NORTH ATLANTIC TREATY ORGANIZATION

SCIENCE AND TECHNOLOGY ORGANIZATION

AC/323(MSG-136)TP/832

www.sto.nato.int

STO TECHNICAL REPORT **TR-MSG-136-Part-V**

Modelling and Simulation as a Service, Volume 2: MSaaS Discovery Service and Metadata

(Modélisation et simulation en tant que service, Volume 2: Service de communication et métadonnées)

Developed by NATO MSG-136.

Published July 2019

Distribution and Availability on Back Cover

NORTH ATLANTIC TREATY ORGANIZATION

SCIENCE AND TECHNOLOGY ORGANIZATION

AC/323(MSG-136)TP/832

www.sto.nato.int

STO TECHNICAL REPORT **TR-MSG-136-Part-V**

Modelling and Simulation as a Service, Volume 2: MSaaS Discovery Service and Metadata

(Modélisation et simulation en tant que service, Volume 2: Service de communication et métadonnées)

Developed by NATO MSG-136.

The NATO Science and Technology Organization

Science & Technology (S&T) in the NATO context is defined as the selective and rigorous generation and application of state-of-the-art, validated knowledge for defence and security purposes. S&T activities embrace scientific research, technology development, transition, application and field-testing, experimentation and a range of related scientific activities that include systems engineering, operational research and analysis, synthesis, integration and validation of knowledge derived through the scientific method.

In NATO, S&T is addressed using different business models, namely a collaborative business model where NATO provides a forum where NATO Nations and partner Nations elect to use their national resources to define, conduct and promote cooperative research and information exchange, and secondly an in-house delivery business model where S&T activities are conducted in a NATO dedicated executive body, having its own personnel, capabilities and infrastructure.

The mission of the NATO Science & Technology Organization (STO) is to help position the Nations' and NATO's S&T investments as a strategic enabler of the knowledge and technology advantage for the defence and security posture of NATO Nations and partner Nations, by conducting and promoting S&T activities that augment and leverage the capabilities and programmes of the Alliance, of the NATO Nations and the partner Nations, in support of NATO's objectives, and contributing to NATO's ability to enable and influence security and defence related capability development and threat mitigation in NATO Nations and partner Nations, in accordance with NATO policies.

The total spectrum of this collaborative effort is addressed by six Technical Panels who manage a wide range of scientific research activities, a Group specialising in modelling and simulation, plus a Committee dedicated to supporting the information management needs of the organization.

- AVT Applied Vehicle Technology Panel
- HFM Human Factors and Medicine Panel
- IST Information Systems Technology Panel
- NMSG NATO Modelling and Simulation Group
- SAS System Analysis and Studies Panel
- SCI Systems Concepts and Integration Panel
- SET Sensors and Electronics Technology Panel

These Panels and Group are the power-house of the collaborative model and are made up of national representatives as well as recognised world-class scientists, engineers and information specialists. In addition to providing critical technical oversight, they also provide a communication link to military users and other NATO bodies.

The scientific and technological work is carried out by Technical Teams, created under one or more of these eight bodies, for specific research activities which have a defined duration. These research activities can take a variety of forms, including Task Groups, Workshops, Symposia, Specialists' Meetings, Lecture Series and Technical Courses.

The content of this publication has been reproduced directly from material supplied by STO or the authors.

Published July 2019

ISBN 978-92-837-2158-1

Table of Contents

	Page
List of Figures	**v**
List of Tables	**vi**
List of Acronyms	**vii**
MSG-136 Membership List	**ix**
Executive Summary and Synthèse	**ES-1**
Chapter 1 – Introduction	**1-1**
1.1 Background and Key Drivers	1-1
1.2 Document Overview	1-1
1.3 Related Activities	1-2
Chapter 2 – Discovery Service Overview	**2-1**
2.1 Discovery Service Needs Analysis	2-2
2.2 Types of Service Discovery	2-3
Chapter 3 – Introduction to Metadata	**3-1**
3.1 Types of Metadata	3-1
3.2 Mapping Between Metadata Standards	3-1
3.3 Metadata Content	3-1
3.4 Metadata Structure	3-2
3.5 Metadata Syntax and Encoding	3-2
3.6 Metadata Context	3-2
Chapter 4 – Evaluation of Metadata Standards	**4-1**
4.1 Dublin Core	4-1
4.2 MSC-DMS	4-1
4.3 FEAT	4-1
4.4 UDDI	4-2
4.5 NDMS	4-2
4.6 NCMS	4-3
4.7 Swagger	4-3
4.8 ISO 19115/19119/19139	4-3
4.9 Department of Defence Discovery Metadata Specification (DDMS)	4-4
4.10 MSG-136 Service Description Template	4-4
4.11 Summary	4-4

Chapter 5 – MSaaS Metadata Information Models 5-1
5.1 Service Description Template 5-1
5.2 The UK MSaaS Information Model 5-7
5.3 The UK Discovery Approach and the SOA Architecture Building Blocks 5-8
5.3.1 MSaaS Information Model Structure 5-9
5.3.2 Information Objects 5-11
5.3.3 Service Types 5-13
5.3.3.1 Participant Services 5-13
5.3.3.2 Supporting Services 5-13
5.3.3.3 Framework Administration Services 5-13
5.3.3.4 Simulation Administration Services 5-13
5.4 Analysis of MSG-136 and UK Approach to Metadata 5-14
5.4.1 UK MSaaS Information Structure 5-14
5.4.2 Results 5-15
Appendix – Appendix 5-1: Mapping of MSG-136 Service Description Template With UK Aims Metadata 5-16

Chapter 6 – Conclusions and Recommendations 6-1

Chapter 7 – References 7-1

List of Figures

Figure		Page
Figure 2-1	Discovery Activities Underpin Steps 3 to 7 of the Distributed Simulation Engineering and Execution Process (DSEEP)	2-1
Figure 2-2	Conceptual View of Discovery	2-1
Figure 4-1	NCMS Notational Layers	4-2
Figure 4-2	Relationship of the DDMS, NDMS, ISO19x, and EDS to the MSaaS Requirements	4-5
Figure 5-1	The UK MSaaS Information Model has a Number of Layers, Enabling Flexibility Within Information and Metadata, While Enabling a Single, Consistent Registry Model to be Used Within a Framework	5-10
Figure 5-2	The UK MSaaS Information Model Specifies a Set of Objects, Their Properties, and the Relationships Between Them	5-11

List of Tables

Table		Page
Table 5-1	Definition of Key Objects in the UK Information Model	5-12
Table 5A1-1	Core Information Elements	5-16
Table 5A1-2	Service Usage Information Elements	5-24
Table 5A1-3	Runtime Infrastructure Requirements	5-26
Table 5A1-4	Service Interface Elements	5-29

List of Acronyms

ABB	Architecture Building Block
ACT	Allied Command Transformation
AIMS	Architectures, Interoperability & Management of Simulations
API	Application Programming Interface
C2	Command and Control
C3	Consultation, Command and Control
C3CT	C3 Classification Taxonomy
CGF	Computer Generated Forces
COI	Community Of Interest
DC	Dublin Core
DCC	Digital Curation Centre
DDMS	DoD Discovery Metadata Specification
DMSCO	Defense Modelling & Simulation Coordination Office
DoD	Department of Defense
DSEEP	Distributed Simulation Engineering and Execution Process
EMBR	Enterprise Metacard Builder Resource
EXCON	Exercise Control
FEAT	Federation Engineering and Agreements Template
FMN	Federated Mission Networking
HLA	High Level Architecture
HTML	Hypertext Markup Language
ID	Identification Descriptor
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
ISO	International Organization for Standardization
IT	Information Technology
M&S	Modelling and Simulation
MISA	Metso Minerals IT Service Architecture
MSaaS	Modelling & Simulation as a Service
MSG	Modelling and Simulation Group
NAF	NATO Architecture Framework
NATO	North Atlantic Treaty Organization
NCMS	NATO Core Metadata Specification
NDMS	NATO Discovery Metadata Specification
NMRR	NATO Metadata Registry and Repository
NMSG	NATO Modelling and Simulation Group
NMSMP	NATO Modelling and Simulation Master Plan
OASIS	Organization for the Advancement of Structured Information Standards
OCD	Operational Concept Document

PoC	Point of Contact
RA	Reference Architecture
RDF	Resource Description Framework
REST	Representational State Transfer
RIM	Registry Information Model
RTI	Runtime Infrastructure
SDT	Service Description Template
SIMCON	Simulation Control
SISO	Simulation Interoperability Standards Organization
SME	Subject Matter Expert
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
TAP	Technical Activity Proposal
TDO	Trusted Data Object
UDDI	Universal Description, Discovery and Integration
UML	Unified Modelling Language
URL	Universal Resource Locator
UUID	Universally Unique Identifier
V&V	Verification and Validation
WMS	Web Map Service
XML	Extensible Markup Language

MSG-136 Membership List

CO-CHAIRS

Dr. Robert SIEGFRIED
aditerna GmbH
GERMANY
Email: robert.siegfried@aditerna.de

Mr. Tom VAN DEN BERG
TNO Defence, Security and Safety
NETHERLANDS
Email: tom.vandenberg@tno.nl

MEMBERS

LtCdr Tevfik ALTINALEV
Turkish Navy
TURKEY
Email: taltinalev@hotmail.com

Mr. Gultekin ARABACI
NATO JFTC
POLAND
Email: gultekin.arabaci@jftc.nato.int

Mr. Anthony ARNAULT
ONERA
FRANCE
Email: anthony.arnault@onera.fr

Col Thierry BELLOEIL
NATO ACT
UNITED STATES
Email: thierry.belloeil@act.nato.int

Dr. Michael BERTSCHIK
DEU Bundeswehr
GERMANY
Email: MichaelBertschik@bundeswehr.org

LtCol Dr. Marco BIAGINI
NATO M&S Centre of Excellence
ITALY
Email: mscoe.cde01@smd.difesa.it

Mr. Maxwell BRITTON
Department of Defence
AUSTRALIA
Email: maxwell.britton1@defence.gov.au

Dr. Solveig BRUVOLL
Norwegian Defence Research Establishment
NORWAY
Email: solveig.bruvoll@ffi.no

Dr. Pilar CAAMANO SOBRINO
CMRE
ITALY
Email: Pilar.Caamano@cmre.nato.int

Prof. Dr. Erdal CAYIRCI
Research Center for STEAM
TURKEY
Email: erdal@dataunitor.com

Mr. Turgay CELIK
MILSOFT Software Technologies
TURKEY
Email: tcelik@milsoft.com.tr

LtCol Roberto CENSORI
NATO M&S CoE
ITALY
Email: mscoe.ms08@smd.difesa.it

Maj Fabio CORONA
NATO M&S Centre of Excellence
ITALY
Email: mscoe.cde04@smd.difesa.it

Dr. Anthony CRAMP
Department of Defence
AUSTRALIA
Email: anthony.cramp@dst.defence.gov.au

Mr. Raphael CUISINIER
ONERA
FRANCE
Email: raphael.cuisinier@onera.fr

Mr. Efthimios (Mike) DOUKLIAS
Space and Naval Warfare Systems Center Pacific
UNITED STATES
Email: mike.d.douklias@navy.mil

Ing Christian FAILLACE
LEONARDO S.p.a.
ITALY
Email: christian.faillace@leonardocompany.com

Dr. Keith FORD
Thales
UNITED KINGDOM
Email: keith.ford@uk.thalesgroup.com

LtCol Stefano GIACOMOZZI
General Defence Staff
ITALY
Email: mscoe.ds02@smd.difesa.it

Mr. Sabas GONZALEZ GODOY
NATO ACT
UNITED STATES
Email: Sabas.Gonzalez@act.nato.int

Ms. Amy GROM
Joint Staff J7
UNITED STATES
Email: amy.m.grom.civ@mail.mil

Mr. Yannick GUILLEMER
French MoD
FRANCE
Email: yannick.guillemer@intradef.gouv.fr

Dr. Jo HANNAY
Norwegian Defence Research Establishment (FFI)
NORWAY
Email: jo.hannay@ffi.no

Mr. Andrew HOOPER
MOD
UNITED KINGDOM
Email: andy.hooper321@mod.uk

Mr. Willem (Wim) HUISKAMP
TNO Defence, Security and Safety
NETHERLANDS
Email: wim.huiskamp@tno.nl

Dr. Frank-T. JOHNSEN
Norwegian Defence Research Establishment (FFI)
NORWAY
Email: frank-trethan.johnsen@ffi.no

LtCol Jason JONES
NATO M&S CoE
ITALY
Email: mscoe.dr02@smd.difesa.it

Lt Angelo KAIJSER
Dutch Ministry of Defence
NETHERLANDS
Email: AJ.Kaijser@mindef.nl

Mr. Daniel KALLFASS
EADS Deutschland GmbH/CASSIDIAN
GERMANY
Email: daniel.kallfass@airbus.com

Col Robert KEWLEY
West Point
UNITED STATES
Email: Robert.Kewley@usma.edu

LtCol Gerard KONIJN
Dutch Ministry of Defence
NETHERLANDS
Email: gerard.konijn@gmail.com

Mr. Niels KRARUP-HANSEN
MoD DALO
DENMARK
Email: nkh@mil.dk

Mr. Vegard Berg KVERNELV
Norwegian Defence Research Establishment (FFI)
NORWAY
Email: vegard.kvernelv@ffi.no

Capt Peter LINDSKOG
Swedish Armed Forces
SWEDEN
Email: peter.j.lindskog@mil.se

Mr. Jonathan LLOYD
Defence Science and Technology Laboratory (Dstl)
UNITED KINGDOM
Email: jplloyd1@dstl.gov.uk

Mr. Jose-Maria LOPEZ RODRIGUEZ
Nextel Aerospace, Defence and Security (NADS)
SPAIN
Email: jmlopez@nads.es

Mr. Rene MADSEN
IFAD TS A/S
DENMARK
Email: Rene.Madsen@ifad.dk

Ms. Sylvie MARTEL
NCIA
NETHERLANDS
Email: Sylvie.Martel@ncia.nato.int

Mr. Gregg MARTIN
Joint Staff J7
UNITED STATES
Email: gregg.w.martin.civ@mail.mil

Mr. Jose Ramon MARTINEZ SALIO
Nextel Aerospace, Defence and Security (NADS)
SPAIN
Email: jrmartinez@nads.es

LtCdr Mehmet Gokhan METIN
Navy Research Centre
TURKEY
Email: m_gokhan_metin@yahoo.com

Mr. Aljosa MILJAVEC
MoD, Slovenian Armed Forces
SLOVENIA
Email: Aljosa.Miljavec@mors.si

Mr. Brian MILLER
U.S. Army
UNITED STATES
Email: brian.s.miller116.civ@mail.mil

Dr. Katherine MORSE
John Hopkins University/APL
UNITED STATES
Email: Katherine.Morse@jhuapl.edu

LtCol Ales MYNARIK
NATO JCBRN Defence COE
CZECH REPUBLIC
Email: mynarika@jcbrncoe.cz

Mr. Rick NEWELL
JFTC
POLAND
Email: rick.newell@jftc.nato.int

Mr. Jeppe NYLOKKE
IFAD TS A/S
DENMARK
Email: jeppe.nylokke@ifad.dk

Mr. Robbie PHILIPPS
Lockheed Martin Corporation
AUSTRALIA
Email: robbie.phillips@lmco.com

Mr. Marco PICOLLO
Finmecanica
ITALY
Email: marco.picollo@finmeccanica.com

Dr. LtCol (Ret) Dalibor PROCHAZKA
University of Defence
CZECH REPUBLIC
Email: dalibor.prochazka@unob.cz

