NATO OTAN

NATO STANDARDIZATION AGENCY
AGENCE OTAN DE NORMALISATION

NSA

12 November 2010 NSA/1188(2010)-AIR/4559

STANAG 4559 AIR (EDITION 3) – NATO Standard ISR Library Interface

References:

a. AC/224-D(2010)0005 dated 23 June 2010
b. NSA/336(2007)-AIR/4559 dated 15 June 2007 (Edition 2)

1. The enclosed NATO Standardization Agreement, which has been ratified by nations under fast track procedure (Ref. a), is promulgated herewith.

2. The references listed above are to be destroyed in accordance with local document destruction procedures.

ACTION BY NATIONAL STAFFS

4. National staffs are requested to examine their ratification status of the STANAG and, if they have not already done so, advise the Defence Investment Division through their national delegation as appropriate of their intention regarding its ratification and implementation.

Cihangir AKSIT, TUR Civ
Director, NATO Standardization Agency

Enclosure:
STANAG 4559 (Edition 3)

NATO Standardization Agency – Agence OTAN de normalisation
B-1110 Brussels, Belgium Internet site: http://nsa.nato.int
E-mail: g.thibaut.@hq.nato.int – Tel 32.2.707.4288 – Fax 32.2.707.4103

STANAG 4559
Edition 3

NORTH ATLANTIC TREATY ORGANISATION
(NATO)

NATO STANDARDIZATION AGENCY
(NSA)

STANDARDIZATION AGREEMENT
(STANAG)

SUBJECT: NATO Standard ISR Library Interface

Promulgated on 12 November 2010

Cihangir AKSIT, TUR Civ
Director, NATO Standardization Agency

RECORD OF AMENDMENTS

No.	Reference/date of amendment	Date Entered	Signature

EXPLANATORY NOTES

AGREEMENT

1. This STANAG is promulgated by the Director NATO Standardization Agency under the authority vested in him by the NATO Standardization Organisation Charter.

2. No departure may be made from the agreement without informing the tasking authority in the form of a reservation. Nations may propose changes at any time to the tasking authority where they will be processed in the same manner as the original agreement.

3. Ratifying nations have agreed that national orders, manuals and instructions implementing this STANAG will include a reference to the STANAG number for purposes of identification.

RATIFICATION, IMPLEMENTATION, AND RESERVATIONS

4. Ratification, implementation and reservation details are available on request or through the NSA websites (internet http://nsa.nato.int; NATO Secure WAN http://nsa.hq.nato.int).

FEEDBACK

5. Any comments concerning this publication should be directed to NATO/NSA – Bvd Leopold III 1110 Brussels – BE.

NATO STANDARDISATION AGREEMENT (STANAG)

NATO STANDARD ISR LIBRARY INTERFACE (NSILI)

1. ANNEXES

The supporting annexes for the NATO Standard ISR Library Interface provide the technical definition and rules of compliance to support the user in developing NSILI servers and/or clients that allow participation in the network exchange of intelligence information.

A. Terms and Definitions
B. NSIL Interface Operational Concept
C. NSIL Interface Process Model
D. NSIL Interface Technical Definition
E. Data Model Attribute Types and Domains
F. Boolean Query Syntax
G. Edition 3 Metadata Model (Version ID: V2.1)
H. Metamodel of the Edition 3 Metadata Model

2. REFERENCED DOCUMENTS

The following STANAGS, ITU-T Recommendations and International Standards contain provisions, which through references in this text constitute provisions of this STANAG. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on this STANAG are encouraged to investigate the possibility of applying the most recent editions of the STANAG, Recommendations and Standards listed below. NATO maintains registers of currently valid STANAGS.

ISO 639-2	-	Codes for the representation of names of languages -- Part 2: Alpha-3 code
ISO/IEC 7498-1	-	Information technology - Open systems interconnection - Basic reference model: The basic model
ISO/IEC 11179-3	-	Information technology -- Metadata registries (MDR) -- Part 3: Registry metamodel and basic attributes
ISO/IEC 12087-5	-	Information technology - Computer graphics and image processing - Image Processing and Interchange (IPI) - Functional specification: Basic Image Interchange Format (BIIF)
ISO/IEC 14750	-	Information technology – Open distributed processing – Interface definition language (IDL)
STANAG 1059	-	Distinguishing Letters Geographic Entities for use in NATO
STANAG 4545	-	NATO Secondary Imagery Format (NSIF)
STANAG 5524	-	NATO Interoperability Standards and Profiles – ADatP-34
STANAG 5525	-	Joint C3 Information Exchange Data Model

AEDP-2		NATO ISR Interoperability Architecture
AIntP-3	-	Military Intelligence Data Exchange Concept
FIPS Pub 10-4	-	Countries, Dependencies, Areas of Special Sovereignty, and Their Principal Administrative Dependencies
GIAS	-	USIGS Geospatial and Imagery Access Services Specification (GIAS) Version 3.5.1, 6 Aug 2001
UCOS	-	USIGS Common Object Specification (UCOS) Version 1.5.1, 6 Aug 2001
CORBA/IIOP	-	OMG document formal/ 00-10, CORBA/IIOP 2.4, The Common Object Request Broker: Architecture and Specification NOTE: STANAG 4559 also allows use of CORBA 2.4 compatible versions.
Related Documents:		
ISO 1000	-	SI units and recommendations for the use of their multiples and of certain other units
STANAG 3377	-	Air Reconnaissance Intelligence Report form
STANAG 5500	-	Concept of NATO Message Text Formatting System (CONFORMETS) - ADatP-3
STANAG 7023	-	Air Reconnaissance Primary Imagery Data Standard
STANAG 7074	-	Digital Geographic Information Exchange Standard (DIGEST)
ATP 47	-	Handbook for Air Reconnaissance Tasking And Reporting
CORBAfacilities	-	CORBAfacilities: The Common Facilities Architecture, Version 4.0, Object Management Group, Framingham, MA, November, 1995
CORBAservices	-	CORBAservices: Common Object Services Specification, Revised Edition, Object Management Group, Framingham, MA, March, 1995
Object Query Service Specification	-	Object Query Service Specification: Joint Submission, Document 95.1.1, Object Management Group, Framingham, MA, December, 1993

3. AIM

The aim of this agreement is to promote interoperability for the exchange of NATO Intelligence, Surveillance and Reconnaissance (ISR) products among NATO accessible C^4I Library Systems. The

NATO Standard ISR Library Interface (NSIL Interface) is the standard interface for querying and accessing heterogeneous product libraries maintained by various nations. This standard specifies a common software interface to be implemented and exist for all NATO interoperable library systems. The interface provides electronic search and retrieval capabilities for distributed users to find products from distributed libraries in support of, but not limited to, rapid mission planning and operation, strategic analysis, and intelligent battlefield preparation. Product Libraries and the NSIL Interface are envisioned as a key technology utilized within existing Request for Information (RFI) procedures. The overall goal is for the users, who may be intelligence analysts, imagery analysts, cartographers, mission planners, simulations and operational users from NATO countries, to have timely access to distributed ISR information if Host Nation operational restrictions and security policies permit this access.

The aim of this document has extended from the exchange of only imagery to the exchange of ISR information to meet the broader interoperability requirements of the NATO ISR Interoperability Architecture (NIIA), which has undergone a similar name change. The second edition of the NSILI STANAG, the specification continued to focus on imagery.

Substantive Changes in this Third Edition of STANAG 4559 address the broader ISR library requirements. The STANAG no longer needs to specify NSIF to define the minimum data model, but the STANAG's ability to allow all standard ISR data formats is the proper state for the STANAG 4559 Edition 3, as a result, Annex E has been revised and renamed as the annex is used to describe the NSIL data types. By incorporating a new metadata model, STANAG 4559 enables web functionality; the Edition 3 Metadata Model details capabilities for the CreationMgr and UpdateMgr supporting more data management options. FTP is now *disallowed* as an exchange protocol in favor of https: or http. More generally the changes migrated into Edition 3 of STANAG 4559 by means of the metadata model break backward compatibility with Editions 1 and 2 of the standard resulting that the STANAG is not compatible with systems implementing Edition 1 or Edition 2 of STANAG 4559; details are provided in the text at appropriate points. While Edition 3 provides a new metadata model, and supporting metamodel, future versions of the STANAG may reference a registry of implementation profiles and metadata models.

4. AGREEMENT

Participating nations agree to exchange ISR products in accordance with this agreement. Nations may propose changes at any time to the control authority where they will be processed in the same manner as the original agreement. Ratifying nations have agreed that national orders, manuals and instructions implementing this STANAG will include a reference to the STANAG number for purposes of identification.

5. DEFINITIONS

The terms and definitions used in this document are listed in Annex A.

6. GENERAL SECTION

This agreement contains eight annexes with associated appendixes. Annex A lists the terms and definitions that apply to this agreement. Annex B explains the NSIL Interface concept of operations. Annex C is the process model; it contains a functional description of the interface in context of interactions between the user and the library. Annex D contains the Technical Definition of the Interface. Annex E contains the Data Model Attribute Types and Domains. Annex F describes and provides specifications for the Boolean Query Syntax (BQS). Annex G and Annex H are new elements with Edition 3 of STANAG 4559; Annex G provides the Edition 3 Metadata Model. Annex H provides the Metamodel of the Edition 3 Metadata Model, naming conventions and compliance rules.

7. DETAILS OF AGREEMENT

The NSIL Interface standardisation agreement is an application program interface (API) allowing data to be accessed in a common way over a network of heterogeneous ISR databases. This interface is consistent with two and three tier client-server architectures of the NATO C3 Technical Architecture as defined by the NATO Open Systems Working Group. The term API is to be understood in a local sense (e.g. API's between components interfaced on a user desktop) as well as in a distributed sense (e.g. interfaces from legacy or external components using an object request broker (ORB) through interface definition language (IDL) interfaces). Application of this interface assumes a network connection in which database access has already been achieved by another software entity.

8. IMPLEMENTATION OF THE AGREEMENT

This STANAG is implemented by a nation when it has issued instructions that all such equipment procured for its forces will be manufactured in accordance with the characteristics detailed in this agreement. It must be noted that implementations of STANAG 4559 Edition 3 are not backwardly compatible with the Minimum Data Model defined in Editions 1 and 2 of this STANAG.

ANNEX A.

TERMS AND DEFINITIONS

ANNEX A. TERMS AND DEFINITIONS

1. ACRONYMS

The following acronyms are used for the purpose of this agreement.

1.1.	API	Application Program Interface
1.2.	ATM	Asynchronous Transfer Mode
1.3.	BE	Basic Encyclopaedia
1.4.	BIIF	Basic Image Interchange Format. See ISO/IEC 12087-5.
1.5.	BNF	Backus-Naur Form (BNF)
1.6.	BQS	Boolean Query Syntax
1.7.	C^4I	Command, Control, Communications, Computers and Intelligence
1.8.	CCT	Computer Compatible Tape
1.9.	CORBA	Common Object Request Broker Architecture
1.10.	COTS	Commercial Off The Shelf
1.11.	DAG	Directed Acyclic Graph
1.12.	DARFI	Direct Access Related File Indicator
1.13.	DVD	Digital Versatile Disk
1.14.	FAF	Fast Access Format
1.15.	FTP	File Transfer Protocol
1.16.	GIAS	Geospatial & Imagery Access Services
1.17.	GIF	Graphic Interchange Format
1.18.	HTTP	Hypertext Transport Protocol
1.19.	IDL	Interface Definition Language
1.20.	IEC	International Electrotechnical Committee
1.21.	IIOP	Internet/Inter-ORB Protocol
1.22.	IOR	Interoperable Object Reference
1.23.	IPL	For the purposes of this standard: Image Product Library, used generically for ISR Product or Intelligence Product Library

1.24.	ISO	International Organisation for Standardisation
1.25.	ISR	Intelligence, Surveillance and Reconnaissance
1.26.	ITU	International Telecommunication Union
1.27.	JC3IEDM	Joint Consultation, Command and Control Information Exchange Data Model
1.28.	JCGISR	Joint Capability Group on Intelligence, Surveillance and Reconnaissance
1.29.	JPEG	Joint Photographic Experts Group
1.30.	LDM	Logical Data Model
1.31.	MAS	Military Agency for Standardisation
1.32.	MIDB	Modernized Integrated Data Base
1.33.	NAFAG	NATO Air Force Armaments Group
1.34.	NATO	North Atlantic Treaty Organization
1.35.	NCIS	NATO Common Interoperability Standards
1.36.	NDMS	NATO DiscoveryMetadata Specification, used in metadata model
1.37.	NGA	National Geospatial-Intelligence Agency (US)
1.38.	NIIA	NATO ISR Interoperability Architecture
1.39.	NISP	NATO Interoperability Standards and Profiles
1.40.	NNEC	NATO Network Enabled Capability
1.41.	NSIF	NATO Secondary Imagery Format
1.42.	NSIL	NATO Standard ISR Library
1.43.	NSILI	NATO Standard ISR Library Interface
1.44.	OMG	Object Management Group
1.45.	ORB	Object Request Broker
1.46.	OSI	Open Systems Interconnect model
1.47.	PNG	Portable Network Graphics-Format
1.48.	STANAG	Standardization Agreement
1.49.	UML	Unified Modeling Language
1.50.	UTC	Universal Time, Coordinated

1.51.	WGS84	World Geodetic System 1984
1.52.	XML	eXtensible Markup Language

2. TERMS AND DEFINITIONS

The following terms and definitions are used for the purpose of this agreement.

2.1. CORBA. CORBA is a major distributed middleware package. It is an open, vendor-independent architecture and infrastructure that computer applications use to work together over networks. Developed by the Object Management Group (OMG), a large, international cooperative standards body, CORBA was standardized in 1991. It has gained industry wide acceptance with many vendor implementations.

Using the standard protocol IIOP, a CORBA-based program from any vendor, on almost any computer, operating system, programming language, and network, can interoperate with a CORBA-based program from the same or another vendor, on almost any other computer, operating system, programming language, and network. CORBA enables a strict separation of interface from implementation through the use of the OMG Interface Definition Language (IDL). This allows a programming language independent specification of the interface, enabling transparent interoperability. OMG IDL is ISO/IEC Standard 14750:1999. The NSIL Interface is defined using the OMG IDL.

2.2. Data communication. The transfer of information between functional units by means of data transmission according to a protocol.

2.3. Image. A two-dimensional rectangular array of pixels indexed by row and column.

2.4. Imagery Associated Data. Data which is needed to properly interpret and render pixels; data which is used to annotate imagery such as text, graphics, etc.; data which describes the imagery such as textual reports; and data which support the exploitation of imagery.

2.5. Interface. (1) A concept involving the definition of the interconnection between two equipment items or systems. The definition includes the type, quantity, and function of the interconnecting circuits and the type, form, and content of signals to be interchanged via those circuits. Mechanical details of plugs, sockets, and pin numbers, etc., may be included within the context of the definition. (2) A shared boundary, e.g., the boundary between two subsystems or two devices. (3) A boundary or point common to two or more similar or dissimilar command and control systems, subsystems, or other entities against which or at which necessary information flow takes place. (4) A boundary or point common to two or more systems or other entities across which useful information flow takes place. (It is implied that useful information flow requires the definition of the interconnection of the systems which enables them to interoperate.) (5) The process of interrelating two or more dissimilar circuits or systems. (6) The point of interconnection between user terminal equipment and commercial communication-service facilities.

2.6. Network. (1) An interconnection of three or more communicating entities and (usually) one or more nodes. (2) A combination of passive or active electronic components that serves a given purpose.

2.7. NSIL Interface Edition 3 Metadata Model. The metadata model defined by the user community represented in Annex G of this STANAG. This Metadata Model is extensible by providing additional views.

2.8. NSIL Interface Metamodel. The UML Diagram of the Required Information Structure of a Model represented in Annex H of this STANAG provides basic interoperability and compliance to the STANAG.

2.9. Open Systems Interconnect Model. This model is defined in ISO/IEC 7498-1.

2.10. Primary Imagery. Unexploited, original imagery data that has been derived directly from a sensor. Elementary processing may have been applied at the sensor. The data stream may include auxiliary data.

2.11. Product. A product is an item of ISR data that is held in a library and can be queried and retrieved via the NSIL Interface. Each product has a reference set of metadata to enable efficient cataloging and querying of the library. A product can be composed of primary or secondary imagery, geospatial information, other ISR data and can be stored electronically or as hard-copy.

2.12. Protocol. (1) In general, a set of semantic and syntactic rules that determines the behavior of functional units in achieving communication. For example, a data link protocol is the specification of methods whereby data communication over a data link is performed in terms of the particular transmission mode, control procedures, and recovery procedures. (2) In layered communication system architecture, a formal set of procedures that are adopted to facilitate functional interoperation within the layered hierarchy. Note: Protocols may govern portions of a network, types of service, or administrative procedures.

Examples of protocols used in NSILI implementations:

The internet/inter-orb protocol (iiop) for CORBA communication
The hypertext transfer protocol (http) for direct access to products.

2.13. Secondary Imagery. Secondary Imagery is digital imagery and/or digital imagery products derived from primary imagery or from the further processing of secondary imagery

2.14. System High. Mode of operation of an information system, wherein all users having access to the system possess a security clearance or authorization, but not necessarily a need-to-know, for all data handled by the system.

ANNEX B.

NSIL INTERFACE OPERATIONAL CONCEPT

ANNEX B.

NSIL INTERFACE OPERATIONAL CONCEPT

TABLE OF CONTENTS

PARAGRAPH	TITLE	PAGE
1.	General ……	B-1
2	Key Assumptions ……	B-1
3	The NSIL Interface for ISR data. ……	B-1
4	User Interaction with the NSIL The following paragraphs describe how users will access and query ISR product libraries and the process to retrieve needed information.....……	B-3
4.1	Client (Interface) ……	B-3
4.2	Log-on Processes ……	B-3
4.3	Shared Information Access ..……	B-3
4.4.	Information Retrieval Process ..……	B-3
4.5	Indices and Their Management ..……..…	B-3
4.6	Information Submission Process…...……	B-3
4.7	Metadata ……...……	B-3
4.8	Search Function ……	B-4
4.9	Accessing Library Holdings ……	B-4
4.10	Data Delivery ……	B-4
4.11	Dissemination Management ……..……	B-4
4.12	Library Software ……..……	B-4
4.13	Equipment ……..	B-4
5	Networks and Communications ……	B-6
6	Summary ……	B-6

FIGURES	TITLE	PAGE
B –1	The NATO ISR Interoperability Architecture	B-2
B –2	NATO Standard ISR Library Interface Architecture Overview	B-5

ANNEX B. NSIL INTERFACE OPERATIONAL CONCEPT

1. GENERAL

Nations' ISR product libraries supporting NATO operations have the capability to provide intelligence, surveillance and reconnaissance information like imagery, geospatial information, imagery products, metadata and other imagery derived information in near real time. In order to exchange this information between the different battlefield nodes that require such information, a standard interface to the individual Nations' systems ISR product libraries is needed. This annex identifies the need for a NATO Standard ISR Library (NSIL) interface definition and describes how products are to be exchanged between member nations and their systems.

2. KEY ASSUMPTIONS

Interoperability between implementations of STANAG 4559 is dependent on some matters of common-ness and willingness to comply with basic level rules of engagement. These key assumptions provide the basic premises of interoperability

2.1. Hard copy archives containing products like imagery and geospatial data will continue to exist. If nations choose to archive ISR products (thereby making them offline) they should provide a digital index mechanism to identify a product.
2.2. Users will have capable tools to display, process, and exploit products, for example special viewers to display digital imagery and geospatial information.
2.3. Regardless of the underlying communications medium, the end-to-end communications system must support the Open Systems Interconnection (OSI) protocol architecture.
2.4. The NSIL interface assumes that servers and clients are installed on System High networks with network and workstation protections appropriate to the classification level of the network. It is assumed that all information loaded into an NSIL library is authorized for access by all users authorized for the network. Neither NSIL libraries nor clients add additional protections beyond that provided by the network and workstation security protections of the operational network.
2.5. Discussion of libraries in this document is confined to libraries with ISR data that have been made available on an NSIL network. Similarly the use of the term products generically referenced in this document is limited to ISR data that can include imagery, both still and motion, moving target indicator data, geospatial information and electronic intelligence.

3. THE NSIL INTERFACE FOR ISR DATA.

The NATO Air Force Armaments Group (NAFAG)/ Joint Capability Group on ISR (JCGISR) is tasked with the identification and development of standardized Intelligence data forms and the means to interoperable exchange of these data. The NAFAG has adopted the AEDP-2: NATO ISR Interoperability Architecture, to define and control ISR products for NATO coalition operations. The NSIL Interface facilitates the discovery and retrieval segment of the NIIA, providing the means to share ISR data within an NSIL network. Figure B- 1 presents the NIIA concept and its functionality in the suite of ISR Standardization Agreements.

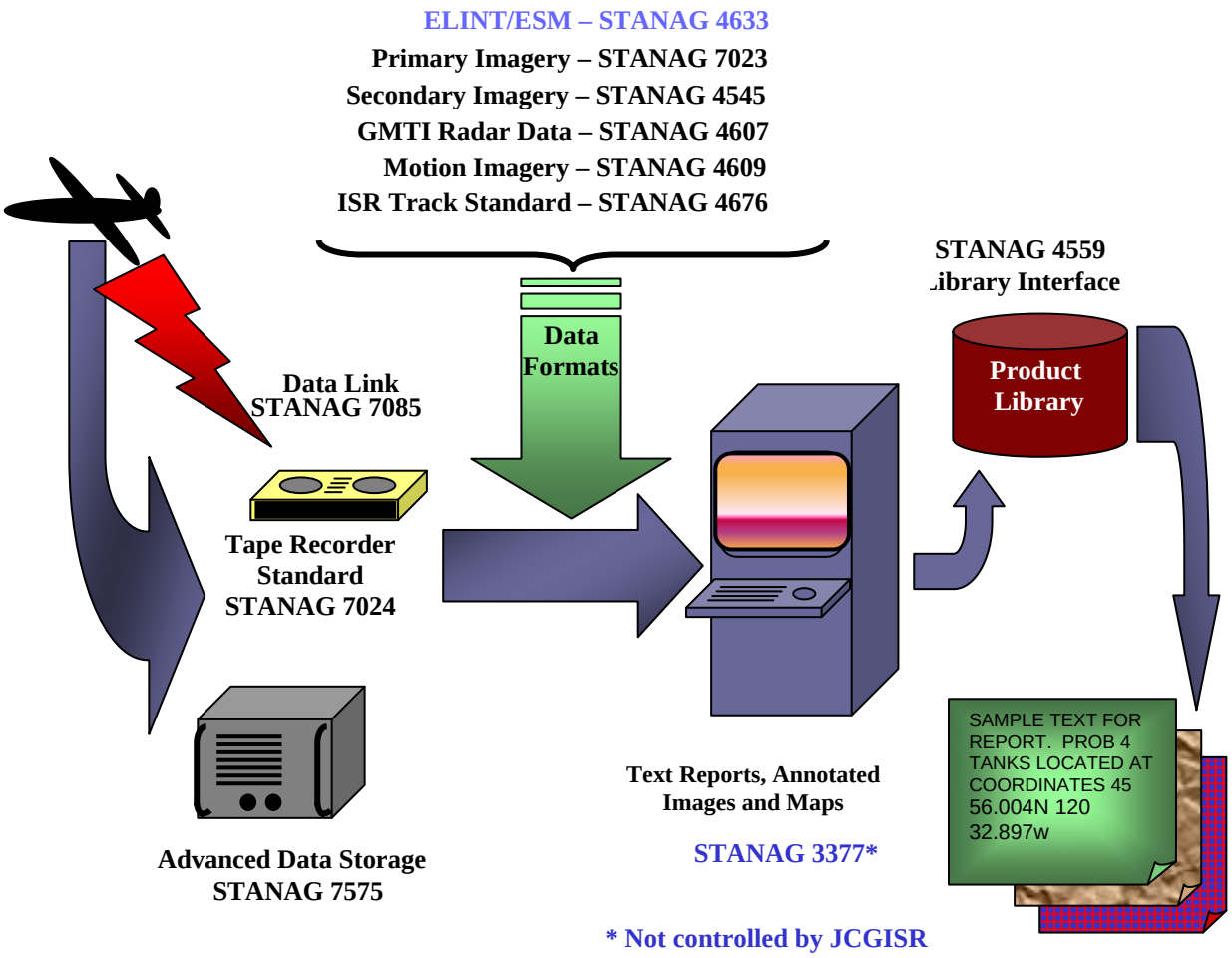

Figure B-1 The NATO ISR Interoperability Architecture

4 USER INTERACTION WITH THE NSIL

The following paragraphs describe how users will access and query ISR product libraries and the process to retrieve needed information.

4.1 Client (Interface). All end users and producers will have access to an ISR library through a client application. This client is a user interface between a library and NATO users. This client will interact with a library through the NSIL Interface. This interface allows the client to use input from the user to search the library holdings, order products, and have digital data transmitted to the user. Users at all echelons will employ client software as a means of accessing library components. Client software may be developed and implemented independently by each NATO Nation, yet by implementing the NSIL Interface interoperability among NATO nation libraries will be achieved.

4.2 Log-on Processes. Once users log-on to a library, they do not need to re-establish their identity to access the data within the library component. The library management will make this determination. Management applications are able to recognize users on the basis of log-on information and will grant access from authorized user retrieval profiles for which they have permission.

4.3 Shared Information Access. Shared information access has the same controls as access to the appropriate network. Individual organizations control user access to their library.

4.4 Information Retrieval Process. Users retrieve information from a library by submitting a user-generated query. To identify and locate library information, users perform searches via queries of metadata contained within library indices. Once the information is located, users will review metadata and preview the information before requesting product delivery.

4.5 Indices and Their Management. Each image library component has an index. Because of the magnitude, diversity and distributed nature of library holdings, users will access each nation's image library through a standard system interface that will allow users to interact with the indices in each system in a common manner. Using these indices, users can identify and locate information, and library administrators can manage the distribution, access and deletion of library holdings efficiently to meet user's needs. Users request information from a library by an ad hoc query request to a specific library component or to all library components. A query is a statement requesting the retrieval of metadata information from an image library.

4.6 Information Submission Process. Products undergo exploitation to provide value-added information to the end user. After this value-added process, the analyst or cartographer places the intelligence products in the appropriate library component. Each NATO nation is responsible for classifying, disseminating and updating its respective products.

4.7 Metadata. Because of the magnitude, diversity and distributed nature of library holdings, users rely on library indices to catalog and provide the location of imagery, geospatial information and other ISR products. These library indices present metadata in a format that enables users to query, link and access library holdings efficiently. Users may make an ad hoc query request to search the metadata index - or catalog - on a subset of user specified metadata attributes. Various metadata fields in the index are mandatory and vary depending on data type. This maximizes the usefulness of the product library, since queries require this metadata to ensure the user can find the products they need. All products stored in an image library component must have a minimum set of metadata associated with them as prescribed in the minimum layer in the Edition 3 Metadata Model, Annex G of this STANAG. The available metadata from products will populate index fields that sufficiently identify items for a later query, access and use. For other ISR products the NSIL Interface allows cataloging and discovery of individual sets of metadata that are consistent with the GIAS specification to be specified in a future edition of this STANAG.

4.8 Search Function. From a user's perspective the principle role of a product library (IPL), generally holding imagery, ISR data, or any form of searchable intelligence data, is to provide a search function that provides the user visibility, within access restrictions, to all NATO ISR library product information. The NSIL Interface provides a standard "bridge" between the user interface that supports this search and display of library product information, and the library itself. Multiple users can simultaneously reference or execute identical queries. After reviewing the search results, selected library holdings may be requested for delivery.

4.9 Accessing Library Holdings. Users create ad hoc queries and send them to the library, through the NSIL Interface. The results are returned from the library through the NSIL Interface. Users then review the returned query results listing and request products for delivery. Single items or multiple items from the listing can be displayed. If the requested "hit" is from the hardcopy index, the requested item can either be digitized and sent or reproduced and physically delivered to the user. The users will be notified of when they can expect an order to be filled. Users are able to review the available metadata from the query results listing before requesting delivery of an item to the workstation. They may also be able to review files that are associated with the main product. For example, in the case of imagery products it may be possible to select a reduced resolution (thumbnail) view of the imagery to determine if the item satisfies the user's need. A user may opt to view such related files because a thumbnail takes less time than the request and delivery of an entire file and allows the user to glance quickly at the item.

4.10 Data Delivery. Users may then request delivery of products from the library. After information has been accessed and delivered, the user can request additional selections from remaining portions of the listing. The listing from a search identifies user-accessible information holdings and will indicate whether this information is immediately accessible or whether a delay will occur. System or network managers restricting access to components for communications or loading reasons may cause such delays. For data delivery, the data will be disseminated either by electronic means or via manual delivery methods specified by the user.

4.11 Dissemination Management. NATO standards will be used when available for data transmitted between and among the various library components and users. Communications capacity and loading between the sites is a determining factor for transmission time. The individual organizations are responsible for access to global communications and internal communications. Information will be disseminated either electronically or manually.

4.12 Library Software. Individual NATO members are responsible for developing and updating all of their IPL software at the appropriate point within their architecture. Commonality and interoperability among and between the various library components is achieved through a NATO standard interface. This interface provides an access façade to each Nation's internal library software.

4.13 Equipment. The hardware and software suites necessary to support the library components are entirely determined by individual NATO nations, based on their own needs and requirements. In general, the suites will include central processing units (CPUs), client and server devices, storage devices, input/output devices, connectors, communications interfaces and other equipment needed to support the library components within the site infrastructure. This equipment is independent of the NSIL Interface.

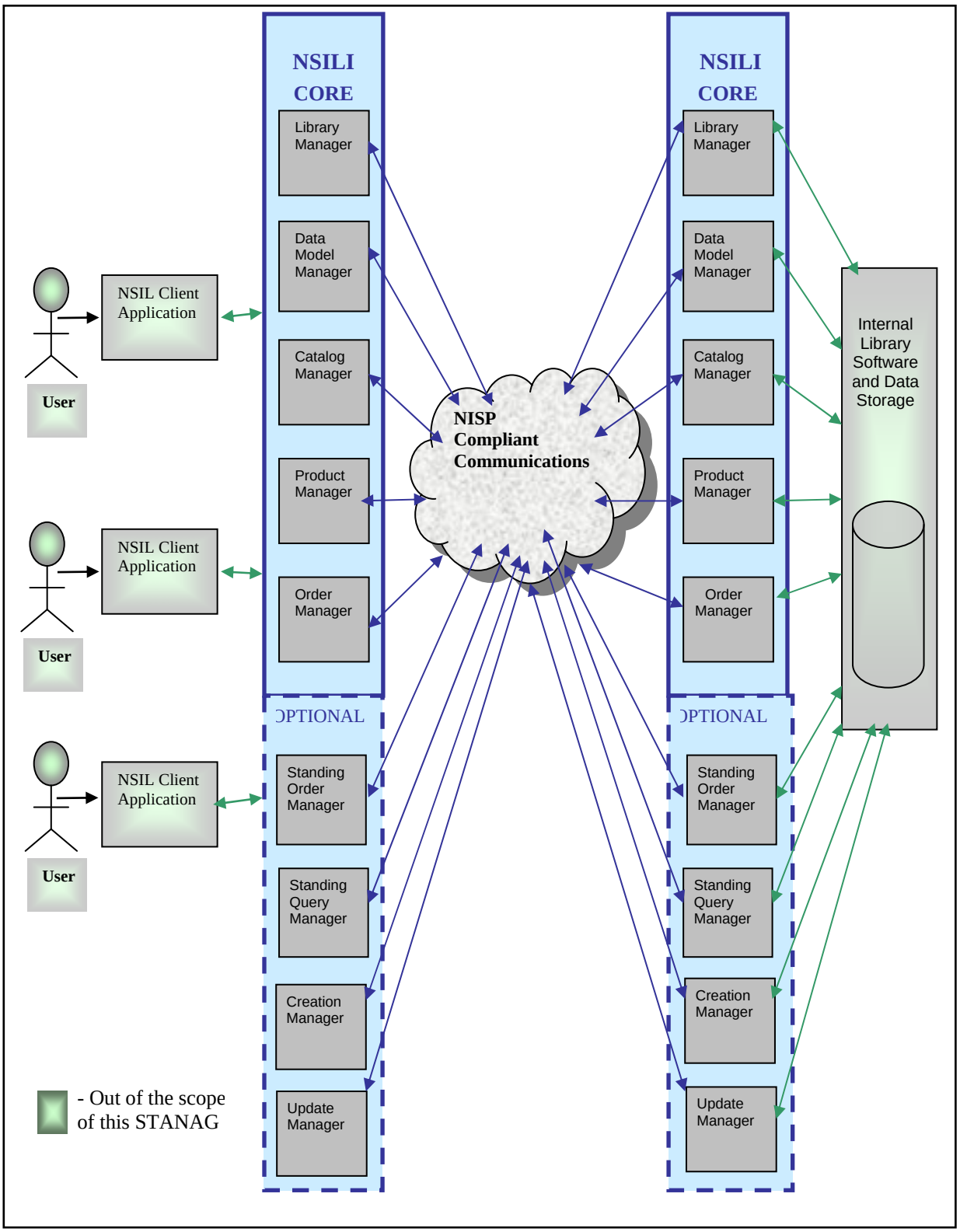

Figure B- 2 NATO Standard ISR Library Interface Architecture Overview

5. Networks and Communications.

Per the key assumptions, the NSIL Interface will function over communications networks that follow OSI protocol architecture. The highly complex NATO military communications environment involves many nations, each with varying military operational concepts and procurement cycles. The overriding requirement in this environment is interoperability among a diverse array of military C4 systems. NATO has adopted a standardization policy to improve interoperability among military systems in the form of the NATO Interoperability Standards Profile (NISP). The NISP provides the necessary guidance and technical components to support project implementations and transition to a NATO Network Enabled Capability (NNEC). The NISP is defined in ADatP-34, NATO Interoperability Standards and Profiles: cover lettered by STANAG 5524, which catalogues C3 Standards usable to NATO. The NISP strategy has been developed to achieve interoperability, maximize the exploitation of COTS and reduce the proliferation of non-standard systems. All future communications and information systems used in NATO will conform to these standards.

