance with their definitions in the latest edition of Webster's New Collegiate Dictionary except when special SISO Product-related technical terms are required.

Acronym or Abbreviation	Meaning
BOM	Base Object Model
CAS	Close Air Support
CAX	Computer-Assisted Exercise
C-BML	Coalition Battle Management Language
CMMI	Capability Maturity Model Integration
COPD	Comprehensive Operations Planning Directive
CRO	Crisis Response Operations
DIF	Data Interchange Format
DSEEP	Distributed Simulation Engineering and Execution Process
EU	European Union
FAC	Forward Air Controller
FOM	Federation Object Model
HLA	High Level Architecture
IEEE	Institute of Electrical and Electronics Engineers
IFF	Identification Friend or Foe
ISO	International Organization for Standardization
JEMM	Joint Exercise Management Module
JLVC	Joint Live Virtual Constructive
JTDS	Joint Training Data Services
LCIM	Levels of Conceptual Interoperability Model
ManPAD	Man Portable Air Defense
MEL	Main Events List
MIL	Main Incidents List
M&S	Modelling and Simulation
MSDL	Military Scenario Definition Language
MSEL	Master Scenario Events List
MSG	Modelling and Simulation Group
NAF	NATO Architecture Framework
NATO	North Atlantic Treaty Organization
NOV	NATO Operational View
PMESII	Political, Military, Economic, Social, Information, Infrastructure
OBS	Order of Battle Service

OMT	Object Model Template
SISO	Simulation Interoperability Standards Organization
SME	Subject Matter Expert
TOEL	Time-Ordered Event List
UML	Unified Modelling Language
V&V	Verification and Validation

4. Scenarios in Distributed Simulation Environments

4.1. Definition “Scenario”

The term “scenario” has many definitions. An exhaustive overview of definitions for the term “scenario” is given in [18] and [30]. The definition used by this guideline in context with the development and execution of simulation environments is the following:

> “A scenario is a description of the hypothetical or real area, environment, means, objectives, and events during a specified time frame related to events of interest.” [18, p. 2-3]

This definition was proposed by the NATO Modelling and Simulation Group MSG-053 “Rapid Scenario Generation for Simulation Applications” and a similar definition can be found in the NATO Code of Best Practice for C2 Assessment [15].

As the term “scenario” may have a different meaning in another context (e.g., CAX), it is stressed that this guideline is primarily concerned with scenarios in the context of (distributed) simulation environments.

Unfortunately, as experience shows again and again, the term “scenario” is used in an ambiguous way, even if there is agreement about the above definition. Two common uses are:

1. Use of the term “scenario” in context with the first steps of the development and execution process of simulation environments (“Define simulation environment objectives” (step 1 of the DSEEP) and “Perform conceptual analysis” (step 2 of the DSEEP), in particular with step 2, activity 2.1 (“Develop Scenario”) of the DSEEP):

   > “The purpose of this activity is to develop a functional specification of the scenario. Depending on the needs of the simulation environment, the scenario may actually include multiple scenarios, each consisting of one or more temporally ordered sets of events and behaviors (i.e., vignettes).” [8, p. 13]

2. Use of the term “scenario” in context with later steps of the development and execution process of simulation environments (“Develop simulation environment” (step 4 of the DSEEP) and “Plan, integrate and test simulation environment” (step 5 of the DSEEP), in particular with step 4, activity 4.2 (“Establish simulation environment agreements”) of the DSEEP):

   > “Once all authoritative data sources that will be used in support of the simulation environment have been identified, the actual data stores are used to transition the functional description of the scenario (developed in Step 2; see Figure 4) to an executable scenario instance (or set of instances).” [8, p. 26]

To clarify the scenario terminology, the following sections describe the scenario development process and the three different types of scenarios that are developed during this process.

4.2. Scenario Development Process

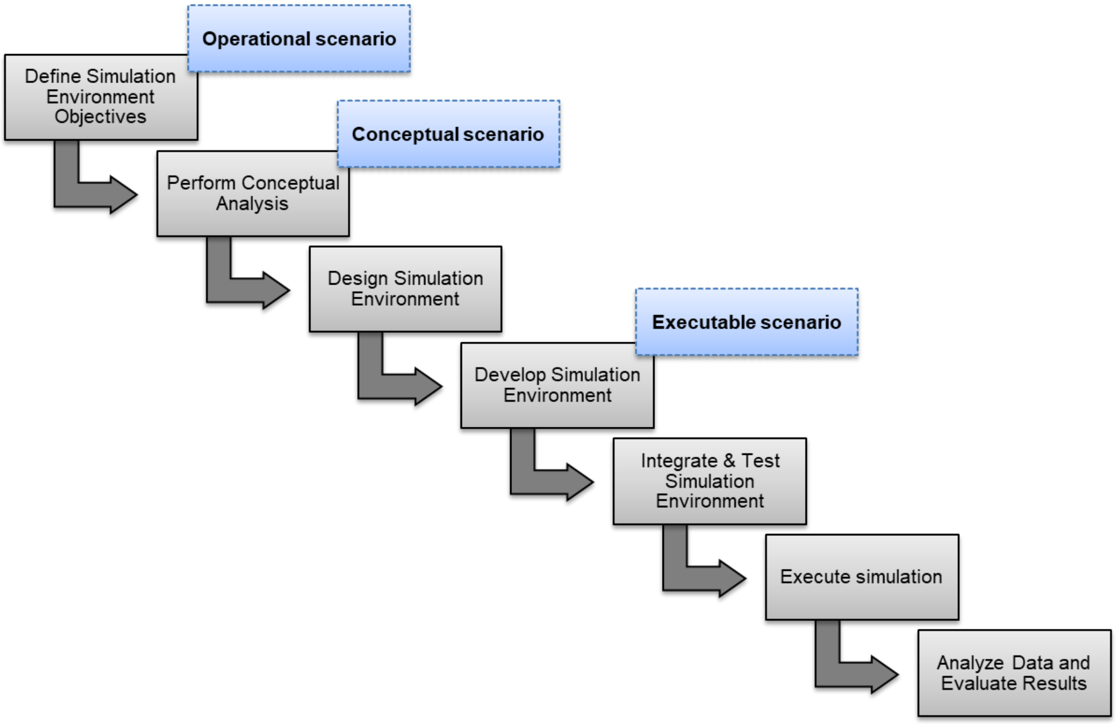
Operational scenario
Define Simulation Environment Objectives
Conceptual scenario
Perform Conceptual Analysis
Design Simulation Environment
Executable scenario
Develop Simulation Environment
Integrate & Test Simulation Environment
Execute simulation
Analyze Data and Evaluate Results

Figure 1: Types of scenarios in the simulation environment engineering process.

During the planning and design process of a simulation environment, the scenario description is continuously refined and augmented with additional information. Figure 1 illustrates the seven steps for developing and executing a simulation environment as defined by the DSEEP. In order to establish a clear understanding of scenarios and their role within the DSEEP steps, three types of scenarios are distinguished:

- Operational scenarios
- Conceptual scenarios
- Executable scenarios

While Figure 1 just associates the different types of scenarios with the most relevant DSEEP step, Table 1 gives an overview of DSEEP activities in which the three types of scenarios are created and which persons are (mainly) involved. Detailed descriptions on each type of scenario are given in the following subsections.

Table 1: Types of scenarios and their relation to DSEEP activities.

Type of scenario	DSEEP activities	Involved persons
Operational scenario	Activity 1.1 “Identify User/Sponsor Needs”, and Activity 1.2 “Develop Objectives”	User/sponsor, subject matter experts (SMEs)
Conceptual scenario	Activity 2.1 “Develop scenario”	M&S experts (assisted by user and SMEs)
Executable scenario	Activity 4.2 “Establish simulation environment agreements”	M&S experts, system operators

Within this guideline the terms “user” and “sponsor” are used in the sense of the DSEEP:

> “The person, agency, or organization who determines the need and scope of a distributed simulation exercise or event, and establishes the funding and other required resources. The user/sponsor also approves the participants, objectives, requirements, and specifications. The user/sponsor appoints the simulation environment manager and verification, validation, and accreditation/acceptance (VV&A) agents.” [8, p. 4]

4.2.1 Operational scenario

Operational scenarios are authoritative descriptions by subject matter experts (SMEs) - using their domain-specific terminology - of those parts of the real world that need to be represented in the simulation environment. Operational scenarios are required to specify what has to be represented in a simulation environment to fulfil the user's needs and objectives. Authoritative means that operational scenarios are binding requirements for the development of a specific simulation environment.

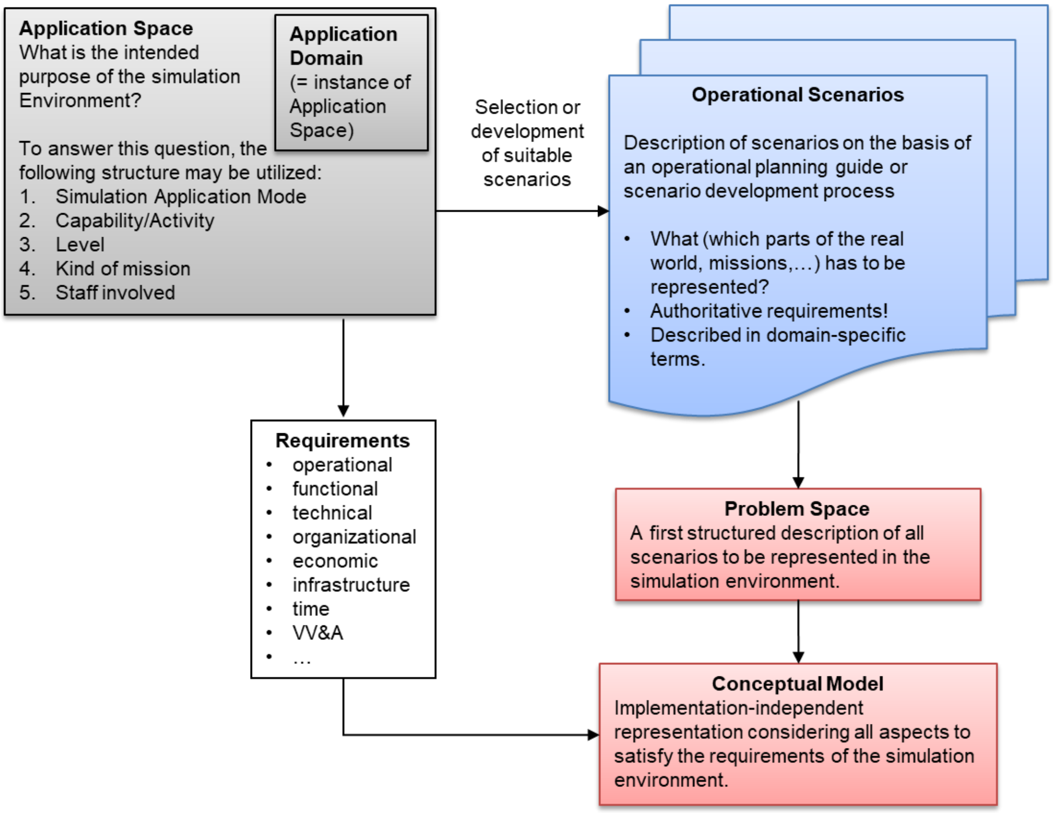
Application Space
What is the intended purpose of the simulation Environment?
To answer this question, the following structure may be utilized:
1. Simulation Application Mode
2. Capability/Activity
3. Level
4. Kind of mission
5. Staff involved
Application Domain
(= instance of Application Space)
Selection or development of suitable scenarios
Operational Scenarios
Description of scenarios on the basis of an operational planning guide or scenario development process
• What (which parts of the real world, missions,…) has to be represented?
• Authoritative requirements!
• Described in domain-specific terms.
Requirements
• operational
• functional
• technical
• organizational
• economic
• infrastructure
• time
• VV&A
• …
Problem Space
A first structured description of all scenarios to be represented in the simulation environment.
Conceptual Model
Implementation-independent representation considering all aspects to satisfy the requirements of the simulation environment.

Figure 2: Role of operational scenarios in the early steps of a simulation environment engineering process.

Operational scenarios are generally provided by the user or sponsor (probably assisted by subject matter experts) at the very beginning of the planning process of a simulation environment (DSEEP step 1 “Define simulation environment objectives”). Operational scenarios are of high importance as they are authoritative sources of requirements for the intended simulation environment. Jointly with further requirements (e.g., regarding cost, schedule or validity) they provide the frame for the M&S experts who are designing and engineering the simulation environment.

As illustrated in Figure 2, the application space (i.e., the totality of possible simulation applications) may be structured using the following 5-dimensional structure (see [17, chapter 3]):

1. Simulation Application Mode (e.g., education, training, exercise, research and development, experimentation, execution support)
2. Capability/Activity (e.g., strategic analysis and planning, operational analysis and planning, conducting operations)
3. Level (e.g., technical (systems), tactical, operational, strategic, political)
4. Kind of mission (e.g., Article 5 (missions related to war), Non-Article 5 (missions related to operations other than war))
5. Staff involved (e.g., CJ1, CJ2 (staff taking care of enemy forces), CJ3 (staff taking care of own forces), CJ4 (logistics staff), ..., CJ9)

A single instance of the application space is denoted as “application domain”. An example for an instance of the application space (thus for an application domain) could be:

- “For Training

- in defense analysis/planning
- on the operational level
- in context with a peace support operation
- for the CJ4 (Logistics)."

Thus, specifying the application domain forces the user to answer the question "For which purpose is a simulation environment supposed to be used?". Based on the specification of the application domain, suitable operational scenarios have to be selected (if available) or developed. In practice, the user often specifies an operational scenario before defining all objectives and questions. In this case, it is important that the application domain is aligned with the given operational scenarios. The alignment process may be iterative, i.e., based on a given operational scenario the application domain and the questions to be answered are specified; these questions may then be used to refine the operational scenarios.

In many cases operational scenarios do not have to be developed from scratch. Rather, the operational scenarios are the collection of existing authoritative descriptions of real world operations or missions of interest that are pertinent to some defined application domain.

As operational scenarios are authoritative sources of requirements for the subsequent development of the simulation environment they provide

- A description of a real or fictitious piece of the world of interest, including the initial state and desired end state,
- The key effects considered to be necessary for a transition from the initial state to the desired end state,
- The required tasks to accomplish these effects,
- The required capabilities to enable these tasks and
- The required entities (e.g., units) to provide these capabilities.

It has to be kept in mind that descriptions of operational scenarios differ depending on the level of the scenario (e.g., political, strategic, operational, tactical). For multi-level simulation environments a consistent hierarchy of operational scenarios is mandatory.

Operational scenarios are described in terms the operational user is familiar with and may be documented in any format. Often a combination of a graphical and a textual description is chosen. Ideally - to capture all aspects of relevance - the descriptions of such operational scenarios should follow an operational planning process or scenario development process as described in [16], [17]. For each selected or developed operational scenario it has to be described how it is related to the application domain and how it supports the original objectives and needs of this application domain.

Each scenario description could (depending on the application domain) embrace all or parts of the following information:

- Reason for the creation and use of the scenario, objectives of the scenario, relation to the application domain
- Relation to higher and/or lower level scenarios (hierarchy of scenarios)
- Historical context, major incidents leading to current situation
- PMESII (political, military, economic, social [cultural, humanitarian, legal], infrastructure, information)

- Types, numbers and behaviors of major entities of all parties (organizations, personnel, equipment, resources (facilities, logistics, and associated sustainment)) that must be represented within the simulation environment
- A description of the characteristics, capabilities, behavior, and relationships between these major entities over time
- Missions (initial state and desired end state), operations, tasks, and associated effects of these entities over time
- Sequence of actions or occurrences relevant to the objectives, e.g. in form of a Time-Ordered Event List (TOEL) where events can change the environment dynamically,, Main Event List / Main Incident List (MEL/MIL) or Master Scenario Events List (MSEL).
- Playbook, game plan (plans, orders, and associated behaviors)
- Constraints (doctrines, rules of engagement, orders, control measures)
- A specification of relevant environmental conditions (terrain (urban terrain versus natural area, type of terrain, positions for physical objects), ocean, space, atmosphere/weather, day/night, climate, etc.) that impact or are impacted by entities in the simulation environment
- Specific geographic regions (areas of operations, areas of interest and geographical locations of objects of interest)

As far as possible the above information should be based on authoritative sources and documents.