Mr. Tomasz ROGULA
NATO Joint Force Training Centre
POLAND
Email: tomasz.rogula@jftc.nato.int

Dr. Martin ROTHER
IABG mbH
GERMANY
Email: rother@iabg.de

Mr. Angel SAN JOSE MARTIN
NATO ACT
UNITED STATES
Email: Angel.SanJoseMartin@act.nato.int

Maj Alfio SCACCIANOCE
NATO M&S CoE
ITALY
Email: mscoe.cde05@smd.difesa.it

LtCol Wolfhard SCHMIDT
JFTC
POLAND
Email: wolfhard.schmidt@jftc.nato.int

Mr. Barry SIEGEL
SPAWAR Systems Center – Pacific
UNITED STATES
Email: Barry.Siegel@navy.mil

Mrs. Louise SIMPSON
Thales
UNITED KINGDOM
Email: louise.simpson@uk.thalesgroup.com

Mr. Neil SMITH
UK MoD Dstl
UNITED KINGDOM
Email: nsmith@dstl.gov.uk

Mr. Per-Philip SOLLIN
Pitch Technologies AB
SWEDEN
Email: per-philip.sollin@pitch.se

Dr. Ralf STÜBER
CPA ReDev mbH
GERMANY
Email: stueber@supportgis.de

Capt Colin TIMMONS
Department of National Defence
CANADA
Email: colin.timmons@forces.gc.ca

Maj Dennis VAN DEN ENDE
Ministry of Defence
NETHERLANDS
Email: d.vd.ende@mindef.nl

Mr. Martin Dalgaard VILLUMSEN
IFAD TS A/S
DENMARK
Email: Martin.Villumsen@ifad.dk

Mr. Brian WARDMAN
Dstl Portsdown West
UNITED KINGDOM
Email: bwardman@dstl.gov.uk

Mr. Andrzej WNUK
Joint Warfare Centre
NORWAY
Email: andrzej.wnuk@jwc.nato.int

ADDITIONAL CONTRIBUTORS

Mr. Andy BOWERS
US Joint Staff J7
UNITED STATES
Email: francis.bowers@gdit.com

Mr. Brent MORROW
US Military Academy
UNITED STATES
Email: Brent.Morrow@usma.edu

Mr. Cory SAYLES
Lockheed Martin
UNITED STATES
Email: Cory.d.sayles@lmco.com

Mr. Roy SCRUDDER
The University of Texas at Austin
UNITED STATES
Email: roy.scrudder@arlut.utexas.edu

Mr. Dennis WILDE
European IAD Centre
UNITED STATES
Email: dennis.wilde@us.af.mil

Modelling and Simulation as a Service, Volume 2: MSaaS Discovery Service and Metadata

(STO-TR-MSG-136-Part-V)

Executive Summary

NATO and nations use simulation environments for various purposes, such as training, capability development, mission rehearsal and decision support in acquisition processes. Consequently, Modelling and Simulation (M&S) has become a critical capability for the alliance and its nations. M&S products are highly valuable resources and it is essential that M&S products, data and processes are conveniently accessible to a large number of users as often as possible. However, achieving interoperability between simulation systems and ensuring credibility of results currently requires large efforts with regards to time, personnel, and budget.

Recent developments in cloud computing technology and service-oriented architectures offer opportunities to better utilize M&S capabilities in order to satisfy NATO critical needs. M&S as a Service (MSaaS) is a new concept that includes service orientation and the provision of M&S applications via the as-a-service model of cloud computing to enable more composable simulation environments that can be deployed and executed on-demand. The MSaaS paradigm supports stand-alone use as well as integration of multiple simulated and real systems into a unified cloud-based simulation environment whenever the need arises.

NATO MSG-136 ("Modelling and Simulation as a Service (MSaaS) – Rapid Deployment of Interoperable and Credible Simulation Environments") investigated the new concept of MSaaS with the aim of providing the technical and organizational foundations to establish th*e Allied Framework for M&S as a Service within NATO and partner Nations. The Allied Framework for M&S as a Service* is the common approach of NATO and Nations towards implementing MSaaS and is defined by the following documents:

- Operational Concept Document;
- Technical Reference Architecture (including service discovery, engineering process and experimentation documentation); and
- Governance Policies.

MSG-136 evaluated the MSaaS concept in various experiments. The experimentation results and initial operational applications demonstrate that MSaaS is capable of realizing the vision that M&S products, data and processes are conveniently accessible to a large number of users whenever and wherever needed. MSG-136 strongly recommends NATO and Nations to advance and to promote the operational readiness of M&S as a Service, and to conduct required Science and Technology efforts to close current gaps.

This document examines different options to handle metadata and discovery in the context of Modelling & Simulation (M&S) as a Service (MSaaS) in alignment with the reference architecture. Metadata are the key element of a formalized description of user requirements and searchable descriptions of available resources. MSG-136 examined several available metadata schemes for their usability in an MSaaS context and finally in a best-of-breed approach suggested a minimum set of metadata elements necessary to enable discovery in an Allied Framework for MSaaS.

Modélisation et simulation en tant que service, Volume 2: Service de communication et métadonnées

(STO-TR-MSG-136-Part-V)

Synthèse

L'OTAN et les pays membres utilisent les environnements de simulation à différentes fins, telles que la formation, le développement capacitaire, l'entraînement opérationnel et l'aide à la décision dans les processus d'acquisition. Par conséquent, la modélisation et simulation (M&S) est devenue une capacité cruciale pour l'Alliance et ses pays membres. Les produits de M&S sont des ressources extrêmement précieuses ; il est essentiel que les produits, données et procédés de M&S soient facilement accessibles à un grand nombre d'utilisateurs aussi fréquemment que possible. Toutefois, l'interopérabilité entre les systèmes de simulation et la crédibilité des résultats ne sont pas encore acquises et nécessitent beaucoup de temps, de personnel et d'argent.

Les évolutions récentes du cloud informatique et des architectures orientées service offrent l'occasion de mieux utiliser les capacités de M&S afin de répondre aux besoins cruciaux de l'OTAN. La M&S en tant que service (MSaaS) est un nouveau concept qui inclut l'orientation service et la fourniture d'applications de M&S via le modèle « en tant que service » du cloud informatique, dans le but de proposer des environnements de simulation plus faciles à composer et pouvant être déployés et exécutés à la demande. Le paradigme du MSaaS permet aussi bien une utilisation autonome que l'intégration de multiples systèmes simulés et réels au sein d'un environnement de simulation dans le cloud, chaque fois que le besoin s'en fait sentir.

Le MSG-136 de l'OTAN (« Modélisation et simulation en tant que service (MSaaS) – Déploiement rapide d'environnements de simulation crédibles et interopérables ») a étudié le nouveau concept de MSaaS afin de fournir les bases techniques et organisationnelles permettant d'établir le « cadre allié de M&S en tant que service » au sein de l'OTAN et des pays partenaires. Le cadre allié de M&S en tant que service est la démarche commune de l'OTAN et des pays visant à mettre en œuvre la MSaaS. Il est défini dans les documents suivant :

- Document de définition opérationnelle ;
- Architecture de référence technique (incluant la communication du service, le processus d'ingénierie et la documentation d'expérimentation) ; et
- Politiques de gouvernance.

Le MSG-136 a évalué le concept de MSaaS au moyen de diverses expériences. Les résultats d'expérimentation et les premières applications opérationnelles démontrent que la MSaaS est capable de rendre les produits, données et processus de M&S commodément accessibles à un grand nombre d'utilisateurs, quels que soient l'endroit et le moment où le besoin s'en fait sentir. Le MSG-136 recommande vivement à l'OTAN et aux pays de faire progresser et d'améliorer l'état de préparation opérationnelle de la M&S en tant que service et de mener les travaux de science et technologie requis pour combler les lacunes actuelles.

Ce document examine différentes options de gestion des métadonnées et de la communication dans le contexte de la modélisation et simulation (M&S) en tant que service (MSaaS), en adéquation avec l'architecture de référence. Les métadonnées sont la clé de la description formalisée des besoins de l'utilisateur et de la possibilité de faire des recherches dans la description des ressources disponibles. Le MSG-136 a étudié la convivialité de plusieurs schémas de métadonnées disponibles dans le contexte de la MSaaS. Il a finalement suggéré un ensemble minimal d'éléments de métadonnées nécessaires à la communication dans un cadre allié de MSaaS.

Chapter 1 – INTRODUCTION

1.1 BACKGROUND AND KEY DRIVERS

NATO and the nations use simulation for various purposes, such as training, mission rehearsal, or decision support in acquisition processes. Achieving interoperability between participating simulation systems and ensuring credibility of results still requires enormous efforts with regards to time, personnel, and budget.

Recent technical developments in the area of cloud computing technology and Service Oriented Architecture (SOA) may offer opportunities to better utilize M&S capabilities to satisfy NATO critical needs. A new concept that includes service orientation and the provision of M&S applications via as-a-service cloud computing may enable more composable simulation environments that can also be deployed more rapidly and on-demand. This new concept is known as "M&S as a Service" (MSaaS).

NATO MSG-136 ("Modelling and Simulation as a Service – Rapid Deployment of Interoperable and Credible Simulation Environments") investigates this new concept with the aim to provide the technical and organizational foundations for a future permanent service-based M&S environment within NATO and partner nations.

MSG-136 focuses on several areas of M&S as a Service within NATO:

- Governance area: the governance concept and roadmap for M&S as a Service within NATO;
- Operational area: the operational concept of M&S as a Service: how does it work from the user point of view; and
- Technical area: the technical concept of M&S as a Service, covering reference architecture, reference services, and reference engineering process.

1.2 DOCUMENT OVERVIEW

This volume describes the options considered for metadata that may be suitable for MSG-136 to discover available M&S services. Metadata literally means 'data about data', i.e., data which carry information about other data like context, meaning, relationships, classifications and so on. In the context of MSG-136 metadata are understood as data used to describe the capabilities of the services.

The requirement for metadata can be derived from M&S Registry Services and M&S Repository Services Architecture Building Blocks (ABBs) and the Simulation Services ABB in Ref. [1] as described below:

> "Simulation Services are a set of capabilities for synthetic representation of (real-world) objects and events. Simulation Services are the service-oriented building blocks of simulations. Simulation services are provided by simulation environment member applications that adhere to the simulation environment's simulation interoperability standard or protocol and that execute underlying models for simulation. Simulation Services can be consumed by other member applications and by consumers outside the simulation environment. Member applications that provide simulation services to consumers outside the simulation environment effectively expose the simulation as a service."

> "The M&S Registry Services provide the capabilities to store, manage, search and retrieve data about (i.e., metadata) simulation resources stored by the M&S Repository Services, such as description of services interface and contract, information about QoS policies, and security and versioning information."

> "The M&S Repository Services provides the capabilities to store, retrieve and manage simulation resources and associations with / references to metadata managed by M&S Registry Services.

> The M&S Repository Services support any simulation resource that may be required for a simulation execution, independent of type and purpose (i.e., service implementations, applications, data files)."

In order to fulfill these requirements on categorizing, discovering, retrieving and evaluating services more information about the services themselves is needed. These additional information elements are called metadata.

1.3 RELATED ACTIVITIES

In 2017, the NATO Modelling & Simulation Group set up an Exploratory Team on "Federated Approach towards NATO Simulation Resources Management". This team was tasked to investigate existing approaches for M&S resource description, discovery, and reuse and to make recommendations for more effective management of simulation resources.

Obviously, this is closely related to the work on metadata presented in this report because metadata form the essential ingredient to describe, discover and thereby manage simulation resources. This sharing of resources was identified as a major driver for an improved Simulation Resource Management. Another finding of ET-47 was that a centralized approach does not seem feasible thus further emphasizing the importance of meaningful and shareable metadata descriptions. More details can be found in ET-47 Final Option Brief Presentation, [2].

Chapter 2 – DISCOVERY SERVICE OVERVIEW

During the design, implementation, and execution of a simulation environment a designer, developer, or integrator needs to search for simulation and data assets to get the environment up and running. Figure 2-1 illustrates how these discovery activities underpin several steps of the Distributed Simulation Engineering and Execution Process (DSEEP) [3].

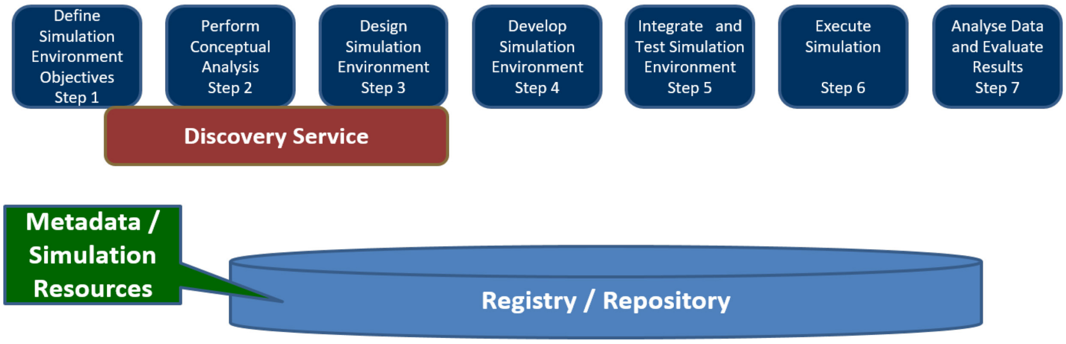

Figure 2-1: Discovery Activities Underpin Steps 3 to 7 of the Distributed Simulation Engineering and Execution Process (DSEEP).

In Step 1 (Define Simulation Environment Objectives), a Simulation Developer searches for previous projects and objectives of executed simulation environments to help defining the objectives of his own simulation environment.

In Step 2 (Perform Conceptual Analysis), Discovery may be performed to reuse existing scenarios.

In Step 3 (Design Simulation Environment), Simulation Developers will search for existing simulation designs, services and simulation assets that satisfy the requirements of the conceptual model developed in the previous Step 2.

It is obvious that the DSEEP plays a central role in the engineering of a simulation environment. Therefore MSG-136 has developed an overlay to DSEEP for supporting MSaaS [4].

The concept of Discovery is shown in Figure 2-2.

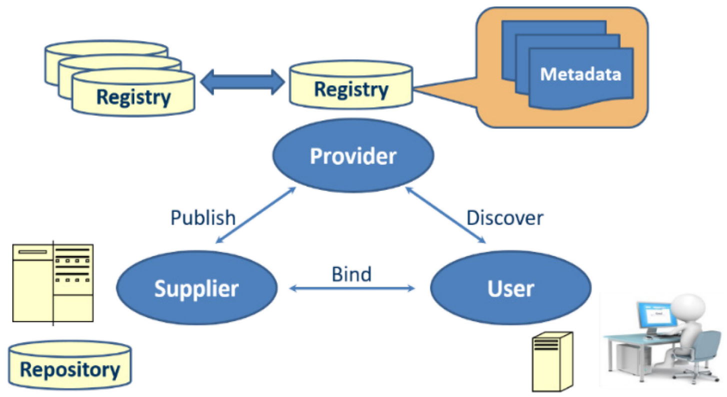

Figure 2-2: Conceptual View of Discovery. Metadata is fed into an M&S Registry; a broker service serves as an interface for queries from simulation developers or for additional metadata fed into the database by resource providers.

Before execution of a Simulation Environment, Service Providers will publish services they want to make available in a M&S Registry. Simulation Developers will query the M&S Registry to discover Simulation Resources that will satisfy their requirements. In order for a Simulation Developer to determine if a particular Simulation Resource satisfies their requirements, metadata has to be provided by the Resource Provider, along with the Simulation Resource, itself. This metadata describes the capabilities of the resource and the required interfaces or data standards needed to access the resource.