6. **Summary**.

In conclusion, the NSIL Interface will provide greater interoperability within NATO to efficiently share ISR Products. The NSIL Interface is a substantial tool that, when implemented, will greatly increase NATO theater intelligence sharing capability and enhance theater operations

.

ANNEX C.

NSIL INTERFACE PROCESS MODEL

ANNEX C.

NSIL INTERFACE PROCESS MODEL

TABLE OF CONTENTS

PARAGRAPH	TITLE	PAGE
1	Introduction.....	C-1
1.1	Purpose ……	C-1
1.2.	Scope ……	C-1
2	NSIL Interface Functional Model.....	C-1
2.1.	Terminology.....	C-1
2.2	Processes.....	C-2
2.3	The Login Process.....	C-2
2.4	The Discovery Process.....	C-2
2.5	The Query Process.....	C-2
2.6	The Order Process.....	C-4
2.7	The Subscription Process……	C-5
2.8	General Utility Processes.....	C-6
3	Non-Functional Considerations.....	C-7
3.1	Communications.....	C-7
3.2	Product Availability	C-7
3.3	Product Identification.....	C-7
3.4	Multiple libraries.....	C-7
3.5	Non-homogeneous Metadata.....	C-8
3.6	Extensibility.....	C-10

FIGURE	TITLE	PAGE
Figure C- 1	The Query Process.....	C-3
Figure C- 2	The Order Process	C-5
Figure C- 3	The Subscription Process	C-6
Figure C- 4	Possible Architectures.....	C-9

ANNEX C. NSIL INTERFACE PROCESS MODEL

1. INTRODUCTION

1.1. Purpose.

This document presents a model of how the NSIL Interface will function in the context of interactions between library and user. It describes the nature of the interactions that are enabled by the interface and discusses the non-functional issues that affect the interpretation of the interface. This document sets the context for understanding the interface specification in terms of both implementation and operation.

1.2. Scope.

The scope of the NSIL Interface is evolving to interface with multiple Intelligence Libraries. As specifications for ISR Libraries are defined, they will be documented for inclusion in subsequent editions of this STANAG. The Subscription Process is also described within this Annex to support registration of standing orders and queries allowing more comprehensive NSIL Interface support to ISR Product Libraries and users. As in this edition of STANAG 4559 metadata models defined for data discovery will be captured in informative annexes incorporated into the STANAG. The existing interface description does not require encapsulation of interactions as formal transactions. Error-handling mechanisms are not described in any detail.

2. NSIL INTERFACE FUNCTIONAL MODEL

2.1. Terminology.

Before considering the interaction processes, it is necessary to explain some terminology that will be used in the descriptions.

2.1.1. User, Client and IPL. For the purposes of this discussion the User is considered to be the person invoking data request interactions with the library. The Client is the application (software) that allows the User to interact with the library by providing a mechanism for translating the User's requests into the correct form for transmission to the library and for translating the responses from the library into a suitable form for presentation to the User. The IPL is the product library implementation that is responsible for executing the request. Note that the user-interface of the Client is outside the scope of the NSIL Interface and is implementation dependent.

2.1.2. Interface Manager. The concept of an "Interface Manager" is central to the description of the NSIL Interface functionality. An Interface Manager acts as an intermediary between Client and IPL, receiving requests from the Client and interacting with the IPL to fulfill those requests. It also stores intermediate results from the IPL before passing them on to the Client. There are several types of interface manager, each responsible for a sub-set of the available services, for example performing queries and query related tasks. The client chooses an appropriate interface manager for the task at hand. The NSIL Interface standard specifies the services that are provided by the Interface Managers and the protocols by which clients can request those services. The mechanism by which the Interface Managers provide the requested service is implementation dependent and does not affect the interface between Client and Interface Manager.

2.1.3. Broker. A Library Broker is an application that allows a User to access multiple libraries without the need for multiple Clients. Interaction between a Broker and an IPL uses the NSIL Interface. The IPL will provide the same services and respond in the

same manner whether a Client or a Broker is accessing it, indeed the IPL need not be aware of the type of application requesting its services. The terms Client and Broker are thus interchangeable as far as the IPL is concerned. As is the case for a Client application, the Broker's user interface is implementation dependent. A discussion of different architectures for libraries and brokers is held in paragraph 3.4 of this Annex.

2.2. Processes.

The interactions between user and library are grouped into four general processes --Login, Discovery, Query, and Order -- along with other general utility processes. The processes are illustrated with process diagrams, which show some of the typical uses of the interface. These generic processes are used to describe the capabilities of the NSIL Interface at a conceptual level and do not necessarily represent the exact sequence of actions that would be required by the interface during operation. Similarly, the names of actions shown are not intended to match the operation names in the NSIL Interface Technical Specification [Annex D].

2.3. The Login Process.

There are two primary activities involved in initiating a session of interaction with an ISR product library, collectively termed the Login Process. These activities must be accomplished prior to accessing the NSIL Interface.

The first action is to create a network connection to the IPL. This action is performed prior to the interactions covered by the NSIL Interface and will be dependent on the security restrictions and the communications and network technology of the day. From the user's perspective this will most likely be no more complex than entering a library name or IP address.

The second process involves logging on to the library and user validation. This process involves interactions with the outside layer of the library interface, prior to the involvement of any of the Interface Managers described in section 2.3. of Annex D. It is this outside layer that provides access to the Interface Managers upon receipt of a valid request.

2.4. The Discovery Process.

The NSIL Interface provides services to allow the Client to discover the metadata model, along with other descriptive parameters. The Client may use these services at the beginning of a session in order to adopt the configuration necessary for successfully interacting with the IPL, if this information has not already been cached. The metadata model represents such things as product formats supported and metadata available for performing queries. Examples of descriptive parameters include the IPL name and the version number of the interface it supports. This process of discovering the IPL's data model is a key element of the NSIL Interface definition since it allows for great flexibility in the internal structure of the IPL. A core set of mandatory, standard definition metadata will be supported by all IPLs, which will allow clients to construct queries without first having to discover the data model in use. Optional metadata can only be accessed via the discovery mechanism. The core set will allow brokering applications to successfully query multiple product libraries. The library implementer must map the metadata of:

- the NSIL Interface Metadata Model as prescribed in the Metamodel in Annex H, and
- metadata that is required to map to their physical database schema.

2.5. The Query Process.

2.5.1. The query process is shown in Figure C-1. This diagram represents a potentially iterative process whereby results from a previous query influence subsequent queries, though each query is a standalone interaction. For the query process the interface

functions very much like an Internet search engine, in that a query is sent, and the results come back according to a request profile. The user will then normally either request some further results, or send a new query.

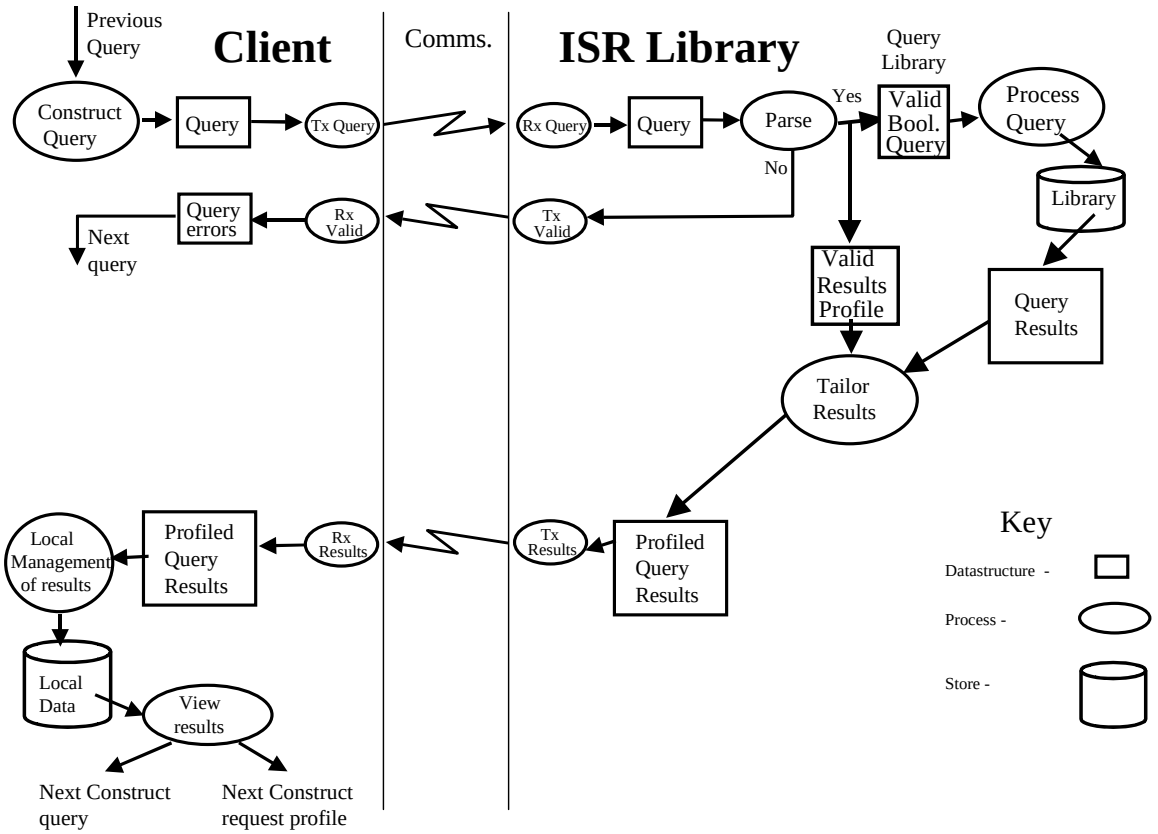

Figure C- 1 The Query Process

2.5.2. The normal information flow shown by the diagram begins with a query being constructed and transmitted to the Interface Manager at the library, where it is parsed. Both user and library will have an interest in the query being syntactically correct. The user checks are assumed to occur prior to transmitting the query and will use knowledge of the library parameters “discovered” during the login process. The syntax of the query is the Boolean Query Syntax (BQS) as defined in Annex F of this document. A Results Profile, comprising a list of desired attributes to be returned including sorting preferences, is sent along with the query. The library checks are performed by the Interface Manager and the Client will be notified if the query is invalid. A validated query will be allocated a reference that will be made known to the Client. Given a syntactically valid query, the system performs the search for matching items in the library at its disposal. Upon completion of the search, the system produces query results, which are sent to the Interface Manager, which stores them ready to send according to the Results Profile. The Profiled Query Results are tailored to include only the attributes that are in the Results Profile

2.5.3. The profiled query results are then sent back to the Client where they can be organized and managed as required, according to the functionality provided by the Client implementation. The user can then view the results and decide how to continue.

The query process can be executed iteratively until the user has identified a set of products that they wish to order, at which time the Order process is invoked.

2.5.4. The NSIL Interface also provides the capability of requesting single or multiple associated files (for example, a thumbnail of an image), which is intended to be viewed before submitting an order.

2.5.5. Notes.

2.5.5.1. The user interface for constructing valid queries is a responsibility of the Client application and so is not considered here.

2.5.5.2. A previous, specialized query has the potential to be processed by the Interface Manager or the client application without performing a new search in the IPL. Information Management issues such as these are implementation dependent and are left open to the designer of the individual library system.

2.5.5.3. This style of interface assumes sufficient availability, in the sense of expected bandwidth, from the library or broker accessed by the Client to provide reasonable response times. In keeping with search engine technology, a broker could keep a compiled database of all libraries' metadata.

2.6. The Order Process.

2.6.1. The Order Process is shown in Figure C- 2. It is similar in many ways to the query process but the Order data structure is different from the Query data structure. An Order will specify a set of product identifiers that uniquely define the required products. These product identifiers are returned as part of the earlier Query process. The nature of the product identifiers is discussed in section 3.3. of this Annex.

2.6.2. To request delivery of a product or set of products, an Order is constructed by the user and submitted to the Interface Manager where it is parsed. As with queries, both user and library will have an interest in the Order being syntactically correct. It is assumed that the Client will perform some checking prior to transmitting the Order. To assist further, the Interface Manager provides an operation to "validate" an order, in the sense of syntax checking, prior to submitting it for completion. When submitting an order it is also validated by the Interface Manager and the client will be notified if the order is invalid. Given a syntactically valid order, the IPL will allocate an OrderRequest for the Client. The Client can request the IPL to fulfill the order using this OrderRequest.

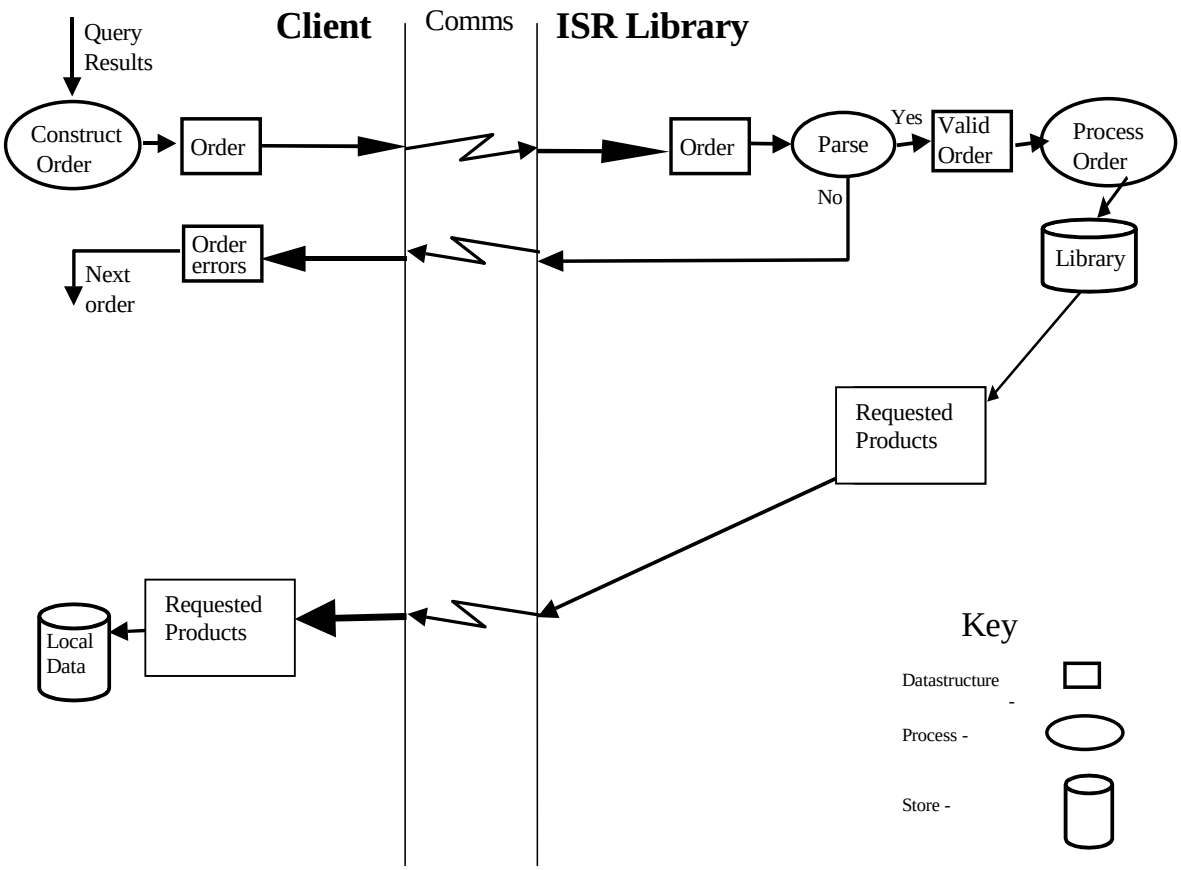

Figure C- 2 The Order Process

2.6.3. In addition to product identifiers, as part of the order the user specifies (for each product) the desired product format including data format and compression, a delivery destination and delivery method (e-mail or physical), details of any alterations to be made or tailoring to be done prior to delivery, delivery priority and, for physical deliveries, the required media type. Several aspects of creating the order could be automated by the Client application, perhaps using a standard profile. The Client will be notified if the order cannot be filled as requested or if any part of the Order is invalid.

2.6.4. Products may also be retrieved using a metadata field that provides a handle for direct access to the product, such as a URL for https transfer. Product retrieval via the direct access handle does not allow any request for product alteration, compression and so on.

2.7. The Subscription Process

2.7.1. The Subscription process is shown in Figure C-3. It is similar to the Query process, because a Client needs to send a query to the library. The library will select the products that match the query. In addition to the Query process the library will continue to monitor new products being added to the library and add those products that match the query to the query results.

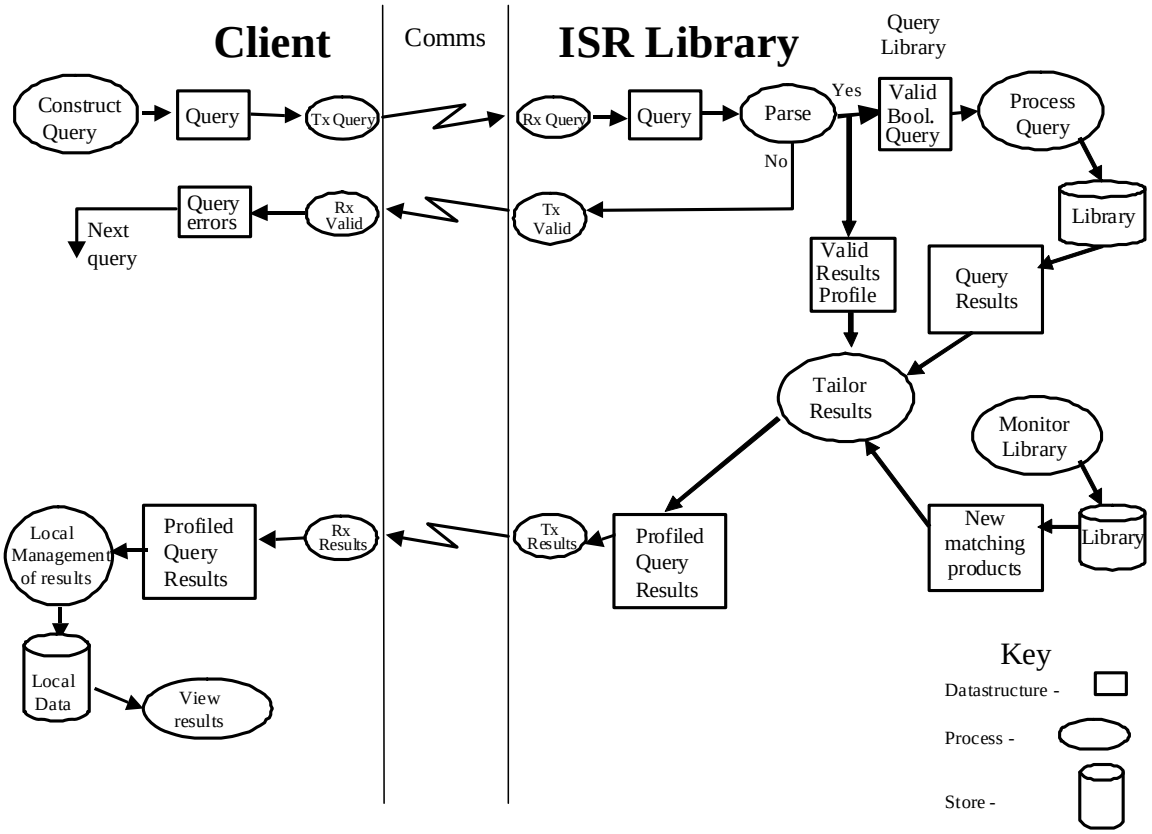

Figure C- 3 The Subscription Process

2.7.2. The query creation is the same as described for the Query process. The syntax of the query is the Boolean Query Syntax (BQS). The Results Profile provided by the client is handled in the same way as for the Query process.

2.7.3. In addition to the basic query parameters, the client provides a query lifespan. This defines when the monitoring of new products starts and ends, and at what interval rate are new matching products reported.

2.7.4. The library can directly inform a client of new matching products by using a callback. Alternatively the client can poll the server.

2.7.5. The client can subscribe to only the metadata, similar to the Query process. Figure C-3 shows this process.

2.7.6. The client can alternatively subscribe to the actual products, similar to the Order process. The client can specify the delivery method (e-mail, HTTP, HTTPS, physical), product format and other delivery details.

2.8. General Utility Processes.

2.8.1. Status. The NSIL Interface allows for enquiries regarding the status of an outstanding query or order. The status of a query or order for which the reference is known may be determined, or alternatively a list of all the outstanding queries or orders may be requested and their status queried in turn. The status request is submitted via

an Interface Manager and a status data structure is returned. Status information will include the condition of a request and an estimate for delivery time for orders.

2.8.2. Cancellation. An operation is provided to allow processing of a query or order to be terminated. The cancellation request is submitted via an Interface Manager.

2.8.3. Query and Order Timeouts. Each query or order request being handled by an Interface Manager has a limited lifetime. A client can determine what requests are active, set a default timeout for requests or modify the lifetime of a specific request.

2.8.4. Exit/Logout process. The Exit/Logout process is implementation dependent and therefore not a concern for the NSIL Interface specification.

3. NON-FUNCTIONAL CONSIDERATIONS

There are several important considerations that affect the interpretation of the NSIL Interface specification. In order to achieve interoperability the specification must be interpreted consistently by all implementations. The purpose of this section is to assist the interpretation by describing the context in which the interface specification must be viewed.

3.1. Communications.

3.1.1. Bandwidth. The bandwidth available to the user from the IPL will vary according to the location and environment of use. Since The NSIL Interface is based in the context of NATO operational use, it must be able to cope with very low bandwidth connections, but be able to take advantage of high-bandwidth links as available.

3.1.2. Continuity and Availability. The transactions between user and IPL should be designed to be as robust as possible against discontinuities of connection, but should be able to support continuous connection as transparently as possible. This raises issues with respect to the nature of the interface and the services it allows. The NSIL Interface allows product delivery to be achieved by HTTP, e-mail and physical delivery, providing the capability for an Internet-like response if the network between user and library can support it, but also allowing graceful degradation to slower response times resulting from limitations in bandwidth and the high demands placed on networks by large-scale imagery. HTTPS is supported optionally with the use of untrusted server certificates to establish a communications channel. Beginning in Edition 3 of STANAG 4559, FTP is specifically disallowed as an exchange protocol.

3.2. Product availability.

3.2.1. Format Conversions. Format conversions or the development of data models are necessary as NSIL Interface compliant libraries support order and export of a wider range of intelligence product formats including STANAG 4545 for imagery.

3.2.2. Legacy Technologies. It should be possible to manage product request and delivery with legacy technologies such as hardcopy prints and reports, as well as archive softcopy. The NSIL Interface allows clients to request an estimate of the delay required to make off-line products available for retrieval.

3.3. Product Identification.

3.3.1. Each product in an IPL has an associated product reference or Unique Product Identification (UID::Product) which is unique within that IPL. This does not guarantee that duplicate products will have the same product reference, i.e. a product may appear more than once on a server, but a product reference will only ever refer to one product.

However, a combination of attributes in the core metadata set will allow client applications to differentiate between products or identify duplicate products. Important instructions and further guidance on product reference usage appears in section 4 UID::PRODUCT USAGE GUIDANCE of Appendix 1 to Annex D.

3.4. Multiple libraries.

3.4.1. It is assumed that there will be a need for accessing multiple libraries with a single application. This will almost certainly result in the introduction of “library brokers” which will enable a set of libraries to be searched with a single user request. This standard is only concerned with the interface to a single IPL, i.e. the NSIL Interface specifies the minimum set of services that an IPL must make available and the protocol by which those services are requested. An IPL will respond to those requests in the same manner regardless of the nature of the entity (client or broker) accessing the IPL. Coping with the problems of multiple library access is a matter for the broker implementer. However, it is necessary to consider multiple library access in order to ensure that the NSIL Interface provides a sufficient level of functionality to allow a broker to resolve those issues and successfully achieve multiple library access.

3.4.2. Figure C- 4 Possible Architectures shows three examples of how multiple libraries may be accessed. In part a), the Client provides a means for user logon to individual libraries, effectively as an individual session for each. Each session is supported by the NSIL Interface. In part b), the Client acts as broker managing login, sending requests to the individual libraries, and collecting the partial results from them. Each Client implementation would be different, and the interaction between user and broker would be non-standard. The NSIL Interface supports the broker-to-library interaction. In part c), the Client accesses a “national broker” which has immediate access to its own national IPL, and can access other national libraries. Again, each national broker implementation would be different and the national broker-to-library interaction is supported by the NSIL Interface.

3.4.3. In general, before any given client establishes communication with any given library, administrative actions are required to configure the communication. Clients with access to multiple libraries may consider whether it is appropriate to make clear to the user which libraries are being accessed.

3.4.4. Duplication of search results is potentially a problem when searching single or multiple libraries, i.e. a search may return multiple “hits” that actually refer to the same image from one or more IPLs. Clients may attempt to utilize information available in the core metadata set to distinguish duplicate products but this is no guarantee that the products are actually dissimilar.

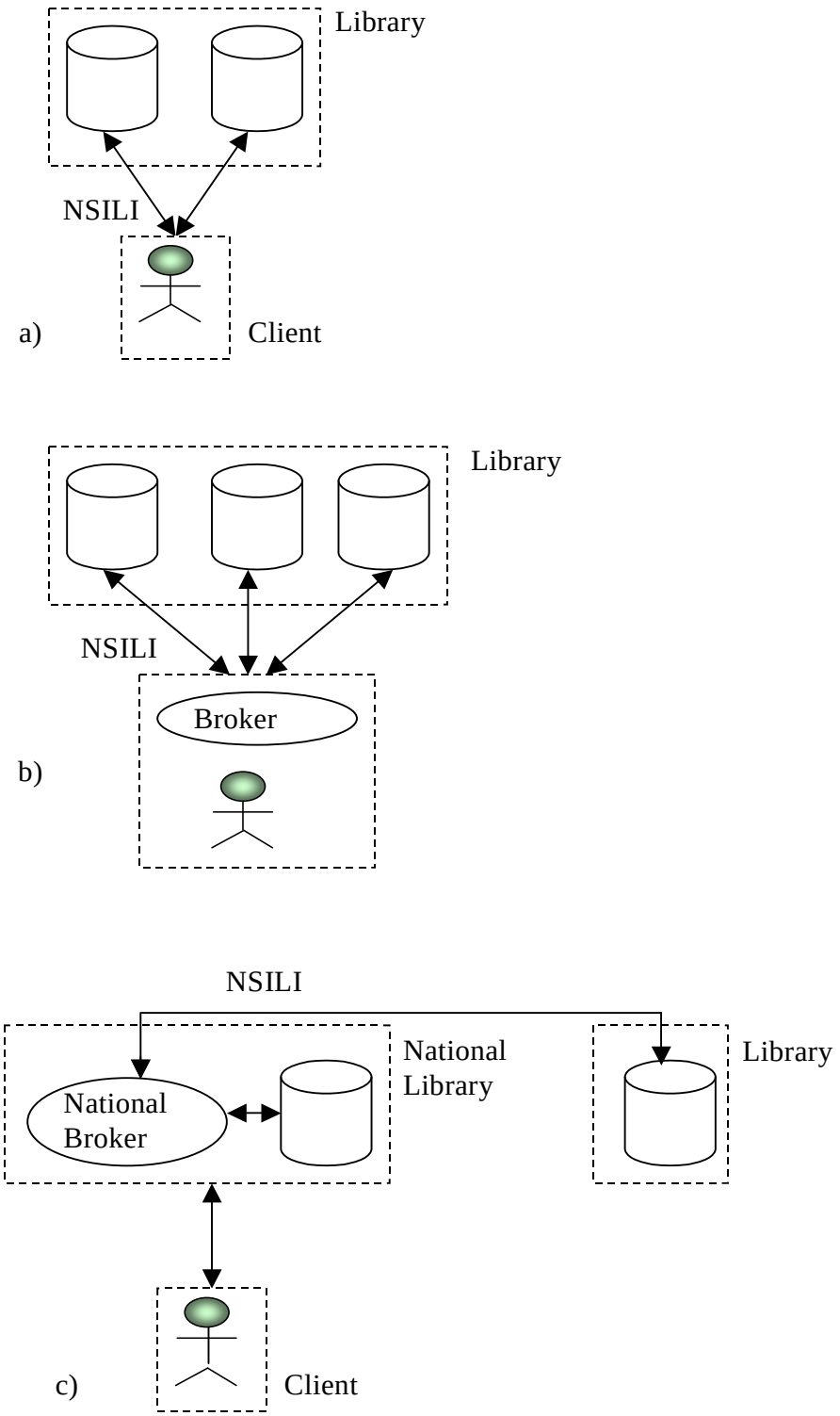

Figure C- 4 Possible Architectures

3.5. Non-homogeneous Metadata.

3.5.1. An important issue is that of non-homogeneous metadata. Metadata available will vary depending on whether the product has been exploited or not, and other factors. It is assumed that some variation in metadata supported by different IPLs is likely to occur.

3.5.2. There may be two types of metadata supported. These are:

Mandatory metadata. All servers are required to support queries involving these metadata and should make these metadata available where possible. They are described in Annex E.

Optional, metadata. These are metadata that may be implemented by a particular IPL, but that are not required by the NSIL Interface. Using the discovery mechanism provided by the NSIL Interface, Clients can discover the optional metadata attributes supported by a particular library.

3.6. Extensibility.

The NSIL Interface provides the flexibility to allow for additional metadata via a discovery mechanism. Servers are not restricted to the core metadata set. Clients can always rely on the core metadata set but can also optionally exploit additional metadata if supported.

ANNEX D.

NSIL INTERFACE TECHNICAL DEFINITION

ANNEX D.

NSIL INTERFACE TECHNICAL DEFINITION

TABLE OF CONTENTS

PARAGRAPH	TITLE	PAGE
1.	Introduction.....	D-1
1.1.	Scope.....	D-1
1.2.	Purpose.....	D-1
1.3.	Document Organization.....	D-1
2.	Interoperability Requirements.....	D-1
2.1.	Distributed Computing Services.....	D-1
2.2.	Object Request Broker.....	D-1
2.3.	Required Interfaces.....	D-1
2.4.	Optional Interfaces.....	D-5
2.5	Implementation Details.....	D-5

TABLES	TITLE	PAGE
Table D- 1	Implementation Detail for Library Interface	D-6
Table D- 2	Implementation Detail for LibraryManager Interface	D-7
Table D- 3	Implementation Detail for Interface RequestManager	D-7
Table D- 4	Implementation Detail for Interface AccessManager : RequestManager	D-8
Table D- 5	Implementation Detail for Interface OrderMgr : LibraryManager, AccessManager	D-8
Table D- 6	Implementation Detail for Interface DataModelMgr : LibraryManager	D-9
Table D- 7	Implementation Detail for Interface CatalogMgr : LibraryManager, RequestManager	D-11
Table D- 8	Implementation Detail for Interface ProductMgr : LibraryManager, AccessManager	D-12
Table D- 9	Implementation Detail for Interface CreationMgr : LibraryManager, RequestManager	D-13
Table D- 10	Implementation Detail for Interface UpdateMgr : LibraryManager, RequestManager	D-14
Table D- 11	Implementation Detail for Request Interface	D-15
Table D- 12	Implementation Detail for Interface OrderRequest : Request	D-16
Table D- 13	Implementation Detail for Interface SubmitQueryRequest : Request	D-17
Table D- 14	Implementation Detail for Interface SetAvailabilityRequest : Request	D-18
Table D- 15	Implementation Detail for Interface HitCountRequest : Request	D-18
Table D- 16	Implementation Detail for Interface GetParametersRequest : Request	D-19
Table D- 17	Implementation Detail for Interface GetRelatedFilesRequest : Request	D-19
Table D- 18	Implementation Detail for Callback Interface	D-20
Table D- 19	Implementation Detail for Interface StandingQueryMgr: LibraryMgr, RequestMgr	D-20
Table D- 20	Implementation Details for Interface SubmitStandingQueryRequest: Request	D-21
Table D- 21	Implementation Detail for Interface QueryOrderMgr: LibraryMgr, RequestMgr	D-22
Table D- 22	Implementation Details for Interface SubmitQueryOrderRequest: Request	D-22
Table D- 23	Properties, values and default settings for non-standard protocol and ports	D-23

FIGURES TITLE..... PAGE

Figure D - 1 UML Static Class Diagram of NSIL Interface Profile of GIAS D-4

APPENDIX 1.