The problem space defines which (or which parts) of the operational scenarios (possibly simplified, adapted, or extended) have to be represented in the simulation environment. While the operational scenarios are defined by the user, the problem space is defined primarily by M&S experts (in collaboration and interaction with users and SMEs). The problem space, together with additional requirements derived from the objectives and needs of the intended application domain, forms the cornerstone for the derivation of the requirements for the development of the conceptual model for a simulation environment.

Thus, operational scenarios are characterized by the following aspects:

- They answer the question “What has to be represented in a simulation environment?”
- They are described by SMEs using domain-specific terminology and guidelines.
- They are human readable.
- They are related to a certain application domain (= instance of the application space).
- They are a basic pre-requisite and authoritative source for the definition of the problem space, thus for the development of the conceptual model of a simulation environment (DSEEP activity 2.1).

4.2.2 Conceptual scenario

The operational scenarios provide a coarse description of the intended situation and its dynamics, but usually do not contain enough information for deriving a conceptual model (that is a fundamental precondition for the development of a simulation environment) and designing a simulation environment. Therefore, operational scenarios need to be refined and augmented with additional information.

This is the focus of DSEEP activity 2.1 (“Develop scenario”) which will most often be carried out by M&S experts (lead) which are assisted by the user and other subject matter experts. The resulting “conceptual scenarios” provide a detailed description of the piece of the world to be simulated and should provide all necessary information for persons who are involved in later steps of the simulation environment engineering process.

Although the conceptual scenarios are primarily developed by M&S experts, a tight integration of the original user and subject matter experts is usually necessary. This helps to reduce misunderstandings and ensures that the right conceptual scenarios are derived from the given operational scenarios.

Like the operational scenarios also the conceptual scenarios are specified in terms the user is familiar with. Yet, as the development of the conceptual scenarios is led and coordinated by M&S experts this transfer of responsibility is also reflected in the scenario description. Especially, the M&S experts should aim for a more structured description of conceptual scenarios and should ensure that all information required for developing a simulation environment is provided. This may be supported by enforcing a more precise use of simulation-related terms. Also the use of specialized tools or methodologies for scenario description should be considered.

Relation of conceptual scenario and operational scenario

The totality of operational scenarios to be represented in a simulation environment determines what is also known as the “mission space”. Based on the operational scenarios (or the “mission space”), the conceptual scenarios are derived (as abstractions of the underlying operational scenarios). The conceptual scenarios provide a description of that piece of the real world which is reflected by the mission space (respectively the operational scenarios).

Relation of conceptual scenario and conceptual model

The DSEEP treats the development of conceptual scenarios in activity 2.1 “Develop Scenario”. This activity is intimately connected to activity 2.2 “Develop Conceptual Model”, and as described by the DSEEP the conceptual scenario and the conceptual model may be developed in parallel.

The conceptual model describes the entities (e.g. units, objects) and their relationships with each other. For this purpose, the conceptual model may define entity classes and their attributes (much like classes are defined in any object-oriented programming language), for example:

Entity class “Helicopter H-1”

Attributes: remainingFuel in liters, ammunition in rounds

Once all entity classes are defined, relationships may be added. These may be distinguished into static and dynamic relationships:

- Static relationships are defined between entity classes and don’t change during a simulation (e.g. a platoon is composed of four squads).
- Dynamic relationships are used to define behavior and interactions between entities (e.g. a platoon commander giving orders to his squad leaders).

Following an object-oriented view, conceptual scenarios may reference and instantiate these entity classes (see Figure 3). It is important to note that other modeling paradigms (e.g., ontology-related approaches [12, 13], architecture approaches using NAF) exist and may be more appropriate for specifying certain conceptual models. The object-oriented paradigm is chosen here only as an example.

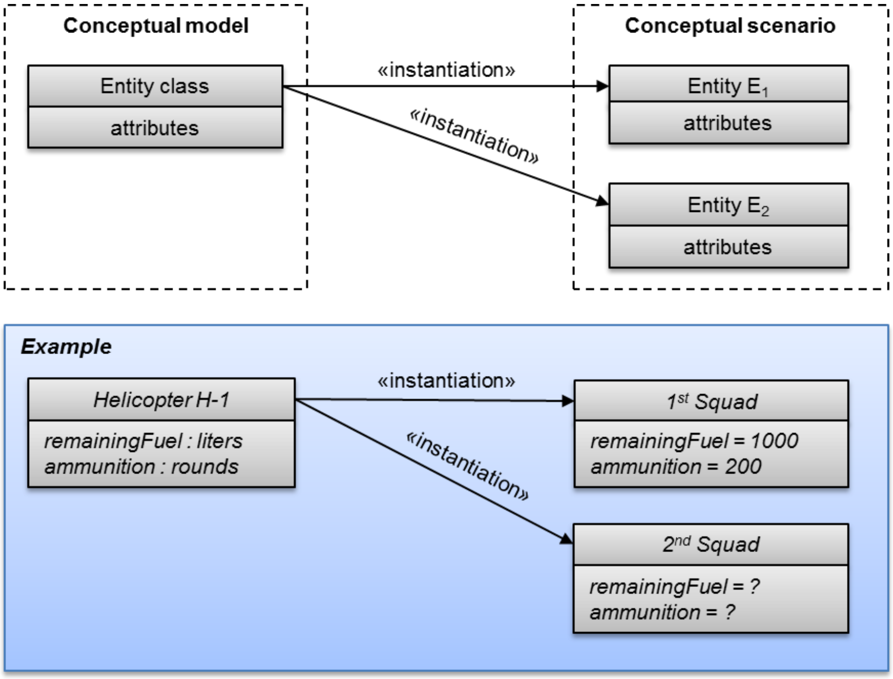

Figure 3: Relation of conceptual scenario and conceptual model regarding entity classes and entities (top), and example (bottom).

A conceptual scenario may contain multiple entities of the same entity class. For consistency and completeness it is important that the conceptual scenario and the conceptual model are aligned. This requires at least that each entity of the conceptual scenario is associated with an entity class defined in the conceptual model. Ensuring this alignment is part of the verification and validation activities accompanying the original simulation environment development process.

It is important to identify all attributes (or parameters) and to specify them unambiguously during development of conceptual scenarios (e.g., an attribute for temperature expressed in °F or °C, or wind speed expressed in km/h or m/s). In addition, where applicable, the definition of valid value ranges for attributes can help avoid misunderstanding (e.g., wind speed has to be a positive value).

As illustrated in Figure 3, a conceptual scenario may assign values to entity attributes. However, assignment of attribute values is not strictly necessary at this point in time and may be delayed until development of executable scenarios.

4.2.3 Executable scenario

Once the simulation environment is designed and set up, the conceptual scenarios have to be made available to all simulation systems and other member applications of the simulation environment. For this purpose the conceptual scenarios need to be transformed into “executable scenarios”. This is done in DSEEP activity 4.2 (“Establish Simulation Environment Agreements”).

An executable scenario is the specification of a specific scenario in a form and containing the level of detail necessary for initialization and execution of a simulation environment. The transformation from conceptual scenarios to executable scenarios is done primarily by the operator personnel of the member applications of the simulation environment (possibly assisted by M&S experts or SMEs). Ideally, the resulting executable scenarios should be specified in a way that they are directly accessible by the member applications (e.g., as a file or via a web service).

The DSEEP uses the term “executable scenario instance”. For an expert this term describes quite well what is meant. However, unfortunately this term is not further defined in the DSEEP, and even M&S experts continue to use just the term “scenario” in this context, again and again generating confusion. This guideline defines “Executable scenario” as follows:

> An executable scenario is a specification of a specific scenario containing the level of detail necessary for initialization and execution of a simulation environment.

In order to get a simulation environment “ready for execution” three basic steps have to be carried out:

1. The entities to be represented in the execution of a simulation environment have to be selected (as specified in the executable scenario that will be executed). Of course these entities have to be defined in the conceptual model of the simulation environment that was derived from given operational scenarios as described above.
2. These entities have to be allocated to individual member applications within a simulation environment (see Figure 4). This decision determines which member application executes which entities.
3. Certain initial values have to be assigned to the attributes of these entities (see Figure 5).

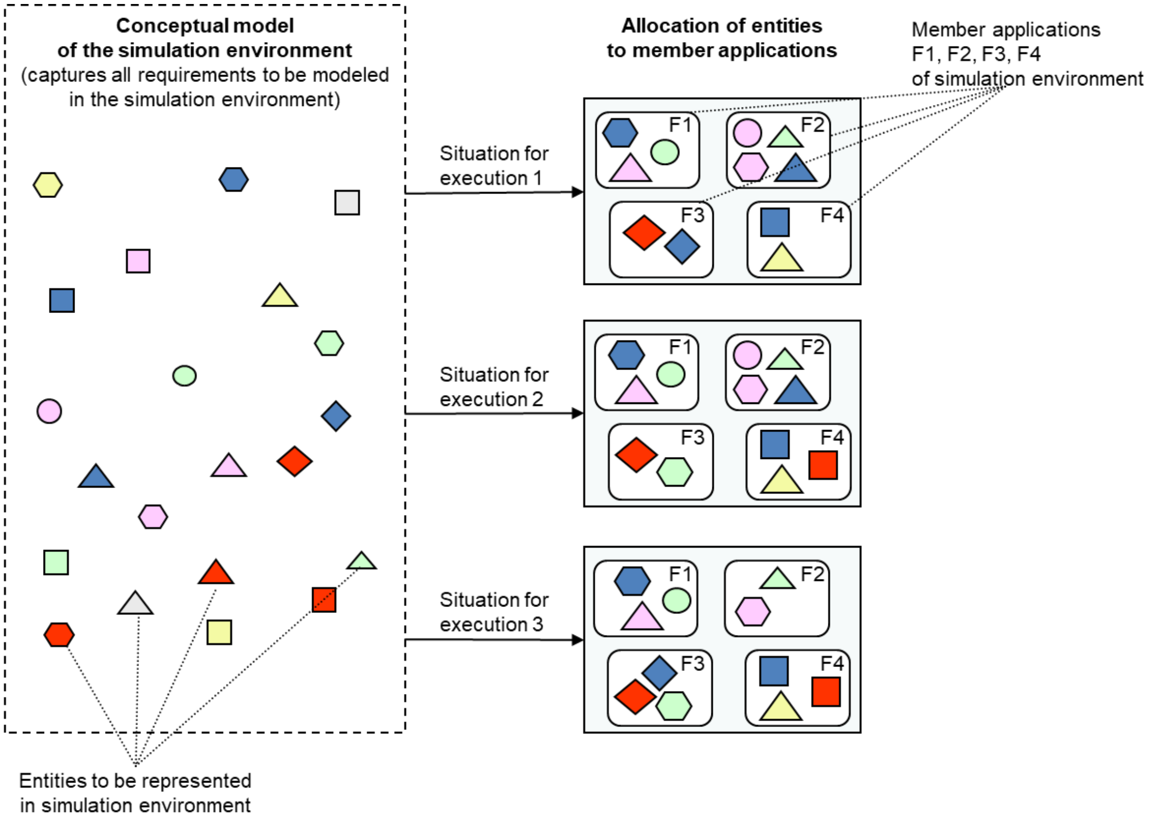
Conceptual model
of the simulation environment
(captures all requirements to be modeled
in the simulation environment)
Allocation of entities
to member applications
Member applications
F1, F2, F3, F4
of simulation environment
Situation for
execution 1
Situation for
execution 2
Situation for
execution 3
F1
F2
F3
F4
Entities to be represented
in simulation environment

Figure 4: Allocation of entities to member applications of a simulation environment for different executions of the simulation environment.

Thus, in this context the term “executable scenario” should always be used. An executable scenario is characterized by the following aspects:

- It represents a subset of the problem space and as such
  - It answers the question "What (which specific entities with which specific characteristics) has to be represented in a specific member application of a specific simulation environment execution?"
  - It contains a description of the characteristics, capabilities, behavior, and relationships between these major entities over time
- It is a basic pre-requisite for the execution of a simulation environment (DSEEP activity 4.2). Therefore, it shall be possible to derive specific configuration files and settings from an executable scenario to configure and initialize each system of a simulation environment.
- It is developed mainly by operator personnel (probably assisted by M&S experts) and described using technical terminology and guidelines.
- Ideally it should be machine-readable and reuseable (e.g., using a specific XML schema).

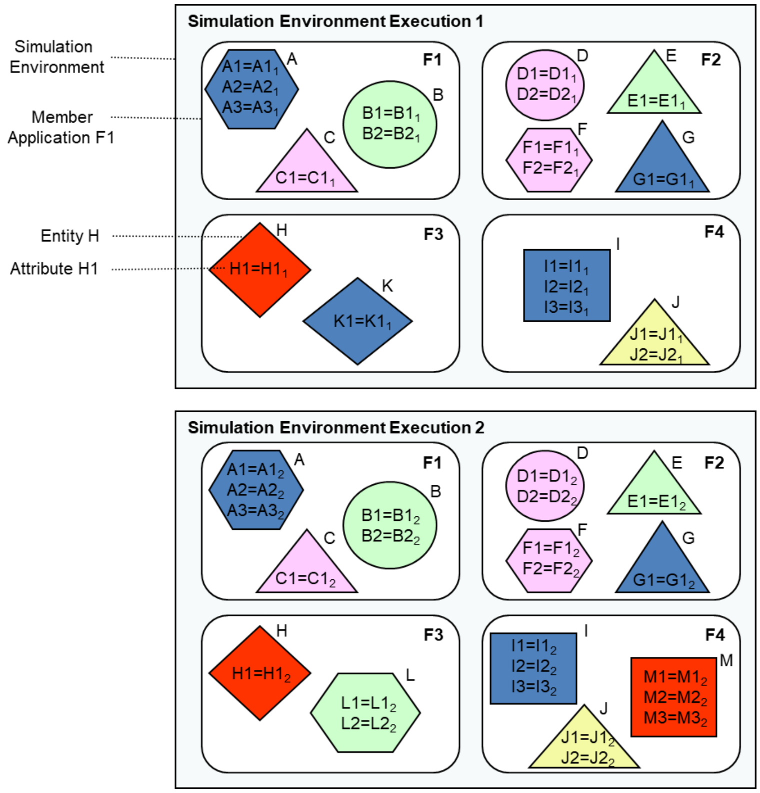

Figure 5: Allocation of values to attributes of entities to be represented by the different member applications for different simulation environment executions.

4.3. Auxiliary Terms

4.3.1 Vignettes

In order to structure scenario descriptions and to allow more flexible reuse of dedicated parts of a scenario, the term “vignette” is regularly used. Currently many definitions exist of which most definitions are to some degree circular [27, p. 89] [8, p. 13f.] [6, p. 158]. Furthermore, a practical drawback of these definitions is that they do not distinguish clearly between a scenario and a vignette. More or less, vignettes are considered to be small scenarios or to be smaller than a regular scenario. This leaves much space for interpretation and, from experience, missing clarity and understandability is a key factor for missing user adoption.

Within this guideline a vignette is defined as follows:

> A vignette is a reusable temporally ordered set of events and behaviors for a specific set of entities.

As described in the EU Core Technical Framework study [27, p. 103ff.] vignettes may be thought of as small, ideally self-contained parts of a scenario. Vignettes provide a way to reuse specific actions in multiple scenarios (e.g., a specific convoy behavior). Chapter 5.5.2 provides more information about reuse of vignettes and assembly of scenarios through using existing vignettes.

4.3.2 Scenario variants

For purposes of experimental design a scenario is often analyzed and executed in different variants. This guideline uses the following definition in this context:

> A scenario variant is a scenario which differs from another scenario by variation of isolated parameters for purposes of experimental design (e.g., different units but otherwise no changes).