When a suitable Simulation Resource has been identified, this will be combined with other resources into a simulation composition and deployment.

To deploy the simulation, a Simulation Developer will use a deployment tool to search for and download the deployment. The deployment tool will interrogate the M&S Registry to determine how to access the Simulation Resources. For a data resource, the M&S Registry will provide a URL for where the data is stored in an M&S Repository. In the case of a service resource that is running 24/7, the M&S Registry will provide the Universal Resource Locator (URL) where the service can be accessed. For services that are not already installed, they will need to be downloaded from a M&S Repository and installed on the simulation infrastructure. The executable code typically is stored in the form of a virtual machine or container image. This situation is very similar to the case of a pure data resource mentioned above.

If the M&S Registry is implemented using international standards it is possible to federate it with other Registries thus allowing a particular query to be forwarded if it cannot be fulfilled locally. This also allows access to a wider range of Simulation Resources.

2.1 DISCOVERY SERVICE NEEDS ANALYSIS

The fundamental idea of the discovery concept in the context of MSaaS is to enable simulation resources to be registered, searched and retrieved on-demand by users. The user interaction with the framework is further detailed by the MSaaS Governance Policies in Ref. [5]. Following this approach an Allied Framework for MSaaS exists independently from the processes involved in composing, deploying and executing a simulation. The framework and its content are supplied and harvested. This happens independently of a concrete Simulation Environment implementation or realization project and enables the realization of an Allied Framework for MSaaS. This decoupling makes the fundamental difference to more traditional systems-engineering-based approaches where requirements are stated, and a single system is designed and developed to meet those particular requirements. It also follows from this fact that simulation developers will need to adopt new working methods to get maximum added value from the framework approach.

This has many implications for the supply of content within an Allied Framework for MSaaS. The supply of M&S resources is no longer tied to an explicitly stated simulation requirement (although, it is possible for users to state a requirement and use a more traditional procurement model to acquire additional components which are not yet available in the framework). Instead, content is made available via the framework independently of any end-user requirement and users are free to use it according to its specific Service Level Agreements (SLA). This does not mean that a user does not have to specify requirements or follow certain processes. There are still requirements to adhere to, but these requirements rise above the scope of a single simulation environment (e.g., standards to comply, too, or certain domain-related agreements like RPR-FOM).

MSG-136 is aware that this is an idealized vision. It is assumed that in the most likely scenario, market forces and the well-known supply-demand paradigm will drive the resources available in the Allied Framework for MSaaS to align with the users’ needs and requirements.

Typical user expectations for building and executing simulation environments in the context of this framework include, but are not limited, to:

- Be able to specify simulation requirements (on entities, synthetic environments and behavior) in a way generic enough to make use of the ‘best-fit’ component that is already available within the eco-system, rather than being forced to develop a specific component suitable for only one-use case.
- Be able to compose a simulation by identifying and defining a set of services that will deliver elements of the simulation requirement.
- Be able to create a simulation environment deployment that defines where and how all of the participating services will be run.
- Be able to execute a specific deployment on demand.
- Be able to record and store information about simulation environment executions, including context information about the execution (e.g., name of an exercise series) as well as any associated simulation data.
- Be able to retrieve existing compositions, deployment and context information on demand.

MSG-136 evaluated the applicability of several existing metadata specifications for capturing the requirements for service Discovery derived from these user expectations. They may be classified into three categories:

- Enable search and reuse by humans and machines;
- Supply Service Contract information and Service Interface information; and
- Align or conform, when appropriate, with existing community metadata “standards”.

In order to meet these high-level requirements, the following challenges need to be dealt with:

- Effective combination of machine and human-readable metadata;
- Discovery via online catalogues and registries;
- Provisioning of enough descriptive metadata for humans to reason about suitability of services;
- Provisioning of sufficiently detailed technical metadata to establish connectivity and interface correctly with the service; and
- Design compatible to existing and adopted Discovery Metadata Schemata (e.g., NCMS and MSC-DMS) to allow low-effort integration and “proper tailoring” at the same time.

2.2 TYPES OF SERVICE DISCOVERY

In the context of this document and the work of MSG-136, the following definitions were accepted:

Discovery is concerned with finding information. It can be compared to a Google search, where the user enters a phrase into a search engine and a set of links or documents related to the search phrase is returned. Thus, a non-specific phrase leads to non-specific search results. The more specific the user-specified search phrase is, the more specific are the results of the query returned by the search engine. In the M&S, domain discovery is related to the process of discovering metadata required to achieve specific M&S goals. This information includes, but is not limited to, information about the services needed to run a specific simulation environment.

Stepping down one level in the hierarchy, *service discovery* can be further divided into *design-time discovery* and *run-time discovery*.

By *design-time service discovery* and selection, we denote the browsing for desired services to be included in a simulation composition by a simulation developer. The metadata can also provide information beyond the pure technical API details, e.g., naming responsible persons for the service, provide some cost figure and so on. This situation is of limited complexity from an implementation point of view because there is a human in the loop to read and interpret the descriptions of the services' semantics.

Run-time discovery occurs when the simulation composition is already deployed and encompasses the machine-to-machine aspects of service discovery. This kind of discovery is concerned with finding running instances of services that fulfill the search criteria or identifying virtual machines or containers to be deployed. Run-time services might be used as part of a load balancing scheme or to dynamically configure the system on start-up following design principles like late binding or loose coupling of software components. Here, the (machine) client needs to be designed to consume services using a specified interface. The actual search and binding to a particular service instance will happen at runtime, using this specified interface.

Chapter 3 – INTRODUCTION TO METADATA

Metadata means *data about data*. All information systems contain, utilize and produce data and information; metadata is a means of managing the data in a way to enable easy storage, retrieval and queries. To ensure interoperability and compatibility between national approaches to metadata management it is necessary to define a high-level concept covering all aspects of metadata. This includes the content, semantics, structure and syntax of metadata, and, in the ideal case, a higher-level concept of metadata handling. MSG-136 has identified several metadata standards that could be used to describe simulation resources.

3.1 TYPES OF METADATA

The following types of metadata have been identified for MSaaS:

- *Discovery Metadata*: This is the minimum amount of information needed to describe the context and content type of the data resources. It essentially answers the "what, why, when, who, where, and how" questions about resources;
- *Use Metadata*: This is the information required to access, transfer, interpret and use a simulation resource in an application; and
- *Management Metadata*: This is the information required to support the management of the data by organizations or by technical infrastructure. Examples are original identifiers, original source, runtime instances of simulations, and their compositions and deployments. This also includes version-tracking data, distribution caveats, references, etc.

Although the majority of metadata standards addresses most of the categories presented above, each application domain may have a different profile that is tailored to their particular needs. The Digital Curation Centre (DCC) website [6], has a good example of a list of domain-specific metadata standards. Other examples can be found on Wikipedia [7].

3.2 MAPPING BETWEEN METADATA STANDARDS

It is possible to maintain compatibility between national approaches that use different standards to address each of these elements as long as each nation fully defines the approach that they are using, and each pair of approaches is formally mapped. This will not guarantee full compatibility among national approaches but adhering to common metadata standards and M&S constructs will make international interoperability more likely.

Content and semantics of the metadata are of particular importance for interoperability. The more two metadata definitions conform regarding represented information, underlying definitions, and allowed semantic variety, the more feasible it is to construct a mapping between these definitions. Clearly, a mapping will be simple as long the metadata definitions differ primarily in structure and syntax. Practical experience shows that this assumption is correct at least for some parts of two metadata definitions whereas there is less overlap for other parts of the same two definitions.

There are many standards that cover one or more aspects of metadata implementation. A comparison between different elements of metadata implementation of practical relevance is given in Ref. [8].

3.3 METADATA CONTENT

A popular standard often used to describe generic content is the Dublin Core (DC) standard, [9], which defines 15 elements of metadata. This standard does not address semantics or syntax. There exists, however, other standards that provide extended implementation specifications based on DC. Some examples are given below.

3.4 METADATA STRUCTURE

Metadata typically uses hierarchical models of nested data structures. This type of structure is the most natural way of representing metadata kind of information and all of the formal encoding standards described below that use it. However, if a participant chooses not to use this type of structure, then this may cause issues when trying to map data to commonly used encodings.

An alternative structure is provided by the Resource Description Framework (RDF), [10]. RDF statements comprise a set of triples to describe entities, their properties and the relations between them. RDF, therefore, is compatible with any scheme of similar structure (entities, properties, relations). In particular, this applies to the compatibility with the schemes proposed by MSG-136 and the schemes used in the UK registry model.

3.5 METADATA SYNTAX AND ENCODING

Standards for encoding data are important in terms of enabling information exchange. As discussed in Section 3.2, it is not necessary that all participants use the same encoding. Rather it is essential that they all use and publish a formal statement of their approach (usually as a schema) to enable other users to translate metadata into their own format. This translation process can be eliminated if every user adheres to the same encoding standard. MSG-136 proposes Extensible Markup Language (XML) encoding of metadata, either following a standardized or an individual XML schema. There may be, however, other approaches for encoding with similar characteristic properties.

3.6 METADATA CONTEXT

Another important area addressed by some metadata standards is the management of metadata within a broader context. An example is the Department of Defence (DoD) Discovery Metadata Specification (DDMS) [11], which contains guidelines on the embedding of standard generic metadata into metadata cards within the Trusted Data Object (TDO) structure.

Chapter 4 – EVALUATION OF METADATA STANDARDS

This chapter gives an overview of several metadata standards analyzed by MSG-136 for their applicability as a description schema for simulation resources. As a result of this analysis, the MSG-136 team developed a service description template. This template is discussed in detail in Section 5.1. During this research also standards were disregarded, which only focus on the details of interfacing with services without addressing semantics (e.g., WSDL).

4.1 DUBLIN CORE

The Dublin Core Metadata Element Set (DC) in Ref. [9] is a relatively small set of vocabulary terms that can be used to describe typical media resources like books, video or images. Information contained in a "classic" DC metadata set refers to title, author, subject, description, publisher, date format, etc. The DC metadata set is internationally standardized by the Internet Engineering Task Force (IETF) and the International Organization for Standardization (ISO).

It is obvious that this very general set of metadata intentionally lacks any domain-specific information and, therefore, can only be a subset of a domain-specific metadata set. However, by using standardized DC as a subset, a specific metadata set immediately gains international compatibility at least at the level of the core metadata elements. MSG-136, therefore, decided to integrate the DC elements into the MSG-136 Service Description Template.

4.2 MSC-DMS

The M&S Community of Interest (COI) Discovery Metadata Specification (MSC-DMS) [11] is used by the US DoD M&S Catalog and maintained by the Defense M&S Coordination Office (DMSCO). It can be accessed *via* the Enterprise Metacard Builder Resource (EMBR) Portal. MSC-DMS is based on the DoD DDMS in Ref. [12].

Analysis of possible mappings between the MSG-136 service description template and the MSC-DMS led to unsatisfactory results. This results from the MSC-DMS lack of a considerable amount of detail required to support meaningful discovery. MSG-136 also considered the possibility of extending the schema through the '##any element' at the end without significant improvement. During this research, the team alternatively also considered using the NATO C3 Taxonomy [12] as part of a possible extension with similar results.

The work done here refers to the publicly available MSC-DMS, i.e., the metadata specification document. The access to the M&S Catalog itself is restricted to US users.

4.3 FEAT

The Federation Engineering Agreements Template (FEAT) [13] provides a standardized format for recording federation agreements to increase their usability and reuse. The template is an XML schema from which compliant XML-based federation agreement documents can be created. The mapping from the MSG-136 service description template to the FEAT proved to be very fragmented. This results from the design of the FEAT as a standardized recording schema for already made agreements. It shows that this design is not suitable as a schema to support discovery of components to be integrated into a federation. However, the FEAT might be useful to record the use of services after the actual discovery process. MSG-136 is also investigating extensions to FEAT to support this activity (see Ref. [4] for more information).

4.4 UDDI

UDDI is an XML-based standard for describing, publishing, and finding Web services [14]. UDDI was disregarded during the initial review phase by MSG-136 team because it has not been as widely adopted as originally intended.

4.5 NDMS

The NATO Discovery Metadata Specification (NDMS) is the metadata schema that supports the NMRR as the specification for its metadata cards [15]. The NMRR stores XML artefacts and is partially aligned with ISO 11179 standard for metadata registries. The metadata searchable in NMRR is a combination of the metadata card and the contents of the NMRR XML artifact.

NDMS is organized in layers, two of which – the security layer and the core layer – are specified in “Guidance on the Use of Metadata Element Descriptions for Use in the NDMS” [16]. Additional layers may be added by COIs as necessary to support their requirements, but the specification of such layers is left to the COI, as illustrated in Figure 4-1. MSG-136 noted the presence of a security layer but did not analyze it further because of the decision of the group to defer security issues to a follow-on activity.

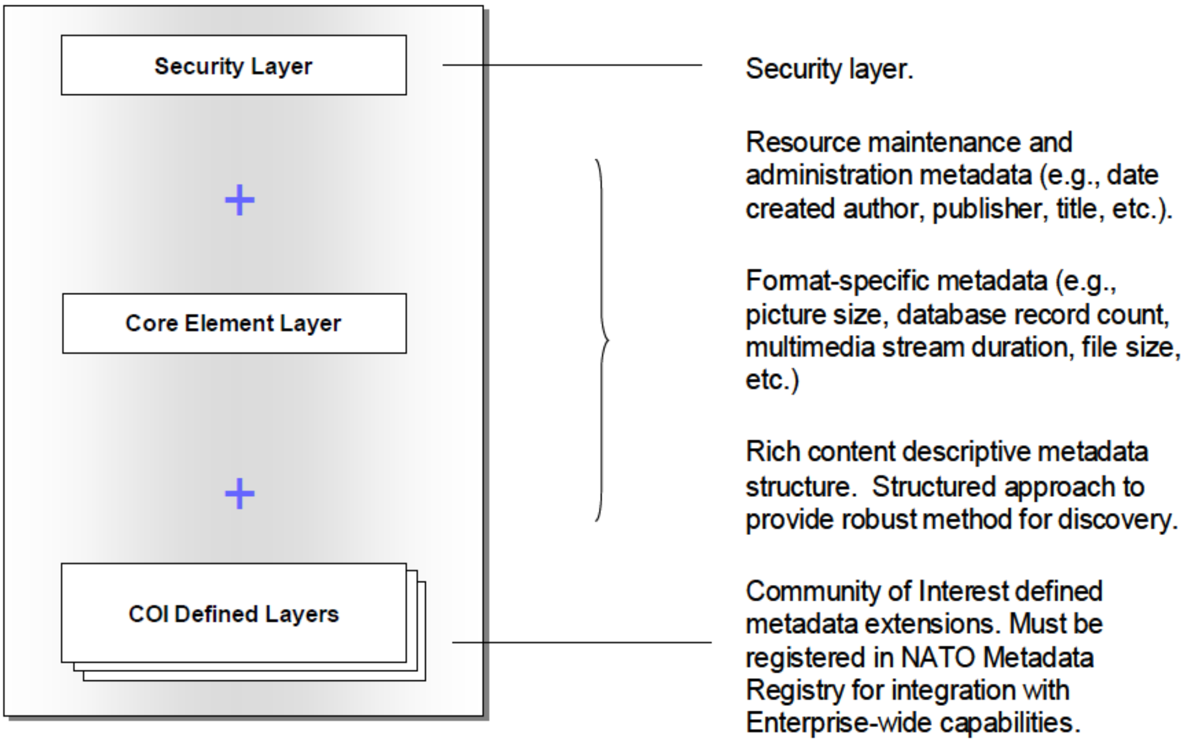

Figure 4-1: NCMS Notational Layers.