RECOMMENDED PRACTICES AND CLARIFICATION INFORMATION

TABLE OF CONTENTS

PARAGRAPH	TITLE	PAGE
1.	OrderContents Structure……	D-1-1
1.1	Media Types……	D-1-1
1.2	Image Specifications……	D-1-1
2.	Image and Ground Coordinates……	D-1-4
2.1.	Image Coordinates……	D-1-4
2.2.	Ground Coordinates……	D-1-5
3.	Data Management……	D-1-6
3.1	Packaging Specifications……	D-1-6
3.2	Media Type Encodings……	D-1-8
3.3	Related File Types……	D-1-8
4.	UID::Product Usage Guidance……	D-1-8
4.1	UID Product Interface……	D-1-8
4.2	Product Association with ORB method……	D-1-8
5.	DAG_Structure for NSIL……	D-1-9
6.	DAG Structure for Edition 3 Metadata Model……	D-1-9
6.1	Data Model Hierarchy……	D-1-9
6.2	NODES in the DAG……	D-1-9
6.3	NODE linking……	D-1-9
6.4	DAG Associations and Products……	D-1-9
6.5	Querying the NSIL_ASSOCIATION_View……	D-1-11

TABLES	TITLE	PAGE
Table D-1- 1	order <OrderContents>.....	D-1-1
Table D-1- 2	order.pSpec <PackagingSpec>	D-1-2
Table D-1- 3	order.prod_list <ProductDetailsList>.....	D-1-2
Table D-1- 4	order.prod_list.mTypes <MediaTypeList>	D-1-2
Table D-1- 5	order.prod_list.aSpec <AlterationSpec>	D-1-2
Table D-1- 6	order.prod_list.aSpec.ps <ImageSpec>	D-1-3
Table D-1- 7	order.del_list <DeliveryDetailsList>	D-1-4
Table D-1- 8	order <QueryOrderContents>.....	D-1-4
Table D-1- 9	Packaging Format and Compression	D-1-6
Table D-1- 10	File Packaging Actions.....	D-1-7
Table D-1- 11	Media Types.....	D-1-8

Table D-1- 12 Standard Related File Types D-1-8

FIGURES TITLE PAGE

Figure D-1- 1 Example of a DAG with attributes from multiple (sub)-entities D-1-10
Figure D-1- 2 Example of a DAG with edges D-1-11
Figure D-1- 3 Example of DAG for NSIL_ASSOCIATION view D-1-12

ANNEX D. NSIL INTERFACE TECHNICAL DEFINITION

1. INTRODUCTION

This Annex is to serve as the Technical Definition of the NSIL Interface. The Technical Definition is based on the Geospatial and Imagery Access Services (GIAS) Specification developed by the United State's National Geospatial-Intelligence Agency (NGA, formerly the National Imagery and Mapping Agency (NIMA)). More specifically it is an Implementation Profile of the GIAS specification tailored for NATO operations. It identifies specific GIAS interfaces and any deviations to those interfaces. It contains implementation details, including all parameter values to be passed to and from the interface objects.

1.1. Scope. The interface defines an application program interface (API) and the services that are provided through the API, as well as communication protocols and data formats.

1.2. Purpose. This implementation profile is intended to define clearly the implementation requirements that must be met by vendors and development contractors providing the components that will make up an ISR product library system. The GIAS Standard is designed to be as broadly applicable as possible and therefore only contains the most general features and data structures. These general features can be used in many different ways by different domains. To guarantee interoperability within a specific domain, developers using these standards, i.e., developers of client and server implementations must use them in a consistent manner. A "standard profile" or "profile of a standard" serves to document a specific community's standards, conventions, and agreed-to procedures on how that general standard is tailored to that community's requirements. Without these lower level implementation documents to specify the details, it is unlikely that any two developers would make exactly the same interpretation of how to use the standards.

1.3. Document Organization. This volume is organized as follows:
Section 1 - Introduction contains overview material for this Implementation Profile.
Section 2 - Interoperability Requirements contains the NSIL Interface implementation requirements of STANAG 4559.
Appendix 1 to Annex D - Provides Recommended Practices and Clarification Information

2 INTEROPERABILITY REQUIREMENTS

2.1 Distributed Computing Services. ISR product library systems shall utilize a common set of Distributed Computing Services for exchanging information as specified in The Common Object Request Broker: Architecture and Specification

2.2 Object Request Broker Interoperability. Object Request Brokers (ORBs) utilized for the NSIL Interface implementations shall support the Internet/Inter-ORB Protocol (IIOP) for inter-ORB communication.

Service IORs and Client/Service Binding. As a matter of notification to implementers, backwards compatibility with STANAG 4559 Edition 2 and 1 will not be supported for FTP, and beginning with Edition 3 FTP is disallowed. CORBA Internet Object References (IOR) are delivered via HTTP or HTTPS to support client binding. The IOR file shall be an ASCII file containing only the IOR string. HTTP authentication is recommended; a specific authentication mechanism can be specified in the profile.

2.3 Required Interfaces. The required interfaces described in this section are a profile, or a subset with some minor deviations in usage, of the interfaces defined in the GIAS Specification. The NSIL Interface defines interface managers that control the functions of the ISR Product Library (IPL) and a top-level interface that provides access to those Interface Managers. Further interfaces define a product reference handle and operations common to request objects used by the interface managers as part of their operations. Optional interfaces are listed in 2.4 of this Annex defines a callback mechanism and interface

managers that are defined by the GIAS and have been deemed valuable to the NSILI client. A client interacts with the IPL to select and request access to a specific interface manager, which it then uses to submit requests to the library. Each request returns a request object that the client can use to monitor progress and retrieve results. The client can also obtain information on a specific request or interface manager. The specification of each interface details the operations, data types and error conditions that provide the functionality. A brief overview of these interfaces is given below. The GIAS specification and Implementation Details. paragraph 2.5 of this Annex provide the details of each interface in table form. The IDL code for these interfaces is specified in the appendices of the GIAS specification and can be compiled directly into various languages. The information in this section is taken largely from the GIAS specification which is made available with the STANAG on the NATO Standardisation Agency website.

2.3.1 Library Interface. The Library interface is the central point of access to the services provided by the IPL. It provides methods to return library identification information and references to requested Interface Managers.

2.3.2 LibraryManager Interface. The LibraryManager Interface is an abstract interface that defines operations common to other Interface Managers. This comprises methods to determine names and current values of various properties applicable to each manager.

2.3.3 RequestManager Interface. The RequestManager Interface is an abstract interface that defines operations common to all managers that use Request objects as part of their operations. The operations defined allow clients to identify active Request objectnd control their lifetimes.

2.3.4 AccessManager Interface. The AccessManager Interface is an abstract interface that defines operations common to managers that allow clients to determine and control the availability of a data set or product, where availability refers to the readiness of that product or data set to be used by other operations provided by the interface manager. It provides operations to determine the usage modes applicable to a specific product or data set, determine or request the availability of a product for those uses, and obtain an estimate of the delay required to make a product available.

2.3.5 OrderMgr Interface. The OrderMgr allows clients to submit orders for products from an IPL and allows orders to be validated prior to submission. Clients use the OrderMgr to specify the desired delivery mechanism and destination, product format, compression, resolution and so on, according to the options available at the IPL.

2.3.6 DataModelMgr Interface. The DataModelMgr Interface provides operations to allow a client to discover and access the data model being used by the library. This Interface provides the mechanism for identifying and using non-standard metadata attributes not defined in the NSIL core metadata set. The DataModelMgr provides two types of operation: access to ancillary data, such as the last modification date and time; and access to the data model itself, such as the set of data views and entities known to the library and the attributes that describe a specific data view or entity.

2.3.7 CatalogMgr Interface. The CatalogMgr Interface provides operations to allow clients to search the IPL for products of interest based on metadata queries, returning a user defined number of hits. It provides an operation to determine the number of hits that would result for a query, prior to its submission, and allows retrieval of query results in a variety of ways including XML documents and NSIL Interface data structures.

2.3.8 ProductMgr Interface. The ProductMgr Interface provides operations that allow the client to determine characteristics of a specific product and retrieve related files associated with that product, such as thumbnail and overview images.

2.3.9 Request Interface. The Request Interface is an abstract interface that defines operations common to Request objects. All specialized Request objects are derived from this interface. The Request Interface defines operations to allow identification of request objects, monitoring of request status, control of request lifetimes, and a facility for registering callback objects to be notified when a request reaches a terminal state. The specialized request objects all provide operations to invoke completion of request processing, according to the service they provide. The specific request objects are created through methods of the specific request managers. Since some managers are optional, some specific request objects are optional. The following is a list of the required specific request interfaces:

OrderRequest
SubmitQueryRequest
SetAvailabilityRequest
HitCountRequest
GetParametersRequest
GetRelatedFilesRequest

The following is a list of the optional specific request interfaces:
SubmitStandingQueryRequest
SubmitQueryOrderRequest

2.3.10 Product Interface. The Product Interface serves as a reference handle for a specific product in an IPL. No operations are defined for this interface.

2.3.11 Usage. The interfaces defined can be used in a variety of ways to suit the end-goal of the user. In a basic interaction the client would use the Library Interface to gain access to a CatalogMgr. An appropriate query would then be submitted and a set of query results would be returned including a Product reference for each result. The ProductMgr could then be used to retrieve information for each result, such as thumbnail images for imagery, allowing the user to select a specific product for retrieval via the OrderMgr. The query submission, related file retrieval and order submission all result in the generation of a request object. Valid queries would be constructed using the NSIL Interface core metadata set, or alternatively after using the DataModelMgr to determine the specifics of any non-standard extension to the data model understood by the IPL.

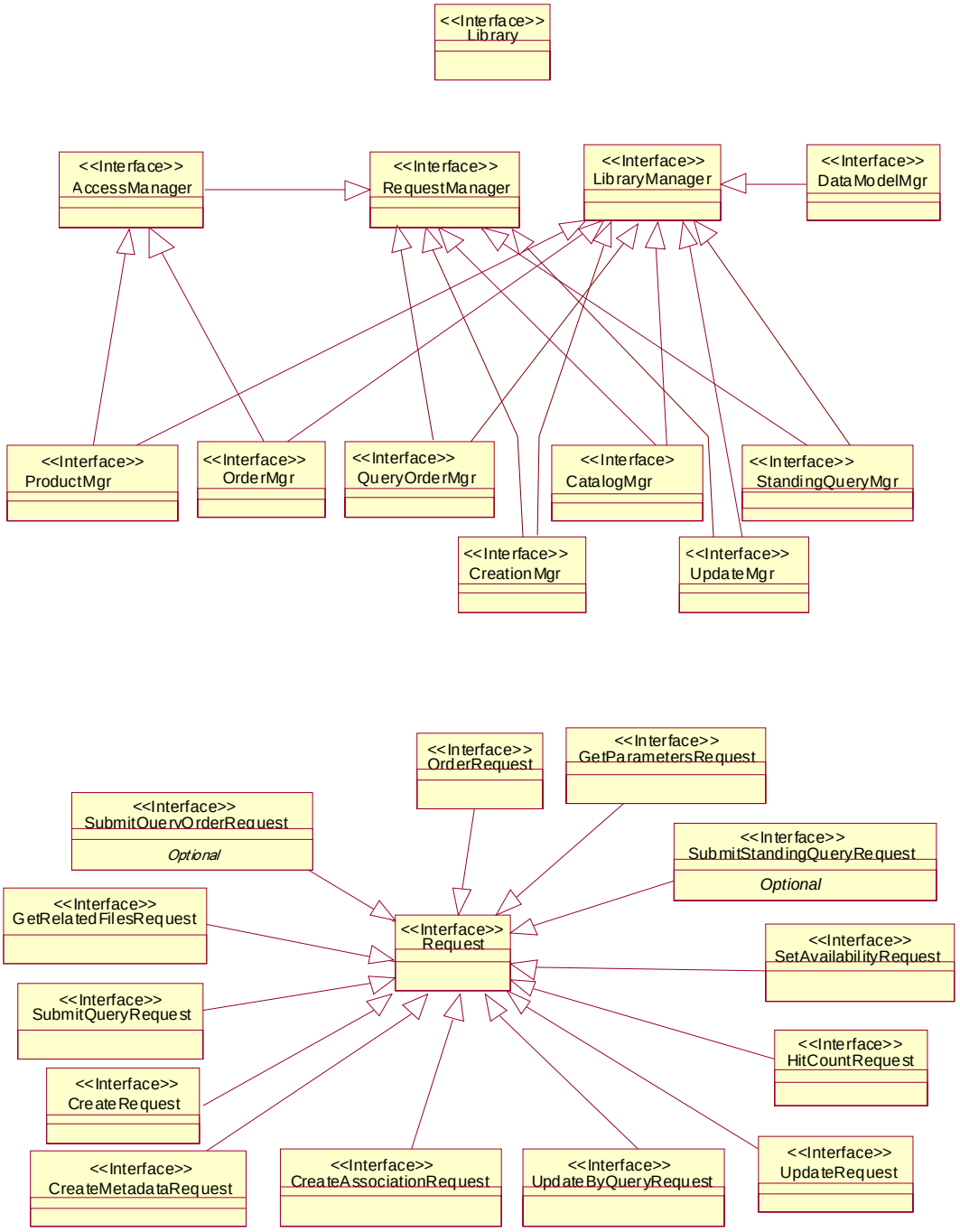

Figure D - 1 UML Static Class Diagram of NSIL Interface Profile of GIAS

2.4 Optional Interfaces. As new interfaces are defined for the NSIL Interface that address the full scope of NIIA data types, it is expected that most will be optional for implementation. However, when new interfaces are adopted, implementations must comply with the interface definitions. Optional interfaces available at ratification of this document are defined below.

2.4.1 StandingQueryMgr Interface. The StandingQueryMgr allows clients to subscribe to metadata of products that match a given query. The client must specify the period during which the library must be monitored for new products matching the query, and the interval time between reports of new products. The client can see the number of hits for each interval that has passed. The query results (metadata) can be retrieved in a variety of ways including XML documents and NSIL Interface data structures. Query results from an interval that have not yet been retrieved can be discarded. Optional

2.4.2 QueryOrderMgr Interface. The QueryOrderMgr allows clients to subscribe to products that match a given query and have them delivered to the client. The client must specify the period during which the library must be monitored for new products matching the query, and the interval time between reports of new products. The matching products will be delivered automatically to the client. The client can retrieve the DeliveryManifest describing the packages and products that have been delivered. Optional.

2.4.3 CreationMgr Interface. The interface allows a client to create products, and associations between existing products in the library. This interface also allows a client to create metadata only products without supplying the product itself.

2.4.4 Callback Interface. The Callback Interface allows an event-driven programming model when using Request objects. The Callback Interface provides operations to notify the client of changes in Request object status and to indicate that a callback object will no longer be used. Descriptive data passed should be server-side request data specifically from the request to which the callback refers.

2.4.5 UpdateMgr Interface. The interface provides the capability for a client to modify existing catalog entries.

2.5 Implementation Details.

2.5.1 Relationship to a Logical Data Model. A NSIL Interface implementation is defined in this document. The LDM represents the natural characteristics of the data and the relationships among the data entities, rather than how the data is stored in the database. Hence, a layer of user view abstraction is required. By using this natural data approach, the impacts of changes in the database design are isolated to the implementation, and do not impact the user's perspective of the data. The LDM defines the attributes (data elements), structure and relationships of the attributes, and the domain of acceptable values for the attributes. The LDM is implementation dependent, but must contain as a minimum the entities, relationships and attributes defined in Annex G as the minimum layer, and as illustrated in Annex H. A "NSIL_CORE" view will also need to be defined which allows a user view of this NSIL Minimum Data Model metadata. By levying this requirement clients will be able to construct valid queries across multiple libraries. IPL implementers have the flexibility to add to the LDM as required and can provide unique views of their own. The implementation details described in section 2.5.2 Interface Details make reference to this LDM.

2.5.2 Interface Details. The implementation details for each interface are listed in the tables below. Each table heading lists the interface followed by any inherited interfaces e.g. CatalogMgr inherits from RequestManager and so is described as "CatalogMgr : RequestManager."

Table D- 1 Implementation Detail for Library Interface

Method	**Return/ Parameter**	**Type**	**Desc/Value(s)**
get_manager_types	[return]	**ManagerTypeList** sequence<string>	"OrderMgr" ; "CatalogMgr" ; "ProductMgr" ; "DataModelMgr" Optional: "QueryOrderMgr" "StandingQueryMgr" "CreationMgr" "UpdateMgr"
get_manager	**LibraryManager**	[Object]	N/A
	manager_type	**ManagerType** <string>	"OrderMgr" ; "CatalogMgr" ; "ProductMgr" ; "DataModelMgr" Optional: "QueryOrderMgr" "StandingQueryMgr" "CreationMgr" "UpdateMgr"
	access_criteria	**AccessCriteria**	Not Used
get_library_ description	[return]	**LibraryDescription**	**Library_name**<string>Valid Library identifier will be implementation dependent. Recommend using hostname. Must be unique within the context of the parent organization/country code specified in Library_description. **Library_description**<string> Organizational affiliation of Library. Implementation unique. Recommend concatenation of country code digraph or trigraph and organizational unit code separated by a '|'. Example "US|AFRL/IFEC" **Library_version** <string>Text string "NSILI" with STANAG version number separated by '|'. Example: "NSILI|1.0"
get_other_libraries			Throw CORBA exception NO_IMPLEMENT.

Table D- 2 Implementation Detail for LibraryManager Interface

Method	Return/ Parameter	Type	Desc/Value(s)
get_property_names			Throw CORBA exception NO_IMPLEMENT.
get_property_values			Throw CORBA exception NO_IMPLEMENT.
get_libraries			Throw CORBA exception NO_IMPLEMENT.

Table D- 3 Implementation Detail for Interface RequestManager

Method	Return/ Parameter	TYPE	Desc/Value(s)
get_active_requests	[return]	RequestList	N/A
get_default_timeout	[return]	<unsigned long>	As defined in GIAS
set_default_timeout	new_default	<unsigned long>	As defined in GIAS
get_timeout	[return]	<unsigned long>	As defined in GIAS
	Request	Request	N/A
set_timeout	**Request**	Request	N/A
	new_lifetime	<unsigned long>	As defined in GIAS
delete_request	**Request**	Request	N/A

Table D- 4 Implementation Detail for Interface AccessManager : RequestManager

Method	Return/ Parameter	Type	Desc/Value(s)
get_use_modes	[return]	UseModeList	"OrderAccess"; Product available for OrderMgr use (i.e. available for ordering)
is_available	[return]	<boolean>	TRUE if product is available for the specified use, otherwise FALSE.
	Product	UID::Product	N/A
	use_mode	UseMode	"OrderAccess"
query_availability_ delay	[return]	<unsigned long>	As defined in GIAS.
	Product	UID::Product	N/A
	Availability_ Requirement	**Availability Requirement**	As defined in GIAS.
	use_mode	UseMode	"OrderAccess"
get_number_of_Priorities()	[return]	CORBA:short	As defined in GIAS.
set_availability	[return]	SetAvailability Request	N/A
	Products	UID::ProductList	N/A
	Availability_ Requirement	**Availability Requirement**	As defined in GIAS.
	priority	CORBA:short	N/A
	use_mode	UseMode	"OrderAccess"

Table D- 5 Implementation Detail for Interface OrderMgr : LibraryManager, AccessManager

Method	Return/ Parameter	Type	Desc/Value(s)
get_package_ specifications	[return]	NameList	Available packaging formats and compression for delivery of products. See Appendix 1 of Annex D, section 3.1 Packaging Specifications for details.
validate_ order	[return]	**ValidationResults**	Results table provided in GIAS Table 2-1 **valid**<boolean>TRUE if order is valid FALSE if order is invalid **warning**<boolean>TRUE if there is a warning message with a valid order, otherwise false ("**details**" contains warning message of proposed order) **details**<string>Contains description of error or warning
	order	OrderContents	The OrderContents structure is defined in GIAS. Details for the use of fields in this structure can be found in Appendix 1 of this Annex, 1 OrderContents Structure
	properties	**PropertyList**	An ordered sequence of Name-Value pairs
Order	[return]	**OrderRequest**	N/A
	order	OrderContents	The OrderContents structure is defined in GIAS.

Method	Return/ Parameter	Type	Desc/Value(s)
			Details for the use of fields in this structure can be found in Appendix 1 of this Annex, section 1 OrderContents Structure
	properties	**PropertyList**	GeographicDatum=<string> Datum for GeoRegion of the Alteration Spec;default="WGS84"

Table D- 6 Implementation Detail for Interface DataModelMgr : LibraryManager

Method	Return/ Parameter	Type	Desc/Value(s)
get_data_model_date	[return]	UCO::AbsTime	Return date time group in GMT when the data model was last changed.
get_view_names	[return]	ViewList	Provides list of views supported by the Library. The list of views must at least include the "NSIL _CORE" view, which sees the entire NSIL Interface Minimum Data Model in Annex E. Other unique views may also be defined to view other parts of the LDM.
	Properties	**PropertyList**	Not Used.
get_attributes	[return]	**AttributeInformation List**	Attribute Information structure is given in Annex E
	view_name	ViewName (<string>)	Valid Logical Data Model view supported by library.
	Properties	**PropertyList**	Not Used. Empty list.
get_queryable_ attributes	[return]	**AttributeInformation List**	AttributeInformation structure is given in Annex E
	view_name	ViewName(<string>)	Valid Logical Data Model view supported by library.
	Properties	**PropertyList**	Not Used.
get_entities	[return]	**UCO::EntityGraph**	EntityGraph containing entities and relationships for the data view requested.
	view_name	ViewName(<string>)	Valid Logical Data Model view supported by library.
	Properties	**PropertyList**	Not Used.
get_entity_attributes	[return]	**AttributeInformation List**	AttributeInformation structure is given in Annex E
	aEntity	Entity(<string>)	Valid LDM entity.
	Properties	**PropertyList**	Not Used.
get_alias_categories	[return]	**UCO::NameList**	This selector operation returns a *NameList* containing user communities known to this Library. A user community contained in the *NameList* is used as a parameter for the selector operation *get_logical_aliases*.
	Properties	**PropertyList**	Not Used.

Method	Return/ Parameter	Type	Desc/Value(s)
get_logical_aliases	[return]	**UCO::NameNameList**	This selector operation returns a NameNameList, which contains a mapping of a specific community's aliases to attribute names from the Logical Data Model.
	category	<string>	“The user community” is a required return value from the get_alias_categories method.
	Properties	**PropertyList**	Not Used.
get_logical_ attribute_name	[return]	<string>	This selector operation returns the name of the logical attribute that is the equivalent of the requested *ConceptualAttributeType* in the requested view *view_name*.
	view_name	ViewName	The requested view.
	attribute_type	ConceptualAttributeType	The conceptual attribute type.
	properties	**PropertyList**	Not Used.
get_associations	[return]	AssociationList	Returns a list of Association structures that contain the descriptions of the associations
	properties	**PropertyList**	Not Used.
get_max_vertices	[return]	<unsigned short>	Returns the maximum number of vertices supported in geospatial queries
	properties	**PropertyList**	Not Used.

Table D- 7 Implementation Detail for Interface CatalogMgr : LibraryManager, RequestManager

Method	**Return/ Parameter**	**Type**	**Desc/Value(s)**
submit_query	[return]	**SubmitQuery Request**	N/A
	aQuery	Query	The query attribute of the Query structure must be a valid Boolean query as defined in ANNEX F. BOOLEAN Query Syntax.
	result_attributes	**UCO::NameList**	Attribute name(s) to be returned. Attributes are from those available in the Logical Data Model
	sort_attributes	**SortAttributeList**	Attribute name(s) to be sorted on with a flag denoting whether ascending or descending. The order in the list will indicate the precedence. Attributes are from those available in the Logical Data Model.
	properties	**PropertyList**	**GeographicDatum**<string> Datum of the geospatial data in the BQS query; default="WGS84"
hit_count	[return]	**HitCountRequest**	N/A
	aQuery	Query	The query attribute of the Query structure must be a valid Boolean query as defined in ANNEX F. BOOLEAN Query Syntax.
	properties	**PropertyList** (UCO::NameValueList)	**GeographicDatum**<string> Datum of the geospatial data in the BQS query; default="WGS84"

Table D- 8 Implementation Detail for Interface ProductMgr : LibraryManager, AccessManager

Method	Return/ Parameter	Type	Desc/Value(s)
get_parameters	[return]	**GetParameters Request**	N/A
	product	UID::Product	Product for which data is being requested
	desired_parameters	**UCO::NameList**	Attribute name(s) from . “ALL” will return all attributes associated with the product. "CORE" will return only attributes in the NSIL Interface Minimum Data Model associated with the product. Node 0 (the root node) shall contain a UID:Product product reference and shall be named “NSIL Interface Image Product”. Node names shall be short form attribute names (e.g., use “FTITLE” not “File Title”) from Logical Data Model. *NOTE: NSILI Minimum Data Model description dependent.* “ORDER” will return attributes associated with product ordering. To be defined by the Logical Data Model.
	properties	**PropertyList**	Not Used
get_related_file_types	[return]	**RelatedFileTypeList**	List of file types available for the product. Related file types are described in Table D-1- 12 Standard Related File Types. Other related file types may appear and are system specific.
	prod	UID::Product	Product for which available file types is being requested
get_related_files	[return]	**GetRelatedFilesRequest**	N/A
	products	UID::ProductList	List of products for which the related files are to be retrieved
	location	UCO::FileLocation	Location where the related files are to be put; FileName will be blank
	type	**RelatedFileType**	The type of related file to be retrieved. The acceptable values for this parameter can be determined via the get_related_file_types method.
	properties	PropertyList	Not Used

Table D- 9 Implementation Detail for Interface CreationMgr : LibraryManager, RequestManager

Method	Return/ Parameter	Type	Desc/Value(s)
Create	[return]	GIAS::CreateRequest	As defined in GIAS.
	new_product	UCO::FileLocation[]	List of product file locations.
	related_files	GIAS::RelatedFile[]	List of files that are relevant to the product.
	creation_metadata	UCO::DAG	A DAG as defined in UCOS holds metadata about product.
	properties	UCO::NameValue[]	Optional list of additional properties that provide information for the server to process the method. The properties are used when specifying HTTP/HTTPS as file transport protocol. The value types of protocol and port properties are defined in Table D- 23 Properties, values and default settings for non-standard protocol and ports
create_metadata	[return]	GIAS::CreateMetaDataRequest	As defined in GIAS.
	creation_metadata	UCO::DAG	A DAG as defined in UCOS holds metadata about product.
	view_name	<string>	Name of the view that the product belongs to.
	related_files	GIAS::RelatedFile[]	List of files that are relevant to the product.
	properties	UCO::NameValue[]	Optional list of additional properties that provide information for the server to process the method. The properties are used when specifying HTTP/HTTPS as file transport protocol. The value types of protocol and port properties are defined in Table D- 23 Properties, values and default settings for non-standard protocol and ports
create_association	[return]	GIAS::CreateAssociationRequest	As defined in GIAS.
	assoc_name	<string>	Name of the association
	view_a_object	UID::Product	Source product that the association is defined from.
	view_b_objects	UID::Product[]	Destination product(s) that the association is defined to.
	assoc_info	UCO::NameValue[]	List of additional information about association in format of UCO::NameValue as defined in UCOS.

Table D- 10 Implementation Detail for Interface UpdateMgr : LibraryManager, RequestManager

Method	Return/ Parameter	Type	Desc/Value(s)
set_lock	lockedProduct	UID::Product	Product that is going to be locked for the update.
Update	[return]	GIAS::UpdateRequest	As defined in GIAS.
	view	<string>	Name of the view that the product belongs to.
	changes	UCO::UpdateDAG[]	List of changes to a DAG in UCO::UpdateDAG format as defined in UCOS.
	relfiles	GIAS::RelatedFile[]	List of files that are relevant to the product.
	properties	UCO::NameValue[]	Optional list of additional properties that provide information for the server to process the method. The properties are used when specifying HTTP/HTTPS as file transport protocol. The value types of protocol and port properties are defined in Table D- 23 Properties, values and default settings for non-standard protocol and ports
update_by_query	[return]	GIAS::UpdateByQueryRequest	As defined in GIAS.
	updated_attribute	UCO::NameValue	Name and value of the attribute in format of UCO::NameValue as defined in UCOS.
	bqs_query	GIAS::Query	As defined in GIAS.
	properties	UCO::NameValue[]	Optional list of additional properties that provide information for the server to process the method. The properties are used when specifying HTTP/HTTPS as file transport protocol. The value types of protocol and port properties are defined in Table D- 23 Properties, values and default settings for non-standard protocol and ports
Release_lock	**lockedProduct**	UID::Product	As defined in GIAS.
delete_product	**prod**	UID::Product	As defined in GIAS.