Scenario variants may be specified for all three types of scenarios, although they are most common in context of executable scenarios. Table 2 illustrates an example for usage of scenario variants during the scenario development process.

Table 2: Example of scenario variants.

Conceptual scenario “Air defense”	**Executable scenarios “Air defense”**		
	Default scenario	**Variant 1: Low altitude approach**	**Variant 2: High altitude approach**

Unit “Enemy helicopter”
Attribute “height : meters”
Attribute “remainingFuel : liters”
Attribute “ammunition : rounds”
	height = **500**
	remainingFuel = 1000
	ammunition = 200
		height = **50**
		remainingFuel = 1000
		ammunition = 200
			height = **2300**
			remainingFuel = 1000
			ammunition = 200

...	...	...	...

4.3.3 MEL/MIL/MSEL

Terms commonly used in context of executable scenarios used for training and exercise purposes are “Main Events List” (MEL), “Main Incidents List” (MIL) and Master Scenario Events List (MSEL). In this terminology, events are major occurrences or a sequence of related incidents, which are actions or situations that provide greater clarity to an event [5]. The only publicly available definition is the following:

> “Master scenario events list (MSEL):
> A chronological list that supplements the exercise scenario with event synopses; expected participant responses; capabilities, tasks, and objectives to be addressed; and responsible personnel. It includes specific scenario events (or injects) that prompt players to implement the plans, policies, and procedures that require testing during the exercise, as identified in the capabilities-based planning process. It also records the methods that will be used to provide the injects (i.e., phone call, facsimile, radio call, e-mail).” [6, p. 121]

Such event lists define courses of action within a surrounding situation. Therefore, they may be used similarly as vignettes to describe dedicated parts of a scenario in a reusable way.

5. Content of a Scenario

Following the scenario definition given above the content of a scenario description consists of three parts:

- Initial state
- Course of events
- Termination conditions

As these contents may be part of all three types of scenarios (operational scenarios, conceptual scenarios and executable scenarios) these three types of scenarios are not distinguished within this section unless explicitly noted otherwise. As illustrated in Figure 6, the content is continuously refined and augmented with additional information (e.g., regarding the actual systems used for executing a scenario).

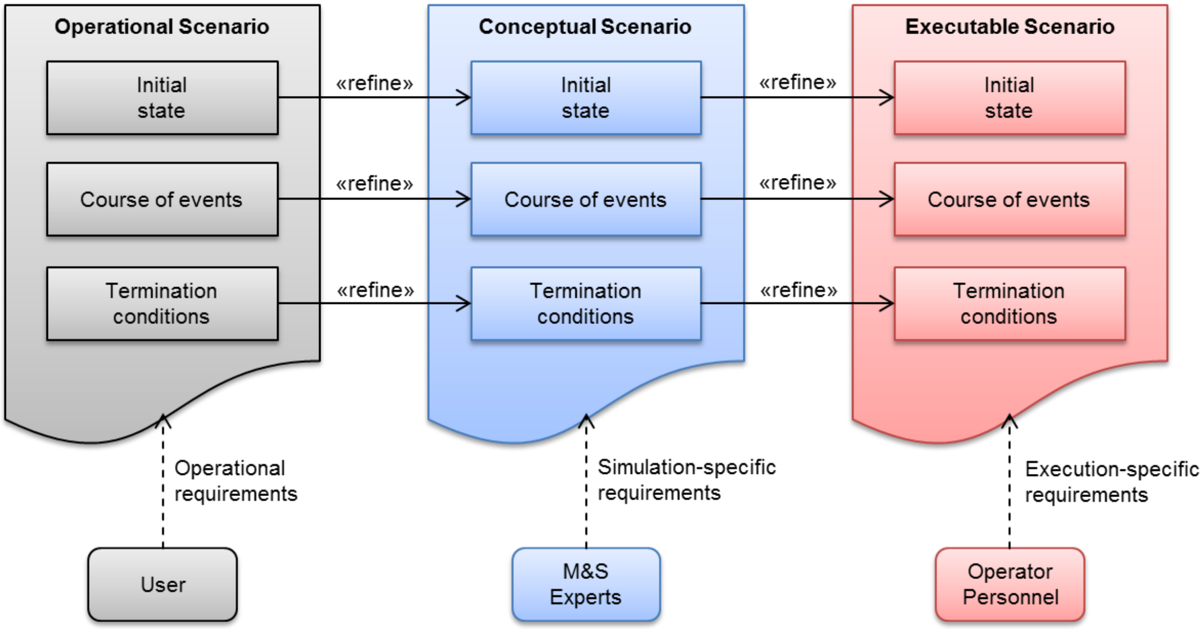

Figure 6: Refinement and extension of the content of a scenario along the scenario development process.

5.1. Preliminary remarks

Scenarios may be described on very different levels (e.g., scenarios on the political level, scenarios on the strategic level, scenarios on the tactical level). Ideally there should be a consistent hierarchy of scenarios from the political level down to the technical level. For example, a scenario on the strategic level should be derived from a political level scenario on top of it; a scenario on a tactical level should be derived from a scenario on a strategic level, and so on. Depending on the purpose of a simulation environment scenarios at a suitable level have to be selected and provided by the user. As many content items of a scenario description are the same for scenarios on all levels, no distinction is made between scenarios of different levels in the following subsections, but generic content items are presented which are required in almost all scenario descriptions.

Furthermore, as the operational scenario is defined by the user, it is usually described in a way (regarding terminology and presentation) the user is familiar with. As the user usually is not an M&S expert, operational scenarios may contain other or much more information than specified in the following sections. During the process of transforming operational scenarios into conceptual scenarios, it is a major task of the M&S

experts to structure the existing operational scenario description, to add missing information, and to strip off too detailed information. Yet, the content items for a scenario description specified in the following sections may also serve well for describing operational scenarios.

5.2. Initial State

The initial state of a scenario defines the situation at the beginning of the scenario timeline. Therefore, the initial state typically contains information regarding the following aspects:

1. Mission Statement
2. Objects and units
3. Forces and force structure
4. Geography
5. Date/Time
6. Surrounding conditions
7. Rules of Engagement

The mentioned set of information items for describing the initial state of a scenario is suitable for a wide range of (military) scenarios. Nevertheless, specific scenarios may require more or other information items. Especially the presented set of information items is suited for scenarios which are directly involving forces and units. It is, for example, less suited for describing scenarios in context of cyberwar.

All information items are described in more detail in the following subsections.

5.2.1 Mission statement

The Mission Statement should capture the high-level essentials of a scenario, e.g., background situation, intended outcome, etc. The mission statement is formulated in domain-specific terms and should communicate concisely the current situation as well as the intended end state.

5.2.2 Objects and units

List and describe all objects and units of this scenario.

The purpose of this information item is to describe which units and objects take part in the scenario. It is particularly important to describe for each unit and each object why its participation in the scenario is necessary. It is necessary here to see the issue in connection with the requirements and justify the participation of the units and objects. In turn, the question should also be answered: what would be the consequence if a specific unit or object was not part of the scenario?

Every unit and object should be described by a common pattern (see Table 3 for an example). The Military Scenario Definition Language (MSDL) [1] describes in great detail which information is necessary to define the initial state of a scenario (including objects, units, their states, attributes etc.). Therefore, MSDL may be taken as a starting point for deriving such a common documentation pattern.

Table 3: Example information set for describing objects and units.

Information	Description
Identifier	Every unit and object should have a unique identifier.

Information	Description
Type	Type of the unit or object. Recommend to use existing type categories, e.g., SISO-REF-010-2016, “Reference for Enumerations for Simulation Interoperability”.
Description	The unit or object has to be described. If possible (or necessary) suitable sources or references should be given.
Number	The numbers of units and objects present in a scenario have to be determined.
Capabilities	All capabilities of the units and objects that are relevant for the scenario have to be described. It has to be justified with a view to the objectives of the simulation environment and the target values and requirements why a specific capability is important.
Behavior	In what way(s) do the units act? Do alternative behaviors exist?
States	Which states (e.g., mobile, partly mobile, degree of damage) have to be distinguished concerning the units and objects?
Attributes	For each object and unit all attributes have to be identified that are required in context of the current scenario. If possible, valid value ranges for all attributes should be specified.

An important remark is that all relevant units and objects of a scenario have to be described. In turn, a scenario shall not contain units or objects that are not described here.

5.2.3 Forces and force structure

Definition of forces and force structures, as well as definition of relationships of objects and assignment of units to forces. Also, command and control hierarchies as well as communication networks and communication channels may need to be specified.

This information is also known as Order of Battle (ORBAT):

> “The identification, strength, command structure, and disposition of the personnel, units, and equipment of any military force.” [22]

5.2.4 Geography

The purpose of this information item is to describe the geographic requirements for a scenario:

1. Area of interest: Specification of the area of interest of this scenario (e.g., as rectangular bounding box).
2. Resolution requirements: Specification of required resolutions (e.g., required terrain resolutions (by region), required imagery resolutions)
3. Special requirements: Requirements regarding buildings (e.g., real buildings or geotypical buildings, need for building externals as well as interiors), vegetation, roads, ditches, etc.
4. Environmental conditions: Specification of all environmental conditions which are of interest within this scenario (e.g., day/night/dawn, rain/fog/snow/dust).
5. Additional information, like need for dynamic terrain, tolerances for terrain correlated for use across constructive and virtual simulations, etc.

Again, the question has to be answered: Why are certain requirements necessary? All requirements shall be derived from the objectives of the simulation environment.

5.2.5 Date/Time

The purpose of this information item is to specify date and time of the beginning of the scenario. It is important to note that this information is related to the scenario itself and not to the date and time when this scenario is executed.

As an example, the date and time of the beginning of a scenario might be specified as “10 August 2007, 0630 Zulu”. This scenario may be executed many times, e.g. 25 October 2015 at 0900 Berlin time. The relationship between scenario time and local time is illustrated in Table 4.

Table 4: Example of relationship between scenario time and local time.

Scenario time	Local time
10 August 2007, 0630 Zulu	25 October 2015 at 0900 Berlin time
10 August 2007, 0645 Zulu	25 October 2015 at 0915 Berlin time
…	…

5.2.6 Surrounding conditions

The surrounding conditions provide an overview of all scenario-related information and are described according to the PMESII approach:

1. Political situation
2. Military situation
3. Economic situation
4. Social situation / Cultural aspects
5. Information
6. Infrastructure

This information item is especially important if live actors are integrated into the simulation environment (operating either virtual or live systems). Unless the behavior of the live actors is completely determined a priori, the surrounding conditions may affect the actions of a live actor (e.g., whether a crowd is considered to be harmful or not, how to respond to specific events).

Especially if real operators are involved, it is important to stress that executable scenarios are not only specific files for initializing constructive simulations but may also be explanations, MSEL, or story boards to “initialize” humans (e.g., personnel operating virtual systems).

5.2.7 Rules of Engagement

A clear definition of rules of engagement is necessary to ensure the correct (i.e., intended) behavior of all participating personnel operating live and virtual systems as well as participating constructive systems.

It has to be noted that no matter how clear and precise the scenario developer is in describing ROE, it does not ensure the correct behavior of all participants. A clear definition of the ROE might ensure that simulation systems exhibit correct behavior (assuming they are developed and configured correctly), but nothing can ensure that virtual or live participants follow the ROE. However, a clear definition of ROE provides a measure to determine the validity of a simulation run.

5.3. Course of Events

Besides a description of the initial state a typical scenario description includes information about the behavior of opposing forces and pre-planned events happening at a specific time. Such events are usually triggering some kind of reaction, either of a participating simulation system or a trainee within the simulation environment. In general, the course of events of a scenario is defined by a set of events and corresponds to the master scenario events list (MSEL) used for training and exercise purposes.

Each event (whether triggered at a pre-planned point in time relative to the starting time of the scenario or triggered by some condition) injects a specific influence into the scenario. Typically, events are used to generate a specific situation onto which the participating member applications (live, virtual or constructive) have to react in a certain way. Intended reactions of the systems may be very different and depend heavily on the purpose of the simulation environment.

Depending on the scenario requirements (e.g., specific training objectives) different event types may be required or more applicable (e.g., PMESII, DIME, cyber events, sensory events). Therefore, the actual set of event types is always scenario-specific and has to be defined by the scenario developers.

As a starting point for scenario developers, this guideline distinguishes the following types of events which are recurring in many scenarios:

1. Communication events
2. Interaction events
3. State change events
4. Environmental events

The following subsections describe the four event types in more detail.

5.3.1 Communication events

Communication events include all types of scenario-related communication (e.g., reconnaissance reports, orders). The sender of a communication may either be a participant of the simulation environment itself (e.g., another member application) or communication may be injected by exercise control (e.g., provision of fictitious news reports to operator personnel of a virtual or live system).

5.3.2 Interaction events

Interaction events define explicit interactions regarding objects and units of the scenario (e.g., one unit attacking another one or explosion of an Improvised Explosive Device).

5.3.3 State change events

State change events are similar to interaction events but concern only a single object or unit (e.g., collapse of a bridge or breakdown of an armored vehicle due to some technical malfunction).

5.3.4 Environmental events

Sub-types of state change events are environmental events which specify state changes within the synthetic natural environment (e.g., beginning of rain, earthquake, volcano eruption).

5.4. Termination Conditions

Within simulation environments conditions should be defined for each scenario determining the achievement of a final state and thus leading to a termination of the simulation run. Although the DSEEP does not include termination conditions in its scenario definition, they are mentioned as one result of activity 2.1 ("Develop scenario") [8, p. 14].

Two typical termination conditions for a scenario can be distinguished:

1. A specific condition is achieved (e.g., destruction of all enemy units, completion of last event on MSEL, completion of required number of simulation runs).
2. A predefined time is elapsed (e.g., two hours of simulation time).

Explicitly defining termination conditions is especially important for simulation environments which are executed in a fully automated fashion. Typical examples for this are constructive simulations to conduct Monte Carlo analysis by running a scenario numerous times with varying input parameters.

5.5. Reuse in the scenario development process

As mentioned, operational scenarios are specified by the user. Therefore, the way to define and describe operational scenarios is usually dominated by domain-specific regulations (e.g., operational planning processes in the military domain). To avoid interference with these planning processes, considerations regarding reuse in the scenario development process are focused on conceptual scenarios and executable scenarios. Nevertheless, operational scenarios may also be subject to reuse and specific operational scenarios should be derived from authoritative sources (e.g., major scenarios defined in defense policies).

Focusing on conceptual scenarios and executable scenarios, two general options for exploiting reuse in the scenario development process may be distinguished [31]:

1. Reuse of a scenario as a whole
2. Reuse of specific parts of a scenario (vignettes)

The first option requires putting in place sufficient storage mechanisms as well as augmenting scenarios with specific metadata (e.g., MSC-DMS [7]) for efficient retrieval. As the scenarios are reused as a whole, this option does not place any specific requirements on the scenario development process and does not impose any constraints on the scenario description. All organizations involved in planning and developing (distributed) simulation environments should aim for at least this level of scenario reuse. Within this guideline, this option is not discussed any further.

Compared to the first option, the second option promises improved reuse possibilities and added flexibility. At the same time, option two requires decomposing a scenario into smaller parts (building blocks) which may then be assembled according to some rules to form a scenario. The remainder of this section is concerned with exactly this option and will discuss how the different parts of a scenario (initial state, course of events, termination conditions) may be split into components.

An obvious requirement for reuse of all kinds is that the applicability of existing components in a new context has to be thoroughly evaluated each time. This is also true for scenario components and always needs to be kept in mind.

5.5.1 Components regarding initial state of a scenario

Almost all parts of the initial state may be considered as components which may be reused separately in multiple scenarios. This is especially true for lists of objects and units as well as for information about forces and force structures. Similarly, the rules of engagement are a candidate for reuse as certain areas of operation are often associated with the same rules of engagement. Depending on the level of detail PMESII-related information (e.g., storybooks, country books) is also a candidate for reuse.