The core layer elements of the NDMS (e.g., title, subject, description) align naturally to the higher-level information requirements of the MSG-136 service description template. The guidance document also includes an “obligation” designation for each element with possible values of “mandatory”, “mandatory if applicable”, “recommended” or “optional”, which the team considered to be useful. This also applies to additional explanatory information such as purpose, comments, examples, syntax, and mapping which can be added to each element. After reviewing and mapping the guidance document the team evaluated the concept of extending the existing layers with COI-specific layers and considered this as a logical approach to add M&S-related metadata. The team also noted the clarity of the guidance, the definitions and the amplifying information. As NDMS is based on the DC Metadata Element Set, the team assumed that all the schemata under consideration were based on DC [9], which would have made the extension process easier. However, the team found this assumption to be wrong.

By further analyzing the feasibility of defining COI-specific layers to cover MSG-136's needs, the team realized that NDMS is going to be superseded by the NATO Core Metadata Specification (NCMS) [17]. This evolution of the NDMS defines a core set of elements organized in a layered structure. NCMS supports NATO information management tasks common to all COIs while allowing these communities to append COI-specific layers to the core specification. NCMS has a wider scope and additional enhancements (e.g., an additional lifecycle support layer), while still supporting NDMS's information discoverability focus. The team, therefore, decided to concentrate further analysis on NCMS.

4.6 NCMS

NCMS is the NATO Core Metadata Specification, which is to be applied at NATO information management level to every piece of information, be it library cards, properties in static files, or even embedded content in streamed data. Its implementation, therefore, must address the specific context in which it will be applied, and which will be different for every use case. This is achieved by using a layer approach with the NCMS at the bottom (core layer) and individual context- or COI-specific metadata extension on top in multiple layers.

NATO currently is collecting input from all COIs in order to produce harmonized metadata as extension layers to NCMS. From MSG-136's point of view, the NCMS is a sound basis, which is already compliant with NATO guidelines. MSG-136, or follow-on activities, therefore, should contribute an M&S COI-specific metadata extension to the NCMS. The semantic precision made possible using the NCMS/NDMS extension mechanism makes this approach appear very promising. The precision results from the sound definitions of the core and security terms, required and optional entries, well-defined syntax and understandable examples.

However, the MSG-136 team also sees challenges with this proposed integration of M&S service specific metadata into NCMS. This type of metadata is required to support the description of M&S-specific services while other non-M&S-specific services relevant for other COIs will require different metadata services. It is well-known from similar situations that the independent development of extensions to a common base by different COIs will almost certainly lead to inconsistencies and interoperability issues. Should a follow-on activity of MSG-136 decide to develop a COI-specific extension to NCMS, special attention has to be paid to this risk.

4.7 SWAGGER

Swagger in Ref. [18] is a simple representation of Representational State Transfer (RESTful) Application Programming Interface (APIs). Since Swagger was designed strictly to represent APIs, it only covers elements of the Service Profile in the Service Description template and none of the other three sections. The MSG-136 team, therefore, decided to disregard Swagger for a deeper analysis.

4.8 ISO 19115/19119/19139

ISO 19115 is a standard for geographic information metadata that uses DC fields for part of its information representation. The related standard ISO 19119 is used for services within the geographic domain and inherits elements from ISO 19115. Although it has been developed with the geo domain in mind, much of its content is not geo-specific. Both standards define content and structure of their respective metadata sets but do not specify an encoding. ISO 19139 also belongs to this family of standards and provides an XML implementation schema to be used in conjunction with other standards of this family.

A combination of these standards has been used for a national implementation of an MSaaS registry in the UK. The implementers opted for these standards because of their generic metadata set, which enables the use of controlled vocabularies (although, do not mandate them) and can be extended, if required.

4.9 DEPARTMENT OF DEFENCE DISCOVERY METADATA SPECIFICATION (DDMS)

The DDMS makes use of DC to define a set of generic metadata within its Primary Category Sets, i.e., the mandatory elements within the core layer. An XML implementation for US DoD and Intelligence Community projects is provided in the form of the "XML Data Encoding Specification for Information Security Marking Metadata Version 9" [19]. However, the document states:

- That other users can adhere to the DDMS, while using a different implementation;
- That the standard *"*…does not provide an interchange specification or substantive implementation guidance";
- That it is "…designed to be platform-, language-, and implementation-independent"*;* and
- That it "…provides flexibility for system developers to generate and retain discovery metadata using any implementation approach, including using Commercial Off-The-Shelf (COTS) products*.*"

The DDMS also specifies the use of TDOs, which defines a structure that enables assertions to be made about a metadata payload. These assertions are used to specify information, such as the origin and security markings of the metadata.

DDMS is now superseded by MSC-DMS. Also, because of the missing service functionality and lack of other features, the MSG-136 team decided not to do further research on DDMS.

4.10 MSG-136 SERVICE DESCRIPTION TEMPLATE

After carefully analyzing existing metadata standards, MSG-136 suggested possible metadata elements for a M&S domain-specific metadata scheme known as the MSG-136 Service Description Template. Although not formally standardized, this template may serve as a source and structuring scheme for metadata content relevant for a standardized description of services. The MSG-136 Service Description Template is presented and discussed in Section 5.1.

4.11 SUMMARY

From the analysis of the individual metadata description standards and their comparison with the MSG-136 service description template, the most promising way forward is to use the NCMS as a basis and add multiple extension layers addressing specific requirements of the M&S COI. Some of these extension layers should derive their metadata content from the NATO C3 Classification Taxonomy (C3CT) [20] to ensure compatibility and ease later integration with other NATO systems. Other layers may then be used to represent more specific metadata relevant only for specific groups of users or specific application contexts.

Figure 4-2 shows the relationship of the standards identified as most promising candidate sources of metadata to fulfill the requirements for MSaaS. Other standards examined by the MSG-136 team (UUDI, Swagger, FEAT, etc.) are not listed here for reasons discussed in the previous sections.

Mandatory, fully defined	Mandatory, partially defined	Optional, fully defined	Optional, partially defined	Not defined

	Required for basic metadata layer implementation			Additional optional MSaaS metadata layer implementation		Required for UK info model implementation		Not required for layered model but useful to achieve broader goals	
	Metadata encoding	Metadata generic structure	Metadata generic content	Extension Structure	Extension Content	Information Layer	Registry Layer	Additional Core Content	High-level Structure
DDMS	Text / XML / HTML examples	Suggested XML spec: Information Security Marking Metadata		Not defined	Allowed in structure, no guidance on implementation	Not defined	Not defined	Not defined	Trusted Data Objects (TDO)
NDMS	XML suggested but not defined	Not defined	Based on Dublin Core	Not defined	CoI Metadata, no guidance on implementation	Not defined	Not defined	Security Layer, Information Lifecycle Layer	Not defined
UK Approach (ISO 19115/ 19119 etc)	XML	Complete formal schema		Complete formal schema	Geo data, and geo services	Not defined, tailored MSaaS model developed	ebRIM Registry Information Model	Not defined	Not defined
EDS	Not defined but references DDMS	Not defined but references DDMS	Dublin Core via DDMS reference	Not defined but references DDMS	Not defined but references DDMS	Conceptual Data Model View – baseline, not MSaaS-specific	Not defined	Not defined	Trusted Data Objects (TDO)

Figure 4-2: Relationship of the DDMS, NDMS, ISO19x, and EDS to the MSaaS Requirements.

Chapter 5 – MSaaS METADATA INFORMATION MODELS

The following sections describe two different approaches to define and use metadata. The first approach developed by MSG-136 focuses on the description of a minimal set of human readable design-time discovery metadata for M&S Services. Section 5.1 describes the Service Description Template generated by MSG-136.

The second approach developed by the UK focuses on the description of human and machine-readable metadata for M&S Services for discovery, composition and execution. Sections 5.2 and 5.3 describe the use of the ISO 19115 standards within the UK approach to metadata definition and discovery. The latter, therefore, comprises a wider approach to describe all the different types of Simulation Resources e.g., services, composition, deployments, whereas the MSG-136 Service Description template focuses on service description only, delegating the support for other types of simulation resources to follow-on activities.

5.1 SERVICE DESCRIPTION TEMPLATE

The approach of MSG-136 to develop a service description template started by gathering data about existing services. The team adopted the Metso Minerals IT Service Architecture (MISA) Generalized Service Template [21], which holds a general description of a service for business/enterprise purpose. MSG-136 concluded that this template contained most of the necessary information but that it lacks a machine-readable format definition.

The MISA Generalized Service Template comprises the following content:

Service Profile

- *Purpose* – What the service does
  - Description
  - C3 Taxonomy Functional Requirement
- *Structure*
  - Identification
    - Service Name
    - Service ID – ID given by simulation infrastructure (e.g., integer number) or a version number
    - Description
      - Business Purpose – Expectations from the client or users, desired result
    - Contacts – Phone numbers, email addresses of persons who are responsible for this service
  - Functionality
    - Parameters – A set of variables to set up the service before runtime/call
    - Input – A set of variables to set up/call the service during runtime
    - Output – A set of variables that is sent to the service caller
    - Preconditions – implementation steps/state of systems needed for the service
    - Effect – The causal result of the output of the service within the simulation
    - Category – Describes in which context the service can operate

- Business Criticality – Depend on the business model/conceptual model
  - Category
    - Critical – Business would fail without service
    - High – Business would be seriously delayed/troubled without service
    - Low – Business would not be affected
  - Justification – Reasons for choice of criticality
- Status
  - Not analyzed – Prototype version
  - Analyzed – First investigations on service performance
  - Approved – Usage of service allowed by accredited business unit
  - Tested – First usage within intended business environment
  - In operation – Service in use
  - Terminated – Service usage ended

Service Resources

- *Purpose* (What is needed to run the service) – Hardware requirements, etc.
- *Structure*
  - Human (skill set) – Skills necessary for user to use the service
  - Technology
    - Applications – Applications/other services necessary to run the service
    - Infrastructure – (Network) infrastructure necessary to run the service
  - Information and Data
    - Metadata (control) – Standards and documents relevant for this service
    - Data (content) – Data/Databases needed for this service

Service Model

- *Purpose* (How the service works)
- *Structure*
  - Responsibilities
    - Owner, business responsible – E.g., company name, MoD
    - Content responsible – Business unit name, contributor
    - Technology responsible – Business unit name, provenance
    - Other responsible – Business unit name
  - Business logic description (data flow chart or similar) – e.g., NATO Architecture Framework (NAF) description
  - Budgeting or financing model – How to pay for the service

- Projecting model – Textual description of the business / organization-dependent projecting model
- Service agreement – Agreement between customer and provider of service
- Service activation – How to get access to this service
- Service validity (annual model, etc.) – Information about the maintenance of the service
- Service cancellation – How to dispose of the service

Service Access

- *Purpose* – Textual description of the interfaces rather than description of variables
- *Structure*
  - Availability – Information about availability with respect to time and resources
  - Address – E.g., website/facility where service is available/located
- *Access Method* – Interfaces used by the service, structure of service call

MSG-136 reviewed this template intensively and identified two requirement gaps:

- A machine-readable version of the template is required for MSaaS service discovery purposes; and
- The template lacks some M&S-specific content.

In order to close these gaps, MSG-136 has reworked and restructured the original template, resulting in the following MSG-136 service description template.

Core Information

- *Service ID* – The unique identifier of the Service, represented by a Universally Unique Identifier (UUID[1]). This ID has to be unique for all services, which are part of the system and will change with a new Version. The ID can be used to locate and reference the service. Every reference to a service should be done using the ID (required).
- *Service Name* – The name of the Service (free text). Should fit the operations of the service (required).
- *Description* – A detailed description of the service, what it is used for and what it does (free text). This description should be written in a user-friendly way so that a person new to the service understands the purpose, scope and limitations of the service (required).
- *Points of Contact (PoC)* – A list of Points of Contact. This list should contain all important PoCs responsible for the service, so if there is a problem, the user knows who to contact about it. Every PoC has the following properties (free text):
  - First name (required for every contact).
  - Last name (required for every contact).
  - Phone – May be an office or cell phone number.
  - Email.
  - Company.
  - Role – Options: developer, owner, supporter, technical authority, content responsible, other (one of the roles must be selected for every contact).

[1] https://tools.ietf.org/html/rfc4122.

    - Notes – Additional notes about the contact, could contain office times, other ways of contact, etc.

- *Lifecycle Stage* – The current state of the service. One of the following options[2] needs to be selected at all times:
    - Proposed – Business needs are identified and assessed.
    - Definition – Requirements are gathered, and design is produced.
    - Development – Specifications are developed, and service is built.
    - Verification – Service is tested and inspected.
    - Production – Service is available for use by intended customers.
    - Deprecated – Service can no longer be used by new customers.
    - Retired – Service is disposed and can no longer be used.
- *Version* – The version number of the current version of the Service. Every new version of the service will result in a new version of the description. The version number will be provided in plain text. Its format will be defined by the owner of the service (required).
- *Previous Version* – A reference to the previous version of the service. Should reference an older version by its service id and grant access to its service description. Only empty on the first description version.
- *History* – Relevant dates in the history of this service. The history contains all important actions of the service like updates, fixes, new scenarios and way more.
    - Date – The date something relevant happened (required for every history entry).
    - Type – Plain text value to enter the type of action which happened on the given date. Examples of possible actions are: created, accepted, updated, retired, used (required for every history entry).
    - Note – Contain further explanations of the event. Every event should have some notes attached to it for a better understanding of what exactly happened. Could be patch notes for updates and fixes, results for executions, explanations for new scenarios, etc. (required for every history entry).
- *Service Access*
    - Pre-Deployed.
        - Address – If service is available without prior deployment.
        - Availability – Information about availability with respect to time and resources.
        - Billing information –How to pay for the service (and how much).
        - Service agreement – Agreement between customer and provider of service (e.g., usage restrictions, number of licenses, etc.).
    - To-Be-Deployed Service.
        - Repository Location – The repository location (e.g., URL) where the service can be obtained (e.g., where Docker image is stored). This URL is required to locate the image of the service that needs to be deployed (required).
        - Billing information – How to pay for the service (and how much).

[2] Available options taken from [5] p. 15.

  - Service agreement – Agreement between customer and provider of service (e.g., usage restrictions, number of licenses, etc.).

- *References*
  - Standards and documents relevant for this service (free text).

Service Usage Information

- *Required Applications* – Applications the user needs to install/provide. Only applications which are mandatory to access or execute the service should be listed.
  - Name – The name of the application (required for every application).
  - Description – Short description what the application does and how it is installed/prepared. Should be written in user-friendly language. Could reference a description/installation guide of the application if its content is satisfying (required for every application).
  - Purpose – Reason why the application is required for using the service.
  - Source – Source where the application can be obtained (required for every application).
- *Required Skills* – Skill requirements for a user of the service (could be linked to existing skill database like for example the one from the German employment exchange).
  - Name – The name of the skill.
  - Description – A description of the skill and what needs to be accomplished to be able to fulfill it.
  - Importance – Options: mandatory, recommended, optional.

Runtime Infrastructure Requirements

- *Hardware Requirements* – The minimum required hardware to run the service properly. These requirements are meant for the server the service will be running on, not the computer used to access the service (required).
  - Processor – The minimal required processor (required).
  - Memory – The minimal required Memory Size (required).
  - Graphics Card – The minimal required Graphic Card. Many services are of non-graphical nature. In these cases, this field may be empty.
  - Other (e.g., required disc space).
- *Required Operating System* – The required operating system to run the service. This information is meant for the server the service will be running on, not the computer used to access the service (required).
  - Options: Windows, Linux, other.
  - Note: More information on specific versions is required (e.g., Windows XP, Ubuntu 10.3, etc.).
- *Network Requirements* – The minimal requirements for the users' local network connection to guarantee a stable access to the service. Should be required for services with a high-information throughput. Not necessarily required for services with nearly no throughput.
  - Minimal download speed – Given in Kbit/s.
  - Minimal upload speed – Given in Kbit/s.
  - Required ports – Ports that need to be opened for the service communication. If any ports are given, the service must be granted access to them. Not doing so can prevent a proper execution of the service.