Table D- 11 Implementation Detail for Request Interface

Method	Return/ Parameter	Type	Desc/Value(s)
get_request_ description	[return]	**UCO::Request Description**	**User_info** <string>Previously supplied by client via set_user_info method [Default is empty] **Request_type**<string> “OrderRequest” | “SubmitQueryRequest” | “SetAvailablityRequest” | “HitCountRequest” | “GetParametersRequest” | “GetRelatedFilesRequest” | *Optional:* “SubmitQueryOrderRequest” | “SubmitStandingQueryRequest” | **request_info**<string> Supplied by server [Default is empty] **request_details<**UCO:: NameValueList> See Tables for specific request interface.
set_user_info	**message**	<string>	Supplied by client [Default is empty]
get_status	[return]	**UCO::Status**	**Completion_state<**UCO::State**>** As defined in UCOS. **Status_message** <string>Human-readable explanatory message
get_remaining_del ay	[return]	DelayEstimate	As defined in GIAS.
Cancel	N/A	N/A	N/A
Register_callback	[return]	**CallbackID**	If callbacks *are not* supported, throw CORBA exception NO IMPLEMENT If callbacks are supported, the callback is assumed to be successfully registered.
	acallback	**CB::Callback**	
free_callback	**id**	**CallbackID**	N/A
get_request_mana ger	[return]	**RequestManag er**	N/A

Table D- 12 Implementation Detail for Interface OrderRequest : Request

Method	Return/ Parameter	Type	Desc/Value(s)
complete	[return]	**UCO::State**	As defined in UCOS.
	prods	**DeliveryManifest**	DeliveryManifest is a structure defined in the GIAS providing the package names and the package elements: **package_name=<**string**>**names of the package PackageElementList: *package_element* <UID::Product> file names <UCO::NameList> The DeliveryManifest will be an ordered sequence: For NSIF: file_id*, file_id.rl, file_id.r2, etc…(note *file_id is RO) See Appendix 1, 3.1.2 for file naming conventions to be used in the DeliveryManifest
get_request_ description	[return]	**UCO::RequestDescription**	**User_info**<string> Previously supplied by the client via the set_user_info method [Default is empty] **Request_type** <string> "OrderRequest" **request_info**<string> Supplied by server [Default is empty] **request_details<UCO::NameValueList>** *order<OrderContents> structure containing order* *GeographicDatum<string>* *OrderSize=<double>Size (in MB/megabytes) of product(s) requested in order.* *TimeSubmitted=<UCO::AbsTime>*

Table D- 13 Implementation Detail for Interface SubmitQueryRequest : Request

Method	**Return/ Parameter**	**Type**	**Desc/Value(s)**
set_number_of_hits	hits	<unsigned long>	As defined in GIAS.
NOTE: Of the following three methods for retrieving query results, **complete_DAG_results** will be **required** for implementation and the rest will be optional. For those methods not implemented, the CORBA exception NO_IMPLEMENT will be thrown. Further instructions on DAG structures are in Appendix 1 to Annex D, Paragraph 5 DAG structure for NSIL Discussion of DAG with the STANAG 4559 edition 3 data model is provided in paragraph 6 DAG structure for STANAG 4559 Edition 3 Metadata Model.			
complete_DAG_results	[return]	UCO::State	As defined in UCOS.
	results	UCO::DAGList	Sequence of DAGs as defined in GIAS. Refer to Appendix 1 of Annex D, paragraph 5 for details.
complete_XML_results	[return]	**UCO::State**	As defined in UCOS.
	results	**UCO::XML.document**	Results are as defined in the XML schema (to be defined)
complete_String DAG_results			Throw CORBA exception NO_IMPLEMENT.
get_request_description	[default]	**UCO::Request Description**	**user_info**<string> Supplied by client [Default is empty] **request_type**<string> "SubmitQueryRequest" **request_info** <string> Supplied by server [Default is empty] **request_details<**UCO::NameValueList> *query*=<Query> query string *result_attributes*<UCO::NameList> *sort_attributes*<SortAttributeList> GeographicDatum<string>

Table D- 14 Implementation Detail for Interface SetAvailabilityRequest : Request

Method	**Return/ Parameter**	**Type**	**Desc/Value(s)**
complete	[return]	**UCO::State**	As defined in UCOS.
get_request_ description	[return]	**UCO::Request Description**	**user_info**<string> Supplied by client [Default is empty] **request_type**<string> "SetAvailabitiyRequest" request_info<string> Supplied by server [Default is empty] **request_details<UCO::NameValueList>** *product*=<Product> product object reference *availability_requirement*<AvailabilityRequirement> *use_mode*= <UseMode**>**

Table D- 15 Implementation Detail for Interface HitCountRequest : Request

Method	**Return/ Parameter**	**Type**	**Desc/Value(s)**
Complete	[return]	**UCO::State**	As defined in UCOS.
	number_of_hits	<unsigned long>	As defined in GIAS.
get_request_ description	[return]	**UCO::RequestDescription**	**User_info**<string> Supplied by client [Default is empty] **Request_type**<string> "HitCountRequest" **Request_info** <string>Supplied by server [Default is empty] **Request_details**<UCO::NameValueList**>** *query*=<Query> query string *GeographicDatum*<string>

Table D- 16 Implementation Detail for Interface GetParametersRequest : Request

Method	Return/ Parameter	Type	Desc/Value(s)
complete	[return]	**UCO::State**	As defined in UCOS.
	parameters	**UCO::DAG**	DAGs as defined in GIAS of type QueryResult. One node shall be a product reference. Node names shall be a subset derived from the Logical Data Model. Cardinality fields of DAG nodes are not used. Cataloging is assuming single image products when imagery is being requested. Refer to Appendix 1 of Annex D, paragraph 5 DAG structure for NSIL for details.
get-request-description	[return]	**UCO::Request Description**	**user_info** <string> Supplied by client [Default is empty] **request_type**<string> "GetParametersRequest **request_info**<string> Supplied by server [Default is empty] **request_details** <UCO::NameValueList> *product*=<UID::ProductList>Product object reference *desired_parameters*<UCO::NameList>

Table D- 17 Implementation Detail for Interface GetRelatedFilesRequest : Request

Method	Return/ Parameter	Type	Desc/Value(s)
complete	[return]	**UCO::State**	As defined in UCOS.
	locations	**UCO::NameList**	As defined in UCOS.
get_request_description	[return]	**UCO::RequestDescription**	**user_info**<string> Supplied by client [Default is empty] **request_type** <string> "GetRelatedFilesRequest" **request_info** <string>Supplied by server [Default is empty] **request_details** <UCO::NameValueList> *products*<UID::ProductList> *location*<UCO::FileLocation> *type*< **RelatedFileType**> Related file types are described in Table D-1- 12 Standard Related File Types.

Table D- 18 Implementation Detail for Callback Interface

Method	Return/ Parameter	Type	Desc/Value(s)
notify	**description**	**RequestDescription**	**user_<string>** Supplied by client [Default is empty] **request_type <string>** description passed should be server-side request type specifically from the request to which the callback refers. **request_info <string>** Request identifier **request_details** <**UCO::NameValueList>** Description passed should be request details specifically from the request to which the callback refers.
release	N/A	N/A	N/A

Table D- 19 Implementation Detail for Interface StandingQueryMgr: LibraryMgr, RequestMgr

Method	Return / Parameter	Type	Desc / Value(s)
get_event_descriptions	[return]	**EventList**	List of events that can be used in the lifespan parameter of submit_standing_query.
submit_standing_query	[return]	**SubmitStandingQueryRequest**	N/A
	aQuery	**Query**	Valid Query structure as defined in *Geospatial and Imagery Access Service (GIAS) Specification.*
	result_attributes	**UCO::NameList**	Attribute name(s) to be returned. Attributes are from those available in the Logical Data Model.
	sort_attributes	**SortAttributeList**	Attribute name(s) to be sorted on with a flag denoting whether ascending or descending. The order in the list will indicate the precedence. Attributes are from those available in the Logical Data Model.
	lifespan	**QueryLifeSpan**	The period of time the query is to be run and the frequency.
	properties	**PropertyList**	**GeographicDatum**<string> Datum of the geospatial data in the BQS query; default="WGS84"

Table D- 20 Implementation Details for Interface SubmitStandingQueryRequest: Request

Method	Return / Parameter	Type	Desc / Value(s)
set_number_of_hits	**hits**	**unsigned long**	As defined in GIAS.
get_number_of_hits	[return]	**unsigned long**	Current total number of hits collected by this Request.
get_number_of_hits_in_interval	[return]	**unsigned long**	Number of hits in the specified interval
	interval	**unsigned long**	N/A
get_number_of_intervals	[return]	**unsigned long**	Number of intervals for which this Request has collected hits.
clear_all	N/A	N/A	Clears all hits currently held by this Request.
clear_intervals	**num_intervals**	**unsigned long**	Clears all hits in the specified interval.
clear_before	**relative_time**	**UCO::Time**	Clears all hits collected before the specified time, i.e. those that are older than the given time.
pause	N/A	N/A	Temporarily suspend the Request.
resume	N/A	N/A	Causes a suspended Request to continue.
get_last_time_executed	[return]	**UCO::AbsTime**	The time of the last performed execution of this standing query.
get_next_time_execution	[return]	**UCO::AbsTime**	The time of the next scheduled execution of this standing query.
NOTE: of the following three methods for retrieving query results, **complete_DAG_results** will be **required** for implementation and the rest will be optional. For those methods not implemented, the CORBA exception NO_IMPLEMENT will be thrown. Further instructions on DAG structures are in Appendix 1, paragraph 5 DAG structure for NSIL.			
complete_DAG_results	[return]	**UCO::State**	As defined in UCOS.
	results	**QueryResults**	Sequence of DAGs as defined in GIAS. Refer to Appendix 1 of Annex D, paragraph 5 DAG structure for NSIL for details.
complete_XML_results	[return]	**UCO::State**	As defined in UCOS.
	results	**UCO::XMLDocument**	Results are as defined in the XML Schema for NSILI results (TBD)
complete_StringDAG_results	[return]	**UCO::State**	throws CORBA exception NO_IMPLEMENT.
get_request_description	[return]	**UCO::RequestDescription**	**user_info**<string> Supplied by client [Default is empty] **request_type**<string> “SubmitStandingQueryRequest” **request_info**<string> Supplied by server [Default is empty] **request_details**<UCO::NameValueList> *query*<Query>

			result_attributes<UCO:NameList> *sort_attributes*<SortAttributeList> *life_span*<QueryLifeSpan> *GeographicDatum*<string>

Table D- 21 Implementation Detail for Interface QueryOrderMgr: LibraryMgr, RequestMgr

Method	**Return / Parameter**	**Type**	**Desc / Value(s)**
get_event_descriptions	[return]	**EventList**	List of events that can be used in the lifespan parameter of submit_query_order.
submit_query_order	[return]	**SubmitQueryOrderRequest**	N/A
	aQuery	**Query**	Valid Query structure as defined in *Geospatial and Imagery Access Service (GIAS) Specification.*
	lifespan	**QueryLifeSpan**	The period of time the query is to be run and the frequency.
	o_type	**OrderType**	If **IMMEDIATE**, the query will run only once and the lifespan parameter is ignored. The matching products will be delivered. If **STANDING** the library will be monitored for the period determined by the lifespan parameter. The existing and new products matching the query will be delivered.
	order	**QueryOrderContents**	TheQueryOrderContents structure is defined in GIAS. Details for the use of fields in this structure can be found in Appendix 1 of this Annex, section 1 **ORDERCONTENTS STRUCTURE**.
	properties	**PropertyList**	**GeographicDatum**<string> Datum of the geospatial data in the BQS query and the AlterationSpec; default="WGS84"

Table D- 22 Implementation Details for Interface SubmitQueryOrderRequest: Request

Method	**Return / Parameter**	**Type**	**Desc / Value(s)**
pause	N/A	N/A	Temporarily suspend the Request.
resume	N/A	N/A	Causes a suspended Request to continue.
complete_list	[return]	**UCO::State**	As defined in UCOS.
	prods	**DeliveryManifestList**	Sequence of DeliveryManifest structures as defined in the GIAS. There is one DeliveryManifest for each separate package that is delivered (generally one per interval). See Appendix 1, section 3.1.2 for file naming conventions to be used in the DeliveryManifest
complete	[return]	**UCO::State**	As defined in UCOS.

	prods	**DeliveryManifest**	DeliveryManifest is a structure defined in the GIAS providing package name and package elements. See Appendix 1, section 3.1.2 for file naming conventions to be used in the DeliveryManifest
get_request_ description	[return]	**UCO::RequestDescripti on**	**user_info**<string> Supplied by client [Default is empty] **request_type**<string> "SubmitQueryOrderRequest" **request_info**<string> Supplied by server [Default is empty] **request_details**<UCO::NameValueList> *query*<Query> *life_span*<QueryLifeSpan> *o_type*<OrderType> *order*<QueryOrderContents> *GeographicDatum*<string>

Table D- 23 Properties, values and default settings for non-standard protocol and ports

PROPERTY	VALUES (type=String*)	DEFAULT
PROTOCOL	HTTP, HTTPS **	HTTP
PORT	21**, 80, 443	*Corresponding to the default protocol*

Notes:

* Both the PROTOCOL and PORT values are expected to be of type String

** The support of FTP is disallowed with Edition 3 of STANAG 4559 and therefore does not support backward compatibility with STANAG 4559 Ed1 and 2.

APPENDIX 1.

RECOMMENDED PRACTICES AND CLARIFICATION INFORMATION

1. ORDERCONTENTS STRUCTURE

The intent of this section is to provide additional or clarifying information on specific interfaces, methods, or files (e.g., formats, products, data types, etc.) to assist developers. Interpretations, recommended procedures, or clarifications are provided as 'value-added' to the limiting structure of Annex D Interface Details, section 2.5.2 and of the tables therein.

The OrderContents structure information is redundant to some of the tables in Annex D., section 2.5.2. The information provided is intended to provide clarification beyond that in the tables, which are limited due to the table formats.

1.1. Media Types. Encodings for standard media types are based on NIIA structures to be used in NSIL Interface implementations. Media types encodings must be coordinated between the NIIA formats, therefore further encodings are to be coordinated through the Metadata Harmonization Technical Support Team to make appropriate reference to the ISO TC211 Geographic Information Standards, including those on Metadata.

1.2. Image Specifications. The policy for cataloging sets of imagery is based on GIAS; each image from a set should be cataloged individually and a separate catalog entry is made to describe the set as a composited unit. Table D-1- 5 order.prod_list.aSpec <AlterationSpec> and Table D-1- 6 order.prod_list.aSpec.ps <ImageSpec> support backward compatibility in this STANAG to the imagery-based implementations.

Table D-1- 1 order <OrderContents>

Element/Type	Desc/Value(s)
originator <string>	UserID of person or system submitting order. Supplied by client.
tSpec <TailoringSpec>	As defined in GIAS. Not Used where cardinality is (0..1)
pSpec <PackagingSpec>	Supplied by client See Table D-1- 2 order.pSpec <PackagingSpec> for details.
needByDate <UCO::AbsTime>	Date product is needed by. As defined in UCOS.
operatorNote <string>	
orderPriority <short>	Priority to be assigned to this order. As defined in GIAS section 2.3.4.3 the priority system is a string value range 1-100 where value = 1 is a crisis or highest priority, and value = 100 is the Null value.
prod_list <ProductDetailsList>	Sequence of ProductDetails structures. See Table D-1- 3 order.prod_list <ProductDetailsList> for details.
del_list <DeliveryDetailsList>	Sequence of DeliveryDetails structures. See Table D-1- 7 order.del_list <DeliveryDetailsList> for details.

Table D-1- 2 order.pSpec <PackagingSpec>

Element/Type	Desc/Value(s)
package_identifier <string>	An identifier for the package so the client can identify the package when it arrives.
packaging_format_ and_compression <string>	Requested format and compression for ordered Product(s). See Table D-1- 9 Packaging Format and Compression for details.

Table D-1- 3 order.prod_list <ProductDetailsList>

Element/Type	Desc/Value(s)
mTypes <MediaTypeList>	A list of structures of type MediaType, these are as defined in GIAS and are also coordinated with the NIIA to reflect the Harmonized Metadata specifications for NIIA. Only relevant for a DestinationType type of PHYSICAL. Table D-1- 4 order.prod_list.mTypes <MediaTypeList>describes the MediaTypeList. Allows media type to be specified on a per product basis. Semantics for a single <MediaTypeList> is contained within the sequence. Table D-1- 11 defines Media Types.
benums <UCO::NameList>	Not Used.
aSpec <AlterationSpec>	Desired format and compression (where applicable) for Product. See Table Table D-1- 5 order.prod_list.aSpec <AlterationSpec> for details.
aProduct <UID::Product>	Product object reference. Note: UID::Product when returned always represents the entire product, whether it is a single image product, a multi-image product, or other ISR product type.
info_system_name <string>	Not Used

Table D-1- 4 order.prod_list.mTypes <MediaTypeList>

Element/Type	Desc/Value(s)
media_type <string>	See Table D-1- 11 for Media Types and encodings Note: Media type must be coordinated between the NIIA formats. Therefore solution to this field is requested from the Metadata Harmonization Technical Support Team to devise an appropriate reference to the ISO TC211 Geographic Information Standards, including those on Metadata.
quantity <unsigned short>	Number of copies.

Table D-1- 5 order.prod_list.aSpec <AlterationSpec>

Element/Type	Desc/Value(s)
pf <ProductFormat>	Format of the delivered product. Usually, “NSIF1.0” shall be used for products found via the NSIL_CORE data view used for NSIL Minimum Data Model. Product formats for non-imagery ISR products may be defined in a future edition of this STANAG.

Element/Type	**Desc/Value(s)**
ps <ProductSpec>	For secondary imagery or NSIF formatted products this CORBA Any contains an ImageSpec data type. See Table D-1- 6 order.prod_list.aSpec.ps <ImageSpec>. For non-NSIF products, if no alteration is required, fill this field with an empty string.
sub_section <GeoRegion>	Not used. (This information is to be specified in the ImageSpec. See Table D-1- 6)
geo_region_type <GeoRegionType>	Not used. (This information is to be specified in the ImageSpec. See Table D-1- 6)

Table D-1- 6 order.prod_list.aSpec.ps <ImageSpec>

Element/Type	Desc/Value(s)
NOTE: This table is applicable to secondary imagery or NSIF formatted products.	
imgform <ImageFormat> typedef string	For NSIF images, recommend using the STANAG 4545 image subheader IMODE values (plus “A” for “As is”) to indicate format of uncompressed images. Valid values are: “A” | -As is “B” | -Band Interleaved by Block, for single band images “P” | -Pixel Interleaved by Pixel, for multiple band images “R” | -Band Interleaved by Row, for multiple band images “S” | -Band Sequential, for multiple band images
imageid <ImageUniqueIdentifier> typedef string	Alphanumeric identification code associated with an image.
comp <Compression> typedef string	IMAGE-COMPRESSION-Codes: "A" | -As is “NC” | -Not Compressed “C1” | -Bilevel “C3” | -JPEG “C4” | -Vector Quantization “C5” | -Lossless JPEG "C6" | -Future multi-component compression algorithm "C7" | -Future complex SAR "C8" | -Future JPEG2000 “I1” | -Downsampled JPEG See STANAG 4545, Annex C, paragraph 17&18 for details.
bpp <BitsPerPixel> typedef short	The maximum number of significant bits for the value in each band of each pixel in which the dataset can be delivered. bpp=0 represents no change
algo <Algorithm> typedef string	Not used. “ “ (Insert an empty string.)
rrds <RsetList> sequence < short >	The degree of reduction for which an image can be manipulated for delivery. Valid values are 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 An empty RsetList means include all available rsets
sub_section <GeoRegion>	Defines geographic subset or “chip” [optional] As defined in GIAS.
geo_region_type <GeoRegionType>	Defines five different types of geo-regions: two line sample types, one lat_lon in decimal degrees and two special cases as defined in GIAS. For the LAT_LON type, the GeoRegion is always in the WGS-84 coordinate system.

Element/Type	Desc/Value(s)
encoding <SupportDataEncoding>	ASCII

Table D-1- 7 order.del_list <DeliveryDetailsList>

Element/Type	Desc/Value(s)
dests <Destination>	To be one of the following types as defined in GIAS: For DestinationType HTTPS: UCO::FileLocation For DestinationType EMAIL: UCO::EmailAddress For DestinationType PHYSICAL: PhysicalDelivery Semantics for a single <Destination> contained within the sequence.
receiver <string>	UserID of person or system receiving order.
shipmentMode <string>	Used to specify the courier service on physical deliveries.

Table D-1- 8 order <QueryOrderContents>

Element/Type	**Desc/Value(s)**
originator <string>	UserID of person or system submitting order. Supplied by client.
tSpec <TailoringSpec>	As defined in GIAS. Not used where cardinality is (0..1)
pSpec <PackagingSpec>	Supplied by client. See Table D-1- 2 order.pSpec <PackagingSpec> for details.
operatorNote <string>	
orderPriority <short>	Priority to be assigned to this order. As defined in GIAS section 2.3.4.3 the priority system is a short value range 1-100 where value = 1 is a crisis or highest priority, and value = 100 is the Null value.
aSpec <AlterationSpec>	Desired format and compression (where applicable) for Product. See Table D-1- 5 order.prod_list.aSpec <AlterationSpec> for details.
del_list <DeliveryDetailsList>	Sequence of DeliveryDetails structures. See Table D-1- 7 for details.

2 IMAGE AND GROUND COORDINATES

The meanings of values of “x” and “y” in the UCO::Coordinate2d data type are not currently specified in the UCOS or GIAS documents. Similarly, the meanings of the “upper left” and “lower right” corners of a rectangle are not currently specified in those documents. The following sections provide definitions when they are used for image coordinates and for ground coordinates.

2.1 Image Coordinates.

When image coordinates are being used, the values of “x” and “y” in the data type UCO::Coordinate2d are interpreted to mean:

2.1.1 The terms used below for the two image pixel indices are “row” and “column” (as used in the NSIF and not dependent on the image sensor technology).

2.1.2 The reference point for image coordinates is the corner of the (full, chippable, or partial) image where the row and column pixel indices are (0, 0). Row and column pixel indices of (0.00, 0.00) apply to the outside corner of this corner pixel (not to the center of this pixel).

2.1.3 The pixel column index is used for the “x” image coordinate value, and the pixel row index is used for the “y” coordinate.

2.1.4 The row and column pixel indices (or “y” and “x”) are in pixel spacing (or pixel) units in the referenced (full resolution, chippable, or reduced resolution) image. (These indices are not given in tiles or FAF 1024x1024 pixel blocks.)

2.1.5 Values for the row and column pixel indices (or “y” and “x”) can be given in either IDL:long or IDL:double data formats.

2.1.6 When GIAS::GeoRegion data is used in a GIAS::AlterationSpec with a GIAS::GeoRegionType value of LINE_SAMPLE_CHIP, all the pixels in the full resolution image are extracted that fall within the pixels specified in the Chippable Image.

2.1.7 When image coordinates are being used, the “upper left” corner of a UCO::Rectangle is the corner where the row and column pixel indices have their numerically smallest values (closest to 0, 0). The “lower right” corner is the diagonally opposite corner, where the row and column pixel indices each take on their maximum values. When integer valued row and column numbers are used for the “lower right” corner, the referenced pixel is included in the rectangle retrieved. When floating point valued row and column numbers are used for the “lower right” corner, those numbers are first rounded to the closest integer, before any conversion to positions in the full resolution image. The pixel referenced by the rounded values is included in the rectangle retrieved.

2.2 Ground Coordinates.

When ground coordinates are being used, the values of “x” and “y” in the UCO::Coordinate2d data type and BQS query are interpreted to mean:

2.2.1 All latitude and longitude values are in the WGS-84 ground coordinate reference system (whenever the related Geographic_datum property has the value “WGS84”, or defaults to that value).

2.2.2 The longitude is used for the “x” ground coordinate value, and the latitude is used for the “y” value.

2.2.3 Longitude values can range from -180 degrees to +180 degrees. Longitude values are positive East of the Greenwich prime meridian, and are negative West of the Greenwich meridian.

2.2.4 Latitude values can range from -90 degrees to +90 degrees. Latitude values are positive North of the equator, and are negative South of the equator.

2.2.5 The longitude and latitude (or “y” and “x”) are in degrees units when recorded as single numbers (not radians or arc-seconds). Note that in the BQS, longitude and latitude can be alternately specified in degrees, minutes, and seconds.

2.2.6 Values for longitude and latitude (or “y” and “x”) can be given in either IDL:long or IDL:double data formats.

2.2.7 When ground coordinates are being used, the “upper left” corner of a UCO::Rectangle and similar data types is interpreted to mean the Northwest corner of the desired rectangle. The “lower right” corner is the diagonally opposite corner, or Southeast corner. (Note that a rectangle cannot be specified to surround the North or South Pole.)

3 DATA MANAGEMENT

3.1 Packaging Specifications. The following describes more specifically the list of packaging specifications to be used by the OrderMgr.

3.1.1 File Formats and Compression. Table D-1- 9 Packaging Format and Compression identifies currently available package options that an NSIL Interface implementation will recognize. The encoded references for compressed and uncompressed file formats are standardized for use in the NIIA to ensure interoperability; minimum compliance to STANAG 4559 requires servers to support .tar (TARUNC) and zip (FILESZIP) compression. These are used in the OrderMgr and QueryOrderMgr packaging specifications.

Table D-1- 9 Packaging Format and Compression

Package format	Description	Filename suffix	Minimum compliance
TARUNC	TAR UNCompressed consists of multiple files into one file with a .tar extension.	.tar	X
FILESUNC	Files UNCompressed is returned as a filelist, no bundling required.	<none>	
TARZIP	TAR ZIPped consists of multiple files into one file with a .tar extension, then that file will be zipped into another file with a .zip extension (tar.zip).	.tar.zip	
FILESZIP	FILES ZIPped consists of multiple files zipped into one archive and shipped with a .zip extension.	.zip	X
TARGZIP	Consists of multiple files into one tar file with a .tar extension, and then the tar file is gzipped. The resulting extension is .tar.gz.	.tar.gz	
FILESGZIP	Consists of multiple files which are individually gzipped. Each will get a .gz extension.	.gz	
TARCOMPRESS	Multiple files are tarred into one single file with a .tar extension, and the result is compressed with the standard Unix compress utility. The resulting extension is .tar.Z.	.tar.Z	
FILESCOMPRESS	Multiple files are compressed individually, each with a .Z extension. Multiple files to multiple files.	.Z	

3.1.2 File Naming conventions to the DeliveryManifest

File Packaging Actions presented in Table D-1- 10 define how to name the files delivered by the OrderMgr and QueryOrderMgr, and how to represent this in the DeliveryManifest structure.

3.1.2.1 The “FILE NAME” column references the field order.del_list.dests.f_dests.file_name. If this field is specified in the delivery details, and is not empty, Table D-1- 10 names this “Specified”.

3.1.2.2 The “PACKAGE ID” column refers to the field order.pSpec.package_identifier.

3.1.2.3 The “PACKAGE FORMAT AND COMPRESSION” column refers to the field order.pSpec.packaging_format_and_compression. “Packaged” in this column refers to formats that package all files in a single file, like TAR or ZIP. “Individual” refers to formats that ship each product as a separate file. The descriptor “compressed” refers to formats that compress the file, e.g. ZIP or GZIP.

3.1.2.4 The “PACKAGE FILE NAME” column identifies the file name to use for the file(s) shipped to the user. For “Individual” formats multiple files can be shipped. In the “xx.yy” placeholder “xx” is a sequence number starting at 1 and “yy” is the total number of files. Table D-1- 9 Packaging Format and Compression defines the filename suffix for formats.

3.1.2.5 The “INTERNAL FILENAME: column refers to the name used within a TAR or ZIP file. It only applies to “Packaged” formats. In the DeliveryManifest produced by an order the PackageElements must be filled with the “PACKAGE FILE NAME”.

Table D-1- 10 File Packaging Actions

FILE NAME	PACKAGE ID	PACKAGE FORMAT AND COMPRESSION	FILE COUNT	PACKAGE FILE NAME	INTERNAL FILE NAME
Specified	Don't care	Individual & uncompressed	1	<filename>	Not used
Specified	Don't care	Individual & uncompressed	Many	<filename>.xx.yy	Not used
Specified	Don’t care	Individual & compressed	1	<filename>.<suffix>	Not used
Specified	Don’t care	Individual & compressed	Many	<filename>.xx.yy.<suffix>	Not used
Not Specified	Specified	Individual & uncompressed	1	<packageid>	Not used
Not Specified	Specified	Individual & uncompressed	Many	<packageid>.xx.yy	Not used
Not Specified	Specified	Individual & uncompressed	1	<packageid>.<suffix>	Not used
Not Specified	Specified	Individual & uncompressed	many	<packageid>.xx.yy.<suffix>	Not used
Not Specified	Not Specified	Individual	Don’t care	<server discretion>	Not used
Specified	Don’t care	Packaged	Don’t care	<filename>.<suffix>	<server discretion>
Not Specified	Specified	Packaged	Don’t care	<packageid>.<suffix>	<server discretion>
Not Specified	Not Specified	Packaged	Don't care	ILLEGAL	ILLEGAL

3.2 Media Type Encodings. Encodings for standard media types are based on NIIA structures to be used in NSIL Interface implementations. Media types encodings must be coordinated between the NIIA formats; therefore, further encodings are to be coordinated through the Metadata Harmonization Technical Support Team to make appropriate reference to the ISO TC211 Geographic Information Standards, including those on Metadata.

Table D-1- 11 Media Types

Media Reference Name	Desc/References
“4MM”	4 Millimeter Tape
“8MM”	8 Millimeter Tape
“CCT”	Computer Compatible Tape
“CD_ROM”	Compact Disc - Read Only Memory
“FILM”	Film
“Floppy”	3 ¼” Floppy Disk
“H/C”	Hard Copy
“D2C”	Large Volume 150 GB Cartridge
”DVD-R”	Digital Versatile Disk dash R
”DVD+R”	Digital Versatile Disk plus R
“ID1_SMALL”	19mm small cassette, STANAG 4238, MIL-STD-2179B
“ID1_MEDIUM”	19mm medium cassette, STANAG 4238, MIL-STD-2179B
“ID1_LARGE”	19mm large cassette, STANAG 4238, MIL-STD-2179B
“Hi8_DIGITAL”	Digital Tape, STANAG 4238, MIL-STD-2179(AS)
“DCRSI”	Digital 1 inch Tape, 240 Mb/s

3.3 Related File Types. The NSIL Interface recognizes standardized related file types that may be supplied with imagery data and used for rapid imagery assessment prior to request, retrieval or exploitation of a full image.

Table D-1- 12 Standard Related File Types

Type	Description
THUMBNAIL	A representative JPEG, PNG or GIF image that is128 x 128 pixels in size.
OVERVIEW	A low resolution, STANAG 4545 formatted image. Maximum image dimension 1024x1024 pixels.

4 UID::PRODUCT USAGE GUIDANCE

This section provides additional information to that contained in the UCOS to assist developers in the interpretation of the Product interfaces intended usage.

4.1 UID Product Interface. The UID::Product interface serves as a reference handle for a data set or product. It applies to a data set or product in a specific IPL and is valid only at that IPL. Product references must be unique within a given server and must be valid for the lifetime of that product within that server. It is not intended to provide a globally unique identifier and attempting to use a Product reference generated by a different IPL will result in unspecified behavior.

4.2 Product association with ORB method. There are a number of mechanisms through which to associate a product reference with a data set or product. The basic principle is to attach a product specific identifier to the product object using an appropriate ORB method for specifying object names or markers. The solution will be implementation dependent and environment specific. This does not compromise interoperability for the reasons stated above.

5 DAG STRUCTURE FOR NSIL

Only the attributes requested in results_attributes will be included in results Directed Acyclic Graph (DAG) structures.If no results attributes are specified, the DAG structure will contain only the root node (containing the product reference) for each result. Where a requested attribute is null, not known or not available it will be omitted from the DAG.

6 DAG STRUCTURE FOR STANAG 4559 EDITION 3 METADATA MODEL

6.1 Data Model Hierarchy. The STANAG 4559 Edition 3 Data Model introduces a hierarchical structure of related entities in all views. In this structure, it must be possible to distinguish the attributes from each of the entities in the DAG structure. Note that this DAG structure is different from NSIL_CORE version of STANAG 4559 Editions 1 and 2 since the Edition 3 Data Model is hierarchical.

The list of nodes contains a root node of type ROOT_NODE representing the top of the DAG. Its name is the name of the main product entity of the view being queried, NSIL_PRODUCT for NSIL_ALL_VIEW view and all other product views, and NSIL_ASSOCIATION for NSIL_ASSOCIATION_VIEW.

The value of the root node is the UID::Product reference for that specific product.

6.2 NODES in the DAG. In the Edition 3 Metadata Model (Annex G) and the Metamodel (Annex H) the main entity has no further attributes and therefore contains no nodes of type ATTRIBUTE_NODE as its children. It can only refer to nodes of type ENTITY_NODE, which represent a related entity, or to nodes of type RECORD_NODE, which represent Products. The ROOT_NODE of a DAG is therefore a special case of a RECORD_NODE. Every entity node can refer to attribute nodes for each of its attributes, and to other entity nodes for its related entities.

6.3 Node Linking. Nodes of type ENTITY_NODE have as name the simple name of the entity without any prefix. Nodes of type ATTRIBUTE_NODE have as name the simple name of the attribute, also without any prefix. Some entity and attribute names can thus appear in multiple nodes, even though they are conceptually different.

The nodes in the DAG are linked using the edges, where each node of type ATTRIBUTE_NODE is the endpoint of an edge with a startpoint being a node of type ENTITY_NODE. An ENTITY node is always the endpoint for an edge with a startpoint either another ENTITY_NODE or a RECORD_NODE (with the ROOT_NODE being a special case of a RECORD_NODE).

In general RECORD_NODE may contain the UID::Product from either the association or the originating product.

6.4 DAG Associations and Products. The DAG Structure of associations is very similar to the DAG Structure of the products, as shown in Fig D-1-2, Example of DAG Edges.