5.5.2 Components regarding course of events of a scenario

Although the term “vignette” is often not clearly distinguished from the term “scenario”, the basic idea behind it is highly valuable. The basic idea is to decompose a scenario into smaller components and to assemble new scenarios from existing components (cp. section 4.3.1).

For the possibility to assemble new scenarios from existing building blocks (i.e., vignettes) and to achieve maximum reuse it is necessary to specify vignettes independently of scenarios. Therefore, a vignette should not contain specific geographic locations or similar information (cp. [27]).

For example, a simple vignette “air refueling operation” might specify the rendezvous of a jet fighter and a tanker aircraft in detail without referring to actual coordinates. Instead, the vignette would describe positions relative to an imaginary starting point. Integrating this vignette into a specific scenario would only require specification of the coordinates of the starting point. This integration of a vignette into a specific scenario is done by introducing “vignette configurations” [27]:

> A vignette configuration places a vignette or set of vignettes in a context of a specific scenario.

As vignettes describe a set of activities in a generic way they may roughly be associated with classes in object-oriented programming languages. Assigning actual values by defining a vignette configuration may then be associated with creating instances of a class (usually called an “object”).

With regard to the example above, the actual values needed for instantiating the vignette are the actual coordinates of the starting point for the air refueling operation.

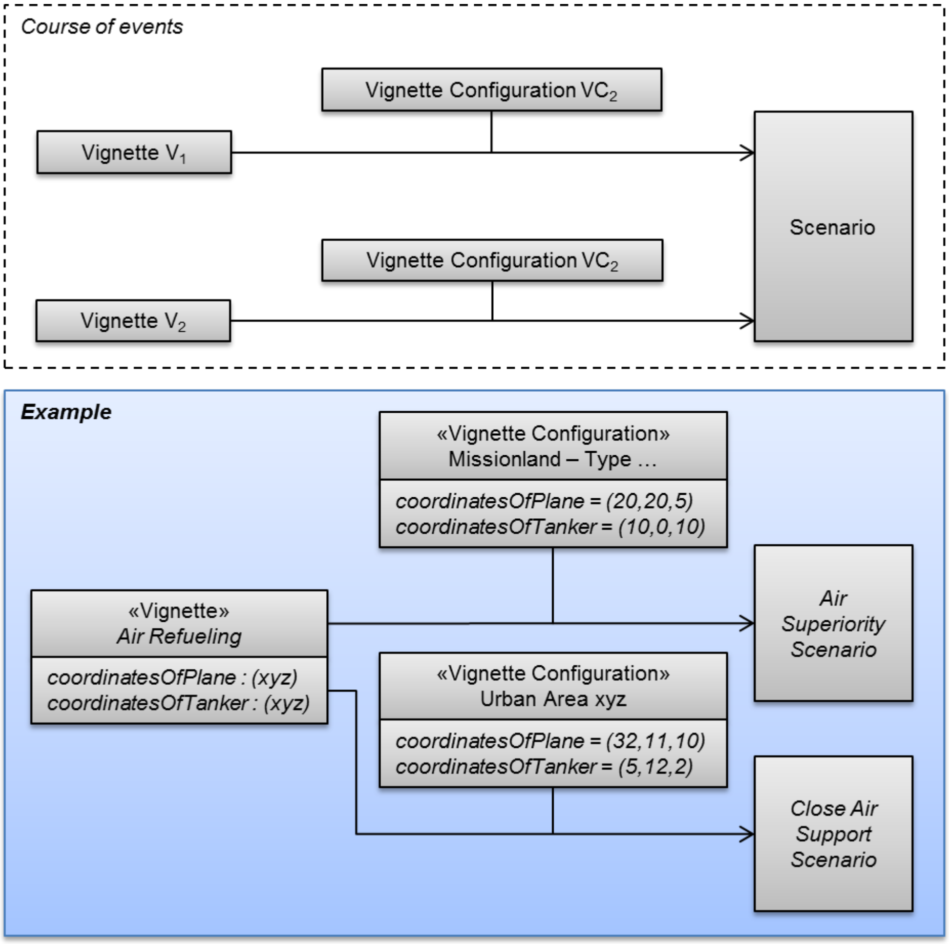
Course of events
Vignette Configuration VC2
Vignette V1
Scenario
Vignette Configuration VC2
Vignette V2
Example
«Vignette Configuration»
Missionland – Type …
coordinatesOfPlane = (20,20,5)
coordinatesOfTanker = (10,0,10)
«Vignette»
Air Refueling
coordinatesOfPlane : (xyz)
coordinatesOfTanker : (xyz)
Air Superiority Scenario
«Vignette Configuration»
Urban Area xyz
coordinatesOfPlane = (32,11,10)
coordinatesOfTanker = (5,12,2)
Close Air Support Scenario

Figure 7: Relationship between vignettes and vignette configurations for composing a scenario.

Figure 7 illustrates the relation of vignettes and vignette configurations in a scenario. Once vignettes are defined, multiple vignettes may be combined and instantiated with actual values to form a specific scenario.

It is obvious that each vignette which is integrated into a scenario imposes certain requirements on the scenario (e.g., with regard to the area, entities). Two examples for vignettes are an air refueling operation and a camp patrol task. Both impose specific requirements on the scenario:

1. Air refueling operation: This vignette requires certain specific units to be defined in a scenario (like a tanker and a jet fighter) as well as a sufficiently large area.
2. Camp patrol task: This vignette requires a specific terrain with a camp for which the patrol task will be executed.

It follows immediately that certain verification activities are required to ensure that selected vignettes and the remainder of the scenario are compatible with each other.

An important aspect is that vignettes should make use of parameters. Each parameter will be associated with an actual value during the process of integrating the vignette into a scenario. Examples:

1. Air refueling operation: Possible parameters may be the starting points of the tanker and the jet fighter to be refueled.

2. Camp patrol task: Possible parameters may be the units executing the patrol task and their initial positions.

In general, the parameters used in vignettes give scenario developers the same degree of freedom as in common programming languages when writing methods. This includes the scenario developer's decision on the degree of parameter usage for specific vignettes, i.e., vignettes might be specified in a way that "everything is parameterized" or vignettes may be defined in a way that "many to all aspects are hard-coded". The following two snippets of pseudo code show an example of a camp patrol task.

```
simple_campPatrolTask(unit u)
   // assume that unit u is at the camp's main gate
   While (scenario is running)
     Move south 500 meters
     Move east 300 meters
     …
   Endwhile
```

```
advanced_campPatrolTask(unit u, waypoint_list waypoints)
   // assume that unit u is at the camp's main gate
   While (scenario is running)
     For each waypoint in waypoints do
        Move to waypoint
      Endfor
   Endwhile
```

The respective benefits of using parameters or hard-coding certain parts are the same as in programming languages. While hard-coded vignettes are quick and easy to create, they require detailed knowledge (e.g., about the camp) and effectively prohibit reuse (e.g., because the patrol task is intimately connected to a specific camp). Therefore, the use of parameters is highly encouraged.

5.5.3 Components regarding termination conditions of a scenario

The reuse of termination conditions is not detailed any further as the potential benefit of reuse seems to be too small compared to the effort for specifying the termination conditions.

6. Maturity Levels of Scenario Description

In order to assess the quality of a scenario description, this guideline introduces maturity levels. Maturity levels are well known from CMMI (Capability Maturity Model Integration) which provides the following definition:

> “Maturity level:
> Degree of process improvement across a predefined set of process areas in which all goals in the set are attained.” [34, p. 445]

With regard to CMMI, the core idea is to use maturity levels as an indicator for process improvement, i.e., a maturity level indicates that all goals defined for this level are achieved. Similarly, maturity levels may be used to indicate the quality of a scenario description.

This guideline uses a 4-level maturity model for scenario descriptions:

- Level 0 – No written scenario description
- Level 1 – Non-standardized scenario description
- Level 2 – Standardized scenario description
- Level 3 – Formal scenario description

Table 5 provides an overview of the maturity levels of scenario descriptions and typical ways to represent scenario descriptions on each maturity level. Detailed information is provided in the following subsections.

Table 5: Maturity levels of scenario description.

Maturity level of scenario description	**Representation of the scenario description**
0 – No written scenario description	Thoughts and ideas within the mind of the military user/SME; oral explanation.
1 – Non-standardized scenario description	Free text.
2 – Standardized scenario description	Documentation which is structured according to a standard or agreed guideline or template.
3 – Formal scenario description	Formal specification of a scenario.

6.1. Level 0 – No written scenario description

Maturity level 0 refers to the situation where no written scenario description is available at all. In this case the scenario is only available within the minds of the participating persons and explained and communicated orally. Obviously, this kind of (unavailable) documentation is error-prone, arbitrarily fuzzy and effectively prevents all types of scenario reuse. Also, traceability and understandability are reduced to a minimum.

For these reasons, achieving only maturity level 0 is in almost all conceivable cases not sufficient. Very seldom exceptions may be very small and focused simulation environments which are planned and set up by a single person (or very few persons).

6.2. Level 1 – Non-standardized scenario description

Maturity level 1 indicates the existence of a scenario description at least in a non-standardized way. In this context “non-standardized” means that the scenario description is not created according to a generally accepted standard but structured according to the likings of the persons participating in planning and designing the simulation environment.

Compared to maturity level 0 the main benefit of a non-standardized scenario description is the availability of a written scenario description at all. The existence of documentation at all allows basic traceability of requirements and improves understandability of the simulation environment for directly involved persons as well as third parties.

However, a non-standardized scenario description still comes along with many problems:

1. Due to a missing standardized documentation template, the familiarization effort for involved parties (user, M&S experts, system operators, etc.) is quite high.
2. Ensuring completeness of the scenario description is complicated (due to missing checklists).
3. Comparability of scenario descriptions is low as each scenario is specified differently.
4. Reuse is hampered due to low comparability of scenario descriptions and thus limited possibilities for efficiently searching and finding suitable existing scenarios.

Maturity level 1 requires only the existence of a written scenario description but does not require that this description is structured according to any specific standard. Therefore, everybody who is describing a scenario may use a different, custom structure for the scenario documentation. The scenario documentation may follow available standards like the DSEEP which give a few hints about the necessary content of a scenario description. As the DSEEP does not provide detailed templates for scenario description, it is sufficient to achieve maturity levels 0 and 1, but is insufficient for achieving higher maturity levels of scenario description.

6.3. Level 2 – Standardized scenario description

Maturity level 2 requires the creation of a standardized scenario description and is a great step forward compared to maturity levels 0 and 1. Within this guideline “standardized” is deliberately interpreted rather broadly and considers any scenario description to be standardized if the standard is agreed upon beforehand jointly by all participants in the engineering process and if the standard is used across multiple simulation environments. Therefore, this broad use of the term “standardized” includes open standards and local standards (as defined by AMSP-01 [24]).

Open standards are defined by AMSP-01 [24] as “Specifications that are developed by a standards organization or a consortium to which membership is open, and are available to the public for developing compliant products (with or without some license fee)”. Examples for standards organizations are SISO, IEEE, and ISO. A key characteristic of open standards is their public availability that allows broad usage of such a standard.

AMSP-01 uses the term “Local standards” to denote specifications and standards that are organization-specific and usually not publicly available. Local standards reflect the fact that standards are used by different communities at different levels.

Only by using a standardized and ideally open (i.e., publicly available) standard for scenario descriptions, potential benefits may be realized. The main benefits of a standardized scenario description are:

1. As all persons involved in the scenario description process are familiar with the standard and the scenario structure defined by this standard, initial familiarization efforts are minimized.

2. Following an analog reasoning, a standardized structure allows easier access to scenario descriptions for all persons working on these scenarios. This includes the M&S experts which set up the simulation environment, the V&V experts responsible for the quality management as well as all readers of the scenario specification in general.
3. A precisely defined scenario description structure simplifies quality management activities (like evaluation of completeness of a scenario specification).
4. Similarly, a clearly defined scenario description structure (ideally accompanied by explanatory texts) simplifies the development of a "good" scenario specification. This availability of such a scenario description structure is especially important if scenario descriptions have to be created by persons with few or zero experience in this area.
5. A precisely defined scenario description structure is the necessary prerequisite for the formal specification of a scenario.

As scenarios are decisive sources of requirements for the intended simulation environment, the provision of at least a standardized (and complete) scenario description is required for achieving interoperability of simulation systems on the pragmatic, dynamic and conceptual level as defined by the Levels of Conceptual Interoperability Model [39].

6.4. Level 3 – Formal scenario description

Maturity level 3 requires the formal description of a scenario. Several possibilities exist for defining such a formal description. One of the most prominent choices, especially in the software engineering domain, is XML Schema [38]. Other options include RELAX NG [25], UML [26] and OWL [37].

The benefits of a rigorous formal scenario description are manifold:

1. Ideally ambiguities are eliminated or at least reduced to a minimum.
2. Secondly and maybe even more important, automated processing of scenario descriptions and extensive utilization of tools are made possible. This opens up a wide range of possible applications:
    a. Automated consistency checks (e.g., no units have overlapping positions; no units are positioned outside the specified geographic boundaries).
    b. Automated initialization of all simulation systems and other member applications (e.g., computer-generated forces) which are part of the simulation environment. Benefits are a reduced risk of errors due to manual misconfiguration or human errors as well as an improved reproducibility.
    c. Semi-automated or fully automated cross-checks with other information products developed during the engineering process (e.g., whether the Federation Object Model (FOM) provides interaction classes for all communication events defined in the scenario).
3. Reuse is simplified. Scenario descriptions may be archived and retrieved for later reuse.
4. Transfer and sharing of scenario descriptions is easily possible.

To realize the benefits, it is strongly recommended to use open standards that are specifically designed to capture scenario information, like MSDL, C-BML and C2-Sim.

6.5. Relation of maturity levels to scenario types

While in principle each maturity level may be assigned to any of the three types of scenarios, in practice only certain combinations are possible or likely. An initial evaluation done by MSG-086 [MSG-086] is shown in Table 6, indicating

- Which maturity levels are generally applicable to which type of scenario (non-empty cells),
- Which maturity levels are currently achieved (C) or at least partially achieved (/C/), and
- Which maturity levels should be targeted for describing the different types of scenarios (T).

Table 6: Relation of maturity levels to the different types of scenarios. [MSG-086]

Type of scenario **Maturity level of scenario description**	**Operational scenario**	**Conceptual scenario**	**Executable scenario**
0 - No written scenario description	C	C	
1 – Non-standardized scenario description	C	C	C
2 - Standardized scenario description	/C/	/C/	/C/
3 - Formal scenario description	T	T	/C/ T

Regarding operational scenarios, non-standardized scenario descriptions often prevail. Although most users follow some kind of schema for describing operational scenarios, standards or templates are generally specific to a certain organization and not used across multiple organizations. For describing operational scenarios the mid-term target should be to achieve maturity level 2 (e.g., using SISO standards such as MSDL and C-BML). A necessary prerequisite for achieving this is the development and adoption of a common standard for describing operational scenarios.

Descriptions of conceptual scenarios are regularly achieving maturity levels 1 and partially also level 2. The short-term goal should be to consistently achieve maturity level 2 across a large number of simulation environment engineering process applications. The mid-term to long-term target should be maturity level 3, i.e., formally described conceptual scenarios. A critical factor for achieving this target will be the availability of specialized tools which allow a formal description of conceptual scenarios by M&S experts on the one hand, and which provide a customized presentation and output of the conceptual scenario for discussion with the user and sponsor on the other hand.

Currently, executable scenarios are most often described on maturity levels 1 and 2. Approaches for formally describing scenarios are available (e.g., MSDL), but are not as regularly used as they could be. A reason for this may be that currently available standards are missing important features or are not providing enough benefits compared to currently used “legacy” standards. The target must be to achieve maturity level 3 for the description of executable scenarios in order to realize the potential benefits.

7. Standards and Tools for Scenario Description

Table 7 lists available standards and tools for scenario description. This compilation is not exhaustive but tries to give an overview. This document does not recommend any specific tool or product. The assignment of standards and tools to the different types of scenarios indicates for which type of scenario a standard or tool is most likely to be used. This assignment does not exclude the use of a standard or tool for a different type of scenario. More detailed explanations on the standards and tools are given in the following subsections and may also be found in AMSP-01 [24].