- *Service Initialization Requirements* – A list of requirements that need to be met before the service is ready to use. These are required for every interface type provided by the service.
  - Description – A detailed, user-friendly description of the requirement that needs to be met. Should contain information about how the requirement can be fulfilled and what it will be used for (required for every initialization requirement).
  - Instruction – Provides the user with sufficient instructions to fulfill the requirement. This instruction should describe all mandatory steps and provide all necessary references (for example a schema for the database that needs to be set up) (required for every initialization requirement).
- *Required Services* – A list of services that are required for this one to be executed. Every service in this list needs to be started when this service needs to be executed.
  - Service ID – The unique identifier of the service. This value matches the Service ID (see page one) of exactly one other service and will be used to locate the required service (required for every required service).
  - Purpose – The reason this service is required.

Service Interface

- *Interfaces* – List of Interfaces which define the possibilities to access the service. The following parameters will be in place for all interface types. Some may require additional data.
  - Type – The type of the interface. Currently, HTML and REST are supported. The list of supported types can be extended by additional ones like SOAP, WMS and more in the future. Depending on the type, additional parameters may be required (e.g., an XML-schema for SOAP) or can be removed (e.g., URL for non-Web services) (required for every interface).
  - URL – The URL to reach the Web service. Should use the following scheme: <network protocol>://<host address>:<Port>/<Path> (required for every Web Service).
  - Preconditions – Conditions that need to be fulfilled before the service can be started.
  - Condition – The condition, that needs to be met (required).
  - Functions – A list of functions the Interface provides. Each function is a separate call with separate input and output. This list should contain every function which is provided by the interface.
    - Name – The name of the function. Should fit the function operations. Provides a first impression of the function's operations (required for every function).
    - Input parameters – A list of all input parameters the function requires to start the execution. Parameters can be simple textual inputs or a whole file of a specified format.
      - Name – The parameter name. The name should generally fit its purpose (required for every parameter).
      - Description – User-friendly description of the parameter. The description can for example contain the purpose of the parameter or the format it needs to have. This should help the user to give valid input to the service (required for every parameter).
      - Type – Plain text input for the parameter type. Describes the data type, examples are integer, String or Boolean.
      - Minimal Value – The minimal Value the parameter is allowed to have. If no value is given, the minimal value could have a default definition according to its data type.
      - Maximal Value – The maximal Value the parameter is allowed to have. If no value is given, the maximal value could have a default definition according to its data type.

- Default Value – The default value for the parameter.
- Required – True, if the parameter needs to be passed, false otherwise. A required parameter is mandatory for the service execution. Every other parameter is optional and could for example be used to filter or improve results (required for every parameter).

- Output – The final output of the Service.
  - Description – User-friendly description of the return value. Should explain the meaning of the output and eventual further usages. The expected results after the completion of the Service execution in a user-friendly description. There may be cases where no direct results exist for the user. In these cases, this field may be left empty (required).
  - Type – Plain text input for the parameter type. Describes the data type, examples are integer, String or Boolean.

The following listing provides the additional parameters for the available interface types:

- HTML (website): No additional parameters are required.
- REST (Web service):
  - Resource interaction type – One of the following HTTP methods to access REST resources:
    - Get – Used to access resources from the Web service. Access happens by the URL defined earlier. The <Path> part defines which resource is accessed. For example, to access a soldier of an exercise with ID “371” the <Path> could be “exercise/troops/371”.
    - Post – Used to save resources or access status changing server logic. For example, a Post call with <Path> “exercise/troops” and an input parameter “Private Ryan” adds the soldier to the resource “troops”.
    - Put – Used to create a not-yet-present resource type. For example, a Put call with path “exercise/tanks” and an input parameter, which defines the schema of a tank, would add the resource type “tanks” for exercises.
    - Delete – Used to remove resources. A call with <Path> “exercise/troops/371” would remove the soldier with ID “371” from the resources.
- Every function of a REST Web service requires an operation type which needs to be selected from this list. The operation type defines some requirements to input parameters and output.

The team also developed a machine-readable XML schema for this template.

5.2 THE UK MSAAS INFORMATION MODEL

The UK’s Architectures, Interoperability & Management of Simulations (AIMS) project is researching the medium-to-long term future as well as the shorter-term delivery options for MSaaS. The work has identified the structure of a capability that can potentially deliver the longer-term vision. It has some specific requirements in terms of user interactions, flexibility and rapid reconfiguration of capability to meet operational needs.

A conclusion of the AIMS research, based on existing SOA models developed within the IT industry as well as the large body of work on modelling and simulation delivery, is that the ‘information and data’ layer is a critical enabler for a MSaaS capability, and that a complete model for all of the information and data that needs to exist to deliver MSaaS does not currently exist. It was concluded that these information and data structures need to mirror the core Allied Framework for MSaaS’ structure as described in the MSaaS Technical Reference Architecture [1].

Based on these conclusions, AIMS has developed a model that identifies generic conceptual, software and hardware constructs that are required to exist to deliver MSaaS, defined their nature and content, and considered requirements for underlying information and metadata management (termed 'discovery' here) across the facility. The guiding principles for the MSaaS information model in the UK are that:

- The model must be sufficiently flexible to permit the UK to incorporate (within reason) data from a range of sources based on different underlying standards;
- The model must enable users to move from high-level overview down to detailed technical implementation within the same facility (although almost certainly not via the same toolsets and applications); and
- The model must be based on the Allied Framework for MSaaS principles as described in standards on SOA such as those from The Open Group or the Organization for the Advancement of Structured Information Standards (OASIS). At a high level, we see this framework approach as fundamentally different to current processes for the specification, design, development and delivery of simulations; the UK MSaaS vision is based on the flexible interactions that can only be delivered via this approach.

The UK model has been implemented as a proof-of-concept Allied Framework for MSaaS.

The information model describes the complete set of information required to define specified objects in full. This will be different for each type of object. The metadata is data that describes the simulation artefacts in a way that enables users to identify and evaluate items of interest without needing to retrieve the object itself. The current metadata model uses the same schema for all objects within the framework. This has been adequate so far, although it is possible that requirements for more sophisticated evaluation will mean that metadata will be object-specific in some cases. This will depend in part on whether the keyword model used in this work is considered appropriate for classification and specification, or whether metadata extensions are used instead.

5.3 THE UK DISCOVERY APPROACH AND THE SOA ARCHITECTURE BUILDING BLOCKS

The UK approach to MSaaS in aligned with the MSaaS Reference Architecture, which is based on SOA standards, and incorporates material from the C3 taxonomy.

The UK MSaaS approach to discovery is based on the premise that this set of functionality is required to deal with the management of all information and metadata within the facility. It, therefore, maps primarily on to the information and governance layers within the SOA reference architecture and is delivered via the five capability layers.

The MSG-136 MSaaS RA concerns itself with:

- M&S-Enabling Services, which are technical capabilities for use when engineering a simulation environment (see below) and executing a simulation in which M&S-Specific Services are brought together for the purpose of that simulation;
- M&S-Specific Services, which are technical capabilities (simulation services) that provide the synthetic representation of (real-world) objects and events; and
- M&S User Applications, which are capabilities for user-facing functionality for accessing simulation environments.

A service persists independently of its various implementations and is, therefore, represented by an implementation-independent *service description*, which consists of a *service interface* and a *service contract.*

The UK discovery approach considers a set of M&S enabling services, but also the wider framework in which these are used, the information structures required to deliver them, and the associated governance functions. In the MSG-136 service description template the cross-cutting information layer is delivered by the M&S information registry services, which provide the means to store, manage and retrieve references to authoritative information required for the execution of simulation environments.

The approach described in this section covers this functionality, but also considers what this authoritative information will need to include to implement a functioning and integrated framework. It does this by:

- Developing a set of stakeholders and use cases to define the proposed framework capability and interactions;
- Understanding the information flows required to support those use cases;
- Developing underlying information objects that represent those information flows;
- Developing metadata model to provide an example of a method to describe authoritative information; and
- Developing a registry model to provide the means to store, manage and retrieve references to authoritative information.

In general, services provide functionality via interfaces. The content, that the services consume and produce, is the information entities that the UK information model is concerned with. This is aligned with the MSG-136 service description template, addressing a different aspect of the requirements for the Allied Framework for MSaaS. Many of the concepts defined within the UK information model are identified within the MSG-136 Reference Architecture (RA), but not described in detail. The UK approach considers an aligned understanding of these objects to be a critical factor in developing a framework that can interoperate across nations.

Based on prior experience, it is considered that individual nations are likely to have their own requirements in terms of implementation, such as service provision, framework access and use of standards. In order to enable nations to implement areas of functionality specific to their own needs, while still being able to interoperate with other nations' MSaaS capability, there must be:

- A common set of underlying concepts; and
- Defined methods to map local implementations to each other via that common set of underlying concepts.

This is what the UK approach provides. It, therefore, complements the areas developed by the MSG-136 working group by contributing additional content in the information and governance layers.

5.3.1 MSaaS Information Model Structure

A key concept of the MSaaS information model is its three-layered structure, as shown in Figure 5-1:

- The **MSaaS Information Layer** contains the M&S information objects defining particular constructs within the framework needed to specify, compose, deploy and execute simulations. An information object can either be a construct in its own right (i.e., a composition or deployment object), which contains details of a particular functional element, or it can be associated with something that has a physical existence (for example, a description of a service).
- The **Metadata Layer** contains all metadata related to M&S assets. The purpose of metadata is to enable discovery functionality to be implemented, including machine-to-machine and framework-querying elements. The UK definition of discovery supports the entire specify-compose-

deploy-execute cycle; it is the full set of functions that enables users to find and reference all other objects within the framework. Metadata is a distillation of the information objects; it always contains a reference to the actual information so does not need to duplicate its entire contents, but only the elements that are needed for discovery functionality.

- The **Registry Layer** is realized using the Registry Information Model (RIM) and is a further distillation of the metadata. Its purpose is to enable implementation of the type of registry required by the UK model, which is one that does not just catalogue, but that actively maintains an up-to-date record not only of objects that exist (both real and conceptual) but also the relationships between them to enable much more sophisticated querying than would otherwise be available. The referenced repository component then contains a full copy of the metadata file and associated annotations.

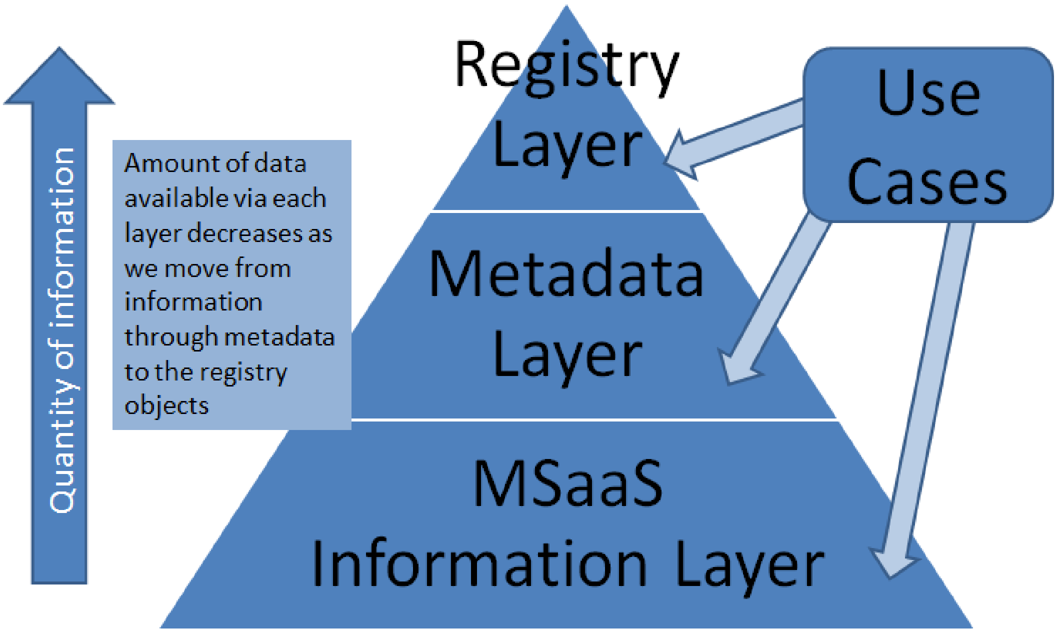

Figure 5-1: The UK MSaaS Information Model has a Number of Layers, Enabling Flexibility Within Information and Metadata, While Enabling a Single, Consistent Registry Model to be Used Within a Framework.

Together, these three layers comprise the MSaaS information model. The information model describes the complete set of information required to define specified objects in full. This will be different for each type of object. The metadata is data that describes the simulation artefacts in a way that enables users to identify and evaluate items of interest without needing to retrieve the object itself. The current metadata model uses the same schema for all objects within the framework. This has been adequate so far, although it is possible that requirements for a more sophisticated evaluation will mean that metadata will be object-specific in some cases. This will depend in part on whether the keyword model used in this work is considered appropriate for classification and specification, or whether metadata extensions are used instead.

The registry layer contains a subset of the metadata that is harvested into and stored in the registry. This is used for identification of objects within the registry, and is not the same as an object's metadata, which can be more extensive. At present, the registry model specifies the same data for all types of objects. In addition to this data, the registry stores particular, defined associations between types of object. In the future, it may be necessary to develop an enhanced model to provide more sophisticated discovery capabilities. The terminology used in this project is slightly different to that used elsewhere, primarily because the data contained within what we call the registry object and the metadata are often treated as identical. Making this distinction is what enables objects described using different metadata schema to be mapped on to the UK registry object structure in a consistent way, providing the flexibility for different groups and nations to choose how they approach metadata definition.

5.3.2 Information Objects

Within the MSaaS framework, there is a range of objects and constructs that support the discovery, composition, deployment and execution functions. An information model is one way to define these objects, which also drive metadata requirements.

An example of a model that contains these elements is the UK information layer, shown in Figure 5-2.