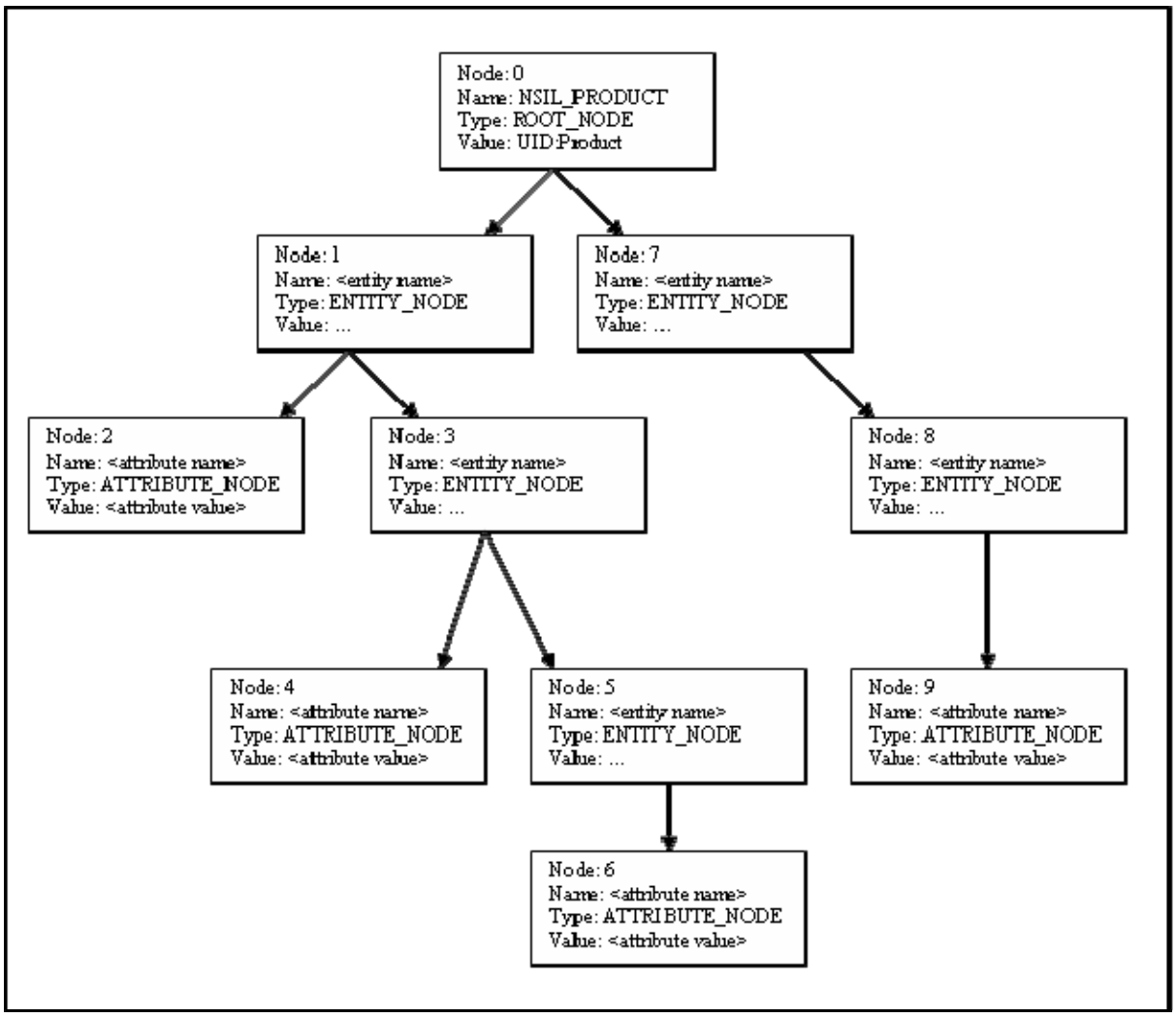

Figure D-1- 1 Example of a DAG with attributes from multiple (sub)-entities

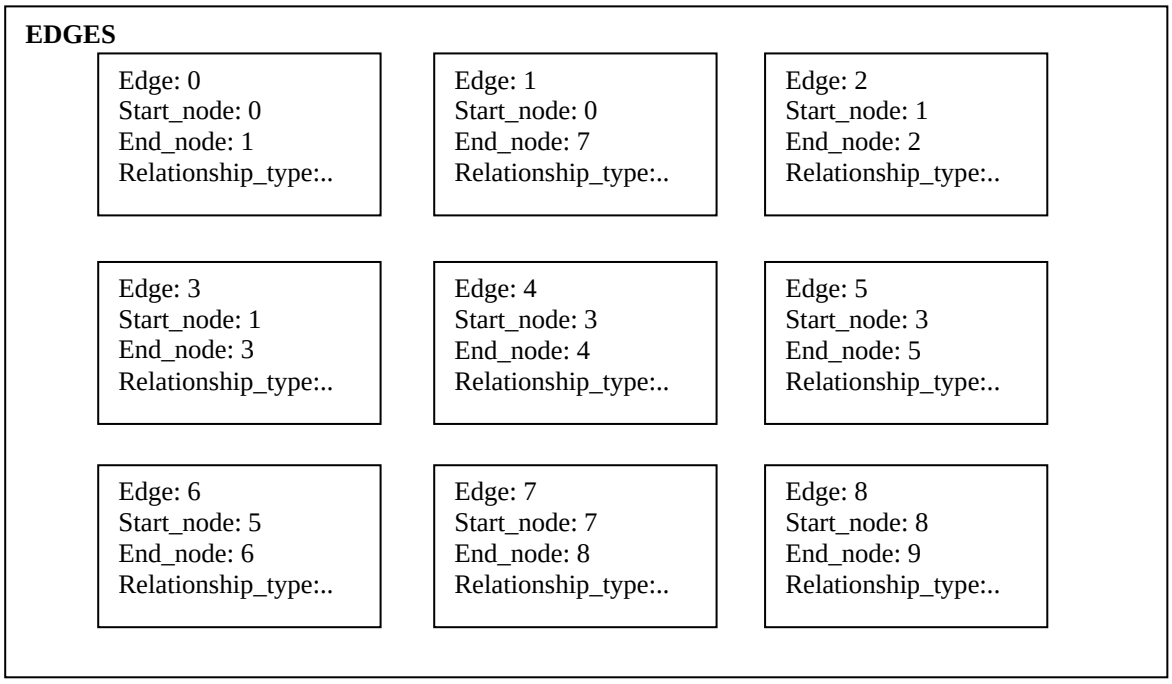

Figure D-1- 2 Example of a DAG with edges

6.5 Querying the NSIL_ASSOCIATION_VIEW

When querying the NSIL_ASSOCIATION_VIEW view, the DAG structure is very similar to the NSIL_ALL_VIEW and other product views as shown in the figure above. The list of nodes contains a root node of type ROOT_NODE representing the top of the DAG. Its name is the name of the main association entity. The value of the root association node is the UID::Product reference for that specific association. The association will have nodes of type ENTITY_NODE for related entities, which can, as in the previous DAG, contain nodes of type ATTRIBUTE_NODE for their attributes and nodes of type ENTITY_NODE for their related entities. In addition, the root node holds the source and destination(s) references of this association product as nodes of type RECORD_NODE. Their names will be either NSIL_SOURCE or NSIL_DESTINATION and as value, they may contain UID::Product of the associated product. These nodes of type RECORD contain as their children, the same entity-attributes-sub-entities structure as in the DAG shown above.

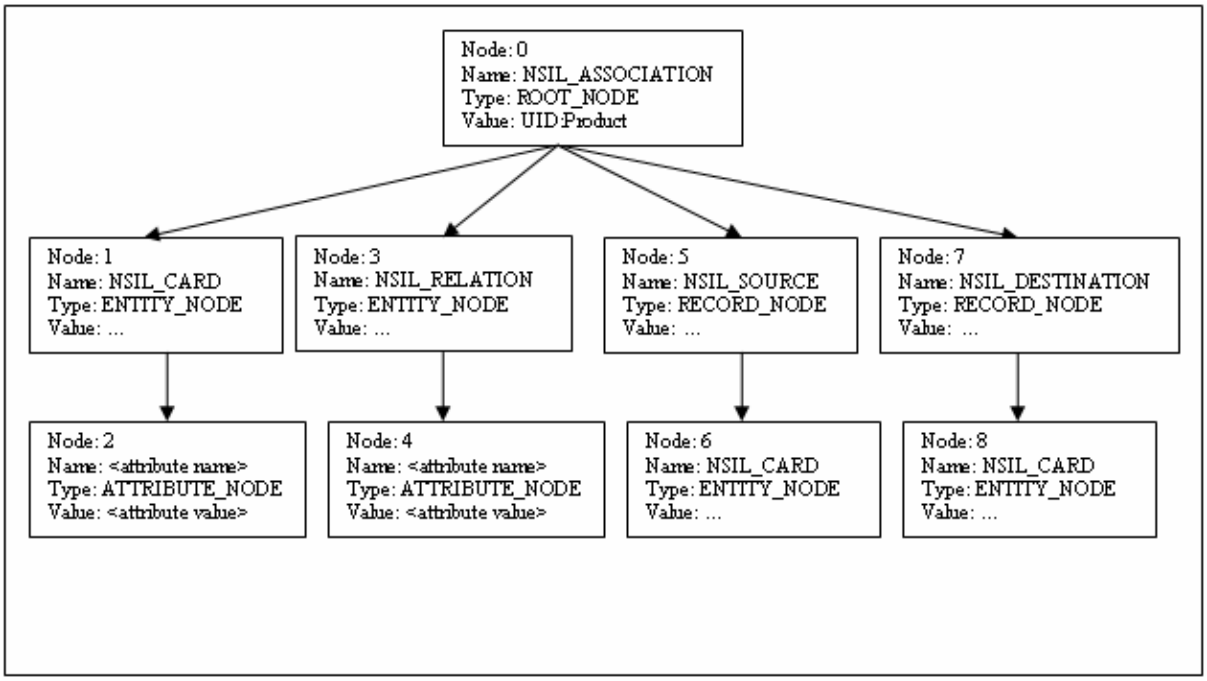

Figure D-1- 3 Example of DAG for NSIL_ASSOCIATION view

ANNEX E.

DATA MODEL ATTRIBUTE TYPES AND DOMAINS

ANNEX E.

DATA MODEL ATTRIBUTE TYPES AND DOMAINS

TABLE OF CONTENTS

PARAGRAPH	TITLE	PAGE
1.	Purpose …….……..	E-1
2	Data Model Attribute Types and Domains ……..	E-1
2.1.	Data Types ……..……	E-1
2.2	The Data Model…….	E-3
3.	Data Domains ……	E-3
3.1.	Relationships between Data Types and Data Domains …….	E-4
3.2	Attribute Information…….	E-5

TABLES	TITLE	PAGE
Table E- 1	Data Types	E-1
Table E- 2	Data Domains	E-3
Table E- 3	AttributeInformation Structure	E-6

ANNEX E. DATA MODEL ATTRIBUTE TYPES AND DOMAINS

1. PURPOSE

The purpose of this Annex is to identify and describe metadata attributes that must be contained in the NSIL_CORE data view. Collectively these attributes describe the entity "NSIL Interface Image Product". Attributes included in the NSIL_CORE are those required by users of the NSIL interface to:

- Perform queries against library holdings;
- Uniquely identify library holdings and recognize different products based on the same sources, i.e. images;
- Assure that library holdings are protected and distributed in an appropriate manner depending on product classification, associated security compartments, and additional security control and/or handling instructions; and
- Retrieve products via the Direct Access method described in Annex C, section 2.6.4 where this capability is provided by the IPL.

2. DATA MODEL ATTRIBUTE TYPES AND DOMAINS

The following guidance applies for data model attribute types and domains.

2.1. Data Types: The AttributeType enumeration defines the valid data types in GIAS 2.2.2.9:

Table E- 1 Data Types

AttributeType	Data Type definition
TEXT:	Represents CORBA:String underlying data type.
INTEGER:	Represents CORBA:Long.
FLOATING_POINT:	Represents CORBA:Double.
UCOS_COORDINATE:	Represents UCO:Coordinate2d, defined in UCOS 2.2.2.1. Underlying data representation is a structure.
UCOS_POLYGON:	Represents UCO:Polygon, defined in UCOS 2.2.2.3. Underlying data representation is a sequence of UCOS:Coordinate2d.
UCOS_ABS_TIME:	Represents UCO:AbsTime, defined in UCOS 2.2.1.5. Underlying data representation is a structure.
UCOS_RECTANGLE:	Represents UCO:Rectangle, defined in UCOS 2.2.2.4. Underlying data representation is a structure.
UCOS_SIMPLE_GS_IMAGE:	Represents UCO:SimpleGSImage, defined in UCOS 2.2.3.2. Underlying data representation is a structure. Does not appear in the data models; not expected to be used.
UCOS_SIMPLE_C_IMAGE:	Represents UCO:SimpleCImage, defined in UCOS 2.2.3.3. Underlying data representation is a structure. Does not appear in the data models; not expected to be used.

AttributeType	Data Type definition
UCOS_COMPRESSED_IMAGE:	Represents UCO:CompressedImage, defined in UCOS 2.2.3.3. Underlying data representation is a structure. Does not appear in the data models; not expected to be used.
UCOS_HEIGHT:	Represents UCO:Height, defined in UCOS 2.2.2.4. Underlying data representation is the UCOS:LinearDimension structure.
UCOS_ELEVATION:	Represents UCO:Elevation, defined in UCOS 2.2.2.4. Underlying data representation is the UCOS:LinearDimension structure.
UCOS_DISTANCE:	Represents UCO:Distance, defined in UCOS 2.2.2.4. Underlying data representation is the UCOS:LinearDimension structure.
UCOS_PERCENTAGE:	Represents UCO:Percentage, defined in UCOS 2.2.1.4. Underlying data representation is a CORBA:Float.
UCOS_RATIO:	Represents UCO:Ratio, defined in UCOS 2.2.1.4. Underlying data representation is a structure. Does not appear in the data models; not expected to be used.
UCOS_ANGLE:	Represents UCO:Angle, doubly defined (consistently) in UCOS 2.2.1.4 and UCOS 2.2.2.4. Underlying data representation is a CORBA:Double.
UCOS_FILE_SIZE:	Represents UCO:FileSize, defined in UCOS 2.2.1.3. Underlying data representation is a CORBA:Double.
UCOS_FILE_LOCATION:	Represents UCO:FileLocation, defined in UCOS 2.2.1.3. Underlying data representation is a structure.
UCOS_COUNT:	No representation, no longer used.
UCOS_WEIGHT:	Represents UCO:Weight, defined in UCOS 2.2.2.4. Underlying data representation is a CORBA:Double. Does not appear in the data models; not expected to be used.
UCOS_DATE:	Represents UCO:Date, defined in UCOS 2.2.1.5. Underlying data representation is a structure.
UCOS_LINESTRING:	Represents UCO:LineString2d, defined in UCOS 2.2.2.2. Underlying data representation is a sequence of UCOS:Coordinate2d. Does not appear in the data models; not expected to be used.
UCOS_DATA_RATE:	Represents UCO:DataRate, defined in UCOS 2.2.1.4. Underlying data representation is a CORBA:Double.

AttributeType	Data Type definition
UCOS_BIN_DATA:	Represents UCO:BinData, defined in UCOS 2.2.1.11. Underlying data representation is a sequence of CORBA:Octet. Appears in data models as UCOS:BIN_DATA.
BOOLEAN_DATA:	Represents: CORBA:Boolean.
UCOS_DURATION:	Represents UCO: Duration, defined in UCOS 2.2.1.5. Underlying data representation is the UCOS:Time structure.

2.2. The data model data dictionaries identify each attribute's Data Type, using the nomenclature from the UCOS (e.g., UCOS:Polygon, UCOS:Angle), or the CORBA nomenclature (e.g., CORBA:String, CORBA:Long).

3. DATA DOMAINS

The DomainTypes enumeration and Domain case switch defines the valid data domains in GIAS 2.2.2.9:

Table E- 2 Data Domains

Domain Type	Domain Case Switch
DATE_VALUE	Indicates that the domain for the attribute is dates, and defines earliest and latest date range boundary for data validation. In practice, the earliest and latest date ranges are left null to leave date attributes unconstrained.
TEXT_VALUE	Indicates that the domain for the attribute is text strings, and defines the maximum length of the string.
INTEGER_RANGE	Indicates that the domain for the attribute is integer numbers and defines upper and lower integer values for data validation.
FLOATING_POINT_RANGE	Indicates that the domain for the attribute is floating-point numbers and defines upper and lower number values for data validation.
INTEGER_VALUE	Indicates that the domain for the attribute is integer numbers and that any integer is valid for the attribute. Should be used when the data model data dictionary specifies an integer data type with no range defined. Clients or libraries should perform no range checking for the attribute. Uses the same structure as INTEGER_RANGE, which includes upper and lower values that should be ignored for this domain.

Domain Type	Domain Case Switch
FLOATING_POINT_VALUE	Indicates that the domain for the attribute is floating-point numbers and that any number is valid for the attribute. Should be used when the data model data dictionary specifies a floating-point data type with no range defined. Clients or libraries should perform no range checking for the attribute. Uses the same structure as FLOATING_POINT_RANGE, which includes upper and lower values that should be ignored for this domain.
LIST	Indicates that the domain for the attribute is a specific list of values and defines the list of valid values.
ORDERED_LIST	Indicates that the domain for the attribute is a specific list of ordered values and defines the list of valid values. This domain is not used.
GEOGRAPHIC	Indicates that the domain for the attribute is geospatial and defines a bounding rectangle describing the valid geographic area. In practice, the geographic rectangle is left null to leave the geographic coordinates unconstrained.
INTEGER_SET	Indicates that the domain for the attribute is sets of integers and defines a sequence of integer ranges describing the ranges of valid values. This domain is not used.
FLOATING_POINT_SET	Indicates that the domain for the attribute is sets of floating-point numbers and defines a sequence of float ranges describing the ranges of valid values. This domain is not used.
GEOGRAPHIC_SET	Indicates that the domain for the attribute is sets of geographic areas and defines a list of bounding rectangles describing the valid geographic areas. This domain is not used.
BINARY_DATA	Indicates that the domain for the attribute is binary data and defines a binary bitmask for the data. In practice, the bitmask is ignored.
BOOLEAN_VALUE	Indicates that the domain for the attribute is True/False.

3.1. Relationships between Data Types and Data Domains

3.1.1. TEXT attributes should use either the TEXT_VALUE or LIST domains.

INTEGER attributes should use:

INTEGER_RANGE domain if the data model data dictionary indicates that the attribute should be validated within a specified range of values
INTEGER_VALUE domain if the attribute has no specified range of values and should be invalidated

3.1.2. FLOATING_POINT, UCOS_HEIGHT, UCOS_ELEVATION, UCOS_DISTANCE, UCOS_PERCENTAGE, UCOS_ANGLE, UCOS_FILE_SIZE, and UCOS_DATA_RATE attributes should use:

FLOATING_POINT_RANGE domain if the data model data dictionary indicates that the attribute should be validated within a specified range of values
FLOATING_POINT_VALUE domain if the attribute has no specified range of values and should be invalidated

3.1.3. UCOS_HEIGHT, UCOS_DISTANCE, UCOS_FILE_SIZE, and UCOS_DATA_RATE attributes should use the FLOATING_POINT_RANGE domain with a lower bound of zero and an upper bound of the maximum value for a CORBA:Double unless the data model data dictionary indicates that the attribute should have a different upper bound value.

3.1.4. UCOS_PERCENTAGE attributes should user the FLOATING_POINT_RANGE domain with a lower bound of zero and an upper bound of 100.0.

3.1.5. UCOS_ANGLE attributes should use:

FLOATING_POINT_RANGE domain with the range of values specified in the classified data model data dictionary if the classified data model is being implemented
FLOATING_POINT_VALUE domain if the unclassified data model is being implemented

3.1.6. UCOS_COORDINATE, UCOS_POLYGON, and UCOS_RECTANGLE attributes should use the GEOGRAPHIC domain.

3.1.7. UCOS_ABS_TIME, UCOS_DATE, and UCOS_DURATION attributes should use the DATE_VALUE domain.

3.1.8. UCOS_FILE_LOCATION attributes should use the TEXT_VALUE domain.

3.1.9. BOOLEAN_DATA attributes should use the BOOLEAN_VALUE domain.

3.1.10. UCOS_BIN_DATA attributes should use the BINARY_DATA domain.

3.2. Attribute Information.

The DataModelMgr uses the AttributeInformation structure to provide information about attributes.

Table E- 3 AttributeInformation Structure

Element	Type
attribute_name	=<string>
attribute_type	=<AttributeType>
attribute_domain	=<Domain>
attribute_units	=<string>
mode	=<RequirementMode>
description	=<string>
sortable	=<Boolean>
updateable	=<Boolean>

ANNEX F.

BOOLEAN QUERY SYNTAX

ANNEX F.

BOOLEAN QUERY SYNTAX

TABLE OF CONTENTS

PARAGRAPH	TITLE	PAGE
1.	Introduction on BQS……..	F-1
2.	The BQS Factor ……..	F-1
3.	Rules for BQS Query Type …….……	F-2
4.	BNF Definition……..……	F-2
4.1	BNF Rules and Constraints…….……	F-3
4.2	Boolean as constant_expression……	F-6

TABLES	TITLE	PAGE
Table F- 1	Mapping Between BQS and UCOS Data Structures.....	F-5

ANNEX F. BOOLEAN QUERY SYNTAX

1 INTRODUCTION ON BQS

The Boolean query syntax (BQS) is a key part of the specification of the NSILI. The intent of the BQS is to formally define the syntax for queries made on geospatial catalogs. It is necessary to define the BQS in the NSILI Standardization Agreement to “decouple” the interfaces used for querying from the implementation details of the catalog. For example, the BQS allows a client to interact with a geospatial catalog in a uniform way regardless of the database or database type underlying the catalog implementation, the native query language of the database and the physical schema or data model of the database. This approach has the dual benefit of simplifying the generation of queries by the client while not constraining the catalog developers in the design choices for the implementation. The catalog implementers must, however, provide the capability to translate the BQS into whatever query language and physical schema they have chosen. A discussion of BQS Semantics will be found in AEDP-5 Edition 2.

2 THE BQS FACTOR

The BQS is based upon the concept of an attribute-operator-value triplet called a factor. Each factor represents a condition of interest to the client. These factors can be assembled into a complete query by relating the factors with the Boolean operators “and”, “or”, and “not”. BQS constructs are case insensitive and BQS logical operators on strings and expressions are case sensitive. The formal definition of the syntax of the BQS, described in Backus-Naur Form (BNF), is detailed in section 3 of this Annex.

2.1 Notes:

2.1.1 Data model attributes (in Entity.Attribute notation) are not case-sensitive. Servers should not expect clients to return data model attribute references in any particular case. However, servers were expected to provide entity and attribute names via the DataModelMgr using the same mixed case form as the Data Model Data Dictionary.

2.1.2 BQS operators are case-sensitive.

2.1.3 Other BQS constructs (hemisphere notation, units, geographic shape description, etc.) are not case-sensitive.

2.1.4 Quoted strings are intended to be not case-sensitive, but not all Relational Database Management Systems (RDBMS) support this capability. Implementations have accepted that a server may handle quoted strings as case-sensitive for performance reasons, although this is not recommended practice.

2.1.5 The most recent GIAS specification was incomplete for date queries – this has been corrected.

2.1.6 Simple attributes are attributes with numeric or date domains. All Boolean operators can be used with these attributes. Attributes with Boolean or enumerated list domains are also simple attributes. Boolean values are expressed as quoted strings “TRUE” and “FALSE” and enumerated list data values are also expressed as quoted strings, which is why the BNF supports a quoted string with the Boolean operators. The valid Boolean operators for Boolean or enumerated list attributes are “=” and “<>”. Some of the server implementations would accept use of the Boolean operators with strings, but clients should not rely on this.

2.1.7 Dates are expressed as quoted strings. Date attributes are considered simple attributes and allow all Boolean operators. Dates can be specified with or without a time. When specified without a time, the following meaning is implied:

date_attribute > ‘2010/01/01’ means “dates equal or after 2010/01/02 00:00:00”

date_attribute < ‘2010/01/01’ means “dates before 2010/01/01 00:00:00”

date_attribute = ‘2010/01/01’ means every time on this date, i.e. it is an implicit “IDATIM >= ‘2010/01/01 00:00:00’ AND IDATIM < ‘2010/01/02 00:00:00”

3 RULES FOR BQS QUERY TYPE

The rules that define which query operator may be used for a given attribute are determined by the AttributeType field of the data model attribute definition. Geographic types (also referred to as Geospatial Query Types in this edition of this STANAG), i.e. UCOS_COORDINATE, UCOS_POLYGON, or UCOS_RECTANGLE shall use the geospatial operators, specified in the BNF definition below as "geo_op" and "rel_geo_op". All other queryable attribute types shall allow querying with simple query operators, specified in the BNF definition below as "comp_op". In addition to this, the TEXT attribute type shall allow "like" and "not like" queries.

4 BNF DEFINITION

The Backus-Naur Form (BNF) for the Boolean query syntax is show below. Note that the individual BQS tokens in a BQS statement are separated from each other by a space.

query ::= term { "or" term }
term ::= factor { "and" factor }
factor ::= ["not"] primary
primary ::= (simple_attribute_name comp_op constant_expression)
(geo_attribute_name geo_op geo_element)
| (geo_attribute_name rel_geo_op number dist_units “of” geo_element)
| (text_attribute_name "like" quoted_string)
| (attribute_name “exists”)
| ("(" query ")")

attribute ::= a member of the set of queryable attribute names (defined in the appropriate GIAS profile)
attribute_name ::= attribute | {entity “:”}entity “.” attribute
simple_attribute_name ::= member of subset of attribute_name for which boolean operators (comp_op) are allowed
geo_attribute_name ::= member of subset of attribute_name for which geospatial operators are allowed
text_attribute_name ::= member of subset of attribute_name for which string operators are allowed (“free text search”)
comp_op ::= "=" | "<" | ">" | "<>" | "<=" | ">="
constant_expression ::= number | quoted_string | date
date ::= “” year “/” month “/” day [“<blank>” hour “:” minute “:” second]”””
year ::= digit digit digit digit
month ::= digit digit

day ::= digit digit
hour ::= digit digit
minute ::= digit digit
second ::= digit digit [“.” digit {digit}]
geo_op ::= “intersect” | “outside”| “inside”
rel_geo_op ::= “within” | “beyond”
dist_units ::= “feet” | “meters” | “statute miles” | “nautical miles” | “kilometers” | “METERS” | “FEET”
geo_element ::= point | polygon | rectangle | circle | ellipse | line | polygon_set
sign ::= “+” | “-”
number ::= [sign] digit_seq ["." [digit_seq]]
digit_seq ::= digit { digit }
digit ::= “0” | “1” | “2” | “3” | “4” | “5” | “6” | “7” | “8” | ” 9”
quoted_string ::= """ { character } """ // Single quotes
character ::= “a”|”b”| .... // All printable ASCII characters To use a """ (single quote) use """ (two single quotes)
Del = “,” // Delimiter
latitude ::= number
longitude ::= number
hemi ::= “N” | “S” | “E” | “W”
DMS ::= [digit] digit digit “:” digit digit “:” digit digit “.” digit hemi
latitude_DMS ::= DMS
longitude_DMS ::= DMS
latlon ::= latitude Del longitude | latitude_DMS Del longitude_DMS
coordinate ::= latlon
point :: = “POINT” “(“ coordinate “)”
polygon ::= “POLYGON” “(“ coordinate Del coordinate Del coordinate {Del coordinate}“)”
rectangle ::= “RECTANGLE” “(“ upper_left Del lower_right “)”
upper_left ::= coordinate
lower_right ::= coordinate
circle ::= “CIRCLE” “(“ coordinate Del radius “)”
radius ::= number dist_units
ellipse ::= “ELLIPSE” “(“ coordinate Del major_axis_len Del minor_axis_len Del north_angle “)”
major_axis_len ::= number dist_units
minor_axis_len ::= number dist_units
north_angle ::= number
number is given in decimal degrees
north is 0 and direction is clockwise
range is 0 to 360
line ::= “LINE” “(“ coordinate Del coordinate { Del coordinate} “)”
polygon_set ::= “POLYGON_SET” “(“ polygon { Del polygon} “)”

4.1 BNF Rules and Constraints

The BNF rules are augmented by the following rules and constraints:

4.1.1 Operator Precedence

The order of precedence for the operators defined in the BQS , from high (evaluated first) to low (evaluated last) is :

( ) - parentheses – highest precedence
comp_op, geo_op, rel_geo_op, “like”, “not like”, exists
not
and
or – lowest precedence

Operators of equal precedence are evaluated from left to right within an expression.

4.1.2 Units

The units applicable to the attributes in a BQS query are those specified by the server implementation. This information is available via the AttributeInformation structure for each attribute. The AttributeInformation structures are accessed through the DataModelMgr.

4.1.3 Strings and Wildcards

Wildcard expressions are allowed using the character "%" to denote a match with 0 or more characters and the character “?” to match with exactly one character.

For example the query: name like 'rob%'
would match the following strings: 'rob' 'robert' 'robin'
where the query: name like 'mik?'
would match the following strings: 'mike' 'miki' 'miko'

The "like" and "not like" operators are the only operators used for text expressions and the only operators supporting wildcards.

Wildcards can be used to implement the effect of many character-matching operations, such as: contains, begins with, ends with, not contains, not begins with, not ends with, and so forth. For example:

attribute like '%contains_this%'
attribute like 'begins_with_this%'
attribute like '%ends_with_this'
attribute not like '%will_not_contain_this%'
attribute not like 'will_not_begin_with_this%'
attribute not like '%will_not_end_with_this'

4.1.4 BQS and UCOS/GIAS Types

To abet developers implementing BQS query parsers, the Table F-1 provides a mapping between the above BNF and UCOS data structures.

Table F- 1 Mapping Between BQS and UCOS Data Structures

BQS	UCOS
Point	Coordinate2d
Coordinate	
Latlon	
Latitude	y
Longitude	x
Coordinate	
Latlon	
Latitude	y
Longitude	x
Polygon (collection of lines)	Polygon (collection of Coordinate2d)
Coordinate	
Latlon	
Latitude	y
Longitude	x
Rectangle	Rectangle
	Coordinate2d
upper_left	upper_left
Coordinate	
Latlon	
	Coordinate2d
Latitude	y
Longitude	x
Lower_right	lower_right
Coordinate	
Latlon	
Latitude	y
Longitude	x
Circle	Circle
	centerpoint
Coordinate	
Latlon	
Latitude	y
Longitude	x
Radius	radius
Number	dimension
	units
	Reference_system
Ellipse	Coordinate
	centerpoint
Coordinate	
Latlon	
Latitude	y
Longitude	x
Minor_axis_len	minor_axis_len
Number	dimension
	units

	Reference_system
Major_axis_len	major_axis_len
Number	dimension
	units
	Reference_system
north_angle	north_angle
Number	float
Line	LineString2d
Coordinate	
Latlon	
Latitude	y
Longitude	x
Coordinate	
Latlon	
Latitude	y
Longitude	x
Polygon_set (collection of polygon's)	PolygonSet (collection of Polygon's)

4.2 Boolean as constant_expression.
Also note that attributes of type Boolean should be described as a constant_expression in its quoted string form; i.e. attribute = 'TRUE', or attribute='FALSE'

ANNEX G.