Table 7: Standards and tools for scenario description.

Maturity level of scenario description	**Type of scenario**					
	Operational scenario		**Conceptual scenario**		**Executable scenario**	
	Standards	**Tools**	**Standards**	**Tools**	**Standards**	**Tools**
0 - No written scenario description						
1 - Non-standardized scenario description	"Everybody uses his own standard." DSEEP	General purpose software	Unified Modelling Language (UML) Systems Modelling Language (SysML)	General purpose software Wiki engines UML editors	Individual documentation (e.g., MEL/MIL, story books)	General purpose software
2 - Standardized scenario description	NATO Comprehensive Operations Planning Directive (formerly: "NATO Guidelines for Operational Planning")	General purpose software	NATO Architecture Framework (NAF) VEVA documentation guidelines (in Germany)	General purpose software Wiki engines UML editors	Proprietary (vendor-specific) data exchange formats	
3 - Formal scenario description	C-BML C2SIM-Tasking Reporting JC3IEDM ADatP-3		Base Object Models (BOM)	JEMM	MSDL C2SIM-Initialize JTDS Order of Battle Service C-BML C2SIM-Tasking Reporting	XML editors

Maturity level of scenario description	Type of scenario					
	Operational scenario		Conceptual scenario		Executable scenario	
	Standards	Tools	Standards	Tools	Standards	Tools
					C2IEDM/JC3 IEDM	

A number of off-the-shelf tools are available to assist the scenario development process.

7.1. Standards and tools for operational scenarios

7.1.1 DSEEP

Type of standard: Open standard, publicly available from SISO and IEEE.

The Distributed Simulation Engineering and Execution Process (DSEEP) [8] describes a generic 7-step process for developing and executing a simulation environment. Each step of the DSEEP is further subdivided into more specific activities, and for each activity inputs and outputs are specified by the DSEEP. As the DSEEP is considered as a generalized, high-level framework which has to be adapted to individual needs, the DSEEP does not provide very detailed guidance or even documentation templates for activity outcomes. It is assumed that each organization that uses the DSEEP provides this detailed guidance as part of the individual adaptation.

Nevertheless, the DSEEP provides basic guidance on the content of a scenario description [8, p. 13ff.].

7.1.2 NATO Comprehensive Operations Planning Directive

Type of standard: Open standard, publicly available.

The NATO Comprehensive Operations Planning Directive (COPD) [16] (formerly: “NATO Guidelines for Operational Planning”) and its national counterparts define operational planning procedures. Although these planning procedures are not dedicated solely to scenario development they are agreed guidelines and are thus standards.

7.1.3 General purpose software

Type of tools: Open tools, publicly available.

Any general-purpose software may be used to document scenarios. Typically used software includes:

- Word processors (e.g., MS Word, OpenOffice Writer)
- Presentation programs (e.g., MS PowerPoint, OpenOffice Impress)
- Graphics editors (e.g., Adobe Photoshop, gimp)

7.1.4 Joint C3 Information Exchange Data Model (JC3IEDM)

Type of standard: Open standard, publicly available from Multilateral Interoperability Programme (MIP)

The Joint Consultation, Command, and Control Information Exchange Data Model (JC3IEDM) [14] is a data exchange model that aims to improve interoperability between systems that exchange Command and

Control (C2) information. The JC3IEDM development is managed by the Multilateral Interoperability Programme (MIP).

Following [14] the scope of this data model is directed at producing a corporate view of the data that reflects the multinational military information exchange requirements for multiple echelons in joint/combined wartime and crisis response operations (CRO). The data model is focused on information that supports:

- Situational awareness
- Operational planning
- Execution
- Reporting

The scope includes data from various functional areas according to the requirements levied by MIP and NATO.

As the JC3IEDM is used to describe operational situations it is obvious that it may also be used to describe operational scenarios.

7.1.5 Allied Data Publication 3 (ADatP-3)

Type of standard: NATO STANAG

As explained by [23] the standard provides the rules, constructions and vocabulary for standardized character-oriented message text formats that can be used in both manual and computer-assisted operational environments. These message formats are to be used for all formatted character-oriented messages within the NATO Command, Control and Information System (NCCIS) unless specifically excluded by multinational agreement. It is concerned solely with the part of a message that contains the thought or idea the originator wishes to communicate. The transmission of formatted messages remains in accordance with the instructions given in relevant Allied Communications Publications.

As ADatP-3 is used to describe operational situations (like JC3IEDM) it is obvious that it may also be used to describe operational scenarios.

7.1.6 Coalition-Battle Management Language (C-BML)

Type of standard: Open standard, publicly available from SISO.

The Coalition-Battle Management Language (C-BML) developed by SISO provides a publicly available standard which may be used for specifying orders and messages within a scenario. It “is a standard language for expressing and exchanging plans, orders, requests, and reports across command and control (C2) systems, live, virtual and constructive (LVC) modeling and simulation (M&S) systems, and autonomous systems participating in Coalition operations.” [5]

The maintenance and further development of C-BML (and MSDL) is done by the SISO C2SIM Product Development Group (PDG). See Ch. 7.1.7 for more details.

7.1.6 Command and Control-Simulation Interoperation (C2SIM)

Type of standard: Open standard, publicly available from SISO (once development is finished).

The Command and Control-Simulation Interoperation (C2SIM) family of products is currently developed by SISO. It will consist of the following products:

- C2SIM Logical Data Model (C2SIM-LDM)
  - Type: Standard
  - Relationship: The logical data model that provides the logical definitions of initialization, tasking, and reporting business elements and associations referenced in the syntactic representation standards below.
- C2SIM Initialization XML Representation (C2SIM-Initialize)
  - Type: Standard
  - Relationship: The XML syntax representation for C2SIM initialization messages. Depends on C2SIM-TaskingReporting for task and report message elements.
- C2SIM Tasking and Reporting XML Representation (C2SIM-TaskingReporting)
  - Type: Standard
  - Relationship: The XML syntax representation for C2SIM tasking and reporting messages.
- Guideline for C2SIM-Initialize Implementation
  - Type: Guideline
  - Relationship: Guideline document for implementing the C2SIM-Initialize standard.
- Guideline for C2SIM-Tasking Reporting Implementation
  - Type: Guideline
  - Relationship: Guideline document for implementing the C2SIM-TaskingReporting standard.

The C2Sim family of products will replace MSDL (to become C2SIM-Initialize) and C-BML (to become C2SIM-TaskingReporting).

7.2. Standards and tools for conceptual scenarios

7.2.1 NATO Architecture Framework (NAF)

Type of standard: Open standard, publicly available.

The NATO Architecture Framework (NAF) [21] defines several views which may be used for documenting a simulation environment engineering process (as described in [27] and [35], similar for the Department of Defense Architecture Framework (DoDAF) in [40, 41]). The NAF defines only the content of the various views but not how these views should be filled (i.e., which methodologies or modelling languages should be used). Furthermore, the NAF is originally intended for developing and describing systems and not specifically for documenting scenarios. Although NAF does not directly address scenario documentation it may be used for this purpose (see Appendix B for an example).

7.2.2 Unified Modeling Language (UML)

Type of standard: Open standard, publicly available from OMG and ISO/IEC.

Also, the Unified Modeling Language (UML) [26] defines several diagram types which may be used to document specific parts of a scenario. Although UML is a generic modelling language that is not specifically designed for documenting scenarios, it can be used for describing scenarios. For example, class diagrams may be used to define unit hierarchies, sequence diagrams may be used to define behavior, and deployment diagrams may be used to document which units are simulated by which simulation system.

7.2.3 Systems Modelling Language (SysML)

Type of standard: Open standard, publicly available from OMG.

The UML language comes from the software engineering world. The system engineering world has derived the Systems Modelling Language (SysML) from UML. Large parts of UML and SysML overlap, but SysML adds two additional diagram types: requirements diagrams and parametric diagrams. The first can be used for requirements analysis and the latter for engineering analysis of critical system parameters. Given that a distributed simulation environment is more than software only, SysML could be beneficial for the development of a distributed simulation environment. Specifically for describing scenarios there is probably not too much difference between using UML or SysML.

7.2.4 Base Object Models (BOMs)

Type of standard: Open standard, publicly available from SISO.

A standard which directly addresses defining and reusing components of models, simulations and federations is the Base Object Model (BOM) Template Specification [2]. As described therein, “BOMs serve to provide an end-state of a simulation conceptual model and can be used as a foundation for the design of executable software code and integration of interoperable simulations.” Regarding scenario documentation:

“The aspects of a simulation conceptual model found in a BOM contain static descriptions of items resident in the real world described in terms of conceptual entities and conceptual events. In addition, those aspects of a simulation conceptual model found in a BOM contain information on how such items relate or interact with each other in the real world in terms of patterns of interplay and state machines.” [2, p. 7]

A detailed analysis regarding usability and practicability of using BOMs for scenario documentation is presented in [32]. Also, Appendixes C and D provide examples for the usage of BOMs for scenario description.

7.2.5 VEVA Process Model

Type of standard: Local standard, used by German Armed Forces, partially publicly available.

Representatives of a different approach are the VEVA documentation guidelines. The VEVA process model is a German approach for operationalizing the DSEEP and provides detailed documentation guidelines covering the whole simulation environment engineering process [28] [29] [33]. Especially, the VEVA documentation guidelines provide specialized documentation templates for scenarios. The documentation templates defined by VEVA are most applicable for documentation of operational scenarios and conceptual scenarios. Obviously, the VEVA is not an official, public standard, but as the VEVA is used regularly within the German Armed Forces it is considered as a standard (in the sense of the definition given in Chapter 6.3).

7.2.6 Joint Exercise Management Module (JEMM)

Type of tool: Local tool, available from NCIA.

JEMM (NC3A Joint Exercise Management Module) [5] is a so-called “Exercise and Scenario Management Tool”.

Exercise and Scenario Management tools can be used for the automation of processes, information management and information exchange throughout an exercise process. They can help in preparation and management of scenarios as well as the Main Events List, Master Incidents List (MEL/MIL) or Master Scenario Events List (MSEL). An exercise management tool like JEMM can also be very useful in synchronizing and managing the flow of an exercise according to the exercise objectives, as well as, planning, collecting and analyzing the observations. MSG-068 used JEMM for this purpose [19].

7.3. Standards and tools for executable scenarios

7.3.1 Military Scenario Definition Language (MSDL)

Type of standard: Open standard, publicly available from SISO.

The most widely known and publicly available standard for specifying executable scenarios is the Military Scenario Definition Language (MSDL) [1] which is developed and maintained by SISO. MSDL allows especially the description of the initial state of a scenario. Due to its original purpose, MSDL is focusing on land-based scenarios and may not be applicable in other application domains (e.g., cyber warfare scenarios).

The maintenance and further development of MSDL (and C-BML) is done by the SISO C2SIM Product Development Group (PDG). See Ch. 7.1.7 for more details.

MSDL is supported by various tools, see Table 7.

7.3.2 Order of Battle Service (OBS)

Type of standard: Local standard used by JTDS.

The Order of Battle Service (OBS) which is part of the US Joint Training Data Services (JTDS) provides pre-populated Order of Battle data sets for use within the Joint Live, Virtual, Constructive (JLVC) Federation [4]. The JTDS OBS XML Schema defines the XML interchange format used to exchange initialization data between JTDS and the JLVC federation. Similar to MSDL, the OBS XML Schema allows representation of force sides, units, facilities, relationships, etc. Due to various reasons, the OBS XML Schema was developed in parallel with MSDL [1].

Appendix A Bibliography (Informative)

The following publications are considered as valuable background information and reports from using this guideline.

#	Title
1.	Lionel Khimeche, Raphael Filipetto, Jean-Paul Mochet, Gildas Premel-Cabic, Achraf Ghorbel, “A disruptive approach for Scenario Generation: an agile reuse bridging the gap between Operational and Executable Scenario”, 2016 Fall SIW, September 2016, Orlando, USA, Paper 2016-SIW-001.
2.	Jose Ruiz, Lionel Khimeche, Philippe Zamoun, Pierre Lagrange, Kevin Seavey, Emilie A. Reitz, Herve Biran, “BOLD QUEST 2015: Lessons Learned from a France-U.S. Close Air Support Training Event Supported by Simulation”, 2016 Fall SIW, September 2016, Orlando, USA, Paper 2016-SIW-005.

Appendix B Scenario Description using NATO Architecture Framework (Informative)

This appendix describes how the development of a scenario may be supported and documented using the NATO Architecture Framework (NAF). Additional information how architecture frameworks may be used in the context of developing and document simulation environments are presented in [40, 41].

B.1 Rationale

The following citation from [21] roughly describes what is commonly understood by the term “architecture”:

> *An architecture is the fundamental organization of a system embodied in its components, their relationships to each other and to the environment and the principles guiding its design and evolution. [...] An architecture is little more than a plan put together in accordance with certain rules.*

When creating an architecture, the architect normally uses a specific type of formal description for his work. Documenting the architecture in a formalized way ensures that every expert capable of understanding the formal notion can also read and understand the architecture, its elements, relations and ideas expressed therein. This fosters coherent design and reuse.

Depending on the specific domain many formal notions of architectures have evolved, e.g. floor plans in the construction domain or UML diagrams in the software domain. The formalized notion for an architecture within a single application domain may be called “architecture framework”. It typically comprises some kind of symbology, elements of the domain, their abstract roles and relationships and the integration of these parts into different views (plans) according to certain templates.

In this sense the NATO Architecture Framework (NAF) provides many different views and related templates to describe system architectures to be used in the NATO military domain for different purposes like for example

- Capabilities Integration, Development and Portfolio
- Planning, Programming and Execution
- Acquisition
- Systems Engineering
- Operations Planning.

Several of these activities will also have to be carried out on the design and execution of distributed simulation. According to [8] for example the following tasks will occur

- Develop Objectives, Initial Planning (Capabilities, Portfolio)
- Develop Scenario (Operations Planning)
- Develop and Integrate Simulation Environment (Systems Engineering).

Due to the similarity of the real world military domain and the design of simulation environments and simulations for the military domain it is very likely, that at least some parts of the NAF will also be helpful to support the design and execution of distributed simulation. Extensive work has been carried out in [35] to analyze and demonstrate the usability of architecture frameworks in this context (using the Ministry of Defence Architecture Framework MoDAF instead of NAF). Following these ideas and convinced of the general usability of NAF in the context of distributed simulation in the military domain in the following sections concentrates on the question on how the development of a scenario may be supported using the NAF.

It should be pointed out, that this use of architectures and the corresponding frameworks must not be confused with simulation architectures like the High Level Architecture (HLA). Although the latter describes federates, object models and federation rules and therefore forms some kind of architecture framework, it concentrates on the construction of distributed simulation systems only and lacks the templates necessary to describe scenarios.

B.2 Architectures in the scenario development process

Chapter 4.2 explains the differences between operational, conceptual and executable scenarios within the scenario development process.

Operational scenarios have to be provided by the user and form the starting point of the scenario development process. They are described in terms the user is familiar with and often are a combination of graphical and textual descriptions of involved entities, actions, tasks and events. The user normally will not be aware, that by creating these graphical and textual descriptions he may actually be creating a NAF NOV-1 High Level Operational Concept Description. Also his description may contain information on doctrine, tactics, and procedures, concepts of operations, environmental conditions, and technical standards to be used and so on. From the NAF point of view such information is captured in templates from the NAV All View, NOV Operational View, NSV System View and NTV Technical View sections. However, the user normally is not familiar with the NAF and will have no support by an architecture expert during the creation of the operational scenario. Although the use of NAF templates even in this early phase of scenario development would be feasible and desirable, normally operational scenarios are being developed without any relations to the NAF.

Conceptual scenarios refine the operational scenarios. They are created by M&S experts by analyzing the operational scenario and extracting the conceptual information necessary to construct a simulation environment matching the operational scenario. It is obvious, that this work will benefit from the application of NAF because the main goal of NAF is the support of the analysis and design of complex systems and finally creating an architecture of the system to be constructed. This will be detailed in the next section. However, it should be noted here, that a M&S expert is not necessarily a NAF-aware system architect. To gain optimal results a system architect familiar with the NAF should be involved in the development of the conceptual scenario, at least as long as the M&S experts are not familiar with the NAF themselves.