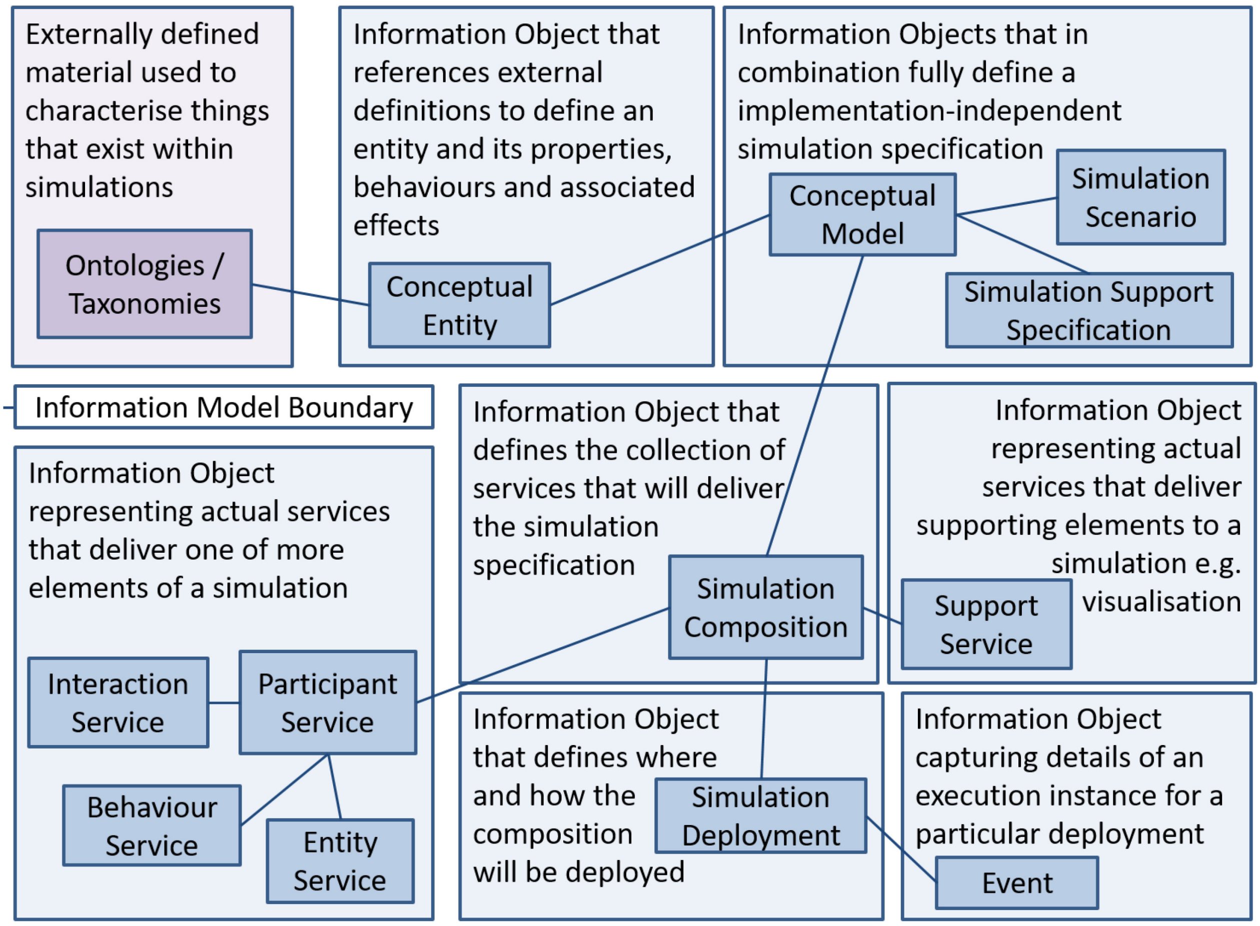

Figure 5-2: The UK MSaaS Information Model Specifies a Set of Objects, Their Properties, and the Relationships Between Them. Linked metadata and registry object models for this structure also exist. The lower levels of the model contain additional detail; this diagram is a high-level overview.

The UK approach begins with understanding who will interact with the capability and how they will do this. Use cases have been developed to describe these interactions, and, based on this, the team has considered how information needs to be structured to enable these interactions to occur. The research concluded that a very modular approach is required in order to enable the flexible use and reuse of services for a range of purposes and as part of multiple simulations.

Defined objects include:

- Objects that allow users to specify an implementation-independent set of requirements (scenario, conceptual model, simulation environment specification);
- Objects that define and describe services of various types;
- Objects that specify a core reference set of abstract entities that all simulation specifications, compositions and services can be linked to define their functionality;
- A simulation composition object, which references a set of specification objects and then provides a list of services that will be used to deliver that specification;
- A simulation deployment object, which references a simulation composition and specifies how those services will be deployed;
- Objects that define physical assets needed for deployment and their properties; and
- An object that defines an event, which is a particular execution of a defined simulation deployment.

A research outcome is that this type of structure is required in order to provide the flexibility needed to be able to use the framework to its full potential.

Definitions of key objects are provided in Table 5-1.

Table 5-1: Definition of Key Objects in the UK Information Model.

Object	Description
Simulation Specification Object(s)	The ability to compose, deploy and execute simulations in a coherent and consistent way depends on a description of the requirements for that simulation being made available. The objects identified for this purpose are the conceptual model, scenario and simulation environment specification, which are, as a group, defined as the simulation specification objects.
Simulation Composition Object	This object is an output of the composition process and describes a specific set of services that can deliver a simulation to meet a set of requirements defined in a conceptual model, scenario, and activity description. The composition provides references to a specific participant and supporting services that can be deployed and run, although it does not itself contain any deployment information. A single combination of a conceptual model, scenario and activity description can have many associated compositions.
Simulation Deployment Object	The simulation deployment object is an information object that defines all necessary data to realize a specific instance of a simulation composition including both set-up and execution. This includes activities to define how it will be deployed, carrying out pre-runtime deployment activities (e.g., setting up VMs), as well as the actual execution itself. Simulation deployments may cover persistent, non-persistent or offline services of any type defined in the MSaaS information model.
Event	An event is an instance of the execution of a particular-simulation deployment. This may correspond to a particular training exercise, mission rehearsal, demonstration, test execution or other simulation execution.

Object	Description
Service Information Object	This is the information describing an available service. The purpose of this object is to provide sufficient information that it can be fully evaluated against a requirement as specified in a conceptual model, scenario and simulation environment specification and then a suitable deployment created. The structure of this object will differ somewhat for different service types. All services will have a core set of metadata, with extensions as appropriate to their type.

5.3.3 Service Types

As a service-based framework, the active entities within the framework are services. There will be a wide range of different services available, and initial research has identified the following types as requiring different descriptive information, while also being critical to the definition and execution of simulations.

5.3.3.1 Participant Services

Services that participate in an executable simulation by representing one or more elements defined in the conceptual model and scenario are referred to as participant services. These services are complex both in terms of their function and their interfaces, and there is a requirement to break down simulation functionality in order to provide robust, well-aligned descriptions that can be understood by humans as well as used for machine-to-machine communication. In order to achieve this for simulation participants, a model of the entities that may participate in a simulation and their interactions is required. Crucially, not only the entities and their properties must be defined, but also the associations between them.

5.3.3.2 Supporting Services

These are services which run as part of a simulation but are not providing elements of the conceptual model or scenario. They may either provide middleware to facilitate the running of the simulation as a whole (e.g., and HLA RTI) or alternatively, they may provide capability to achieve a particular outcome, as in the activity description, such as data recording or visualization.

Other services will also be required; previously, two other categories of services have been defined. Although these types are not considered in this work they are described below.

5.3.3.3 Framework Administration Services

These services are provided for the purpose of managing and maintaining the simulation framework facility. As such, they do not play any direct role in composing or executing simulations; their only purpose is to ensure the availability of the framework itself. Services for discovery of assets and data are framework administration services.

5.3.3.4 Simulation Administration Services

Services that compose, deploy and execute simulations may be provided either by the simulation framework or by simulation specifiers. It is envisaged that in a mature service-based facility, many of the functions required to turn a capability requirement into a working simulation could be performed automatically. The simulation administration services interact directly with the framework administration services; the main difference is the originator of the requirement that they are addressing. Simulation administration services exist to perform functions that are required in direct response to a simulation composition request, whereas framework administration services are performed continuously in order to ensure availability of a ready-to-use framework. These types of components will rely on discovery components to provide metadata in order that they can function.

5.4 ANALYSIS OF MSG-136 AND UK APPROACH TO METADATA

5.4.1 UK MSaaS Information Structure

The UK approach to managing information and objects within the MSaaS framework makes use of a number of underlying principles developed by the UK team which are not currently aligned with the approach taken in the MSG-136 service description template. Three areas are of particular note, and these are described briefly below:

1) The UK approach begins with understanding which users will interact with the capability and how they will do this. Use cases have been developed to describe these interactions, and based on these, the UK team has considered how information needs to be structured to enable these interactions to occur. AIMS concluded that a very modular approach is required in order to enable the flexible use and reuse of services for a range of purposes and as part of multiple simulations.

   Defined objects include:

   - Objects that allow users to specify an implementation-independent set of requirements (scenario, conceptual model, activity description);
   - Objects that define and describe services (and the UK has also started to look at the differences between services that participate in a simulation at runtime and other services that provide offline functionality such as preparation and analysis);
   - Objects that specify a core reference set of entities, properties, behaviours and interactions that all simulation specifications, compositions and services can be linked to define their functionality;
   - A simulation composition object which references a set of specification objects and then provides a list of services that will be used to deliver that specification;
   - A simulation deployment object which references a simulation composition and specifies how those services will be deployed;
   - Objects that define physical resources needed for deployment and their properties; and
   - An object that defines an event, which is a particular execution of a defined simulation deployment.

   This type of structure is required in order to provide the flexibility needed to be able to use the framework to its full potential. AIMS has developed these concepts within an Enterprise Architect model, which was updated throughout the life of the project.

   The concept of having a service object is part of the AIMS model, but the team also considered what other objects need to exist and how they will interact with services through the specify-design-deploy-execute cycle.

2) Another key idea is splitting the information and data needed within the framework into three distinct layers:

   - Information objects contain all information about a particular construct within the framework needed to compose, deploy and execute simulations. An information object can either be a construct in its own right (i.e., a composition or deployment object) which contains details of a particular functional element, or it can be associated with something that has a physical existence (a service or resource, for example).
   - Metadata is data about information objects. The purpose of metadata is to enable all discovery functionality to be implemented, including machine-to-machine and framework querying elements. The UK definition of discovery supports the entire

specify-design-deploy-execute cycle; it is the full set of functions that enable users to find and reference all other objects within the framework. Metadata is a distillation of the information objects; it always contains a reference to the actual information so does not need to duplicate its entire contents, but only the elements that are needed for discovery functionality.

- Registry objects are a further distillation of the metadata and have the purpose of enabling implementation of the type of registry required by the UK model, which is one that does not just catalogue, but that actively maintains an up-to-date record not only of objects that exist (both real and conceptual) but also the relationships between then to enable much more sophisticated querying than would otherwise be available. The referenced repository component then contains a full copy of the metadata file and associated annotations.

3) The UK concept has a core requirement to relate many of the concepts used within it back to authoritative reference sources such as controlled vocabularies, ontologies or similar data sets. This is embedded into our proposed model, with the concept of abstract entities existing as information objects within the framework being an example of the use of this type of data to enable conceptual interoperability (i.e., that users and stakeholders can exchange objects knowing that they are referencing the same underlying information). We see this as essential to achieve machine-to-machine communication, but also believe that even human readability is limited if we have no way of understanding other interpretations of the concepts involved. For example, a service that provides a simulated aircraft may be useful for a range of purposes – but not if we have no way of formally specifying the behaviours, properties and level of fidelity that it provides such that others can evaluate whether it meets their own requirements.

The supplied comparison of the UK approach with the Service Description Template (SDT) points out where the SDT differs – in particular, it combines many of the elements that the UK approach treats as individual modular components together, and it also does not make the same use of controlled vocabularies, etc.

5.4.2 Results

The outcome of the mapping from the MSG-136 Service Description Template is shown in Appendix 5-1. The comparison is based on the UK approach to MSaaS delivery, which uses the ISO 19115/19119/19139 family of standards for metadata [22] and is currently documented in an Enterprise Architect model (see Figure 5-2). The UK assessment of metadata standards showed that these are suitable for use within MSaaS as they support all the Dublin Core elements, they are formally extensible, and have an XML encoding standard to support machine interpretation. As well as the Dublin Core elements, the standards support sophisticated geographic elements, data quality, distribution information, and in the case of 19119, supports detailed descriptions of Services as resources.

Appendix 5-1: MAPPING OF MSG-136 SERVICE DESCRIPTION TEMPLATE WITH UK AIMS METADATA

The following tables provide a mapping of the MSG-136 Service Description Template with UK AIMS Metadata. The mapping is structured in core information elements (Table 5A1-1), service usage information elements (Table 5A1-2), runtime infrastructure requirements (Table 5A1-3), and service interface elements (Table 5A1-4).

Table 5A1-1: Core Information Elements.

#	Field Name	Description	Required	Value Restrictions	Example Values	Sources	UK Comment
1	Service ID	The unique identifier of the Service is represented by a Universally Unique Identifier (UUID). This ID must be unique for all services, which are part of the system and will change with a new version. The ID can be used to locate and reference the service. Every reference to a service should be done using the ID.	Yes.	UUID format.		https://tools.ietf.org/html/rfc4122	UK registry object, metadata, service description and actual service (deployed or non-deployed) all have a unique ID.
2	Service Name	The name of the Service. Should fit the operations of the service.	Yes.	Free text.			Metadata includes a citation (recommended reference to be used for the object) and Service Type (type name from a registry of services). Service description template has a name.

#	Field Name	Description	Required	Value Restrictions	Example Values	Sources	UK Comment
3	Description	A detailed description of the service, what it is used for and what it does. This description should be written in a user-friendly way so that a person new to the service understands the purpose, scope and limitations of the service.	Yes.	Free text.			Maps to generic metadata field abstract.
4	Points of Contact (PoC)	A list of Points of Contact. This list should contain all important PoCs responsible for the service, so if there is a problem, the user knows who to contact about it.	Yes, at least one PoC must be provided.				Maps to generic metadata field contact of type CI_Responsible Party, which may include the following fields: • Individual Name; • Organization Name; • CI_Contact: phone, address, email; and • Role (from codelist).
5	First Name	The first name of the PoC.	Yes, for every PoC.	Free text.			See line 4.
6	Last Name	The last name of the PoC.	Yes, for every PoC.	Free text.			See line 4.
7	Phone	May be an office or cellphone number of the PoC.	No.	Free text.			See line 4.
8	Email	An email address of the PoC.	No.	Email format required.			See line 4.

#	Field Name	Description	Required	Value Restrictions	Example Values	Sources	UK Comment
9	Company	Name and short description of the company the PoC is working for. Could contain an address.	No.	Free text.			See line 4.
10	Role	Options: developer, owner, supporter, technical authority, content responsible, other.	Yes, for every PoC. One of the options must be selected.	Predefined options (see description).			See line 4.
11	Notes	Additional notes about the contact, could contain office times, other ways of contact, etc.	No.	Free text.			"Free form" information as notes are not generally used within the UK model as our goal is to make material both human and machine-readable. The gmd:CI_Citation provides an equivalent "free form" element called "otherCitationDetails". Possible use of "distributor" details could provide a better location for things like other contact details, office times etc.

#	Field Name	Description	Required	Value Restrictions	Example Values	Sources	UK Comment
12	Lifecycle Stage	The current state of the service. One of the following options must be selected at all times: • Proposed – Business needs are identified and assessed; • Definition – Requirements are gathered, and design is produced; • Development – Specifications are developed, and service is built; • Verification – Service is tested and inspected; • Production – Service is available for use by intended customers; • Deprecated – Service can no longer be used by new customers; and • Retired – Service is disposed and can no longer be used.	Yes, one of the options must be selected.	Predefined options (see description).		Available Options taken from Ref. [16], p. 15.	Not currently included in this format in either metadata or service description; to accommodate we could extend the codelist to include the specific values. The Registry Information Model could also be updated to support this lifecycle.
13	Version	The version number of the current version of the Service. Every new version of the service will result in a new version of the description. The version number will be provided in plain text.	Yes.	Free text.			Maps to generic metadata fields.

#	Field Name	Description	Required	Value Restrictions	Example Values	Sources	UK Comment
	Version (cont'd)	Its format will be defined by the owner of the service.					
14	Previous Version	A reference to the previous version of the service. Should reference an older version by its service ID and grant access to its service description. Only empty on the first description version.	For all versions succeeding the first.	Free text.			Does not map directly to metadata, but UK registry supports versioning where full metadata on the previous version is held.
15	History	Relevant dates in the history of this service. The history contains all important actions of the service like updates, fixes, new scenarios and way more.	No. History can be empty.				Functional elements overlap with other UK objects – definitely events, plus scenario/conceptual models and potentially others. We believe that events that execute a simulation needed to be stored separately and linked to relevant services and other objects, as needed. Also, scenarios, compositions and data objects all need to be defined and stored separately. Software/application history will be part of the service description template, not the metadata. Additionally, to record service updates, there is a Data Quality component that provides a formalized "lineage" including a history of what has happened to the dataset/service.