EDITION 3 METADATA MODEL

ANNEX G.
EDITION 3 METADATA MODEL

TABLE OF CONTENTS

PARAGRAPH	TITLE	PAGE
1	INTRODUCTION……	G-5
2.	MINIMUM LAYER……	G-6
3	REPORTING LAYER…….....	G-70
4	CCIRM LAYER…….....	G-74
5	EXPERIMENTAL LAYER……	G-81
6	ASSOCIATIONS.....	G-89
7	RELATED FILES.....	G-90
8	BACKWARD COMPATIBILITY WITH STANAG 4559 EDITION 1 AND 2	G-91
9	DATA DEFINITIONS – 4559 Edition 3 CONTENT.....	G-92

TABLES	TITLE	PAGE
Table G- 1	NSIL_CARD.identifier	G-8
Table G- 2	NSIL_CARD.sourceDateTimeModified	G-8
Table G- 3	NSIL_CARD.dateTimeModified	G-9
Table G- 4	NSIL_CARD.status	G-9
Table G- 5	NSIL_CARD.numberOfParts	G-9
Table G- 6	NSIL_CARD.publisher	G-10
Table G- 7	NSIL_CARD.sourceLibrary	G-10
Table G- 8	NSIL_COMMON.descriptionAbstract	G-11
Table G- 9	NSIL_COMMON.identifierMission	G-11
Table G- 10	NSIL_COMMON.identifierUUID	G-13
Table G- 11	NSIL_COMMON.identifierJC3IEDM	G-13
Table G- 12	NSIL_COMMON.language	G-14
Table G- 13	NSIL_COMMON.source	G-15
Table G- 14	NSIL_COMMON.subjectCategoryTarget	G-16
Table G- 15	NSIL_COMMON.targetNumber	G-16
Table G- 16	NSIL_COMMON.type	G-17
Table G- 17	NSIL_COVERAGE.spatialCountryCode	G-18
Table G- 18	NSIL_COVERAGE.spatialGeographicReferenceBox	G-19
Table G- 19	NSIL_COVERAGE.temporalEnd	G-20
Table G- 20	NSIL_COVERAGE.temporalStart	G-21
Table G- 21	NSIL_FILE.archived	G-22
Table G- 22	NSIL_FILE.archiveInformation	G-22
Table G- 23	NSIL_FILE.creator	G-22
Table G- 24	NSIL_FILE.dateTimeDeclared	G-23
Table G- 25	NSIL_FILE.extent	G-23
Table G- 26	NSIL_FILE.format	G-24
Table G- 27	NSIL_FILE.formatVersion	G-25
Table G- 28	NSIL_FILE.productURL	G-25
Table G- 29	NSIL_FILE.title	G-26
Table G- 30	NSIL_FILE.isProductLocal	G-26
Table G- 31	NSIL_GMTI.identifierJob	G-27
Table G- 32	NSIL_GMTI.numberOfTargetReports	G-27
Table G- 33	NSIL_IMAGERY.category	G-28
Table G- 34	NSIL_IMAGERY.cloudCoverPercentage	G-28

Table G- 35 NSIL_IMAGERY.comments G-29
Table G- 36 NSIL_IMAGERY.decompressionTechnique G-30
Table G- 37 NSIL_IMAGERY.identifier G-30
Table G- 38 NSIL_IMAGERY.NIIRS G-31
Table G- 39 NSIL_IMAGERY.numberOfBands G-31
Table G- 40 NSIL_IMAGERY.numberOfRows G-32
Table G- 41 NSIL_IMAGERY.numberOfColumns G-32
Table G- 42 NSIL_IMAGERY.title G-33
Table G- 43 NSIL_MESSAGE.recipient G-34
Table G- 44 NSIL_MESSAGE.subject G-34
Table G- 45 NSIL_MESSAGE.messageBody G-35
Table G- 46 NSIL_MESSAGE.messageType G-35
Table G- 47 NSIL_METADATASECURITY.classification G-36
Table G- 48 NSIL_METADATASECURITY.policy G-36
Table G- 49 NSIL_METADATASECURITY.releasability G-37
Table G- 50 NSIL_PART.partIdentifier G-38
Table G- 51 NSIL_RELATED_FILE.creator G-39
Table G- 52 NSIL_RELATED_FILE.dateTimeDeclared G-40
Table G- 53 NSIL_RELATED_FILE.extent G-40
Table G- 54 NSIL_RELATED_FILE.fileType G-41
Table G- 55 NSIL_RELATED_FILE.URL G-41
Table G- 56 NSIL_RELATED_FILE.isFileLocal G-42
Table G- 57 NSIL_RELATION.amplification G-42
Table G- 58 NSIL_RELATION.contributor G-43
Table G- 59 NSIL_RELATION.dateTimeDeclared G-43
Table G- 60 NSIL_RELATION.description G-44
Table G- 61 NSIL_RELATION.relationship G-44
Table G- 62 NSIL_SECURITY.classification G-45
Table G- 63 NSIL_SECURITY.policy G-46
Table G- 64 NSIL_SECURITY.releasability G-47
Table G- 65 NSIL_STREAM.archived G-48
Table G- 66 NSIL_STREAM.archiveInformation G-49
Table G- 67 NSIL_STREAM.creator G-49
Table G- 68 NSIL_STREAM.dateTimeDeclared G-50
Table G- 69 NSIL_STREAM.standard G-50
Table G- 70 NSIL_STREAM.standardVersion G-51
Table G- 71 NSIL_STREAM.sourceURL G-51
Table G- 72 NSIL_STREAM.programID G-52
Table G- 73 NSIL_VIDEO.averageBitRate G-52
Table G- 74 NSIL_VIDEO.category G-53
Table G- 75 NSIL_VIDEO.encodingScheme G-54
Table G- 76 NSIL_VIDEO.frameRate G-54
Table G- 77 NSIL_VIDEO.numberOfRows G-55
Table G- 78 NSIL_VIDEcO.numberOfColumns G-55
Table G- 79 NSIL_VIDEO.metadataEncodingScheme G-56
Table G- 80 NSIL_VIDEO.MISMLevel G-56
Table G- 81 NSIL_VIDEO.scanningMode G-57
Table G- 82 NSIL_ALL_VIEW G-59
Table G- 83 Conceptual attributes present in 'NSIL_ALL_VIEW' G-60
Table G- 84 NSIL_IMAGERY_VIEW G-62
Table G- 85 Conceptual attributes present in 'NSIL_IMAGERY_VIEW' G-62
Table G- 86 NSIL_GMTI_VIEW G-64
Table G- 87 Conceptual attributes present in 'NSIL_GMTI_VIEW' G-64
Table G- 88 NSIL_MESSAGE_VIEW G-66
Table G- 89 Conceptual attributes present in 'NSIL_MESSAGE_VIEW' G-66
Table G- 90 NSIL_VIDEO_VIEW G-68

Table G- 91 Conceptual attributes present in 'NSIL_VIDEO_VIEW' G-68
Table G- 92 NSIL_ASSOCIATION_VIEW G-69
Table G- 93 Conceptual attributes present in 'NSIL_ASSOCIATION_VIEW'..... G-70
Table G- 94 NSIL_REPORT.originatorsRequestSerialNumber..... G-70
Table G- 95 NSIL_REPORT.priority G-70
Table G- 96 NSIL_REPORT.type G-71
Table G- 97 NSIL_REPORT_VIEW..... G-73
Table G- 98 Conceptual attributes present in 'NSIL_REPORT_VIEW' G-73
Table G- 99 NSIL_CXP.status G-74
Table G- 100 NSIL_RFI.forAction G-74
Table G- 101 NSIL_RFI.forInformation G-75
Table G- 102 NSIL_RFI.serialNumber..... G-75
Table G- 103 NSIL_RFI.status..... G-76
Table G- 104 NSIL_RFI.workflowStatus G-76
Table G- 105 NSIL_TASK.comments G-77
Table G- 106 NSIL_TASK.status G-77
Table G- 107 NSIL_CCIRM_VIEW G-80
Table G- 108 Conceptual attributes present in 'NSIL_CCIRM_VIEW'..... G-80
Table G- 109 NSIL_APPROVAL.approvedBy G-81
Table G- 110 NSIL_APPROVAL.dateTimeModified..... G-81
Table G- 111 NSIL_APPROVAL.status G-82
Table G- 112 NSIL_EXPLOITATION_INFO.description..... G-82
Table G- 113 NSIL_EXPLOITATION_INFO.level..... G-83
Table G- 114 NSIL_EXPLOITATION_INFO.autoGenerated G-83
Table G- 115 NSIL_EXPLOITATION_INFO.subjectiveQualityCode G-84
Table G- 116 NSIL_SDS.operationalStatus..... G-84
Table G- 117 NSIL_TDL.activity G-85
Table G- 118 NSIL_TDL.messageNumber..... G-85
Table G- 119 NSIL_TDL.platform G-86
Table G- 120 NSIL_TDL.trackNumber..... G-86
Table G- 121 NSIL_TDL_VIEW G-88
Table G- 122 Conceptual attributes present in 'NSIL_TDL_VIEW'..... G-88
Table G- 123 Supported associations..... G-89
Table G- 124 Related files G-90
Table G- 125 Mapping metadata to STANAG 4559 edition 1 and 2 NSIL_CORE model G-91
Table G- 126 Supported data types..... G-92

FIGURE TITLE PAGE

Figure G- 1 NSIL_ALL_VIEW G-58
Figure G- 2 NSIL_IMAGERY_VIEW G-61
Figure G- 3 NSIL_GMTI_VIEW..... G-63
Figure G- 4 NSIL_MESSAGE_VIEW..... G-65
Figure G- 5 NSIL_VIDEO_VIEW G-67
Figure G- 6 NSIL_ASSOCIATION_VIEW G-69
Figure G- 7 NSIL_REPORT_VIEW..... G-72
Figure G- 8 NSIL_CCIRM_VIEW G-79
Figure G- 9 NSIL_TDL_VIEW G-87

ANNEX G - EDITION 3 METADATA MODEL

1. INTRODUCTION

The following sections describe the STANAG 4559 Edition 3 Metadata Model, otherwise referred to in this document as Edition 3 Metadata Model, and its views. The reader should be aware that this entire chapter is auto-generated from an XML document that defines the metadata model.

1.1. The Edition 3 Metadata Model is influenced by existing metadata definitions from several other communities:

1.1.1. The Intelligence Project Implementation Working Group (IPIWG) exchange metadata models

1.1.2. The NATO Discovery Metadata Specification (NDMS) 1.1, November 2005 as approved by NC3B, 23.03.2006 (AC/322-D(2006)0007-AS1)

1.1.3. The ISO 191xx series in particular ISO 19115 and ISO 19115 part 2

1.2. The Edition 3 Metadata Model was designed with the objective to map into IPIWG, NDAG and the ISO 19100 Geographic information metadata standards 19115. Mapping to the IPIWG metadata exchange model is a priority as it will allow implementations of the Edition 3 metadata model to link to existing NATO ISR product libraries such as BICES.

1.3. Naming conventions for the metadata attributes is based on the NDMS (and hence ISO/IEC 11179-3), directly mapping attributes were chosen and used whenever possible. For elements not described by NDMS, new descriptive names are derived from the source specifications in paragraph 1.1 of this Annex. For enumerations (lists) the values shall always be in 'UPPERCASE'.

1.4. Since the Edition 3 Metadata Model supports container files with multiple parts, the process of sorting library results (products) has become more complicated. As an example, consider a file containing two images, one dated the year of 2000 and the other one dated the year of 2008, will this file be sorted before or after another file (product) with a single image dated 2004? To avoid causing extra implementation complexity for the servers it will be optional for a server to support sorting on part-specific attributes. The default values for the 'SORTABLE' qualifier of these attributes are set to 'NO' in the metadata model, but this may be changed by a server to 'YES' (for DataModelMgr discovery) if it chooses to support part-specific sorting. If a server chooses to support sorting of part-specific attributes then it is left to the individual server implementation to decide how to resolve the ambiguity of multiple parts in a container file, e.g. make the decision to use the date from the first or second image. Clients need to be aware that there is no defined "correct" way of performing sorting on multi-part products.

1.5. A STANAG 4559 Edition 3 compliant server shall, in a query result, always be able to provide any of the metadata of requirement mode 'Mandatory' on request from a client. In case of a client ingesting products into the NSILI server, the client shall always provide all the metadata attributes of 'Mandatory' requirement mode. Attributes that have been marked as optional and non-updateable cannot be set by the client through the CreationMgr. For these attributes, the IPL server will generate the attribute value itself.

1.5.1. AEDP-8 Ed2r06 for Motion Imagery was used for identification of metadata from video streams/files. It seems that this revision of the AEDP is strongly US based

and it could be assumed that future revisions of the STANAG 4609 will have corrected this. Examples of problems caused by the US-bias are security instructions where the security classification is categorized as US classification and Non-US classification (it should be NATO and non-NATO classification). Note also that AEDP-8 Ed2r06 has not defined a generic metadata set, and only provides an example of a metadata set, the metadata set used for Predator generated video. It is assumed that a generic (non-Predator specific) metadata set will be developed, resulting in an updated and new mapping at some point.

1.6. Note that there is a difference between the 'Attribute type' and the 'Data type'. 'Attribute type' is from the GIAS:AttributeType enumeration. Like 'Domain', it is included in the GIAS:AttributeInformation structure. See GIAS351, section 2.2.2.9. 'Data type' is not related to GIAS DataModelMgr, that is, not discoverable. It is intended to indicate the type of the object that can be expected in the DAG. There is not always a one-to-one mapping from 'Attribute type' to 'Data type' (e.g. an INTEGER type could be realized in a DAG as int, long, ulong or short). The 'Data type' defines the type to be used by servers when returning DAGs. It is advisable also to use the same type when a client creates or updates metadata, but a server should allow more leniency if other, but similar types are used.

1.7. The table entry 'Updateable' indicates that the attribute value can be updated (through the UpdateMgr) by the client.

2. MINIMUM LAYER

2.1. MINIMUM LAYER ENTITIES

The complete data model is capable of strong metadata for several relatively independent categories of ISR products. A few of the categories have been assessed and refined multiple times as part of laboratory experiments and live-fly exercises providing a degree of maturity, while others are less mature. Also, depending on the responsibilities of the ISR library implementing this STANAG, it may not need to support all of the types represented in the complete model. For these reasons, the Edition 3 Metadata Model Entities and Views have been segregated into layers. The Minimum Layer provides the foundational components of the metadata model, such as the core “library card’ attributes, file and metadata security attributes, temporal and spatial coverage and associations. The Minimum Layer also supports the mature ISR types Imagery, Video and GMTI.

The Reporting Layer requires the existence of the Minimum Layer. This can represent six types of XML report, which support the data types represented in the Minimum Layer.

The Collection Coordination and Information Requirements Management (CCIRM) Layer allows the Library implementing this STANAG to support some aspects of the CCIRM process. This Layer, along with the Reporting Layer and Associations in the Minimum Layer allow

1) A CCIRM system to use the Library to provide tasking to sensor systems and exploitation systems
2) Sensor and exploitation systems to provide the results of the tasking back to the CCIRM system via the Library

The Experimental Layer provides support for Tactical Data Link products. Since TDLs, such as Link 16, are networks on the battlefield that carry all manner of real-time messages, this category encompasses diverse types such as Participant Position Location and Identification (PPLI) and tracking management messages. Because this Layer is not as mature, and because it does not represent a logical data type, but a

trasmission mechanism, this Layer is likely to change in future editions of STANAG 4559.

An example application links CCIRM layers to the viewsSome comments to Entity tables below:

2.1 NSIL_ASSOCIATION

This is only a pseudo-entity.
The following section lists attributes of the 'NSIL_ASSOCIATION' entity.

2.2 NSIL_CARD

The following section lists attributes of the 'NSIL_CARD' entity.

Table G- 1 NSIL_CARD.identifier

Description	Alphanumeric identification code associated with the product. The code must universally uniquely identify the product. The identifier must obey the following regular expression [A-Za-z0-9_\-\.]{1,50}
Requirement mode	OPTIONAL
Updateable	NO
IPL Assigned	YES
Attribute name	identifier
Attribute type	TEXT
Domain	TEXT_VALUE
Max field length	50
Data type	CORBA string
Units	None (empty string)
Sortable	YES
Queryable	YES

Table G- 2 NSIL_CARD.sourceDateTimeModified

Description	The date and time upon which information (cataloguing metadata) about the product was last stored or changed in the source IPL. For any STANAG 4559 Edition 3 server that is not the source of the product, but which have received this product through synchronization, the value of this attribute shall stay unchanged as it is in the source STANAG 4559 server. This then means that the date-time value represent the instant in time when the product was uploaded or changed at the product-source STANAG 4559 Server.
Requirement mode	OPTIONAL
Updateable	NO
IPL Assigned	YES
Attribute name	sourceDateTimeModified
Attribute type	UCOS_ABS_TIME
Domain	DATE_VALUE (Implementation dependent.)
Data type	UCO
AbsTime	
Units	UTC
Sortable	YES
Queryable	YES

Table G- 3 NSIL_CARD.dateTimeModified

Description	The date and time upon which information (cataloguing metadata) about the product was last stored or changed in the local IPL. This attribute is used when synchronizing metadata, to capture when the metadata was synchronized (and not when the metadata was created at the source STANAG 4559 Server). For local products, not received through synchronization, the value of this attribute shall be identical to the value of the 'sourceDateTimeModified' attribute.
Requirement mode	OPTIONAL
Updateable	NO
IPL Assigned	YES
Attribute name	dateTimeModified
Attribute type	UCOS_ABS_TIME
Domain	DATE_VALUE (Implementation dependent.)
Data type	UCO
AbsTime	
Units	UTC
Sortable	YES
Queryable	YES

Table G- 4 NSIL_CARD.status

Description	The status of the product metadata indicating if the metadata entry is new, has been changed or is marked as obsolete. Note: 1. The Library should remove the URL (productURL) when it sets the status to 'OBSOLETE'. Also the ordering of an obsolete product via OrderMgr should not be allowed. 2. A metadata entry should never be removed. 3. The product can never be modified after it has been posted to a library. The 'CHANGED' status refers to the metadata only.
Requirement mode	OPTIONAL
Updateable	NO
IPL Assigned	YES
Attribute name	status
Attribute type	TEXT
Domain	List: [NEW, CHANGED, OBSOLETE]
Data type	CORBA:string
Units	None (empty string)
Sortable	YES
Queryable	YES

Table G- 5 NSIL_CARD.numberOfParts

Description	A number specifying the number of parts belonging to the product
Requirement mode	OPTIONAL
Updateable	NO
IPL Assigned	YES
Attribute name	numberOfParts
Attribute type	INTEGER
Domain	INTEGER_RANGE: 0 .. 99999
Data type	CORBA:long
Units	None
Sortable	YES
Queryable	YES

Table G- 6 NSIL_CARD.publisher

Description	The name of the organization responsible for making the resource (product) available in an IPL. By doing so the publisher enables the discovery of that resource by a requestor entity (client). Examples of organizations are 'CJTF', 'LCC', 'ACC' etc.
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	publisher
Attribute type	TEXT
Domain	TEXT_VALUE
Max field length	30
Data type	CORBA:string
Units	None (empty string)
Sortable	YES
Queryable	YES

Table G- 7 NSIL_CARD.sourceLibrary

Description	Alphanumeric identification code of the library in which the product was initially ingested. This identifier is NOT identifying the library the client is currently connected to.
Requirement mode	OPTIONAL
Updateable	NO
IPL Assigned	YES
Attribute name	sourceLibrary

Attribute type	TEXT
Domain	TEXT_VALUE
Max field length	30
Data type	CORBA: string
Units	None (empty string)
Sortable	YES
Queryable	YES

2.3 NSIL_COMMON

The following section lists attributes of the 'NSIL_COMMON' entity.

Table G- 8 NSIL_COMMON.descriptionAbstract

Description	A summary of the content of the resource (product).
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	descriptionAbstract
Attribute type	TEXT
Domain	TEXT_VALUE
Max field length	800
Data type	CORBA:string
Units	None (empty string)
Sortable	NO
Queryable	YES

Table G- 9 NSIL_COMMON.identifierMission

Description	An alphanumeric identifier that identifies the mission (e.g. a reconnaissance mission) under which the product was collected/generated. As an example, for products collected by sensors on an aircraft, the mission identifier should be the 'Mission Number' from the Air Tasking Order (ATO).
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	identifierMission
Attribute type	TEXT
Domain	TEXT_VALUE

Max field length	40	
Data type	CORBA: string	
Units	None (empty string)	
Sortable	NO	
Queryable	YES	
Aliases	Category: Name:	NSIL_CORE MISNID
	Category: Name:	STANAG4607 /GMTI/PacketHeader/MissionID
	Category: Name:	STANAG4609 EpisodeNumber

Table G- 10 NSIL_COMMON.identifierUUID

Description	A Universally Unique Identifier (UUID) is an identifier (standardized by the Open Software Foundation (OSF)) that can be generated by systems to uniquely identify information without any central coordination. Thus, anyone can create a UUID and use it to identify something with reasonable confidence that the identifier will never be unintentionally used by anyone for anything else.
Requirement mode	MANDATORY
Updateable	YES
IPL Assigned	NO
Attribute name	identifierUUID
Attribute type	TEXT
Domain	TEXT_VALUE
Max field length	36
Data type	CORBA: string
Units	None (empty string)
Sortable	NO
Queryable	YES

Table G- 11 NSIL_COMMON.identifierJC3IEDM

Description	An additional identifier that can be used for cross-referencing into peripherally accessed JC3IEDM databases.
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	identifierJC3IEDM
Attribute type	INTEGER
Domain	INTEGER_RANGE: 0 .. 2147483647
Data type	CORBA: unsignedLong
Units	None
Sortable	NO
Queryable	YES

Table G- 12 NSIL_COMMON.language

Description	A list of 3 character language codes according to ISO 639-2 for the intellectual content of the resource. Multiple languages are separated by BCS Comma (code 0x2C). Note: There are two codes for special situations: * mul (for multiple languages) should be applied when several languages are used and it is not practical to specify all the appropriate language codes. * und (for undetermined) is provided for those situations in which a language or languages must be indicated but the language cannot be identified.
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	language
Attribute type	TEXT
Domain	TEXT_VALUE (Specification: ISO 639-2)
List separator	','
Max field length	12
Data type	CORBA: string
Units	None (empty string)
Sortable	NO
Queryable	YES

Table G- 13 NSIL_COMMON.source

Description	References to assets (e.g. platform IDs) from which the tagged data asset is derived. Sources may be derived partially or wholly, and it is recommended that an identifier (such as a string or number from a formal identification system) be used as a reference. In case of multiple sources these shall be separated by BCS Comma (could happen for container files like AAF, MXF and NSIF). 'Source' is different from 'creator' in that the 'source' entity is the provider of the data content, while the 'creator' is the entity responsible for assembling the data (provided by "sources") into a file.	
Requirement mode	OPTIONAL	
Updateable	YES	
IPL Assigned	NO	
Attribute name	source	
Attribute type	TEXT	
Domain	TEXT_VALUE	
Max field length	200	
Data type	CORBA: string	
Units	None (empty string)	
Sortable	NO	
Queryable	YES	
Aliases	Category: Name:	STANAG4545 ISORCE
	Category: Name:	NSIL_CORE ISORCE
	Category: Name: Comment:	STANAG4607 /GMTI/PacketHeader/PlatformID Unique source identification will also include /GMTI/PacketHeader/Nationality. In case more than one source is listed expose only the first source in NSIL_CORE.
	Category: Name:	STANAG4609 ImageSourceDevice
	Category: Name:	NACT_L16 /HJ3/SOURCE_TRK_NBR

Table G- 14 NSIL_COMMON.subjectCategoryTarget

Description	A target category from STANAG 3596 that should reflect the main subject of the resource. In case of multiple categories these shall be separated by the BCS Comma (code 0x2C) character.
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	subjectCategoryTarget
Attribute type	TEXT
Domain	TEXT_VALUE (Specification: STANAG 3596)
List separator	','
Max field length	50
Data type	CORBA: string
Units	None (empty string)
Sortable	NO
Queryable	YES

Table G- 15 NSIL_COMMON.targetNumber

Description	This field shall contain the identification of the primary target in the product. In case of multiple target identifiers these shall be separated by the BCS Comma (code 0x2C) character.	
Requirement mode	OPTIONAL	
Updateable	YES	
IPL Assigned	NO	
Attribute name	targetNumber	
Attribute type	TEXT	
Domain	TEXT_VALUE	
List separator	','	
Max field length	159	
Data type	CORBA: string	
Units	None (empty string)	
Sortable	NO	
Queryable	YES	
Aliases	Category: Name:	NSIL_CORE TGTID

Table G- 16 NSIL_COMMON.type

Description	The nature or genre of the content of the resource.
Requirement mode	MANDATORY
Updateable	YES
IPL Assigned	NO
Attribute name	type
Attribute type	TEXT
Domain	List: [COLLECTION/EXPLOITATION PLAN, DOCUMENT, GEOGRAPHIC AREA OF INTEREST, GMTI, IMAGERY, INTELLIGENCE REQUIREMENT, MESSAGE, OPERATIONAL ROLES, ORBAT, REPORT, RFI, SYSTEM ASSIGNMENTS, SYSTEM DEPLOYMENT STATUS, SYSTEM SPECIFICATIONS, TACTICAL SYMBOL, TASK, TDL DATA, VIDEO]
Data type	CORBA: string
Units	None (empty string)
Sortable	NO
Queryable	YES

2.3.1 NSIL_COVERAGE

The following section lists attributes of the 'NSIL_COVERAGE' entity.

Table G- 17 NSIL_COVERAGE.spatialCountryCode

Description	List of standards-based abbreviation (three character codes) of a country names describing the area over which the product was collected separated by BCS Comma (code 0x2C). The three character code shall be according to STANAG 1059.	
Requirement mode	OPTIONAL	
Updateable	YES	
IPL Assigned	NO	
Attribute name	spatialCountryCode	
Attribute type	TEXT	
Domain	TEXT_VALUE (Specification: STANAG 1059)	
List separator	','	
Max field length	20	
Data type	CORBA: string	
Units	None (empty string)	
Sortable	NO	
Queryable	YES	
Aliases	Category: Name: Comment:	STANAG4545 TGTID Use last two characters only. FIPS 10-4/STANAG 1059 mapping required.
	Category: Name: Comment:	NSIL_CORE CNTRYCODE FIPS 10-4/STANAG 1059 mapping required.
	Category: Name: Comment:	STANAG4609 ObjectCountryCode

Table G- 18 NSIL_COVERAGE.spatialGeographicReferenceBox

Description	Geographic location of the dataset. Always in WGS-84 reference system, and using decimal degrees. The first coordinate represents the most North-Western corner, the second the most South-Eastern corner. The x-value in a UCOS: Coordinate2D struct represents the longitude, the y-value represents the latitude.	
Requirement mode	OPTIONAL	
Updateable	YES	
IPL Assigned	NO	
Attribute name	spatialGeographicReferenceBox	
Attribute type	UCOS_RECTANGLE	
Domain	GEOGRAPHIC ('upper left x', 'upper left y', 'lower right x' and 'lower right y' values are implementation dependent/>)	
Data type	UCO: Rectangle	
Units	WGS-84/Decimaldegrees	
Sortable	NO	
Queryable	YES	
Aliases	Category: Name:	STANAG4545 IGEOLO
	Category: Name:	NSIL_CORE IGEOLO
	Category: Name: Comment:	STANAG4607 /GMTI/DwellSegment/DwellArea Use the union of all dwell areas in the revisit
	Category: Name: Comment:	STANAG4609 FrameCenterLatitude + FrameCenterLongitude Use the union of all frame centers in the video clip

Table G- 19 NSIL_COVERAGE.temporalEnd

Description	End time of a period of time for the content of the dataset (end time of content acquisition). For products capturing a single instant in time, start time and end time will be equal (or the end time could be omitted)	
Requirement mode	OPTIONAL	
Updateable	YES	
IPL Assigned	NO	
Attribute name	temporalEnd	
Attribute type	UCOS_ABS_TIME	
Domain	DATE_VALUE (Implementation dependent.)	
Data type	UCO: AbsTime	
Units	UTC	
Sortable	NO	
Queryable	YES	
Aliases	Category: Name:	STANAG4545 IDATIM
	Category: Name: Comment:	STANAG4607 /GMTI/MissionSegment/ReferenceTime + /GMTI/DwellSegment/DwellTime Use the dwell time of the last dwell in the revisit
	Category: Name: Comment:	STANAG4609 TimingReconciliationMetadataSet/UserDefinedTimeStamp Use the last timestamp found in the video clip file
	Category: Name:	NACT_L16 /HJ9/TIME_STAMP

Table G- 20 NSIL_COVERAGE.temporalStart

Description	Start time of a period of time for the content of the dataset (start time of content acquisition). For products capturing a single instant in time, start time and end time will be equal (or the end time could be omitted).	
Requirement mode	OPTIONAL	
Updateable	YES	
IPL Assigned	NO	
Attribute name	temporalStart	
Attribute type	UCOS_ABS_TIME	
Domain	DATE_VALUE (Implementation dependent.)	
Data type	UCO: AbsTime	
Units	UTC	
Sortable	NO	
Queryable	YES	
Aliases	Category: Name:	STANAG4545 IDATIM
	Category: Name:	NSIL_CORE IDATIM
	Category: Name: Comment:	STANAG4607 /GMTI/MissionSegment/ReferenceTime + /GMTI/DwellSegment/DwellTime Use the dwell time of the first dwell in the revisit
	Category: Name: Comment:	STANAG4609 TimingReconciliationMetadataSet/UserDefinedTimeStamp Use the first timestamp found in the video clip file
	Category: Name:	NACT_L16 /HJ9/TIME_STAMP

2.3.2 NSIL_DESTINATION

Please note that this entity is only a helper entity for realization of product associations.

The 'NSIL_DESTINATION:NSIL_CARD.sourceLibrary' attribute of the product is identified by the 'view_b_product' parameter of 'create_association' method.

This is a placeholder for attributes of the 'NSIL_DESTINATION' entity.

2.3.3 NSIL_FILE

The following section lists attributes of the 'NSIL_FILE' entity.

Table G- 21 NSIL_FILE.archived

Description	Indicates whether the file has been archived. Set to "YES" if the file has been archived, otherwise set to "NO".
Requirement mode	OPTIONAL
Updateable	NO
IPL Assigned	YES
Attribute name	Archived
Attribute type	BOOLEAN_DATA
Domain	BOOLEAN_VALUE (YES/NO default=NO)
Data type	CORBA: Boolean
Units	None
Sortable	YES
Queryable	YES

Table G- 22 NSIL_FILE.archiveInformation

Description	Additional archiving information in free text form allowing the archived data (product files) to be found and possibly be restored to the server storage. This information could contain the date when the data was archived and point of contact for retrieving the archived data.
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	archiveInformation
Attribute type	TEXT
Domain	TEXT_VALUE
Max field length	200
Data type	CORBA: string
Units	None (empty string)
Sortable	YES
Queryable	YES

Table G- 23 NSIL_FILE.creator

Description	An entity primarily responsible for making the content of the resource. Creator is responsible for the intellectual or creative content of the

	resource. With raw sensor products a creator is the unit that runs the sensor. With exploited products it is the exploitation station, with an Information Request (IR) it is an IR Management system/service.	
Requirement mode	MANDATORY	
Updateable	YES	
IPL Assigned	NO	
Attribute name	Creator	
Attribute type	TEXT	
Domain	TEXT_VALUE	
Max field length	200	
Data type	CORBA: string	
Units	None (empty string)	
Sortable	YES	
Queryable	YES	
Aliases	Category: Name:	STANAG4545 OSTAID
	Category: Name:	NSIL_CORE OSTAID

Table G- 24 NSIL_FILE.dateTimeDeclared

Description	Date and time on which the resource was declared, filed or stored (the date the content of the product (dataset) was compiled).	
Requirement mode	MANDATORY	
Updateable	YES	
IPL Assigned	NO	
Attribute name	dateTimeDeclared	
Attribute type	UCOS_ABS_TIME	
Domain	DATE_VALUE (Implementation dependent.)	
Data type	UCO: AbsTime	
Units	UTC	
Sortable	YES	
Queryable	YES	
Aliases	Category: Name:	STANAG4545 FDT
	Category: Name:	NSIL_CORE FDT

Table G- 25 NSIL_FILE.extent

Description	(Estimated) size of product in the specified transfer format, expressed in

	megabytes. The transfer size is > 0.0.	
Requirement mode	OPTIONAL	
Updateable	NO	
IPL Assigned	YES	
Attribute name	Extent	
Attribute type	FLOATING_POINT	
Domain	FLOATING_POINT_RANGE (0.0 .. 3e38)	
Data type	CORBA: double	
Units	MB	
Sortable	YES	
Queryable	YES	
Aliases	Category: Name:	STANAG4545 FL
	Category: Name:	NSIL_CORE FL

Table G- 26 NSIL_FILE.format

Description	The MIME type for the product object to which this metadata applies with the necessary extension to cater for ISR specific file types not yet MIME registered.	
Requirement mode	OPTIONAL	
Updateable	NO	
IPL Assigned	YES	
Attribute name	Format	
Attribute type	TEXT	
Domain	TEXT_VALUE	
Max field length	50	
Data type	CORBA: string	
Units	None (empty string)	
Sortable	YES	
Queryable	YES	
Aliases	Category: Name: Comment:	STANAG4545 FHDR Use character 0-3 from the start of the file to identify the NSIF (or NITF) format.

Table G- 27 NSIL_FILE.formatVersion

Description	Version of product format. E.g. "1.0". In case of an XML document this attribute refers to the version of the XML schema that the document refers to (and not to the version of the XML format).	
Requirement mode	OPTIONAL	
Updateable	NO	
IPL Assigned	YES *(but optional)*	
Attribute name	formatVersion	
Attribute type	TEXT	
Domain	TEXT_VALUE	
Max field length	10	
Data type	CORBA: string	
Units	None (empty string)	
Sortable	YES	
Queryable	YES	
Aliases	Category: Name: Comment:	STANAG4545 FHDR Use character 4-8 from the start of the file to identify the NSIF (or NITF) format version.
	Category: Name:	STANAG4607 /GMTI/PacketHeader/VersionID

Table G- 28 NSIL_FILE.productURL

Description	The address and access protocol in the form of an URL by which a user can directly access a product without the user placing an order for delivery. Note, This attribute is not queryable.	
Requirement mode	OPTIONAL	
Updateable	NO	
IPL Assigned	YES *(but optional)*	
Attribute name	productURL	
Attribute type	TEXT	
Domain	TEXT_VALUE	
Max field length	0	
Data type	CORBA:string	
Units	URL	
Sortable	YES	
Queryable	NO	
Aliases	Category: Name:	NSIL_CORE DAID

Table G- 29 NSIL_FILE.title

Description	Product (dataset) title.	
Requirement mode	OPTIONAL	
Updateable	YES	
IPL Assigned	NO	
Attribute name	Title	
Attribute type	TEXT	
Domain	TEXT_VALUE	
Max field length	100	
Data type	CORBA: string	
Units	None (empty string)	
Sortable	YES	
Queryable	YES	
Aliases	Category: Name:	STANAG4545 FTITLE
	Category: Name:	NSIL_CORE FTITLE

Table G- 30 NSIL_FILE.isProductLocal

Description	Indicates whether the product file is available locally in the library, or resides in a remote library.
Requirement mode	OPTIONAL
Updateable	NO
IPL Assigned	YES
Attribute name	isProductLocal
Attribute type	BOOLEAN_DATA
Domain	BOOLEAN_VALUE (YES/NO default=YES)
Data type	CORBA: Boolean
Units	None
Sortable	YES
Queryable	YES

2.3.4 NSIL_GMTI

The following section lists attributes of the 'NSIL_GMTI' entity.