Executable scenarios finally contain all information necessary for preparation, initialization and execution of the simulation environment. They contain the most detailed description of the scenario and are derived from the corresponding conceptual scenarios by M&S experts and system operators, preferably in machine readable form. The use of NAF templates to describe these detailed scenarios probably won’t be feasible, because the information to be captured in most cases will be too specific to be described by NAF templates. In addition there is no standardized machine-readable format for NAF.

B.3 NAF templates for conceptual scenarios

The analysis in the previous section shows, that the derivation of conceptual scenarios from operational scenarios will benefit most from the use of NAF.

The NAF does not specify a specific sequence in which the templates should be populated, nor will all templates featured by the NAF have to be created for the description of a particular architecture. Instead the decision on the templates to be created and the sequence of template creation has to be made on a case by case basis to gain optimal results. For a description of conceptual scenarios using templates of the NAF a good starting point would be the creation of the NAF operational view, because the operational scenario is already present and will provide a good starting point. Therefore the following sequence of steps is proposed as a guideline.

B.3.1 NOV Operational View

The creation of the operational view mainly consists of a capturing and structuring of information already present in the operational scenario in an unstructured form. Following and augmenting the recommendations from [35], the NOV templates from the NAF may be used in this way:

Table 8: Usability of NAF Operational View templates for the description of scenarios.

NAF Template	Usage
NOV-1 High-Level Operational Concept	Describe the context, major entities and actions involved in the operation. May already be available in the operational scenario description. Additional constraints or preferences e.g. on environmental conditions may be stated here.
NOV-2 Operational Node Connectivity	Describe operational nodes ★ and their roles and collaboration patterns from an operational point of view. Describe activities to be performed by nodes.
NOV-3 Operational Information Requirements	Describe operational information to be exchanged among the nodes.
NOV-4 Organizational relationships	Describe organizational relationships if relevant for the operation.
NOV-5 Operational Activities	Describe activities that are relevant for the operational scenario.
NOV-6 Operational Activities Sequence & Timing	Describe the sequencing of activities and events.

★ According to NAF, operational nodes are logical collections of operational activities. Operational nodes produce or consume information and may represent an operational realization of capabilities.

It is explicitly noted here that this proposed structuring of operational information in NAF templates has nothing to do with simulation. It is the same task as designing e.g. a defense architecture using real weapon systems on the basis of a real world operational scenario. The transformation to a simulation environment occurs at a later stage by replacing the real systems by simulation models with the appropriate capabilities.

Given the operational scenario from Appendix D for example the NOV-5 activity tree may look as depicted in Figure 8.

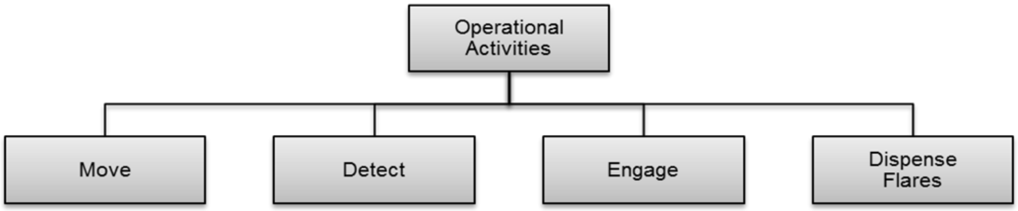

Figure 8: NOV-5 Operational Activity Tree for scenario example from Appendix D.

For the example operational scenario (see Appendix D) two operational nodes may be identified, the AH-1 helicopter and the infantry troop, and assign operational activities to the nodes. To express the opponent role of the helicopter different colors for the operational nodes are used. No operational information is exchanges between the nodes, so (in NAF terminology) a "need line" between the nodes is missing. Additionally the nodes may be attributed with the activities performed at the nodes. The NOV-2 operational node connectivity diagram then might look as shown in Figure 9.

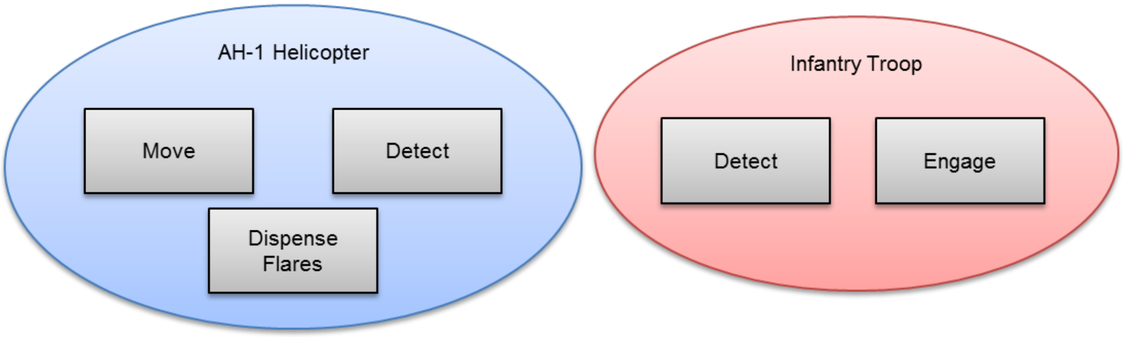

Figure 9: NOV-2 Operational Nodes for scenario example from Appendix D.

B.3.2 NSV System View

In a next step some templates of the NAF System View may be deduced from the operational view. In this phase special M&S knowledge will be necessary, because the systems will not be real world weapon systems but will be simulation systems or individual simulation models and therefore their specific requirements will have to be taken into account.

According to [35] the NSV templates documented in Table 9 are useful.

Table 9: Usability of NAF System View templates for the description of scenarios.

NAF Template	**Usage**
NSV-1 System Interfaces	Describe the interfaces (simulated operational interfaces and simulation specific interfaces) of the systems.
NSV-2 System Communication	Describe system or model communication, message exchanges and (simulation) communication quality requirements.
NSV-5 System Functions to Operational Activity Tracing	Describe system functions (e.g., model capabilities) necessary to perform the required operational activities.

NAF Template	**Usage**
NSV-4 System Functionality	Describe systems or models to be present and their functionality to be represented.
NSV-10 Rules, Sequence & Timing	Describe system rules, state transitions and event traces.
NSV-11 System Data Model	Describe data to be represented in the simulation environment, may be focused on exchanged data (FOM, in HLA terminology).

Continuing the example from above, the list of system functions (first column) and their mapping to operational activities (first row) as NSV-5 matrix may appear as presented in Table 5 below.

Table 5: NSV-5 Operational Activities to System Function Traceability Matrix for scenario example from Appendix D.

Operational activities (->) **System functions**	move	detect	engage	dispense flares
detect with radar		X		
detect visually		X		
detect with IR			X	
flight of helicopter	X			
flight of missile	X			
flight of flare	X			X
emit IR		X		X
reflect radar		X		

Each of the operational activities thereby requires one or several system functions to be implemented in the system. These system functions therefore may be identified with individual functions or capabilities of simulation models. Similar to the assignment of operational activities to operational nodes in the operational view, these system functions may be assigned to individual systems. From the assignment of system functions to system nodes and from the necessary data flows between the system functions then follows the NSV-2 System Communication template.

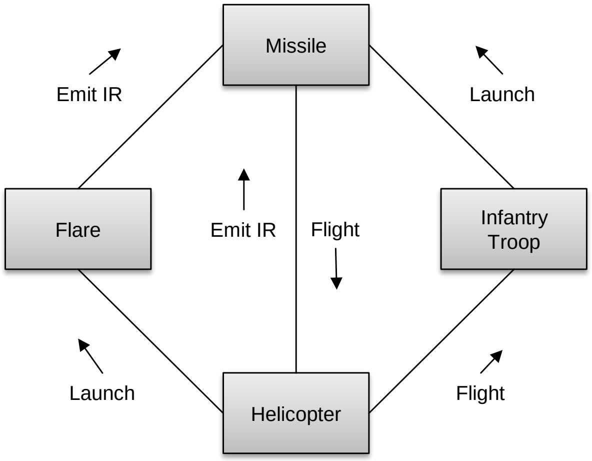

Figure 10: NSV-2 System Communication template for scenario example from Appendix D.

Here the green boxes denote the individual simulation models, which implement the system functions listed in the table. Each of the models may contain sub-models not shown in this figure, e.g. the missile model may be composed of a movement model (implementing the “flight missile” system function) and an IR seeker model (implementing the “IR detection” system function). In contrast to the NOV-2 operational node connectivity diagram there is intensive data exchange between the individual systems/models. This is depicted by the connection lines in the figure above, the transmitted information and direction of data flow indicated by the arrows and their labels.

For example the infantry troop model needs information on position and movement of the helicopter to implement the visual detection system function. On detection it will launch the missile (model). The helicopter model will need the position information from the missile model to implement radar detection of the missile and to launch the flare. The missile will need position information from the helicopter and the flare to implement target detection, guidance and distortion of detection/guidance by IR emissions from the flare.

It is obvious, that these information exchanges form a starting point for the definition of the data (exchange) model NSV-11 (the FOM in HLA terminology) or a description of event sequence and timing NSV-10.

B.3.3 Other views and templates

In addition to the templates listed above several other templates from the NAF may be used to further detail the conceptual scenario description. Examples are

- NTV-1 Technical Standards Profile, to describe the (simulation and operational) technical standards to be followed
- NTV-3 Standard Configuration, to describe standard system configurations that are known to work
- NAV-2 Integrated Dictionary

Appendix C Scenario Description using Base Object Models (Informative)

C.1 Overview of the BOM standard

As stated in [2], “Base Object Models (BOMs) provide a component framework for facilitating interoperability, reuse, and composability”. For this purpose the “Base Object Model (BOM) Template Specification” defines the format and syntax for describing the elements of a template for representing BOMs.

A Base Object Model is composed of four major components:

- Model Identification (Metadata)
- Conceptual Model Definition
- Object Model Definition
- Model Mapping

All four components are described in more detail in the following subsections.

C.1.1 Model Identification

The “Model Identification” component of a BOM associates important metadata (i.e., information about the base object model itself) with the model. This information is necessary for purposes of search and retrieval of BOMs as well as for maintaining traceability and facilitating reuse of BOMs.

C.1.2 Conceptual Model Definition

The “Conceptual Model Definition” component of a BOM is used for specifying the conceptual model represented by this BOM. For defining the conceptual model the BOM Template Specification defines four template components as subcomponents of the “Conceptual Model Definition”:

- Entity Type
- Pattern of Interplay
- State Machine
- Event Type

The entity types define the types of conceptual entities required for representing all aspects of the conceptual model.

The pattern of interplay defines a set of patterns of interplay (including pattern actions, variations, and exceptions) which are required to represent the activities and interactions within the conceptual model.

The state machine identifies the conceptual entities and their respective states as well as the state transitions which are required within this conceptual model.

The event types define types of conceptual events (directed, undirected) which are required for representing the pattern action of a pattern of interplay.

C.1.3 Object Model Definition

"The Object Model Definition defines the structure of object and interaction classes, and their associated attributed and parameters" [2, p. 48]. In its current version the BOM Template Specification refers to the HLA OMT for defining object and interaction classes (e.g., using the classes "HLA Object Class" and "HLA Interaction Class" as respective root classes for objects and interactions).

C.1.4 Model Mapping

The "Model Mapping" component of a BOM template provides the link between the entities and events of the Conceptual Model Definition and the classes specified by the Object Model Definition.

C.1.5 Documentation of BOMs

Each of the four main components of a BOM may be documented and represented in many ways (e.g., textual, tabular, graphical). Additionally, the BOM Template Specification describes the BOM Data Interchange Format (DIF). The BOM DIF is specified as XML Schema and provides the possibility to represent a BOM as XML.

Basically, the BOM Template Specification uses table-based approach for specifying data. However, almost all tables may be transformed into respective UML diagrams (e.g., sequence diagrams for patterns of interplay, state diagrams for state machines). Figure 11 shows an example for a table-based specification and a graphical representation.

State Machine Name	**Conceptual Entity**	**State**			**Note**
		State Name	**Exit Condition**		
			Exit Action	**Next State**	
ShooterStates	**FiringEntity**	Ready	CommandToFire	Fire	na
		Fire	WeaponFire	MunitionFlight	na
		MunitionFlight	MunitionDetonation	Ready	na
Note	**na**				

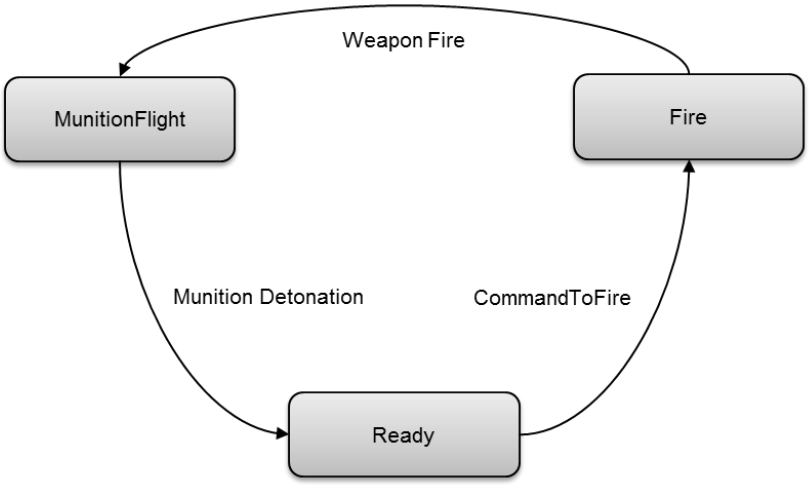

Figure 11: Example specification of state machine as a table (top) and as a diagram (bottom) [2].

Appendix D Example Scenario “Air Defense” (Informative)

The example scenario “Air Defense” is concerned with the study of the operational effectiveness of man portable air defense (ManPAD) systems for defending a maneuvering unit against airborne attacks.

A more detailed description of this scenario including an evaluation of the use of Base Object Models (BOMs) for scenario description is available in [32].

The first section provides an example of a fictitious operational scenario. Based on this operational scenario, the second section presents a textual description of a conceptual scenario and the third section describes how the BOM Template is used for specifying the conceptual scenario (in contrast to the textual description of the conceptual scenario outlined in the previous section).

All example descriptions follow the scenario documentation template outlined in Chapter 5.

D.1 Description of the operational scenario

Initial State
A ManPAD team consisting of a commander, an observer and a gunner, is supporting a maneuvering unit. The ManPAD team is deployed around 400 meters behind the maneuvering unit on high ground. At the instant when the ManPAD commander receives an early warning with the assumed target location, the maneuvering unit is heading north and the ManPAD team is behind the unit. The ManPAD observer starts searching the sector from which the aircraft is approaching.

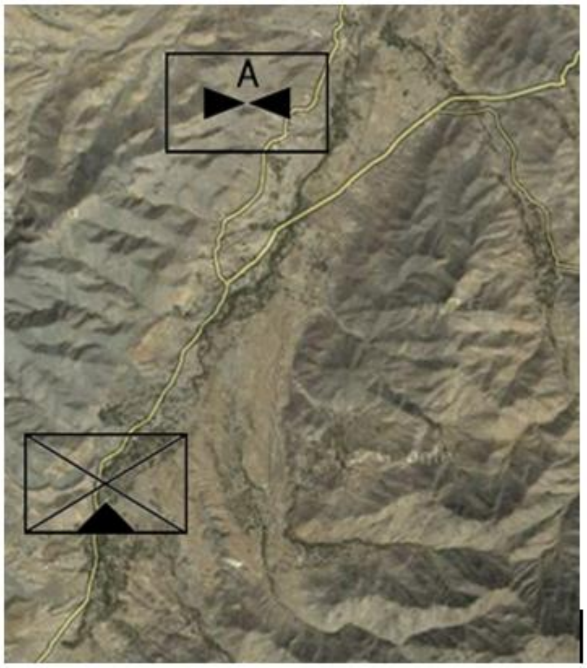

Figure 12: Area of interest.