#	Field Name	Description	Required	Value Restrictions	Example Values	Sources	UK Comment
16	Date	The date something relevant happened.	Yes, for every history entry.	Date in any format.			See line 15.
17	Type	Plain text value to enter the type of action which happened on the given date.	Yes, for every history entry.	Free text.	Created, accepted, updated, retired, used.		See line 15.
18	Note	Contains further explanations of the event. Every event should have some notes attached to it for a better understanding of what exactly happened. Could be patch notes	Yes, for every history entry.	Free text.			See line 15.
19	Service Access	Information about how and under which circumstances the service can be accessed. Also contains billing information. This information differs between already deployed services and those that still need to be deployed.	Yes, either pre-deployed or to-be-deployed.				Some of the access restrictions could be managed by LegalConstraints and SecurityConstraints elements that support: • useLimitation; • accessConstraints; • useConstraints; and • otherConstraints.

#	Field Name	Description	Required	Value Restrictions	Example Values	Sources	UK Comment
19	Service Access (cont'd)						We can accommodate fees, etc., by the srv:accessProperties element. It is likely that some of this information will be stored in the UK SDT only and not included in metadata. Also, the issue of whether the service is pre-deployed, or not, is handled differently and scope overlaps with the UK deployment object to some degree.
20	Pre-Deployed						See line 19.
21	Address	The address under which the service is available.	Yes.	Free text.			See line 19.
22	Availability	Availability information with respect to time and resources.	Yes.	Free text.			"Availability" is too poorly defined for our purposes. It isn't clear here what might limit access, either. Partially overlaps with metadata accessProperties and restrictions.
23	Billing Information	Ways to pay for the service and pricing.	Yes.	Free text.			"Billing information" – we envisage that some services will have valid commercial agreements that mean they can be accessed on a pay-per-use basis and automatically billed which will need a fuller and more precise description.

#	Field Name	Description	Required	Value Restrictions	Example Values	Sources	UK Comment
24	Service Agreement	Agreement between customer and provider of service.	Yes.	Free text.	Usage restrictions, number of licenses, etc.		"Service agreement" needs to be split down further if the more transparent services are to be used in a generic way, as in UK model.
25	To-be-deployed						See line 24.
26	Repository Location	The repository location (e.g., URL) where the service can be obtained (e.g., where Docker image is stored). This URL is required to locate the image of the service that needs to be deployed.	Yes.	Free text.			See line 24.
27	Billing Information	Ways to pay for the service and pricing.	Yes.	Free text.			See line 24.
28	Service Agreement	Agreement between customer and provider of service.	Yes.	Free text.	Usage restrictions, number of licenses, etc.		See line 24.
29	References	Standards and documents relevant for this service.	No.	Free text.			Maps to generic metadata field citation.

Table 5A1-2: Service Usage Information Elements.

#	Field Name	Description	Required	Value Restrictions	Example Values	Sources	UK Comment
30	Required Applications	Applications the user needs to install/provide. Only applications, which are mandatory to access or execute the service should be listed.	No, there may be no required applications.				A service should be a standalone entity with clearly-defined functionality and interfaces. It may have dependencies, but these should really be dealt with by referencing the other service, which will also be registered. It should not be permitted to supply content to the framework without also making provision for supply of dependencies. Where dependencies cannot themselves be treated as a service, then they should be packaged with the service that requires them.
31	Name	The name of the application.	Yes, for every application.	Free text.			See line 30.
32	Description	Short description what the application does and how it is installed/prepared. Should be written in user friendly language. Could reference a description/installation guide of the application if its content is satisfying.	Yes, for every application.	Free text.			See line 30.

#	Field Name	Description	Required	Value Restrictions	Example Values	Sources	UK Comment
33	Purpose	Reason why the application is required for using the service.	No.	Free text.			See line 30.
34	Source	Source where the application can be obtained.	Yes, for every application.	Free text.			See line 30.
35	Required Skills	Skill requirements for a user of the service (could be linked to existing skill database, for example, similar to the one from the German employment exchange).	No, there may be no required skills.				This is only relevant in a system with significant manual intervention required. UK, therefore, does not have this field (although could be attached as a free text comment as temporary measure).
36	Name	The name of the skill.	Yes, for every skill.	Free text.			See line 35.
37	Description	A description of the skill and what needs to be accomplished to be able to fulfill it.	Yes, for every skill.	Free text.			See line 35.
38	Importance	Options: mandatory, recommended, optional.	Yes, for every skill.	Predefined options (see description).			See line 35.

Table 5A1-3: Runtime Infrastructure Requirements.

Note: AIMS cannot properly accommodate this section as metadata at present. It could be stored in the registry as an annotation attached to the Participant Service/Supporting Service objects, which can be retrieved on request. Comments in the table discuss to the way in which these items relate to the overall UK approach.

#	Field Name	Description	Required	Value Restrictions	Example Values	Sources	UK comment
39	Hardware Requirements	The minimal required hardware to run the service properly. These requirements are meant for the server the service will be running on, not the computer used to access the service.	Yes.				The UK SDT will contain this information. Likely that fields will map across to some degree although UK will consider machine-readability, as well as human.
40	Processor	The minimal required processor.	Yes.	Free text, could become a list of processors later on.			See line 39.
41	Memory	The minimal required memory size.	Yes.	Gigabyte (GB) value.			See line 39.
42	Graphics Card	The minimal required Graphic Card. Many services are of non-graphical nature. In these cases, this field may be empty.	No.	Free text, could become a list of graphic cards later on.			See line 39.
43	Other	Other hardware requirements that could come up.	No.	Free text.	Required disc space.		See line 39.

#	Field Name	Description	Required	Value Restrictions	Example Values	Sources	UK comment
44	Required Operating System	The required operating system to run the service. This information is meant for the server the service will be running on, not the computer used to access the service.	Yes.				The UK SDT will contain this information. Likely that fields will map across to some degree although UK will consider machine-readability, as well as human.
45	Operating System	Options: Windows, Linux, other.	Yes.	Predefined options (see description).			See line 44.
46	Note	More information on specific version that is required (e.g., Windows XP, Ubuntu 10.33, etc.)	No.				See line 44.
47	Network Requirement	The minimal requirements for the users' local network connection to guarantee a stable access to the service. Should be required for services with a high information throughput. Not necessarily required for services with nearly no throughput.	No, the service may not be accessed over a network.				The UK SDT will contain this information. Likely that fields will map across to some degree although UK will consider machine-readability, as well as human.
48	Minimal download speed		No.	Given in Kbit/s.			See line 47.
49	Minimal upload speed		No.	Given in Kbit/s.			See line 47.
50	Required ports	Ports that need to be opened for the service communication. If any ports are given the service must be granted access to them. Not doing so can prevent a proper execution of the service.	No.	Number from 0 to 65535.			See line 47.

#	Field Name	Description	Required	Value Restrictions	Example Values	Sources	UK comment
51	Service initialization requirements	A list of requirements that need to be met before the service is ready to use. These are required for every interface type provided by the service.	No, there may be no initialization steps before the service. can be used.				The UK SDT will contain this information. Likely that fields will map across to some degree although UK will consider machine-readability, as well as human.
52	Description	A detailed, user friendly description of the requirement that needs to be met. Should contain information about how the requirement can be fulfilled and what it will be used for.	Yes, for every initialization requirement.	Free text.			See line 51.
53	Instruction	Provides the user with sufficient instruction to fulfill the requirement. This instruction should describe all mandatory steps and provide all necessary references.	Yes, for every initialization requirement.	Free text.	Schema for the database that needs to be set up.		See line 51.
54	Required Services	A list of services that are required for this one to be executed. Every service in this list needs to be started when this service needs to be executed.	No, there may be no other services required.				In general, a service should be a standalone entity. If it has a dependency on another service, then that can be handled by referencing the service ID as is done here.
55	Service ID	The unique identifier of the service. This value matches the Service ID	Yes, for every	UUID format			See line 54

#	Field Name	Description	Required	Value Restrictions	Example Values	Sources	UK comment
55	Service ID (cont'd)	(see Section 5.1) of exactly one other service and will be used to locate the required service.	required service.				.
56	Purpose	The reason this service is required.	No.	Free text.			Will be included in service documentation; not relevant to information model.

Table 5A1-4: Service Interface Elements.

Note: AIMS cannot properly accommodate this section as metadata at present. It could be stored in the registry as an annotation attached to the Participant Service/Supporting Service objects, which can be retrieved on request. Comments in the table discuss the way in which these items relate to the overall UK approach.

#	Field Name	Description	Required	Value Restrictions	Example Values	Sources	UK comment
57	Interfaces	List of Interfaces which define the possibilities to access the service. The following parameters will be in place for all interface types. Some may require additional data.	Yes, at least one per service.				The UK SDT will contain this information, although not in exactly the same format. We note that the source of this service interface description appears to be based on an early version of the OWL-S ontology, although the ontological elements have been modified. UK approach anticipates that some kind of ontology for defining interfaces, and their properties, will be required. OWL-S is a good start for Web services; we need to think about how simulation interfaces (e.g., HLA) would work.

#	Field Name	Description	Required	Value Restrictions	Example Values	Sources	UK comment
58	Type	The type of the interface. Currently, HTML and REST are supported. The list of supported types can be extended by additional ones like SOAP, WMS and more in the future. Depending on the type, additional parameters may be required (e.g., an XML-schema for SOAP) or can be removed (e.g., URL for non-Web services).	Yes	Free text.			See line 57.
59	Path	The path under which the interface is located. Should be an URL path for Web services.	Yes, if it is a Web service; otherwise no.	In case of a Web service it should provide the required network protocol and the URL path. The network address and port are determined by the deployment.			See line 57.
60	Preconditions	Conditions that need to be fulfilled before the service can be started.	UUID format.				See line 57.
61	Condition	The condition, that needs to be met.	Yes, for every pre condition.	Free text.			See line 57.

#	Field Name	Description	Required	Value Restrictions	Example Values	Sources	UK comment
62	Functions	A list of functions the Interface provides. Each function is a separate call with separate input and output. This list should contain every function which is provided by the interface.	Yes, each interface should at least provide one function.				See line 57.
63	Name	The name of the function. Should fit the function operations. Provides a first impression of the functions' operations.	Yes, for every function.	Free text.			See line 57.
64	Input Parameters	A list of all input parameters the function requires to start the execution. Parameters can be simple textual inputs or a whole file of a specified format.	No, the interface could provide a function without parameter.				See line 57.
65	Name	The parameter name. The name should generally fit its purpose.	Yes, for every input parameter.	Free text.			See line 57.
66	Description	User friendly description of the parameter. The description can for example contain the purpose of the parameter or the format it needs to have. This should help the user to give valid input to the service.	Yes, for every input para-meter.	Free text.			See line 57.

#	Field Name	Description	Required	Value Restrictions	Example Values	Sources	UK comment
67	Type	Plain text input for the parameter type. Describes the data type.	No.	Free text.	Integer, String, Boolean.		See line 57.
68	Minimal Value	The minimal Value that the parameter is allowed to have. If no value is given, the minimal value could have a default definition according to its data type.	No.	free text, restricted by Type.			See line 57.
69	Maximal Value	The maximal Value that the parameter is allowed to have. If no value is given, the maximal value could have a default definition according to its data type.	No.	free text, restricted by Type.			See line 57.
70	Default Value	The default value for the parameter.	No.	Free text, restricted by Type.			See line 57.
71	Required	True, if the parameter needs to be passed, false otherwise. A required parameter is mandatory for the service execution. Every other parameter is optional and could for example be used to filter or improve results.	Yes, for every input parameter.	True or False.			See line 57.

#	Field Name	Description	Required	Value Restrictions	Example Values	Sources	UK comment
72	Output	The final output of the Service.	No, not every function results in usable output.				See line 57.
73	Description	User friendly description of the return value. Should explain the meaning of the output and eventual further usages. The expected results after the completion of the Service execution in a user-friendly description. There may be cases where no direct results exist for the user. In these cases, this field may be left empty.	Yes, for every output.	Free text.			See line 57.
74	Type	Plain text input for the parameter type. Describes the data type.	No.	Free text.	integer, String, Boolean		See line 57.
75	The following listing provides the additional parameters for the available interface types:						
76	HTML (Website)	No additional parameters are required.					See line 57.
77	REST (Web Service*)*				http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm		See line 57.

#	Field Name	Description	Required	Value Restrictions	Example Values	Sources	UK comment
78	Resource Interaction Type	One of the following HTTP methods to access REST resources: • Get – Used to access resources from the Web service. Access happens by the URL defined earlier. The <Path> part defines which resource is accessed. For example, to access a soldier of an exercise with id "371" the <Path> could be "exercise/troops/371". • Post – Used to save resources or access status changing server logic. For example, a Post call with <Path> "exercise/troops" and an input parameter "Private Ryan" adds the soldier to the resource "troops". • Put – Used to create a not yet present resource type. For example, a Put call with path "exercise/tanks" and an input parameter, which defines the schema of a tank, would add the resource type "tanks" for exercises. • Delete – Used to remove resources. A call with <Path> "exercise/troops/371" would remove the soldier with id "371" from the resources.	Yes.	Every function of a REST Web service requires an operation type, which needs to be selected from this list. The operation type defines some requirements to input parameters and output.			See line 57.

Chapter 6 – CONCLUSIONS AND RECOMMENDATIONS

MSG-136 has reviewed many different publicly available standards for the description and structuring of metadata. It has been found that none of the reviewed standards completely satisfies the requirements on metadata in an Allied Framework for MSaaS.

One approach is a new metadata schema, i.e., the Service Description Template, which has been proposed, implemented and tested successfully. Alternatively, the UK approach, comprising the combination of open standards from the geospatial domain, has been discussed and reviewed. Information modelling carried out by the UK delivers cross-cutting information and governance elements of the framework that will be required in a production system.

Experimentation has proven that it is possible to exchange metadata between the two approaches. This enables different organizations and nations to take their own approach to implementation within the guidelines and broader design, while still maintaining compatibility with other aligned systems.

It is recommended that the following activities are conducted to continue and extend this work:

- Detailed mapping of national approaches for discovery and metadata handling based on different standards and implementations of the underlying future MSaaS framework;
- Implement a programme of experimentation using mappings to enable discovery to be performed between federated registries that use different metadata schemas; and
- Further extending and standardizing the Service Description Template to emphasize its role as a minimum set of metadata required for interoperability of national MSaaS implementations.

It is also recommended that elements of the Allied Framework for MSaaS developed by different nations that support elements of the MSaaS concept are integrated further with the ABBs described in the MSaaS Reference Architecture. Further recommendations may also be found in the Option Brief Presentation of MSG ET-47, [2].

Follow-on activities, therefore, should contribute an M&S COI-specific metadata extension to the NCMS.

Chapter 7 – REFERENCES

[1] NATO STO, Modelling and Simulation as a Service – Volume 1: MSaaS Technical Reference Architecture. STO Technical Report STO-TR-MSG-136-Part-IV. To be published.

[2] NATO Modelling & Simulation Group, ET-47, Federated Approach Towards NATO Simulation Resources Management Option Brief Presentation. Accessed 11 Oct 2017.

[3] Institute of Electrical and Electronics Engineers, IEEE Std. 1730–2010, Distributed Simulation Engineering and Execution Process (DSEEP), 2010.