Table G- 31 NSIL_GMTI.identifierJob

Description	A platform-assigned number identifying the specific requestor task to which the packet pertains. The Job ID shall be unique within a mission.	
Requirement mode	MANDATORY	
Updateable	YES	
IPL Assigned	NO	
Attribute name	identifierJob	
Attribute type	FLOATING_POINT	
Domain	FLOATING_POINT_RANGE (0.0 .. 4294967295.0)	
Data type	CORBA: double	
Units	None	
Sortable	NO	
Queryable	YES	
Aliases	Category: Name:	STANAG4607 /GMTI/PacketHeader/JobID

Table G- 32 NSIL_GMTI.numberOfTargetReports

Description	The total number of target reports within all the dwells in the file.	
Requirement mode	MANDATORY	
Updateable	YES	
IPL Assigned	NO	
Attribute name	numberOfTargetReports	
Attribute type	INTEGER	
Domain	INTEGER_RANGE: 0 .. 2147483647	
Data type	CORBA: unsignedLong	
Units	None	
Sortable	NO	
Queryable	YES	
Aliases	Category: Name: Comment:	STANAG4607 /GMTI/DwellSegment/TargetReportCount Summarize over all dwells in the radar revisit

2.3.5 NSIL_IMAGERY

This section provides details for the metadata for a single IMAGERY segment from a NSIF file. Hint for the implementation:

The attribute 'decompressionTechnique' represents an important property of the product file itself. Therefore, the attribute needs to be "correct". Since a product file cannot change, it is not updatable. For the creationMgr, if a client does not provide the 'decompressionTechnique'

attribute, and the server has no functionality to extract it from the NSIF file, the server must give an error. Otherwise if the client does not provide 'decompressionTechnique' attribute the server must fill in the value. If the client provides the 'decompressionTechnique' attribute, and the server can also extract it, and detects that the client provided value is incorrect, the server could throw an exception.

The following section lists attributes of the 'NSIL_IMAGERY' entity.

Table G- 33 NSIL_IMAGERY.category

Description	A valid indicator of the specific category of image, raster, or grid data. The specific category of an image reveals its intended use or the nature of its collector.	
Requirement mode	MANDATORY	
Updateable	YES	
IPL Assigned	NO	
Attribute name	category	
Attribute type	TEXT	
Domain	List: [VIS, SL, TI, FL, RD, EO, OP, HR, HS, CP, BP, SAR, SARIQ, IR, MS, FP, MRI, XRAY, CAT, VD, BARO, CURRENT, DEPTH, WIND, MAP, PAT, LEG, DTEM, MATR, LOCG]	
Data type	CORBA: string	
Units	None (empty string)	
Sortable	NO	
Queryable	YES	
Aliases	Category: Name:	STANAG4545 ICAT
	Category: Name:	NSIL_CORE ICAT

Table G- 34 NSIL_IMAGERY.cloudCoverPercentage

Description	A code that indicates the percentage of the image obscured by cloud cover.
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	cloudCoverPercentage
Attribute type	INTEGER
Domain	INTEGER_RANGE: 0 .. 100
Data type	CORBA: short
Units	None
Sortable	NO

Queryable	YES

Table G- 35 NSIL_IMAGERY.comments

Description	Comments related to the image	
Requirement mode	OPTIONAL	
Updateable	YES	
IPL Assigned	NO	
Attribute name	comments	
Attribute type	TEXT	
Domain	TEXT_VALUE	
Max field length	800	
Data type	CORBA: string	
Units	None (empty string)	
Sortable	NO	
Queryable	YES	
Aliases	Category: Name: Comment	STANAG4545 ICOM Concatenate all 10 ICOM fields (0-9)
	Category: Name:	NSIL_CORE ICOM

Table G- 36 NSIL_IMAGERY.decompressionTechnique

Description	Specification of algorithm or process to apply to read or expand the image to which compression techniques have been applied.	
Requirement mode	MANDATORY	
Updateable	YES	
IPL Assigned	NO	
Attribute name	decompressionTechnique	
Attribute type	TEXT	
Domain	List: [NC, NM, C1, M1, I1, C3, M3, C4, M4, C5, M5, C8, M8]	
Data type	CORBA: string	
Units	None (empty string)	
Sortable	NO	
Queryable	YES	
Aliases	Category: Name:	STANAG4545 IC

Table G- 37 NSIL_IMAGERY.identifier

Description	This field shall contain a valid alphanumeric identification code associated with the image. The valid codes are determined by the application.	
Requirement mode	MANDATORY	
Updateable	YES	
IPL Assigned	NO	
Attribute name	identifier	
Attribute type	TEXT	
Domain	TEXT_VALUE	
Max field length	10	
Data type	CORBA: string	
Units	None (empty string)	
Sortable	NO	
Queryable	YES	
Aliases	Category: Name	STANAG4545 IID1
	Category: Name:	NSIL_CORE IID1

Table G- 38 NSIL_IMAGERY.NIIRS

Description	NIIRS - National Image Interpretability Rating Scales (NIIRS) -defines and measures the quality of images and the performance of imaging systems.
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	NIIRS
Attribute type	INTEGER
Domain	INTEGER_RANGE: 1 .. 9
Data type	CORBA: short
Units	None
Sortable	NO
Queryable	YES

Table G- 39 NSIL_IMAGERY.numberOfBands

Description	The number of data bands within the specified image.	
Requirement mode	MANDATORY	
Updateable	YES	
IPL Assigned	NO	
Attribute name	numberOfBands	
Attribute type	INTEGER	
Domain	INTEGER_RANGE: 0 .. 99999	
Data type	CORBA: long	
Units	None	
Sortable	NO	
Queryable	YES	
Aliases	Category: Name:	STANAG4545 NBANDS

Table G- 40 NSIL_IMAGERY.numberOfRows

Description	Number of significant rows in Image. This field shall contain the total number of rows of significant pixels in the image. It could be that an image has been padded with fill data (to fill out the last part of the block in a blocked image). The meaning of 'significant' is then only the rows conveying image data excluding all the rows with padded fill data.	
Requirement mode	OPTIONAL	
Updateable	YES	
IPL Assigned	NO	
Attribute name	numberOfRows	
Attribute type	INTEGER	
Domain	INTEGER_RANGE: 1 .. 2147483647	
Data type	CORBA: unsignedLong	
Units	None	
Sortable	NO	
Queryable	YES	
Aliases	Category: Name:	STANAG4545 NROWS

Table G- 41 NSIL_IMAGERY.numberOfColumns

Description	Number of significant columns in Image. This field shall contain the total number of columns of significant pixels in the image. It could be that an image has been padded with fill data (to fill out the last part of the block in a blocked image). The meaning of 'significant' is then only the columns conveying image data excluding all the columns with padded fill data.	
Requirement mode	OPTIONAL	
Updateable	YES	
IPL Assigned	NO	
Attribute name	numberOfColumns	
Attribute type	INTEGER	
Domain	INTEGER_RANGE: 1 .. 2147483647	
Data type	CORBA: unsignedLong	
Units	None	
Sortable	NO	
Queryable	YES	
Aliases	Category: Name:	STANAG4545 NCOLS

Table G- 42 NSIL_IMAGERY.title

Description	This field shall contain the title of the image.	
Requirement mode	OPTIONAL	
Updateable	YES	
IPL Assigned	NO	
Attribute name	title	
Attribute type	TEXT	
Domain	TEXT_VALUE	
Max field length	80	
Data type	CORBA: string	
Units	None (empty string)	
Sortable	NO	
Queryable	YES	
Aliases	Category: Name	STANAG4545 IID2
	Category: Name:	NSIL_CORE IID2

2.3.6 NSIL_MESSAGE

The purpose of this entity is to allow cataloguing any type of message traffic, where each individual message is a server product. Typically this will be used for cataloguing XMPP conference room chat, but could also be used to store other types of message traffic. The following section lists attributes of the 'NSIL_MESSAGE' entity.

Table G- 43 NSIL_MESSAGE.recipient

Description	Identify the receiving entity of the message. This will typically be an XMPP conference room identifier (including the fully qualified domain name extension), but could also be an individual/personal XMPP or e-mail account. In the case a message is sent to more than one recipient, this shall be supported by separating the recipients by BCS Comma. Note that the 'from' information is captured in NSIL_COMMON.source
Requirement mode	MANDATORY
Updateable	YES
IPL Assigned	NO
Attribute name	recipient
Attribute type	TEXT
Domain	TEXT_VALUE
Max field length	200
Data type	CORBA: string
Units	None (empty string)
Sortable	NO
Queryable	YES

Table G- 44 NSIL_MESSAGE.subject

Description	Data that specifies the topic of the message.
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	subject
Attribute type	TEXT
Domain	TEXT_VALUE
Max field length	200
Data type	CORBA: string
Units	None (empty string)
Sortable	NO
Queryable	YES

Table G- 45 NSIL_MESSAGE.messageBody

Description	The body of the message. In case the message body text exceeds the maximum length of this metadata attribute the message text will be truncated. . The complete message is available through the URL in NSIL_FILE entity, but only the characters stored in this attribute can be used for free text search.
Requirement mode	MANDATORY
Updateable	YES
IPL Assigned	NO
Attribute name	messageBody
Attribute type	TEXT
Domain	TEXT_VALUE
Max field length	2048
Data type	CORBA: string
Units	None (empty string)
Sortable	NO
Queryable	YES

Table G- 46 NSIL_MESSAGE.messageType

Description	Identification of message type (protocol). Initially only XMPP is recognized.
Requirement mode	MANDATORY
Updateable	YES
IPL Assigned	NO
Attribute name	messageType
Attribute type	TEXT
Domain	List
[XMPP]	
Data type	CORBA: string
Units	None (empty string)
Sortable	NO
Queryable	YES

2.3.7 NSIL_METADATASECURITY

The following section lists attributes of the 'NSIL_METADATASECURITY' entity.

Table G- 47 NSIL_METADATASECURITY.classification

Description	NATO Security markings that determine the physical security given to the metadata.
Requirement mode	MANDATORY
Updateable	YES
IPL Assigned	NO
Attribute name	classification
Attribute type	TEXT
Domain	List: [COSMIC TOP SECRET, SECRET, CONFIDENTIAL, RESTRICTED, UNCLASSIFIED, NO CLASSIFICATION]
Data type	CORBA: string
Units	None (empty string)
Sortable	YES
Queryable	YES

Table G- 48 NSIL_METADATASECURITY.policy

Description	This field shall contain valid values indicating the national or multinational security policies used to classify the metadata. To indicate a national security policy, Country Codes per STANAG 1059 shall be used. In cases where the country is not listed by a three character country code, but as a six character region code, the six character province code from STANAG 1059 shall be used (e.g. Kosovo is listed as a region in Yugoslavia with six character province code YUKM-). In all cases when STANAG 1059 is not sufficient to describe the security policy, additional enumeration labels can be used (e.g. 'NATO/PFP' and 'NATO/EU'). Note: Although STANAG 1059 includes the 'XXN' entry for NATO, the full 'NATO' name shall be used to indicate NATO security policy to avoid any confusion from established and common used terms.
Requirement mode	MANDATORY
Updateable	YES
IPL Assigned	NO
Attribute name	policy
Attribute type	TEXT
Domain	TEXT_VALUE (Specification: STANAG 1059)
List separator	','
Max field length	20
Data type	CORBA: string

Units	None (empty string)
Sortable	YES
Queryable	YES

Table G- 49 NSIL_METADATASECURITY.releasability

Description	An additional marking to limit the dissemination of the product metadata information. Typical values include one or more country codes as found in STANAG 1059 separated by the BCS Comma (code 0x2C) character. Default value should be 'NATO'. Note: Although STANAG 1059 includes the 'XXN' entry for NATO, the full 'NATO' name shall be used to indicate NATO releasability to avoid any confusion from established and common used terms.
Requirement mode	MANDATORY
Updateable	YES
IPL Assigned	NO
Attribute name	releasability
Attribute type	TEXT
Domain	TEXT_VALUE (Specification: STANAG 1059)
List separator	','
Max field length	50
Data type	CORBA: string
Units	None (empty string)
Sortable	YES
Queryable	YES

2.3.8 NSIL_PART

The following section lists attributes of the 'NSIL_PART' entity.

Table G- 50 NSIL_PART.partIdentifier

Description	String describing the part of the containing Product.
Requirement mode	MANDATORY
Updateable	YES
IPL Assigned	NO
Attribute name	partIdentifier
Attribute type	TEXT
Domain	TEXT_VALUE
Max field length	0
Data type	CORBA: string
Units	None (empty string)
Sortable	NO
Queryable	YES

2.3.9 NSIL_PRODUCT

Helper entity.

The following section lists attributes of the 'NSIL_PRODUCT' entity.

2.3.10 NSIL_RELATED_FILE

The NSIL_RELATED_FILE entity provides information on the related files for a product. For each of the related files types for a product as can be discovered with ProductMgr.get_related_file_types, there will be an instance of this entity with the related file type and a direct access URL to that related file. Related files can only be added via the RelatedFileList in the CreationMgr and updated via the RelatedFileList in the UpdateMgr. It is not possible to change them via the attributes in the NSIL_RELATED_FILE entity, hence they are all server assigned.

2.3.10.1 The NSIL_RELATED_FILE entity provides the same kind of capability as the ProductMgr methods.

Requesting the NSIL_RELATED_FILE.fileType attribute via the CatalogMgr, and requesting ProductMgr.get_related_file_types both provide the available related file types for a product. NSIL_RELATED_FILE.url provides the related file via a pull-mechanism.

ProductMgr.get_related_file provides it via a push-mechanism.

2.3.10.2 A client will not know the file sizes of the related files, nor will the client have access to any metadata about the format of these related files. For the thumbnail and overview files it is assumed that the file sizes are irrelevant (small files) and that the format of the files is known (thumbnail files are JPEG files

and overview files are NSIF files).

2.3.10.3 Since a product can have multiple Related Files of a particular type (e.g. Thumbnail) if the products have multiple parts of the same type (e.g. multiple images within an NSIF file) it is recommended to use a file naming convention for these related files. The recommended naming convention is '{RELATED-FILE-TYPE}-{PART-IDENTIFIER}'. As an example THUMBNAIL-4 would refer to the thumnail file of the image identified in 'part 4'. The following section lists attributes of the 'NSIL_RELATED_FILE' entity.

Table G- 51 NSIL_RELATED_FILE.creator

Description	An entity primarily responsible for making the content of the related file. Creator is responsible for the intellectual or creative content of the resource.	
Requirement mode	MANDATORY	
Updateable	YES	
IPL Assigned	NO	
Attribute name	creator	
Attribute type	TEXT	
Domain	TEXT_VALUE	
Max field length	200	
Data type	CORBA: string	
Units	None (empty string)	
Sortable	YES	
Queryable	YES	
Aliases	Category: Name:	STANAG4545 OSTAID
	Category: Name:	NSIL_CORE OSTAID

Table G- 52 NSIL_RELATED_FILE.dateTimeDeclared

Description	Date and time on which the related file was declared, filed or stored (the date the content of the related file was compiled).
Requirement mode	MANDATORY
Updateable	YES
IPL Assigned	NO
Attribute name	dateTimeDeclared
Attribute type	UCOS_ABS_TIME
Domain	DATE_VALUE (Implementation dependent.)
Data type	UCO: AbsTime
Units	UTC
Sortable	YES
Queryable	YES

Table G- 53 NSIL_RELATED_FILE.extent

Description	(Estimated) size of the file, expressed in megabytes. The transfer size is > 0.0.	
Requirement mode	OPTIONAL	
Updateable	NO	
IPL Assigned	YES	
Attribute name	extent	
Attribute type	FLOATING_POINT	
Domain	FLOATING_POINT_RANGE (0.0 .. 3e38)	
Data type	CORBA: double	
Units	MB	
Sortable	YES	
Queryable	YES	
Aliases	Category: Name:	STANAG4545 FL
	Category: Name:	NSIL_CORE FL

Table G- 54 NSIL_RELATED_FILE.fileType

Description	The related file type.
Requirement mode	OPTIONAL
Updateable	NO
IPL Assigned	YES
Attribute name	fileType
Attribute type	TEXT
Domain	TEXT_VALUE
Max field length	30
Data type	CORBA: string
Units	None (empty string)
Sortable	NO
Queryable	YES

Table G- 55 NSIL_RELATED_FILE.URL

Description	An unambiguous reference (in form of an URL) to the related file.
Requirement mode	OPTIONAL
Updateable	NO
IPL Assigned	YES *(but optional)*
Attribute name	URL
Attribute type	TEXT
Domain	TEXT_VALUE
Max field length	0
Data type	CORBA: string
Units	URL
Sortable	NO
Queryable	NO

Table G- 56 NSIL_RELATED_FILE.isFileLocal

Description	Indicates whether the related file is available locally in the library, or resides in a remote library.
Requirement mode	OPTIONAL
Updateable	NO
IPL Assigned	YES
Attribute name	isFileLocal
Attribute type	BOOLEAN_DATA
Domain	BOOLEAN_VALUE (YES/NO default=YES)
Data type	CORBA: boolean
Units	None
Sortable	YES
Queryable	YES

2.3.11 NSIL_RELATION

The following section lists attributes of the 'NSIL_RELATION' entity.

Table G- 57 NSIL_RELATION.amplification

Description	Additional information on the association. This could be used to add additional information for very specific associations. This information is set through the 'assoc_info' parameter of 'create_association' method, where name="amplification".
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	amplification
Attribute type	TEXT
Domain	TEXT_VALUE
Max field length	0
Data type	CORBA: string
Units	None (empty string)
Sortable	YES
Queryable	YES

Table G- 58 NSIL_RELATION.contributor

Description	An entity responsible for contributing the reference. This will be removed in a later edition.
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	contributor
Attribute type	TEXT
Domain	TEXT_VALUE
Max field length	30
Data type	CORBA: string
Units	None (empty string)
Sortable	YES
Queryable	YES

Table G- 59 NSIL_RELATION.dateTimeDeclared

Description	Date and time when the relationship was created. Note that this value must not change when synchronizing the metadata to another library.
Requirement mode	OPTIONAL
Updateable	NO
IPL Assigned	YES
Attribute name	dateTimeDeclared
Attribute type	UCOS_ABS_TIME
Domain	DATE_VALUE (Implementation dependent.)
Data type	UCO: AbsTime
Units	UTC
Sortable	YES
Queryable	YES

Table G- 60 NSIL_RELATION.description

Description	Further describes or explains the relationship between the products. This information is set through the 'assoc_info' parameter of 'create_association' method, where name="description"
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	description
Attribute type	TEXT
Domain	TEXT_VALUE
Max field length	255
Data type	CORBA: string
Units	None (empty string)
Sortable	YES
Queryable	YES

Table G- 61 NSIL_RELATION.relationship

Description	Defines the kind of relationship between the described product and the associated product. This value is specified by the 'assoc_name' parameter of 'create_association' method.
Requirement mode	OPTIONAL
Updateable	NO
IPL Assigned	YES
Attribute name	relationship
Attribute type	TEXT
Domain	List: [HAS PART, IS VERSION OF, REPLACES, IS SUPPORT DATA TO, ORIGINATING FROM, FOLLOWS]
Data type	CORBA: string
Units	None (empty string)
Sortable	YES
Queryable	YES

2.3.12 NSIL_SECURITY

The following section lists attributes of the 'NSIL_SECURITY' entity.

Table G- 62 NSIL_SECURITY.classification

Description	NATO Security markings that determine the physical security given to the information in storage and transmission, its circulation, destruction and the personnel security clearance required for access as required by [C-M(2002)49].	
Requirement mode	MANDATORY	
Updateable	YES	
IPL Assigned	NO	
Attribute name	classification	
Attribute type	TEXT	
Domain	List: [COSMIC TOP SECRET, SECRET, CONFIDENTIAL, RESTRICTED, UNCLASSIFIED, NO CLASSIFICATION]	
Data type	CORBA: string	
Units	None (empty string)	
Sortable	NO	
Queryable	YES	
Aliases	Category: Name: Comment:	STANAG4545 ISCLAS ISCLAS is for NSIL_PART: NSIL_SECURITY. Use FSCLAS for NSIL_PRODUCT: NSIL_SECURITY
	Category: Name: Comment	NSIL_CORE PSCLAS Map PSCLAS to NSIL_PRODUCT: NSIL_SECURITY
	Category Name:	STANAG4607 /GMTI/PacketHeader/Security/Classification
	Category: Name	STANAG4609 SecurityClassification

Table G- 63 NSIL_SECURITY.policy

Description	This field shall contain valid values indicating the national or multinational security policies used to classify the Product. To indicate a national security policy Country Codes per STANAG 1059 shall be used. In cases where the country is not listed by a three character country code, but as a six character region code, the six character province code from STANAG 1059 shall be used (e.g. Kosovo is listed as a region in Yugoslavia with six character province code YUKM-). In all other cases when STANAG 1059 is not sufficient to describe the security policy, additional enumeration labels could be used (e.g. 'NATO/PFP' and 'NATO/EU'). Note: Although STANAG 1059 includes the 'XXN' entry for NATO, the full 'NATO' name shall be used to indicate NATO security policy to avoid any confusion from established and common used terms.	
Requirement mode	MANDATORY	
Updateable	YES	
IPL Assigned	NO	
Attribute name	policy	
Attribute type	TEXT	
Domain	TEXT_VALUE (Specification: STANAG 1059)	
List separator	','	
Max field length	20	
Data type	CORBA: string	
Units	None (empty string)	
Sortable	NO	
Queryable	YES	
Aliases	Category: Name: Comment:	STANAG4545 ISCLSY ISCLSY is for NSIL_PART: NSIL_SECURITY. Use FSCLSY for NSIL_PRODUCT: NSIL_SECURITY. FIPS 10-4/STANAG 1059 mapping required.
	Category: Name: Comment	NSIL_CORE PSCLSY Map PSCLSY to NSIL_PRODUCT: NSIL_SECURITY. FIPS 10-4/STANAG 1059 mapping required.
	Category: Name:	STANAG4607 /GMTI/PacketHeader/Security/ClassificationSystem

Table G- 64 NSIL_SECURITY.releasability

Description	An additional marking to further limit the dissemination of classified information in accordance with [C-M (2002)49]. Values include one or more three character country codes as found in STANAG 1059 separated by a single BCS Comma (code 0x2C). Default value should be NATO. Note: Although STANAG 1059 includes the 'XXN' entry for NATO, the full 'NATO' name shall be used to indicate NATO releasability to avoid any confusion from established and common used terms.	
Requirement mode	MANDATORY	
Updateable	YES	
IPL Assigned	NO	
Attribute name	releasability	
Attribute type	TEXT	
Domain	TEXT_VALUE (Specification: STANAG 1059)	
List separator	','	
Max field length	50	
Data type	CORBA: string	
Units	None (empty string)	
Sortable	NO	
Queryable	YES	
Aliases	Category: Name: Comment:	STANAG4545 ISREL ISREL for NSIL_PART: NSIL_SECURITY. Use FSREL for NSIL_PRODUCT NSIL_SECURITY. FIPS 10-4/STANAG 1059 mapping required.
	Category: Name: Comment	NSIL_CORE PSREL Map PSREL to NSIL_PRODUCT: NSIL_SECURITY. PSREL separates the entities in the list with a single BCS Space character (code 0x20) while 'releasability' separates the entities with BCS Comma (code 0x2C). FIPS 10-4/STANAG 1059 mapping required.
	Category: Name:	STANAG4609 ReleasingInstructions

2.3.13 NSIL_SOURCE

Please note that this entity is only a helper entity for realization of product associations. The 'NSIL_SOURCE:NSIL_CARD.sourceLibrary' attribute of the product is identified by the 'view_a_product' parameter of 'create_association' method. The following section lists attributes of the 'NSIL_SOURCE' entity.

2.3.14 NSIL_STREAM

This entity will be used for discovery of streaming data (video, GMTI, etc) that is normally streamed from a source EXTERNAL to the library server. These streams are normally live streams. The responsibility for ingesting and maintaining the metadata for the streams and for determining the time period to store data for streaming lies fully with the external streaming service.

This entity addresses a form of product delivery and as such is similar in nature to the NSIL_FILE entity. Situations existing were a product could be delivered in multiple ways are plausible; the product could be delivered as a file or it could be streamed to the requesting client with the effect of reducing the network traffic load. An example of a product type where both delivery mechanisms could be used is with video clips.

Notes:

- In case of streaming video, a streaming product could also include the NSIL_PART/NSIL_VIDEO entity to provide additional metadata about the video.
- For GMTI the NSIL_PART/NSIL_GMTI entity will not be used since it provides information pertaining to one particular radar revisit; a GMTI stream would contain numerous radar revisits.
- For TDL the NSIL_PART/NSIL_TDL entity will not be used since it provides information pertaining to only a few particular messages.
- For TDL it is assumed that all producers stream to the same port. There should only be one entity managing the TDL stream information in the STANAG 4559 network. This means that for TDL streaming there needs to be one single entity maintaining the streaming information. It could for instance be Networked Interoperable Real-Time Information Services (NIRIS) doing this as part of the LAN-to-LAN forwarding task. The following section lists attributes of the 'NSIL_STREAM' entity.

Table G- 65 NSIL_STREAM.archived

Description	Indicates whether the stream has been archived. Set to "YES" if the file has been archived, otherwise set to "NO".
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	archived
Attribute type	BOOLEAN_DATA
Domain	BOOLEAN_VALUE (YES/NO default=NO)
Data type	CORBA: boolean
Units	None
Sortable	YES
Queryable	YES

Table G- 66 NSIL_STREAM.archival information

Description	Additional archiving information in free text form allowing the archived data (stream data) to be found and possibly be restored to the server storage. This information could contain the date when the data was archived and point of contact for retrieving the archived data.
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	archiveInformation
Attribute type	TEXT
Domain	TEXT_VALUE
Max field length	200
Data type	CORBA: string
Units	None (empty string)
Sortable	YES
Queryable	YES

Table G- 67 NSIL_STREAM.creator

Description	An entity responsible for providing the stream. Creator is responsible for the intellectual or creative content of the stream.	
Requirement mode	MANDATORY	
Updateable	YES	
IPL Assigned	NO	
Attribute name	Creator	
Attribute type	TEXT	
Domain	TEXT_VALUE	
Max field length	200	
Data type	CORBA: string	
Units	None (empty string)	
Sortable	YES	
Queryable	YES	
Aliases	Category: Name:	STANAG4545 OSTAID
	Category: Name:	NSIL_CORE OSTAID

Table G- 68 NSIL_STREAM.dateTimeDeclared

Description	Date and time on which the stream was declared, filed or stored.	
Requirement mode	MANDATORY	
Updateable	YES	
IPL Assigned	NO	
Attribute name	dateTimeDeclared	
Attribute type	UCOS_ABS_TIME	
Domain	DATE_VALUE (Implementation dependent.)	
Data type	UCO: AbsTime	
Units	UTC	
Sortable	YES	
Queryable	YES	
Aliases	Category: Name:	STANAG4545 FDT
	Category: Name:	NSIL_CORE FDT

Table G- 69 NSIL_STREAM.standard

Description	The standard that the streamed data complies with.
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	Standard
Attribute type	TEXT
Domain	List: [STANAG4609, STANAG4607, STANAG5516_NACT]
Data type	CORBA: string
Units	None (empty string)
Sortable	NO
Queryable	YES

Table G- 70 NSIL_STREAM.standardVersion

Description	Version of product standard E.g. "1.0".	
Requirement mode	OPTIONAL	
Updateable	YES	
IPL Assigned	NO	
Attribute name	standardVersion	
Attribute type	TEXT	
Domain	TEXT_VALUE	
Max field length	10	
Data type	CORBA: string	
Units	None (empty string)	
Sortable	YES	
Queryable	YES	
Aliases	Category: Name:	STANAG4607 /GMTI/PacketHeader/VersionID

Table G- 71 NSIL_STREAM.sourceURL

Description	The address and access protocol in the form of an URL by which a user can directly access the data stream. For RTSP, HTTP, HTTPS and JPIP a standard URL syntax will be used (could include port number). For UDP/RTP, a URL notation using a '@-character-notaton' is used to allow a URL to define a broadcasted stream or a multicasted stream. A UDP broadcast source on port 8105 would be defined as 'udp://@:8105', a RTP broadcast source would be defined as 'rtp://@:8105'; similarly multicasted UDP/RTP source would be defined as 'udp://@multicast_address:port' or 'udp://@multicast_address:port'. This UDP/RTP @-syntax is identical to what is being used by the VideoLAN VLC application. For the broadcast situation it is assumed that the network takes care about forwarding the broadcasts between the LANs (e.g. using NIRIS) so the stream consumer does not need of the LAN-location of the or origial stream source. Note 1: Contrary to the productURL this attribute is queryable. This would allow clients to only discover streams using protocols that the client can consume. Note 2: When a stream is archived and no longer online, the sourceURL attribute shall be removed.
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	sourceURL
Attribute type	TEXT
Domain	TEXT_VALUE

Max field length	0
Data type	CORBA: string
Units	URL
Sortable	NO
Queryable	YES

Table G- 72 NSIL_STREAM.programID

Description	The program ID determines the stream ID within the Transport Stream that contains the PMT (Program Mapping Table) of the stream. In cases where the Transport Stream would contain more than one stream, this would allow clients to find quickly the program being advertised.
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	programID
Attribute type	INTEGER
Domain	INTEGER_RANGE: 3 .. 16383
Data type	CORBA: short
Units	None
Sortable	NO
Queryable	YES

2.3.15 NSIL_VIDEO

This section provides details for the metadata for a video file. Hints for implementation:

• The attribute 'encodingScheme' represents an important property of the product file itself. Therefore, this attribute needs to be "correct". It is needed for clients to know if they can play the file at all. Since a product file cannot change, it cannot be updated. For the creationMgr, if a client does not provide the encodingScheme attribute, and the server has no functionality to extract it from the video file, the server must give an error. Otherwise if the client does not provide it the server must fill in the value. If the client provides it, and the server can also extract it, and detects that the client provided value is incorrect, the server could throw an exception.

• The attributes 'averageBitRate', 'frameRate', 'frameResolutionRows', 'frameResolutionColumns' and 'scanningMode' are non-updatable, since they are directly related to the product file, which cannot be updated either. They are optional, because they are less important than the 'NSIL_VIDEO.encodingScheme'. They are not IPL-assigned to prevent servers from being required to interpret the whole video file. The following section lists attributes of the 'NSIL_VIDEO' entity.

Table G- 73 NSIL_VIDEO.averageBitRate

Description	The bit rate of the motion imagery product when being streamed.
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	averageBitRate
Attribute type	FLOATING_POINT
Domain	FLOATING_POINT_RANGE (0.0 .. 3e38)
Data type	CORBA: double
Units	Mbps
Sortable	NO
Queryable	YES

Table G- 74 NSIL_VIDEO.category

Description	An indicator of the specific category of motion image (information about the bands of visible or infrared region of electromagnetic continuum being used). The specific category of an image reveals its intended use or the nature of its collector.
Requirement mode	MANDATORY
Updateable	YES
IPL Assigned	NO
Attribute name	Category
Attribute type	TEXT
Domain	List: [VIS, IR, MS, HS]
Data type	CORBA: string
Units	None (empty string)
Sortable	NO
Queryable	YES

Table G- 75 NSIL_VIDEO.encodingScheme

Description	A code that indicates the manner in which the motion imagery was encoded.
Requirement mode	MANDATORY
Updateable	YES
IPL Assigned	NO
Attribute name	encodingScheme
Attribute type	TEXT
Domain	List: [264ON2, MPEG-2]
Data type	CORBA: string
Units	None (empty string)
Sortable	NO
Queryable	YES

Table G- 76 NSIL_VIDEO.frameRate

Description	The standard rate (in frames per second (FPS)) at which the motion imagery was captured. Note: The Motion Imagery Standards Profile identifies several values, such as: 23.98 FPS, 24 FPS, 29.97 FPS, 30 FPS, 59.94 FPS.
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	frameRate
Attribute type	FLOATING_POINT
Domain	FLOATING_POINT_RANGE (0.0 .. 3e38)
Data type	CORBA: double
Units	FPS
Sortable	NO
Queryable	YES

Table G- 77 NSIL_VIDEO.numberOfRows

Description	The frame vertical resolution.
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	numberOfRows
Attribute type	INTEGER
Domain	INTEGER_RANGE: 1 .. 2147483647
Data type	CORBA: unsignedLong
Units	None
Sortable	NO
Queryable	YES

Table G- 78 NSIL_VIDEcO.numberOfColumns

Description	The frame horizontal resolution.
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	numberOfColumns
Attribute type	INTEGER
Domain	INTEGER_RANGE: 1 .. 2147483647
Data type	CORBA: unsignedLong
Units	None
Sortable	NO
Queryable	YES

Table G- 79 NSIL_VIDEO.metadataEncodingScheme

Description	A code that indicates the manner in which the metadata for the motion imagery was encoded.
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	metadataEncodingScheme
Attribute type	TEXT
Domain	List: [KLV, NONE]
Data type	CORBA: string
Units	None (empty string)
Sortable	NO
Queryable	YES

Table G- 80 NSIL_VIDEO.MISMLevel

Description	From [AEDP-8]: The "Motion Imagery Systems (Spatial and Temporal) Matrix" (MISM) defines an ENGINEERING GUIDELINE for the simple identification of broad categories of Motion Imagery Systems. The intent of the MISM is to give user communities an easy to use, common shorthand reference language to describe the fundamental technical capabilities of NATO motion imagery systems.
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	MISMLevel
Attribute type	INTEGER
Domain	INTEGER_RANGE: 0 .. 12
Data type	CORBA: short
Units	None
Sortable	NO
Queryable	YES

Table G- 81 NSIL_VIDEO.scanningMode

Description	Indicate if progressive or interlaced scans are being applied.
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	scanningMode
Attribute type	TEXT
Domain	List: [INTERLACED, PROGRESSIVE]
Data type	CORBA: string
Units	None (empty string)
Sortable	NO
Queryable	YES

2.4 MINIMUM LAYER VIEWS

2.4.1 All the defined views, shown in the bullet list below, are individually described in the following sections.

- NSIL_IMAGERY_VIEW, Section • 2.4.3
- NSIL_GMTI_VIEW, Section 2.4.4
- NSIL_MESSAGE_VIEW, Section 2.4.5
- NSIL_VIDEO_VIEW, Section 2.4.6
- NSIL_ASSOCIATION_VIEW, Section 2.4.7

2.4.2 NSIL_ALL_VIEW

The 'NSIL_ALL_VIEW' is a superset of all the other views and hence includes all the defined metadata entities and attributes. Please note the following points:

• All the "part specific" parts like NSIL_GMTI, NSIL_IMAGERY, NSIL_VIDEO, etc in this view are shown to have a 0..1 cardinality. Be aware that the "part specific" parts are mutually exclusive; only one of them can exist at any time for one single part.