Course of Events
The ManPAD observer catches a glimpse of a blade flash from rotating helicopter blades approaching from North. Since the ManPAD team is in Weapons Free status, the ManPAD gunner starts an interrogation procedure. As soon as the target is in range ring, he triggers an IFF (identification friend or foe) operation. As the target is identified as hostile, the ManPAD Commander orders a Fire Command.

At the instant of fire, the enemy helicopter is at 500 meters altitude and has a speed of 45 meters per second with straight flight. The ManPAD gunner launches the missile from 80% of range ring, and the missile approaches the target from the front. As soon as the helicopter detects the engagement, it throws a dozen flares to protect against the missile when it is within the last kilometer.

The ManPAD observer then evaluates the first missile and reports the result to the MANPAD commander for consecutive action.
Termination Conditions
Not explicitly specified in this example.

D.2 Textual description of the conceptual scenario

D.2.1 Initial state

The initial state specifies the situation at the beginning of the scenario time line which includes participating units, forces and the force structure, geography, surrounding conditions and the rules of engagement. Below is the initial state of the example Air Defense scenario.

Units

Table 6: Units and their initial states

Unit	Attribute	Value
Maneuvering unit	Initial position	400, 0, 0 in Local NED
ManPAD Team	Initial Position	0, 0, 0 (Local NED origin)
	Sub Units	ManPAD Commander, ManPAD Observer, ManPAD Gunner
	Equipment and Weapons	IFF and ManPAD-X
	Status	Weapons Free
Target	Type	AH-1 similar helicopter
	Altitude	500 meters
	Speed	45 m\s to South (-45, 0, 0)
	Maneuver	Straight flight
	Position	5500, 0, -500 in Local NED
	Engagement Ring	2500m

Forces and Force Structure

ManPAD Team

- Composed of: ManPAD commander, ManPAD observer, ManPAD gunner.
- Command structure: ManPAD observer and ManPAD gunner are under the command of ManPAD commander.
- Spatial position: All three persons are located in the same area next to each other.
- Command and Control requirements: Ability to receive voice messages and commands over radio.

Geography

Table 7: Geography

Attribute	Value
Area	Hypothetical area
Terrain	Flat earth
Atmosphere	ICAO Standard
Wind	5 m/s from East

Surrounding Conditions

Not applicable.

Rules of Engagement

ManPAD Team:

1. If any approaching object is identified and the status is Weapons Free then IFF operation will be triggered as soon as object heads into the range ring.
2. If object is identified as hostile and the object is in 80% of range ring, weapon is fired.

Helicopter:

1. Apply any means of soft kill (flares, maneuver etc.) as soon as a missile attack is detected.
2. Attack any maneuvering target within engagement ring.

D.2.2 Course of events

Besides specifying the initial conditions, the conceptual scenario shall also specify pre-planned events that are triggered at a specific time or due to a specific condition. These events may include communication, interaction, state change or environmental events. The course of events in the example air defense scenario is provided below.

Communication events

Table 8: Communication events

Time	Event
00:00	ManPAD commander receives a voice message with early warning information for a target at a specific position.

Interaction events

Table 9: Interaction events

Event type	Attribute(s)	Trigger/condition
Target Identification	Target Position	Within 5km of ManPAD team
IFF Operation	Target Position	In the range ring of ManPAD missile
Missile Fire	IFF Status	Foe
	Target Position	In 80% of range ring
Missile Detection	Missile Position	Within 1.5km of helicopter
Flare Dispense	Missile Slant Range	1400 m
	Dispense Number	12
	Initial Dispense Time	0.6 s
	Dispense Interval Time	0.1 s

D.2.3 Termination conditions

Each conceptual scenario shall specify termination conditions which define the end of a scenario. Typical termination conditions can be that a predefined time has elapsed or that a specific condition is achieved. Below is the termination condition of the example air defense scenario.

Termination condition: The missile either hits the target or misses and self-destructs.

D.3 Using BOMs for scenario description

D.3.1 Model Identification

The “Model Identification” component is a very important component which should always be specified in “real world” scenario development efforts. Although it might be argued that it provides little additional value in the context of this example, the “Model Identification” component of the BOM Template Specification is deliberately not omitted.

Table 10: BOM Model Identification

Category	**Information**
Name	Air defense of a maneuvering unit using ManPADs
Type	BOM
Version	1.0
Modification Date	2012-12-10
Security Classification	Unclassified
Release Restriction	Publicly releasable
Purpose	Analysis of the operational effectiveness of man portable air defense (ManPAD) systems for defending a maneuvering unit against airborne attacks.
Application Domain	Analysis
Description	This BOM specifies interactions between infantry using ManPADs against helicopters.
Use Limitation	This BOM is not applicable for airborne attacks carried out by jet fighters or bomber units.
Use History	
Keyword Taxonomy Keyword Value	
POC POC Type POC Name POC Organization POC Telephone POC Email	 Primary author Halit Oguztüzün … … …
POC POC Type POC Name POC Organization POC Telephone POC Email	 Technical POC Umut Durak, Aylin Hatip … … …
Reference Type Identification	None.
Other	None.

Category	Information
Glyph Type Alt Height Width	No glyph provided.

D.3.2 Conceptual Model Definition

As described in Chapter 4.2.2 conceptual scenarios are tightly connected to the conceptual model of a simulation environment. Strictly speaking, the definition of entities (and their properties) is part of the conceptual model and not part of a conceptual scenario. However, a conceptual scenario will usually reference parts of a conceptual model (e.g., entities).

If a separate documentation of a conceptual model is available for a simulation environment, a reference to this documentation of the conceptual model may be included in the “Reference” sub-component of the “Model Identification” component. Otherwise, the BOM Template provides the possibility to include information about entity types etc. directly into the BOM.

D.3.2.1 Entity Types

The BOM is made of several key elements; one of them is the conceptual model. Our conceptual model includes

- Commander Entity,
- Observer Entity,
- Gunner Entity,
- Hostile Object Entity,
- Firing Entity,
- Target Entity, and
- Missile Entity.

Table 11: Overview of entity types

Entity Type			
Name	**Description**	**Characteristics**	**Description**
Commander Entity _identifier	Thing that is the commander information	ID	Unique ID for entity
		Location	Physical position of the entity
Observer Entity _identifier	Thing that is the observer information	ID	Unique ID for entity

Entity Type			
Name	**Description**	**Characteristics**	**Description**
		Location	Physical position of the entity
Gunner Entity _identifier	Thing that is the gunner information	ID	Unique ID for entity
		Location	Physical position of the entity
Hostile ObjectEntity _identifier	Thing that is the hostile object information	ID	Unique ID for entity
		Location	Physical position of the entity
		Velocity	Velocity for the entity
Firing Entity _identifier	Thing that fires a weapon at a target	ID	Unique ID for entity
		Location	Physical position of the entity
Target Entity_identifer	Thing that is the target information	ID	Unique ID for entity
		Location	Physical position of the entity
		Velocity	Velocity for the entity
MissileEntity_ identifier	Thing that is the missile information	ID	Unique ID for entity
		Location	Physical position of the entity
		Velocity	Velocity for the entity

D.3.2.2 Pattern of Interplay

The major pattern of interplay within this air defense scenario is the sequence from observing the airspace until detection, identification and engagement of the target.

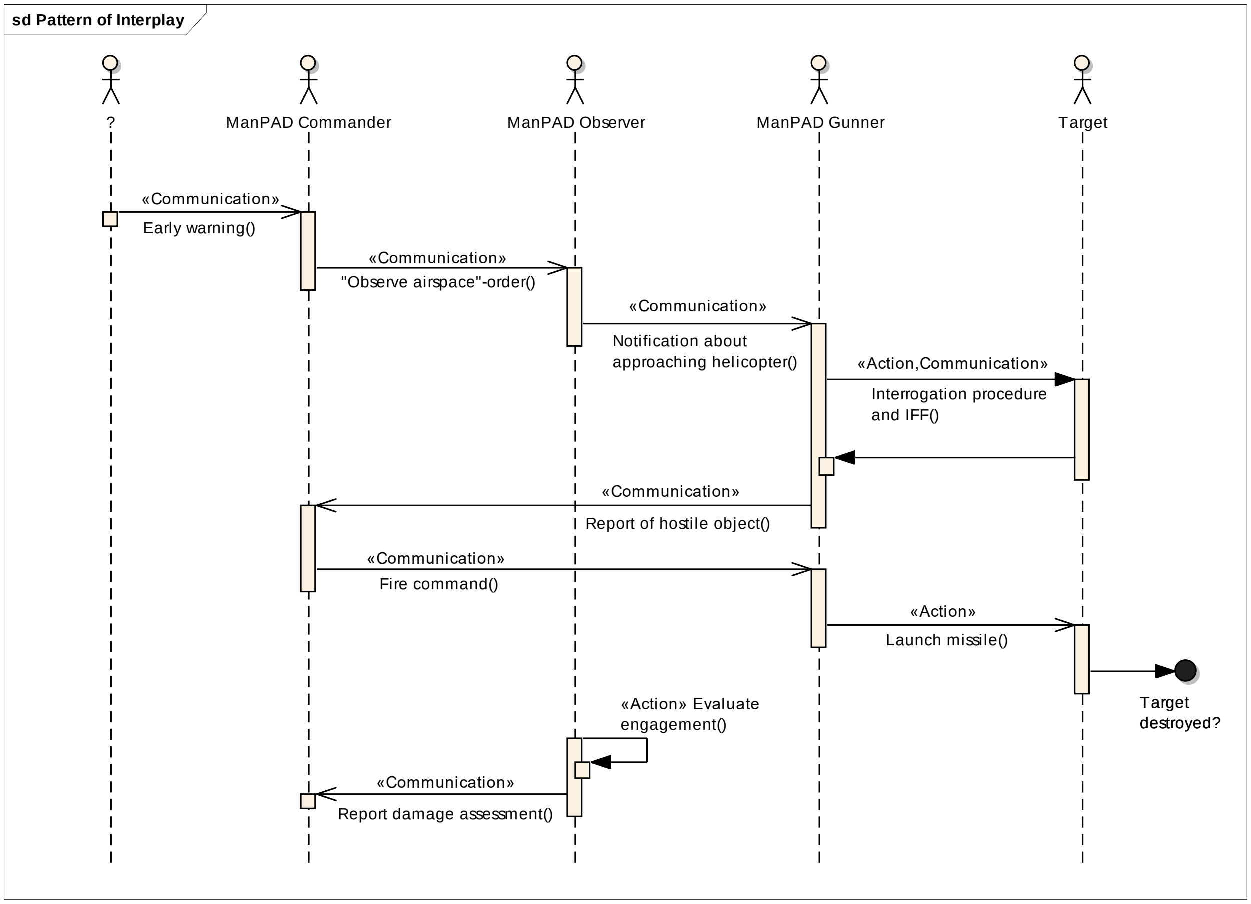

Figure 13: Overview of pattern of interplay in example scenario "Air defense".

This pattern of interplay is illustrated in Figure 13 using a UML sequence diagram. The transitions (arrows) are annotated with the communication and interaction events between actors. Two types of transitions (indicated by stereotypes in Figure 13) are distinguished:

- Communication (represents exchange of information between two parties)
- Action

Currently, the distinction is used only for clarifying the scenario developers' intention and the chosen types are not described any further. Ideally, a scenario developer may choose from a set of well-defined transition types which are standardized within an organization or across the M&S community (e.g., as SISO standard).

During development of the sequence diagram, a couple of shortfalls in the original operational scenario could be identified:

- Who sends the early warning? The missing information is indicated in Figure 13 by using "?" for the actor that sends the early warning.
- What precisely is the "Interrogation procedure and IFF"? Within the sequence diagram this transition is annotated with the stereotypes "Action" and "Communication" to indicate that it may be an interaction (e.g., sending an IFF request) or a communication (e.g., radio communication). However, a clear description of this procedure is missing.

- What precisely is the content of the “Report of hostile object”?
- The ManPAD Observer has to evaluate the engagement. How is this activity triggered?

In summary, specifying a scenario using a sequence diagram has proven very useful as it requires a scenario developer to be much more precise compared to textually specifying a scenario.

D.3.2.3 State Machines

The Pattern of Interplay gives an overview over the whole scenario, but does not provide detailed information about behavior of single entities. Specification of behavior of single entities may be done using UML state machines. In the following, each entity of the example scenario is described and its behavior is defined using state machines.

ManPAD team

The state machines of the ManPAD team are provided in Figures 14-16.

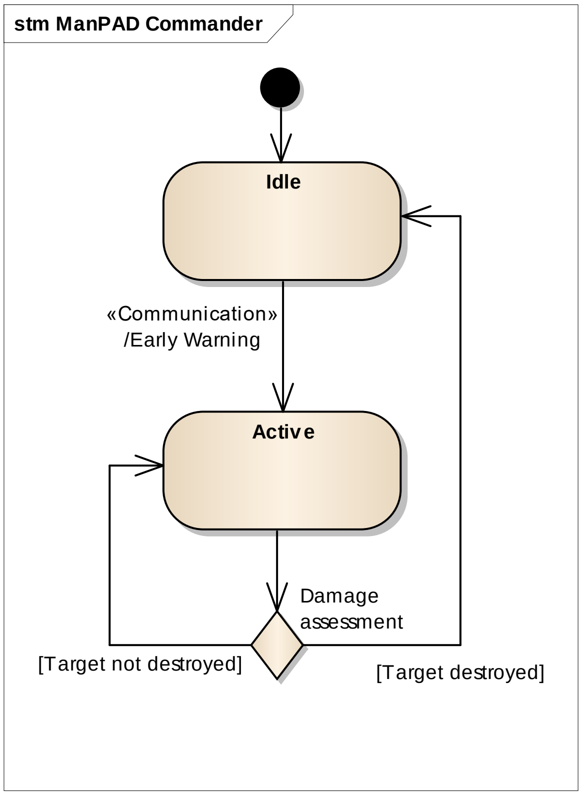

Figure 14: State machine of ManPAD Commander.

Figure 14 shows the state machine of the ManPAD Commander. As can easily be conceived from the state machine, the ManPAD Commander has a very simple behavior. Basically, he is just responding to the reception of an early warning message and enters the “Active”-state once he receives this communication. He stays in “Active”-state until the target was successfully destroyed.

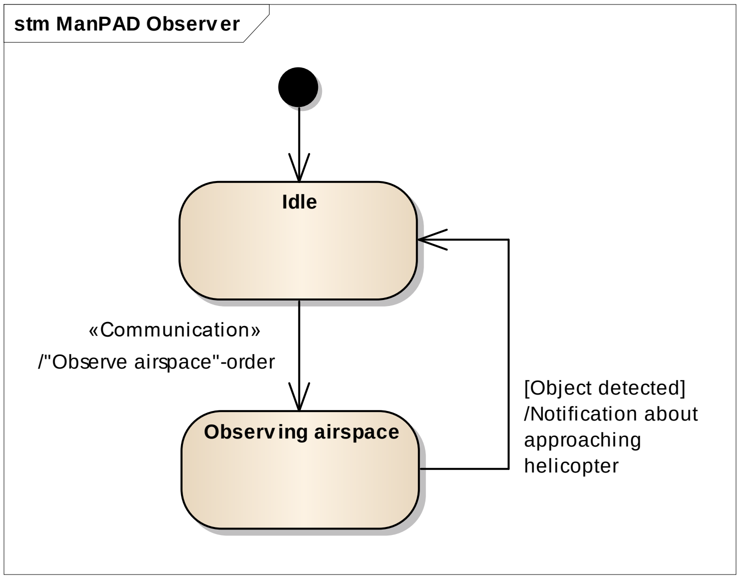

Figure 15: State machine of ManPAD Observer.

Figure 15 shows the state machine of the ManPAD Observer. The observer is “Idle” until he receives an “Observe airspace”-order. The observation of the airspace is continued until an object is detected. In this case, a notification is send to the ManPAD gunner and the ManPAD observer goes back into “Idle”-state.