[4] NATO STO, Modelling and Simulation as a Service – Volume 3: MSaaS Engineering Process. STO Technical Report STO-TR-MSG-136-Part-VI. To be published.

[5] NATO, AMSP-0: Allied Framework for Modelling and Simulation as a Service (MSaaS) Governance Policies. Edition (A), Version 1. To be published.

[6] Digital Curation Centre (DCC), List of Metadata Standards, http://www.dcc.ac.uk/resources/metadata-standards/list. Accessed 09 Nov 2017.

[7] Metadata Standard. Wikipedia, https://en.wikipedia.org/wiki/Metadata_standard. Accessed 09 Nov 2017.

[8] Riley, J., Understanding Metadata. National Information Standards Organization (NISO), 2017.

[9] Dublin Core Metadata Initiative, http://dublincore.org/. Accessed 09 Nov, 2017.

[10] World Wide Web Consortium (W3C), Resource Description Format (RDF), https://www.w3.org/RDF. Accessed 09 Nov, 2017.

[11] Department of Defense (DoD), DoD Discovery Metadata Specification (DDMS), Version 5, 2014.

[12] Defense Modeling and Simulation Coordination Office (DMSCO), M&S Community of Interest Discovery Metadata Specification (MSC-DMS), Version 1.5, December 2012.

[13] Simulation Interoperability Standards Organization (SISO), Federation Engineering Agreements Template (FEAT*)*, SISO-STD-012-2013, 2013.

[14] Organization for the Advancement of Structured Information Standards (OASIS), Universal Description, Discovery, and Integration (UDDI*)*, Version 3.0.2, 19 October 2004. http://www.uddi.org/pubs/uddi-v3.0.2-20041019.htm.

[15] North Atlantic Treaty Organization (NATO), NATO Discovery Metadata Specification (NDMS), Version 1.1. North Atlantic Treaty Organization (NATO), 2006.

[16] North Atlantic Treaty Organization (NATO) Data Administration Group (NDAG), Guidance On The Use Of Metadata Element Descriptions for Use in the NATO Discovery Metadata Specification (NDMS), Version 1.1., 2005.

[17] North Atlantic Treaty Organization (NATO), NATO Core Metadata Specification (NCMS), Version 1.0, 2014.

[18] Swagger, Version 2.0, http://swagger.io/, September 2014.

[19] Office of the Director of National Intelligence, XML Data Encoding Specification for Need-to-Know Metadata, Version 9, 5 April 2013, https://www.dni.gov/index.php/about/organization/chief-information-officer/need-to-know-metadata.

[20] NATO Allied Command Transformation (ACT) C4ISR Technology and Human Factors (THF) Branch, NATO C3 Classification Taxonomy, Baseline 1.0, June 2012.

[21] Hänninen, H., Service Description Template Method for SOA Modelling – What You Need to Know Before Starting a Web Service Project. Fortex Consulting, 2005.

[22] International Organization for Standardization, ISO 19115, https://www.iso.org/obp/ui/#iso:std:iso:19115:-1:ed-1:v1, April 2014.

REPORT DOCUMENTATION PAGE

1. Recipient's Reference	2. Originator's References	3. Further Reference	4. Security Classification of Document
	STO-TR-MSG-136-Part-V AC/323(MSG-136)TP/832	ISBN 978-92-837-2158-1	PUBLIC RELEASE

5. Originator

Science and Technology Organization
North Atlantic Treaty Organization
BP 25, F-92201 Neuilly-sur-Seine Cedex, France

6. Title

Modelling and Simulation as a Service, Volume 2: MSaaS Discovery Service and Metadata

7. Presented at/Sponsored by

Developed by NATO MSG-136.

8. Author(s)/Editor(s)	9. Date
Multiple	July 2019

10. Author's/Editor's Address	11. Pages
Multiple	74

12. Distribution Statement

There are no restrictions on the distribution of this document. Information about the availability of this and other STO unclassified publications is given on the back cover.

13. Keywords/Descriptors

Cloud computing
Composability
Distributed simulation
Interoperability
Live, Virtual, Constructive (LVC) Modelling
Modelling and Simulation (M&S)
Modelling and Simulation as a Service (MSaaS)
M&S Services
NATO C3 Classification Taxonomy
Reference architecture
Service-Oriented Architecture (SOA)
Simulation
Simulation Architecture
Simulation Environments
Simulation Interoperability

14. Abstract

M&S as a Service (MSaaS) is a concept that combines service orientation and the provision of M&S applications via the as-a-service model of cloud computing to enable more composable simulation environments that can be deployed and executed on-demand. NATO MSG-136 investigated the concept of MSaaS and provided technical and organizational foundations to establish the Allied Framework for M&S as a Service within NATO and partner nations. The Allied Framework for M&S as a Service is the common approach of NATO and nations towards implementing MSaaS and is defined by the Operational Concept Document, Technical Reference Architecture, and MSaaS Governance Policies.

This document analyses different options to handle metadata and discovery in the context of MSaaS in alignment with the reference architecture. Metadata are the key element of a formalized description of user requirements and searchable descriptions of available resources. MSG-136 examined several available metadata schemes for their usability in an MSaaS context and finally in a best-of-breed approach suggested a minimum set of metadata elements necessary to enable service discovery in the Allied Framework for MSaaS.

NORTH ATLANTIC TREATY ORGANIZATION

NATO OTAN

BP 25

F-92201 NEUILLY-SUR-SEINE CEDEX • FRANCE

Télécopie 0(1)55.61.22.99 • E-mail mailbox@cso.nato.int

SCIENCE AND TECHNOLOGY ORGANIZATION

DIFFUSION DES PUBLICATIONS
STO NON CLASSIFIEES

Les publications de l'AGARD, de la RTO et de la STO peuvent parfois être obtenues auprès des centres nationaux de distribution indiqués ci-dessous. Si vous souhaitez recevoir toutes les publications de la STO, ou simplement celles qui concernent certains Panels, vous pouvez demander d'être inclus soit à titre personnel, soit au nom de votre organisation, sur la liste d'envoi.

Les publications de la STO, de la RTO et de l'AGARD sont également en vente auprès des agences de vente indiquées ci-dessous.

Les demandes de documents STO, RTO ou AGARD doivent comporter la dénomination « STO », « RTO » ou « AGARD » selon le cas, suivi du numéro de série. Des informations analogues, telles que le titre est la date de publication sont souhaitables.

Si vous souhaitez recevoir une notification électronique de la disponibilité des rapports de la STO au fur et à mesure de leur publication, vous pouvez consulter notre site Web (http://www.sto.nato.int/) et vous abonner à ce service.

CENTRES DE DIFFUSION NATIONAUX

ALLEMAGNE
Streitkräfteamt / Abteilung III
Fachinformationszentrum der Bundeswehr (FIZBw)
Gorch-Fock-Straße 7, D-53229 Bonn

BELGIQUE
Royal High Institute for Defence – KHID/IRSD/RHID
Management of Scientific & Technological Research for Defence, National STO Coordinator
Royal Military Academy – Campus Renaissance
Renaissancelaan 30, 1000 Bruxelles

BULGARIE
Ministry of Defence
Defence Institute "Prof. Tsvetan Lazarov"
"Tsvetan Lazarov" bul no.2
1592 Sofia

CANADA
DGSIST 2
Recherche et développement pour la défense Canada
60 Moodie Drive (7N-1-F20)
Ottawa, Ontario K1A 0K2

DANEMARK
Danish Acquisition and Logistics Organization (DALO)
Lautrupbjerg 1-5
2750 Ballerup

ESPAGNE
Área de Cooperación Internacional en I+D
SDGPLATIN (DGAM)
C/ Arturo Soria 289
28033 Madrid

ESTONIE
Estonian National Defence College
Centre for Applied Research
Riia str 12
Tartu 51013

ETATS-UNIS
Defense Technical Information Center
8725 John J. Kingman Road
Fort Belvoir, VA 22060-6218

FRANCE
O.N.E.R.A. (ISP)
29, Avenue de la Division Leclerc
BP 72
92322 Châtillon Cedex

GRECE (Correspondant)
Defence Industry & Research General Directorate, Research Directorate
Fakinos Base Camp, S.T.G. 1020
Holargos, Athens

HONGRIE
Hungarian Ministry of Defence
Development and Logistics Agency
P.O.B. 25
H-1885 Budapest

ITALIE
Ten Col Renato NARO
Capo servizio Gestione della Conoscenza
F. Baracca Military Airport "Comparto A"
Via di Centocelle, 301
00175, Rome

LUXEMBOURG
Voir Belgique

NORVEGE
Norwegian Defence Research Establishment
Attn: Biblioteket
P.O. Box 25
NO-2007 Kjeller

PAYS-BAS
Royal Netherlands Military Academy Library
P.O. Box 90.002
4800 PA Breda

POLOGNE
Centralna Biblioteka Wojskowa
ul. Ostrobramska 109
04-041 Warszawa

PORTUGAL
Estado Maior da Força Aérea
SDFA – Centro de Documentação
Alfragide
P-2720 Amadora

REPUBLIQUE TCHEQUE
Vojenský technický ústav s.p.
CZ Distribution Information Centre
Mladoboleslavská 944
PO Box 18
197 06 Praha 9

ROUMANIE
Romanian National Distribution Centre
Armaments Department
9-11, Drumul Taberei Street
Sector 6
061353 Bucharest

ROYAUME-UNI
Dstl Records Centre
Rm G02, ISAT F, Building 5
Dstl Porton Down
Salisbury SP4 0JQ

SLOVAQUIE
Akadémia ozbrojených síl gen. M.R. Štefánika, Distribučné a informačné stredisko STO
Demänová 393
031 06 Liptovský Mikuláš 6

SLOVENIE
Ministry of Defence
Central Registry for EU & NATO
Vojkova 55
1000 Ljubljana

TURQUIE
Milli Savunma Bakanlığı (MSB)
ARGE ve Teknoloji Dairesi Başkanlığı
06650 Bakanliklar – Ankara

AGENCES DE VENTE

The British Library Document Supply Centre
Boston Spa, Wetherby
West Yorkshire LS23 7BQ
ROYAUME-UNI

Canada Institute for Scientific and Technical Information (CISTI)
National Research Council Acquisitions
Montreal Road, Building M-55
Ottawa, Ontario K1A 0S2
CANADA

Les demandes de documents STO, RTO ou AGARD doivent comporter la dénomination « STO », « RTO » ou « AGARD » selon le cas, suivie du numéro de série (par exemple AGARD-AG-315). Des informations analogues, telles que le titre et la date de publication sont souhaitables. Des références bibliographiques complètes ainsi que des résumés des publications STO, RTO et AGARD figurent dans le « NTIS Publications Database » (http://www.ntis.gov).

NORTH ATLANTIC TREATY ORGANIZATION

BP 25
F-92201 NEUILLY-SUR-SEINE CEDEX • FRANCE
Télécopie 0(1)55.61.22.99 • E-mail mailbox@cso.nato.int

SCIENCE AND TECHNOLOGY ORGANIZATION

DISTRIBUTION OF UNCLASSIFIED STO PUBLICATIONS

AGARD, RTO & STO publications are sometimes available from the National Distribution Centres listed below. If you wish to receive all STO reports, or just those relating to one or more specific STO Panels, they may be willing to include you (or your Organisation) in their distribution.

STO, RTO and AGARD reports may also be purchased from the Sales Agencies listed below.

Requests for STO, RTO or AGARD documents should include the word 'STO', 'RTO' or 'AGARD', as appropriate, followed by the serial number. Collateral information such as title and publication date is desirable.

If you wish to receive electronic notification of STO reports as they are published, please visit our website (http://www.sto.nato.int/) from where you can register for this service.

NATIONAL DISTRIBUTION CENTRES

BELGIUM
Royal High Institute for Defence – KHID/IRSD/RHID
Management of Scientific & Technological Research for Defence, National STO Coordinator
Royal Military Academy – Campus Renaissance
Renaissancelaan 30
1000 Brussels

BULGARIA
Ministry of Defence
Defence Institute "Prof. Tsvetan Lazarov"
"Tsvetan Lazarov" bul no.2
1592 Sofia

CANADA
DSTKIM 2
Defence Research and Development Canada
60 Moodie Drive (7N-1-F20)
Ottawa, Ontario K1A 0K2

CZECH REPUBLIC
Vojenský technický ústav s.p.
CZ Distribution Information Centre
Mladoboleslavská 944
PO Box 18
197 06 Praha 9

DENMARK
Danish Acquisition and Logistics Organization (DALO)
Lautrupbjerg 1-5
2750 Ballerup

ESTONIA
Estonian National Defence College
Centre for Applied Research
Riia str 12
Tartu 51013

FRANCE
O.N.E.R.A. (ISP)
29, Avenue de la Division Leclerc – BP 72
92322 Châtillon Cedex

GERMANY
Streitkräfteamt / Abteilung III
Fachinformationszentrum der Bundeswehr (FIZBw)
Gorch-Fock-Straße 7
D-53229 Bonn

GREECE (Point of Contact)
Defence Industry & Research General Directorate, Research Directorate
Fakinos Base Camp, S.T.G. 1020
Holargos, Athens

HUNGARY
Hungarian Ministry of Defence
Development and Logistics Agency
P.O.B. 25
H-1885 Budapest

ITALY
Ten Col Renato NARO
Capo servizio Gestione della Conoscenza
F. Baracca Military Airport "Comparto A"
Via di Centocelle, 301
00175, Rome

LUXEMBOURG
See Belgium

NETHERLANDS
Royal Netherlands Military Academy Library
P.O. Box 90.002
4800 PA Breda

NORWAY
Norwegian Defence Research Establishment, Attn: Biblioteket
P.O. Box 25
NO-2007 Kjeller

POLAND
Centralna Biblioteka Wojskowa
ul. Ostrobramska 109
04-041 Warszawa

PORTUGAL
Estado Maior da Força Aérea
SDFA – Centro de Documentação
Alfragide
P-2720 Amadora

ROMANIA
Romanian National Distribution Centre
Armaments Department
9-11, Drumul Taberei Street
Sector 6
061353 Bucharest

SLOVAKIA
Akadémia ozbrojených síl gen M.R. Štefánika, Distribučné a informačné stredisko STO
Demänová 393
031 06 Liptovský Mikuláš 6

SLOVENIA
Ministry of Defence
Central Registry for EU & NATO
Vojkova 55
1000 Ljubljana

SPAIN
Área de Cooperación Internacional en I+D
SDGPLATIN (DGAM)
C/ Arturo Soria 289
28033 Madrid

TURKEY
Milli Savunma Bakanlığı (MSB)
ARGE ve Teknoloji Dairesi Başkanlığı
06650 Bakanliklar – Ankara

UNITED KINGDOM
Dstl Records Centre
Rm G02, ISAT F, Building 5
Dstl Porton Down, Salisbury SP4 0JQ

UNITED STATES
Defense Technical Information Center
8725 John J. Kingman Road
Fort Belvoir, VA 22060-6218

SALES AGENCIES

The British Library Document Supply Centre
Boston Spa, Wetherby
West Yorkshire LS23 7BQ
UNITED KINGDOM

Canada Institute for Scientific and Technical Information (CISTI)
National Research Council Acquisitions
Montreal Road, Building M-55
Ottawa, Ontario K1A 0S2
CANADA

Requests for STO, RTO or AGARD documents should include the word 'STO', 'RTO' or 'AGARD', as appropriate, followed by the serial number (for example, AGARD-AG-315). Collateral information, such as title and publication date, is desirable. Full bibliographical references and abstracts of STO, RTO and AGARD publications are given in "NTIS Publications Database" (http://www.ntis.gov).

ISBN 978-92-837-2158-1