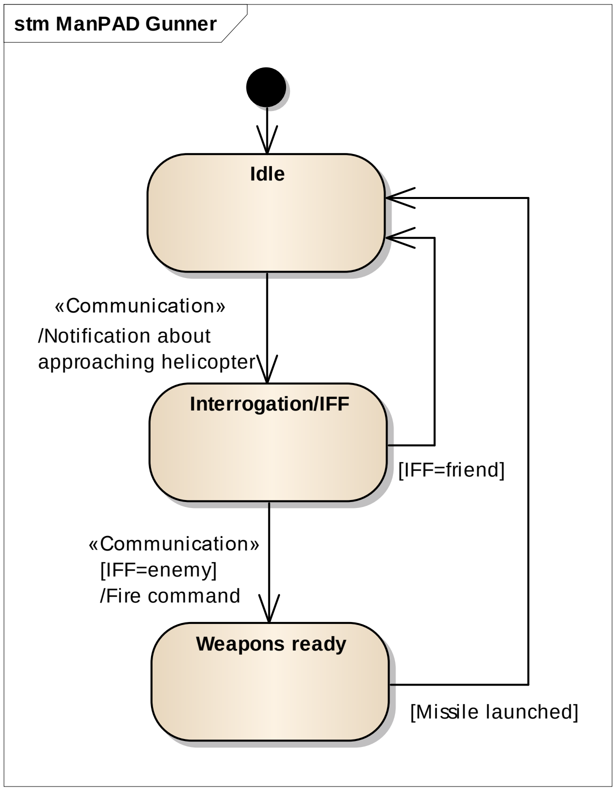

Figure 16: State machine of ManPAD Gunner.

Figure 16 presents the state machine of the ManPAD Gunner. The gunner is "Idle" until he receives a "Notification about approaching helicopter". Upon reception of this communication, the gunner starts the interrogation procedure and IFF. The result of the IFF is reported back to the ManPAD Commander (see Figure 13). If the ManPAD Commander sends the "Fire command" the gunner launches a missile and goes back into "Idle".

The case of an approaching helicopter being identified as friendly was not described in the original operational scenario. Yet, when developing the conceptual scenario and creating the state machine for the ManPAD Gunner, it became obvious that such a state transition (as indicated in Figure 16 by "IFF=friend") is necessary. Otherwise, the ManPAD Gunner would be stuck once he is in "Interrogation/IFF"-state.

Target

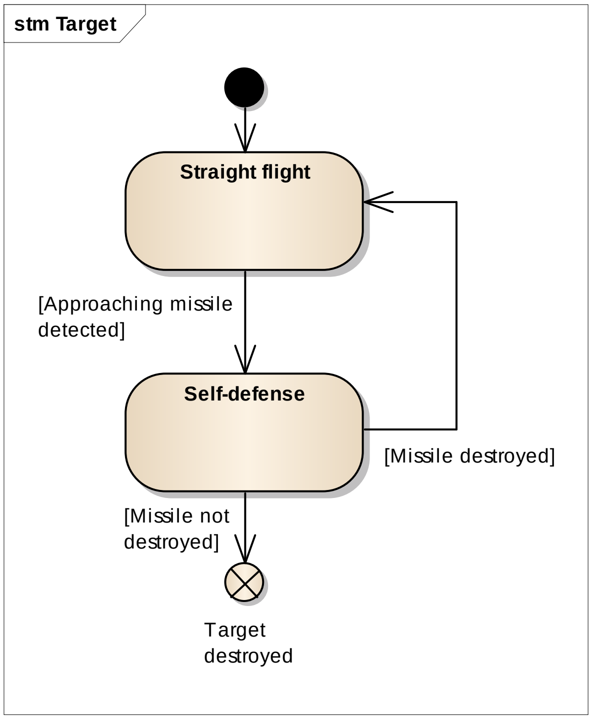

Figure 17: State machine of Target.

Figure 17 shows the state machine of a Target. As described in the operational scenario, a target is in straight flight until an approaching missile is detected. Upon detection of an approaching missile, the target goes into "Self-defense" state and throws its flares. Depending on the result of the defensive actions either the missile is destroyed (respectively does not hit and destroy the target) or the target is hit and destroyed.

Missile

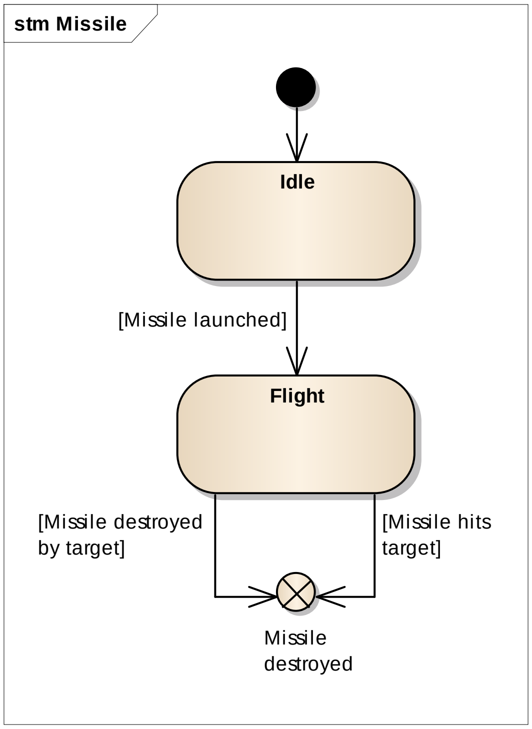

Figure 18: State machine of Missile.

Finally, Figure 18 shows the state machine of the Missile. The behavior of a missile is rather simple. Once launched, a missile flies towards the target and either destroys the target or misses the target. Independently of the result, the missile is destroyed.

D.3.3 Object Model Definition and Model Mapping

The Object Model Definition and Model Mapping components of the BOM Template Specification are neither suitable nor necessary for documenting conceptual scenarios.

Appendix E Example Scenario “Close Air Support” (Informative)

In this appendix an example is given how the information products related to scenario can be applied to a real world scenario. A Close Air Support scenario is described, where a Forward Air Controller (FAC) team on the ground and a flight of fighter aircraft work together to deliver a laser guided weapon on a target. See Joint Publication 3-09.3 [10] for more details about Close Air Support procedures. This example is based on work also described in ‘Effective Realism I/ITSEC paper [11]’, and ‘Effective Use of Simulation Means in Collective Mission Simulation I/ITSEC paper [36]’.

In the next sections the operational scenario and conceptual scenario for this case are given. The operational scenario is presented as it could be given by an operational person. This operational scenario is given as free text, so it is using a non-standardized scenario description.

E.1 Description of the operational scenario

A friendly infantry unit is being attacked by insurgents from a building. A flight of two F-16 aircraft is on a close air support mission and is flying above the mission area. The aircraft are equipped with laser guided bombs. The forward air controller (FAC) team of the infantry unit requests air support from the F-16 flight.

The flight lead determines which of the two aircraft will deliver the close air support. The attacking pilot contacts the FAC team on the ground. The FAC team requests a Type 1 CAS attack and delivers the information about the target using the standard 9 liner procedure.

Using his targeting pod (TGP) and while communicating with the FAC team the pilot identifies the target. After successful identification the pilot starts his attack run. He will only release his weapon after the FAC team announced CLEARED HOT. The pilot will announce when the weapon is 10 seconds before impact and at the moment the FAC team will start to illuminate the target with their laser.

After weapon impact the damage is assessed by the pilot and the FAC team. And if needed another attack is made.

E.2 Textual description of the conceptual scenario

E.2.1 Initial state

F-16 flight

- Structure
  - Flight lead
  - Wingman
- Equipment
  - Radio
  - Laser guided bomb
  - Targeting pod
- Initial position
  - In a holding pattern somewhere above the battlefield

The UML use case diagram below shows the actors within this scenario and indicates that both the flight lead and the wingman can become the attacking aircraft when close air support is requested. The flight lead will determine which aircraft is in the best position to deliver the support once the request is made.

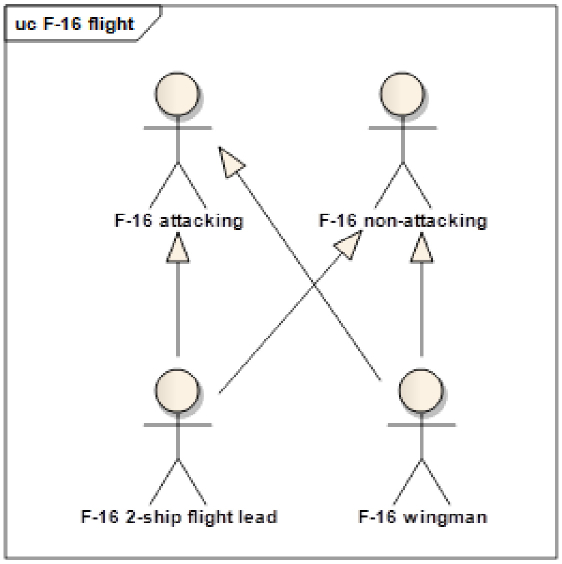

Figure 19: Actors

FAC team

- Structure
    - Team leader
    - Laser operator
- Equipment
    - Radio
    - Laser designator
- Initial position
    - Near the infantry unit
    - Within visual range of the insurgents

The UML diagram below shows the two roles in the FAC team graphically.

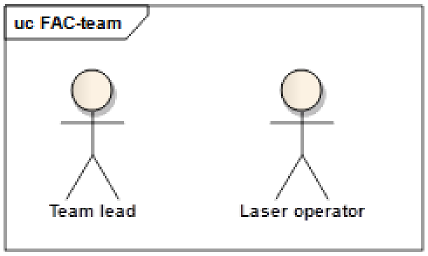

Figure 20: Two roles in the FAC team

Infantry unit

- Initial position

    - Near insurgents

Insurgents

- Initial position
    - Inside a building

Geography, date/time

- Rural environment
    - Day time

Surrounding conditions

- Clear weather
- No wind

Rules of engagement

- F-16
    - Must have identified the target.
    - Must have identified the location of friendly forces around the target.
    - Must have received CLEARED HOT instruction from FAC team.

E.2.2 Course of action

Communication events

- FAC team leader requests CAS from F-16 flight lead
- F-16 flight lead acknowledges CAS request
- FAC team leader provides basic target information to F-16 flight lead
- F-16 flight lead reports to F-16 wingman who is going to be the attacking aircraft
- FAC team leader provides detailed target information to the attacking F-16
- Attacking F-16 reports ready for attack run, once target has been identified
- FAC team leader reports CLEARED HOT when allowed to employ weapon
- Attacking F-16 reports to FAC team leader 10 seconds before expected weapon impact

Interaction events

- Insurgents fire at infantry unit
- Attacking F-16 identifies target (building with insurgents)
- Attacking F-16 releases laser guided bomb
- FAC team laser operator illuminates target with the laser designator

- Laser guided bomb acquires laser signal
- Laser guided bomb impacts on target or ground

Environmental events

- N/A

The UML activity diagram below shows the basic interactions between the different actors in the scenario.

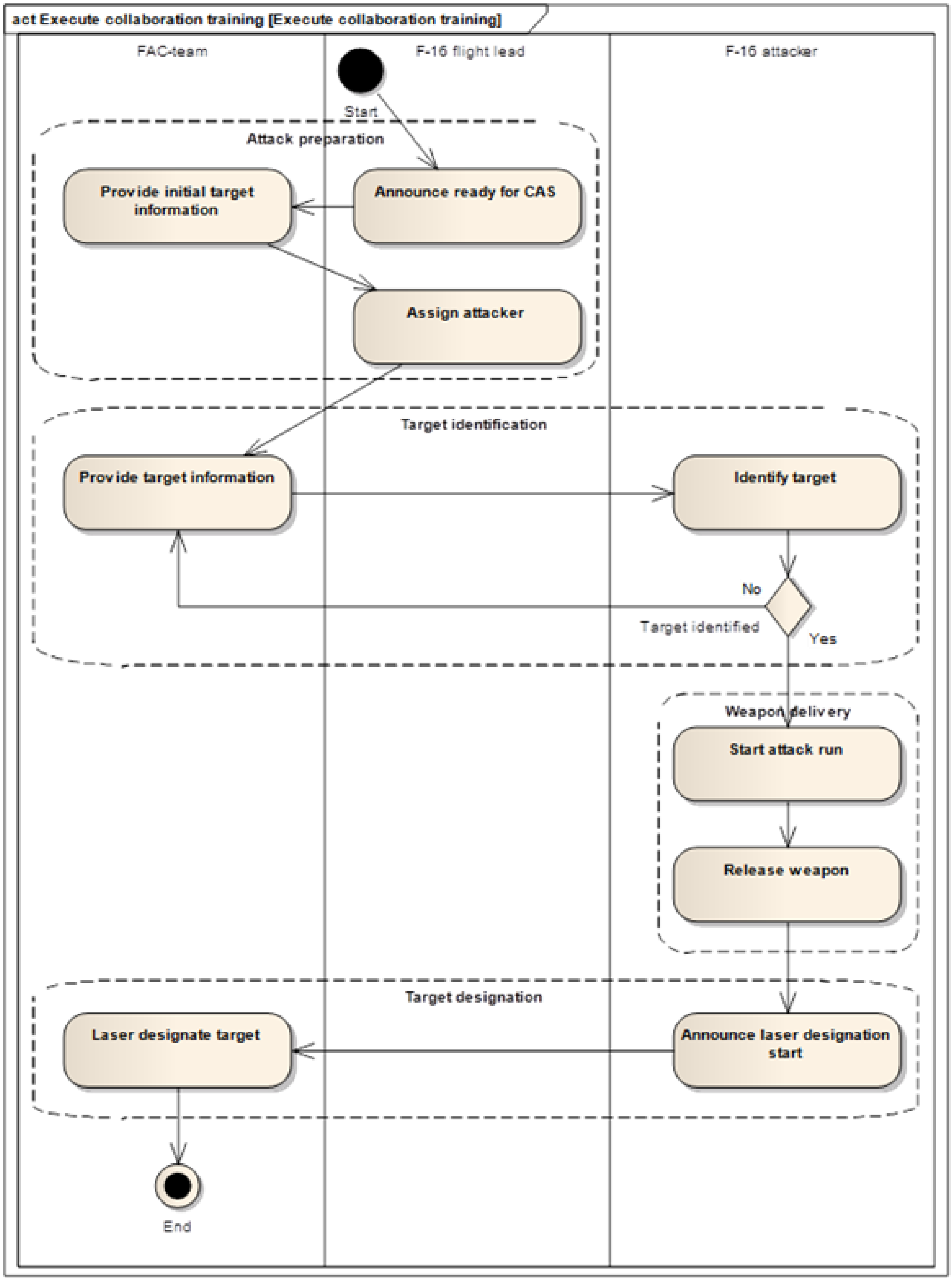

Figure 21: Scenario interactions between different actors.

E.2.3 Termination criteria

- Weapon has detonated and damage has been assessed.

E.3 Discussion

Compared to the other example scenario this example has chosen a different level of detail to describe certain elements of the conceptual scenario. This section discusses why this decision has been made. The required amount of detail should be balanced for each specific scenario.

On one hand as much detail as possible should be provided, since that eases the task to transform a conceptual scenario into an executable scenario. Any detail that is not included in the conceptual scenario will have to be determined by the developer producing the executable scenario.

On the other hand if different variants of the same scenario are used, it might be useful to make the conceptual scenario more generic, since that allows usage of the same conceptual scenario for deriving multiple executable scenarios.

In the specific case of this example scenario two variants of the same scenario were used in a real-world project. One variant situated the close air support mission in a training context, while the other variant put it in a mission rehearsal context. As a result of this the conceptual scenario does not provide very detailed initial position and related information of the different units.

When describing the course of events the different interactions have been described slightly more detailed than the other example scenario. This has been done because the different interactions among the entities were important aspects in the project on which the scenario is based. Specifying them in more detail helped with the development of the conceptual model.