• To be able to query for associations in combination with some file product specific attributes, the NSIL_ASSOCIATION pseudo-entity is also included in the NSIL_ALL_VIEW view. When querying the result will contain association attributes and file product metadata attributes in a single DAG, provided a client explicitly mentions its interest in any of the NSIL_SOURCE and/or NSIL_DESTINATION entities.

• In the case of STANAG 4559, Edition 3 datamodel, depending on the number of additional layers supported beyond the minimum layer, some servers may choose not to implement all entities of the NSIL_ALL_VIEW. This view contains all possible types of products.

This view is orderable through the OrderMgr.

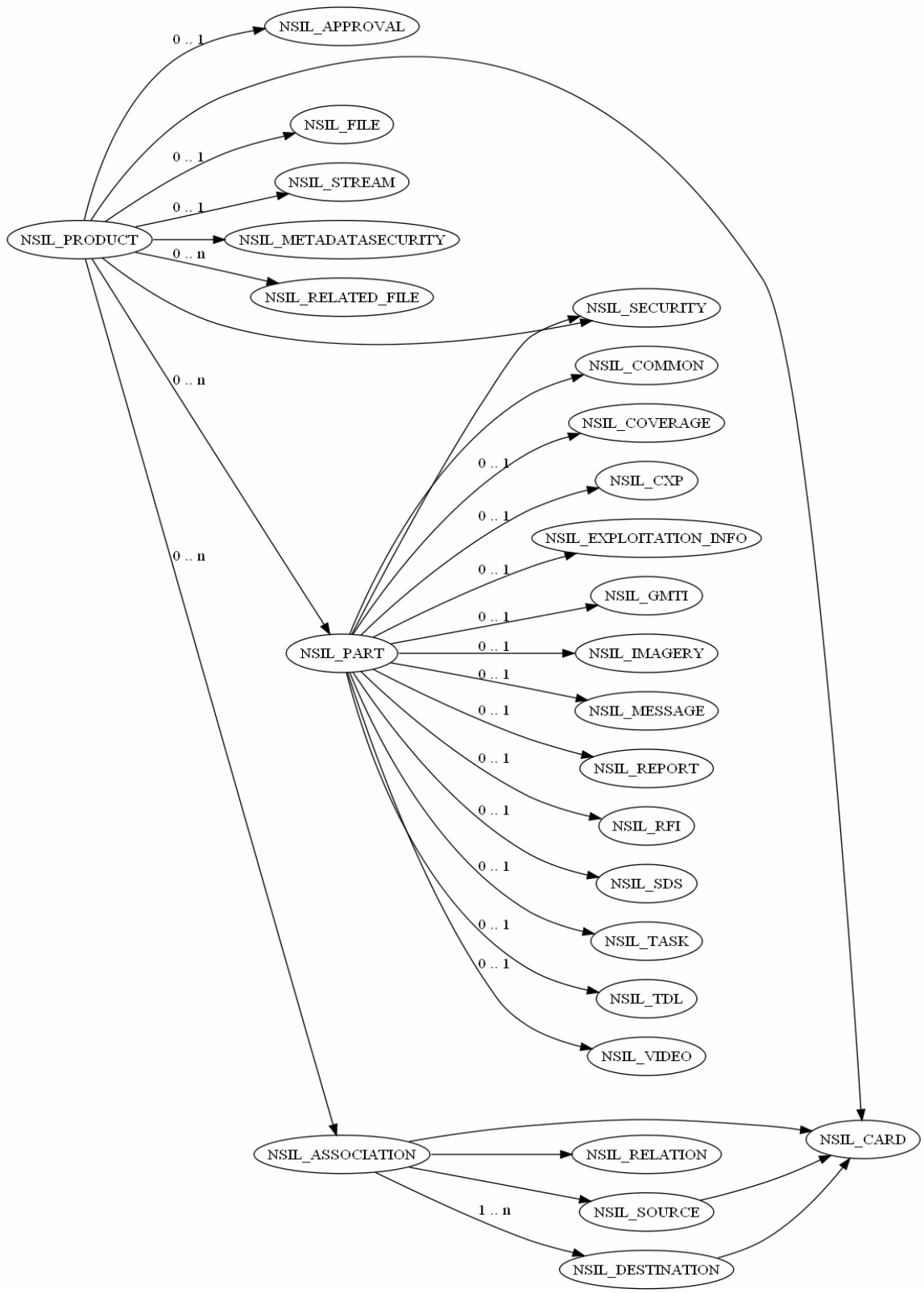

Figure G- 1 NSIL_ALL_VIEW

2.4.2.1 The layout of the full entity graph, showing all entity relationships for 'NSIL_ALL_VIEW', is shown in the table below.

Table G- 82 NSIL_ALL_VIEW

Node list	NSIL_PRODUCT, NSIL_APPROVAL, NSIL_CARD, NSIL_FILE, NSIL_STREAM, NSIL_METADATASECURITY, NSIL_RELATED_FILE, NSIL_SECURITY, NSIL_PART, NSIL_COMMON, NSIL_COVERAGE, NSIL_CXP, NSIL_EXPLOITATION_INFO, NSIL_GMTI, NSIL_IMAGERY, NSIL_MESSAGE, NSIL_REPORT, NSIL_RFI, NSIL_SDS, NSIL_TASK, NSIL_TDL, NSIL_VIDEO, NSIL_ASSOCIATION, NSIL_RELATION, NSIL_SOURCE, NSIL_DESTINATION		
Start node	**End node**	**Forward cardinality**	**Reverse cardinality**
NSIL_PRODUCT	NSIL_ASSOCIATION	ONE_TO_ZERO_OR_MORE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_APPROVAL	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_CARD	ONE_TO_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_FILE	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_STREAM	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_METADATASECURITY	ONE_TO_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_RELATED_FILE	ONE_TO_ZERO_OR_MORE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_SECURITY	ONE_TO_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_PART	ONE_TO_ZERO_OR_MORE	ONE_TO_ONE
NSIL_PART	NSIL_COMMON	ONE_TO_ONE	ONE_TO_ONE
NSIL_PART	NSIL_COVERAGE	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PART	NSIL_SECURITY	ONE_TO_ONE	ONE_TO_ZERO_OR_ONE
NSIL_PART	NSIL_EXPLOITATION_INFO	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PART	NSIL_CXP	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PART	NSIL_GMTI	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PART	NSIL_IMAGERY	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PART	NSIL_MESSAGE	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PART	NSIL_REPORT	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PART	NSIL_RFI	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PART	NSIL_SDS	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PART	NSIL_TASK	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PART	NSIL_TDL	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PART	NSIL_VIDEO	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_ASSOCIATION	NSIL_CARD	ONE_TO_ONE	ONE_TO_ZERO_OR_MORE
NSIL_ASSOCIATION	NSIL_SOURCE	ONE_TO_ONE	ONE_TO_ZERO_OR_MORE
NSIL_ASSOCIATION	NSIL_DESTINATION	ONE_TO_ONE_OR_MORE	ONE_TO_ZERO_OR_MORE
NSIL_ASSOCIATION	NSIL_RELATION	ONE_TO_ONE	ONE_TO_ONE
NSIL_SOURCE	NSIL_CARD	ONE_TO_ONE	ONE_TO_ZERO_OR_ONE
NSIL_DESTINATION	NSIL_CARD	ONE_TO_ONE	ONE_TO_ZERO_OR_ONE

Table G- 83 Conceptual attributes present in 'NSIL_ALL_VIEW'

Concept type	Attribute
CLASSIFICATION	NSIL_SECURITY.classification
DATASETTYPE	NSIL_PART: NSIL_COMMON.type
DATASIZE	NSIL_FILE.extent
DIRECTACCESS	NSIL_FILE.productURL
FOOTPRINT	NSIL_PART: NSIL_COVERAGE.spatialGeographicReferenceBox
MODIFICATIONDATE	NSIL_CARD.dateTimeModified
PRODUCTTITLE	NSIL_FILE.title
UNIQUEIDENTIFIER	NSIL_CARD.identifier

2.4.3 NSIL_IMAGERY_VIEW

The NSIL_IMAGERY_VIEW provides imagery specific metadata attributes in addition to the core metadata set. While this view could allow a server to expose imagery in formats different from STANAG 4545/NSIF (in which case the MIME type attribute would reflect an image type different from NSIF), the server expects all imagery in this view to be in STANAG 4545 format. By introducing the 'part' structure the new revision of the metadata model the server now supports cataloguing NSIF files with more than one image. Each image within the file is individually described in the metadata. The metadata model also supports description of reports with the NSIF file.

Notes:

• An NSIF file doesn't need to include an image segment to be a valid file according to STANAG 4545. However, to be catalogued under the NSIL_IMAGERY_VIEW it will be required that the NSIF file contains a minimum of one image segment.

• In the case of a streaming image the NSIL_IMAGERY entity is optional in this view.

• In the case of a file-product the NSIL_IMAGERY entity is mandatory.

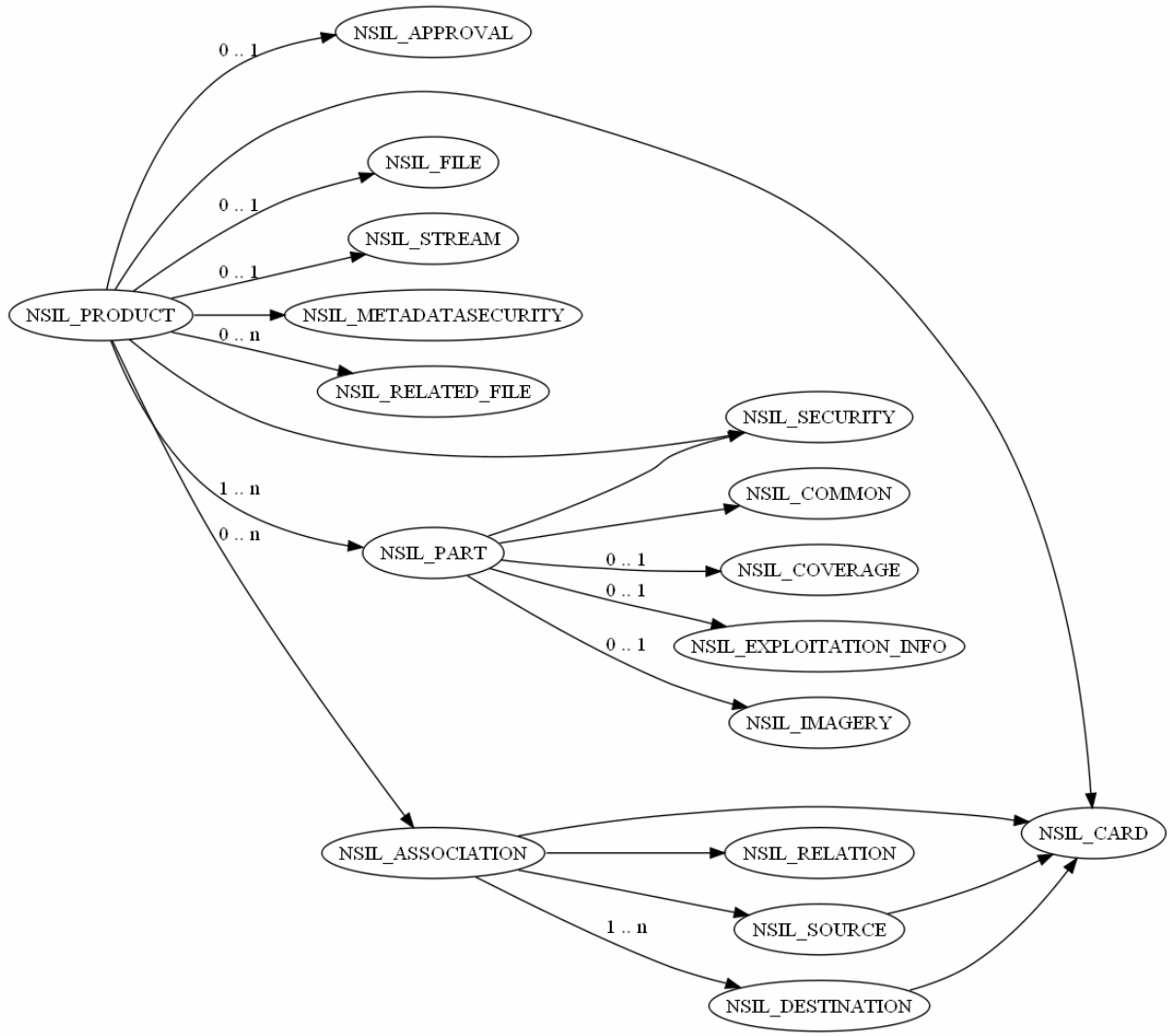

Figure G- 2 NSIL_IMAGERY_VIEW

2.4.3.1 Products with the following types of parts are present in 'NSIL_IMAGERY_VIEW'.

NSIL_PART:NSIL_COMMON.type = 'IMAGERY'

2.5.3.2 This view is orderable through the OrderMgr. The layout of the full entity graph, showing all entity relationships for 'NSIL_IMAGERY_VIEW', is shown in the table below. Note that the cardinality marked by grey cell and italic text is different from the cardinality between the same two entities in the NSIL_ALL view.

Table G- 84 NSIL_IMAGERY_VIEW

Node list	NSIL_PRODUCT, NSIL_APPROVAL, NSIL_CARD, NSIL_FILE, NSIL_STREAM, NSIL_METADATASECURITY, NSIL_RELATED_FILE, NSIL_SECURITY, NSIL_PART, NSIL_COMMON, NSIL_COVERAGE, NSIL_EXPLOITATION_INFO, NSIL_IMAGERY, NSIL_ASSOCIATION, NSIL_RELATION, NSIL_SOURCE, NSIL_DESTINATION		
Start node	**End node**	**Forward cardinality**	**Reverse cardinality**
NSIL_PRODUCT	NSIL_ASSOCIATION	ONE_TO_ZERO_OR_MORE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_APPROVAL	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_CARD	ONE_TO_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_FILE	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_STREAM	*ONE_TO_ZERO_OR_ONE*	ONE_TO_ONE
NSIL_PRODUCT	NSIL_METADATASECURITY	ONE_TO_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_RELATED_FILE	ONE_TO_ZERO_OR_MORE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_SECURITY	ONE_TO_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_PART	*ONE_TO_ONE_OR_MORE*	ONE_TO_ONE
NSIL_PART	NSIL_COMMON	ONE_TO_ONE	ONE_TO_ONE
NSIL_PART	NSIL_COVERAGE	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PART	NSIL_SECURITY	ONE_TO_ONE	ONE_TO_ZERO_OR_ONE
NSIL_PART	NSIL_EXPLOITATION_INFO	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PART	NSIL_IMAGERY	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_ASSOCIATION	NSIL_CARD	ONE_TO_ONE	ONE_TO_ZERO_OR_MORE
NSIL_ASSOCIATION	NSIL_SOURCE	ONE_TO_ONE	ONE_TO_ZERO_OR_MORE
NSIL_ASSOCIATION	NSIL_DESTINATION	ONE_TO_ONE_OR_MORE	ONE_TO_ZERO_OR_MORE
NSIL_ASSOCIATION	NSIL_RELATION	ONE_TO_ONE	ONE_TO_ONE
NSIL_SOURCE	NSIL_CARD	ONE_TO_ONE	ONE_TO_ZERO_OR_ONE
NSIL_DESTINATION	NSIL_CARD	ONE_TO_ONE	ONE_TO_ZERO_OR_ONE

Table G- 85 Conceptual attributes present in 'NSIL_IMAGERY_VIEW'

Concept type	**Attribute**
CLASSIFICATION	NSIL_SECURITY.classification
DATASETTYPE	NSIL_PART:NSIL_COMMON.type
DATASIZE	NSIL_FILE.extent
DIRECTACCESS	NSIL_FILE.productURL
FOOTPRINT	NSIL_PART:NSIL_COVERAGE.spatialGeographicReferenceBox
MODIFICATIONDATE	NSIL_CARD.dateTimeModified
PRODUCTTITLE	NSIL_FILE.title
UNIQUEIDENTIFIER	NSIL_CARD.identifier

2.4.4 NSIL_GMTI_VIEW

Notes:

- In the case of streaming GMTI the NSIL_GMTI entity will not be used since this entity describes data particular for only one particular radar revisit and not for a stream of GMTI messages.
- In the case of a file-product the NSIL_GMTI entity is mandatory.

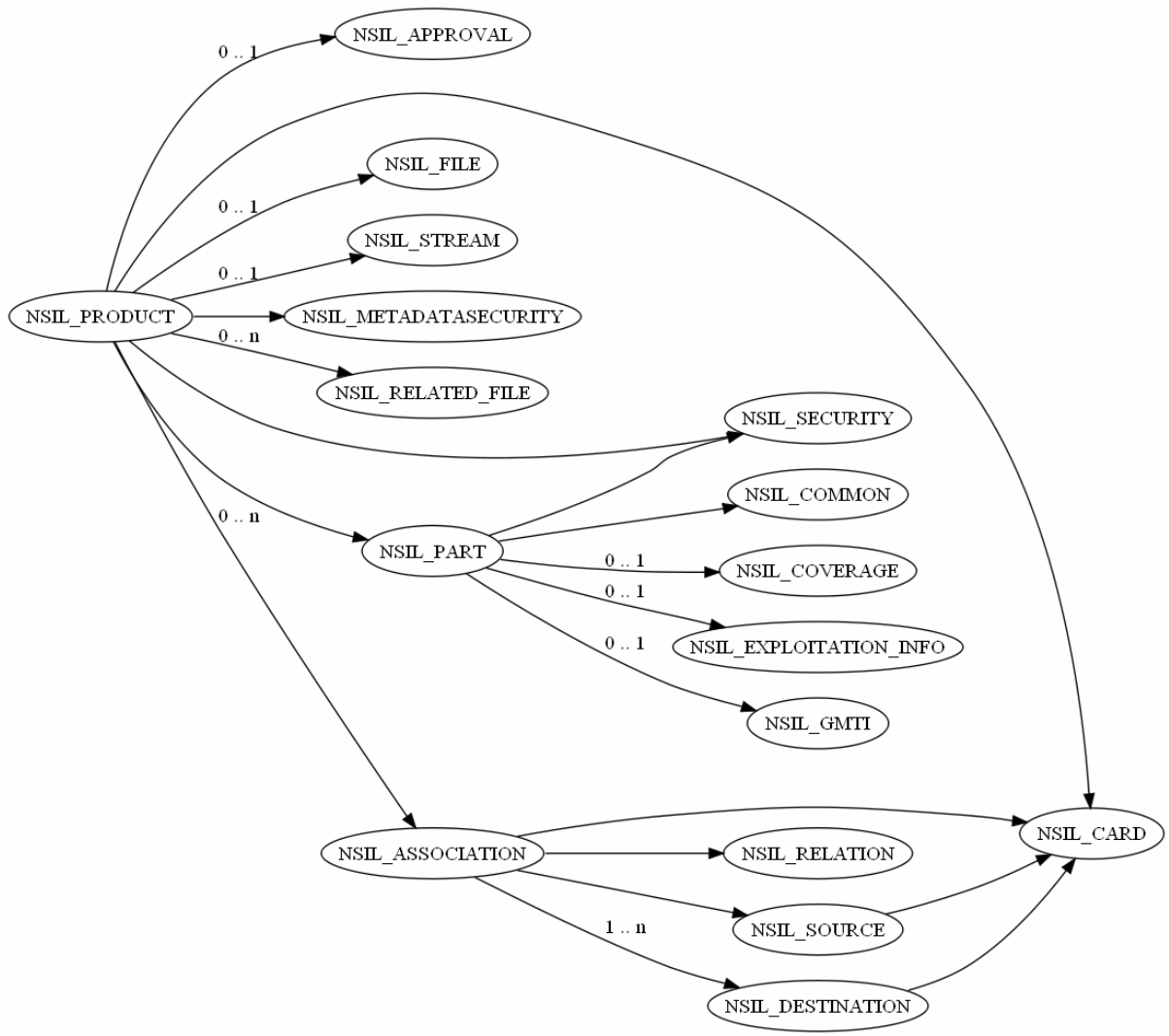

Figure G- 3 NSIL_GMTI_VIEW

2.4.4.1 Products with the following types of parts are present in 'NSIL_GMTI_VIEW'.

NSIL_PART:NSIL_COMMON.type = 'GMTI'

2.4.4.2 This view is orderable through the OrderMgr. The layout of the full entity graph, showing all entity relationships for 'NSIL_GMTI_VIEW', is shown in the table below. Note that the cardinality marked by grey cell and italic text is different from the cardinality between the same two entities in the NSIL_ALL view.

Table G- 86 NSIL_GMTI_VIEW

Node list	NSIL_PRODUCT, NSIL_APPROVAL, NSIL_CARD, NSIL_FILE, NSIL_STREAM, NSIL_METADATASECURITY, NSIL_RELATED_FILE, NSIL_SECURITY, NSIL_PART, NSIL_COMMON, NSIL_COVERAGE, NSIL_EXPLOITATION_INFO, NSIL_GMTI, NSIL_ASSOCIATION, NSIL_RELATION, NSIL_SOURCE, NSIL_DESTINATION		
Start node	**End node**	**Forward cardinality**	**Reverse cardinality**
NSIL_PRODUCT	NSIL_ASSOCIATION	ONE_TO_ZERO_OR_MORE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_APPROVAL	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_CARD	ONE_TO_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_FILE	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_STREAM	*ONE_TO_ZERO_OR_ONE*	ONE_TO_ONE
NSIL_PRODUCT	NSIL_METADATASECURITY	ONE_TO_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_RELATED_FILE	ONE_TO_ZERO_OR_MORE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_SECURITY	ONE_TO_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_PART	*ONE_TO_ONE*	ONE_TO_ONE
NSIL_PART	NSIL_COMMON	ONE_TO_ONE	ONE_TO_ONE
NSIL_PART	NSIL_COVERAGE	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PART	NSIL_SECURITY	ONE_TO_ONE	ONE_TO_ZERO_OR_ONE
NSIL_PART	NSIL_EXPLOITATION_INFO	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PART	NSIL_GMTI	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_ASSOCIATION	NSIL_CARD	ONE_TO_ONE	ONE_TO_ZERO_OR_MORE
NSIL_ASSOCIATION	NSIL_SOURCE	ONE_TO_ONE	ONE_TO_ZERO_OR_MORE
NSIL_ASSOCIATION	NSIL_DESTINATION	ONE_TO_ONE_OR_MORE	ONE_TO_ZERO_OR_MORE
NSIL_ASSOCIATION	NSIL_RELATION	ONE_TO_ONE	ONE_TO_ONE
NSIL_SOURCE	NSIL_CARD	ONE_TO_ONE	ONE_TO_ZERO_OR_ONE
NSIL_DESTINATION	NSIL_CARD	ONE_TO_ONE	ONE_TO_ZERO_OR_ONE

Table G- 87 Conceptual attributes present in 'NSIL_GMTI_VIEW'

Concept type	**Attribute**
CLASSIFICATION	NSIL_SECURITY.classification
DATASETTYPE	NSIL_PART:NSIL_COMMON.type
DATASIZE	NSIL_FILE.extent
DIRECTACCESS	NSIL_FILE.productURL
FOOTPRINT	NSIL_PART:NSIL_COVERAGE.spatialGeographicReferenceBox
MODIFICATIONDATE	NSIL_CARD.dateTimeModified
PRODUCTTITLE	NSIL_FILE.title
UNIQUEIDENTIFIER	NSIL_CARD.identifier

2.4.5 NSIL_MESSAGE_VIEW

This view is provided to be used for searching for messages and ingest of messages.

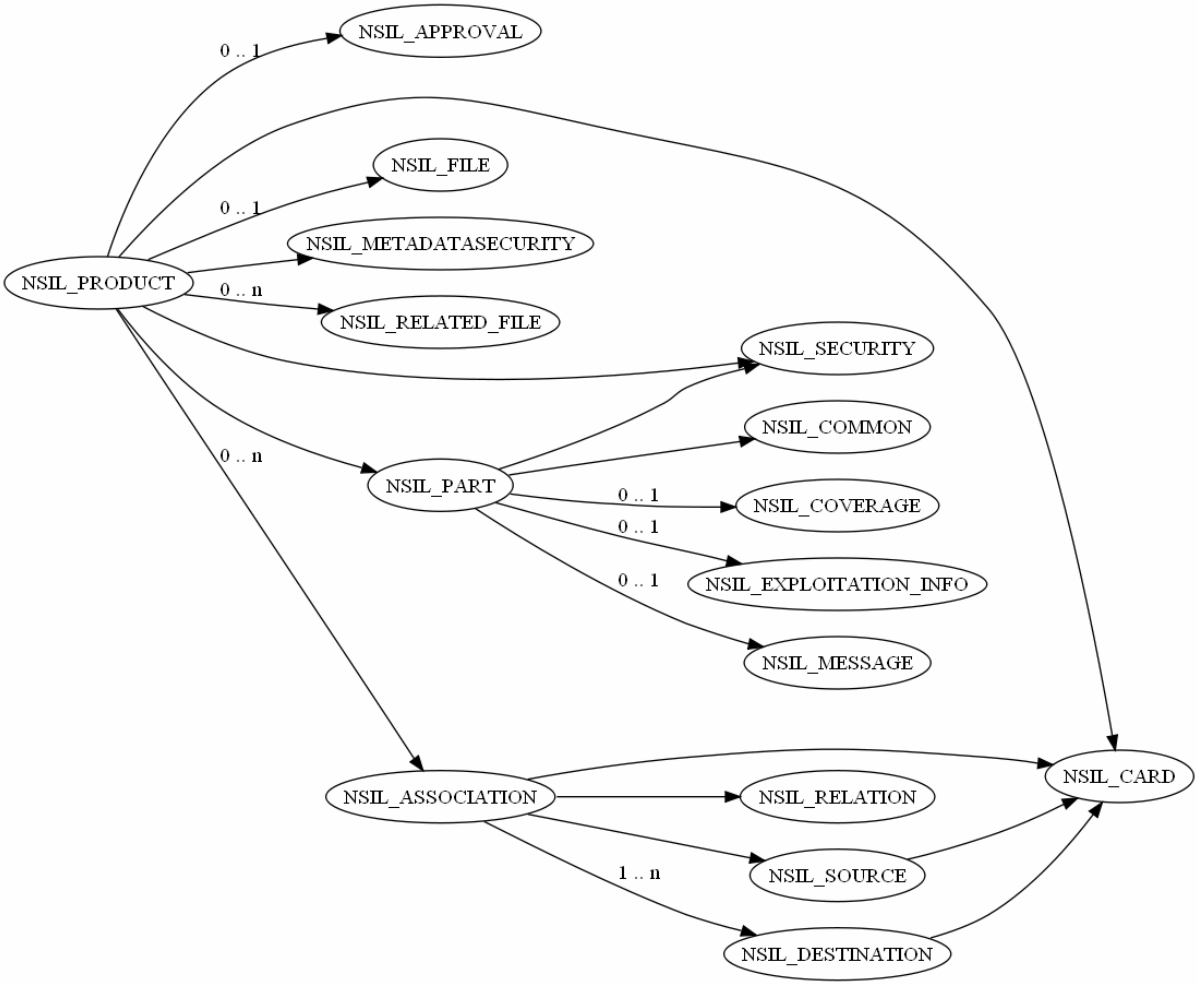

Figure G- 4 NSIL_MESSAGE_VIEW

2.4.5.1 Products with the following types of parts are present in 'NSIL_MESSAGE_VIEW'.
NSIL_PART:NSIL_COMMON.type = 'MESSAGE'

2.4.5.2 This view is orderable through the OrderMgr. The layout of the full entity graph, showing all entity relationships for 'NSIL_MESSAGE_VIEW', is shown in the table below. Note that the cardinality marked by grey cell and italic text is different from the cardinality between the same two entities in the NSIL_ALL view.

Table G- 88 NSIL_MESSAGE_VIEW

Node list	NSIL_PRODUCT, NSIL_APPROVAL, NSIL_CARD, NSIL_FILE, NSIL_METADATASECURITY, NSIL_RELATED_FILE, NSIL_SECURITY, NSIL_PART, NSIL_COMMON, NSIL_COVERAGE, NSIL_EXPLOITATION_INFO, NSIL_MESSAGE, NSIL_ASSOCIATION, NSIL_RELATION, NSIL_SOURCE, NSIL_DESTINATION		
Start node	**End node**	**Forward cardinality**	**Reverse cardinality**
NSIL_PRODUCT	NSIL_ASSOCIATION	ONE_TO_ZERO_OR_MORE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_APPROVAL	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_CARD	ONE_TO_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_FILE	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_METADATASECURITY	ONE_TO_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_RELATED_FILE	ONE_TO_ZERO_OR_MORE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_SECURITY	ONE_TO_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_PART	*ONE_TO_ONE*	ONE_TO_ONE
NSIL_PART	NSIL_COMMON	ONE_TO_ONE	ONE_TO_ONE
NSIL_PART	NSIL_COVERAGE	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PART	NSIL_SECURITY	ONE_TO_ONE	ONE_TO_ZERO_OR_ONE
NSIL_PART	NSIL_EXPLOITATION_INFO	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PART	NSIL_MESSAGE	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_ASSOCIATION	NSIL_CARD	ONE_TO_ONE	ONE_TO_ZERO_OR_MORE
NSIL_ASSOCIATION	NSIL_SOURCE	ONE_TO_ONE	ONE_TO_ZERO_OR_MORE
NSIL_ASSOCIATION	NSIL_DESTINATION	ONE_TO_ONE_OR_MORE	ONE_TO_ZERO_OR_MORE
NSIL_ASSOCIATION	NSIL_RELATION	ONE_TO_ONE	ONE_TO_ONE
NSIL_SOURCE	NSIL_CARD	ONE_TO_ONE	ONE_TO_ZERO_OR_ONE
NSIL_DESTINATION	NSIL_CARD	ONE_TO_ONE	ONE_TO_ZERO_OR_ONE

Table G- 89 Conceptual attributes present in 'NSIL_MESSAGE_VIEW'

Concept type	**Attribute**
CLASSIFICATION	NSIL_SECURITY.classification
DATASETTYPE	NSIL_PART:NSIL_COMMON.type
DATASIZE	NSIL_FILE.extent
DIRECTACCESS	NSIL_FILE.productURL
FOOTPRINT	NSIL_PART:NSIL_COVERAGE.spatialGeographicReferenceBox
MODIFICATIONDATE	NSIL_CARD.dateTimeModified
PRODUCTTITLE	NSIL_FILE.title
UNIQUEIDENTIFIER	NSIL_CARD.identifier

2.4.6 NSIL_VIDEO_VIEW

While this view allows a server to expose motion imagery in formats different from STANAG 4609, the server expects all motion imagery in this view to be in a STANAG 4609 approved format.

Notes:

• In the case of streaming video the NSIL_VIDEO entity is optional in this view.

• In the case of a file-product the NSIL_VIDEO entity is mandatory.

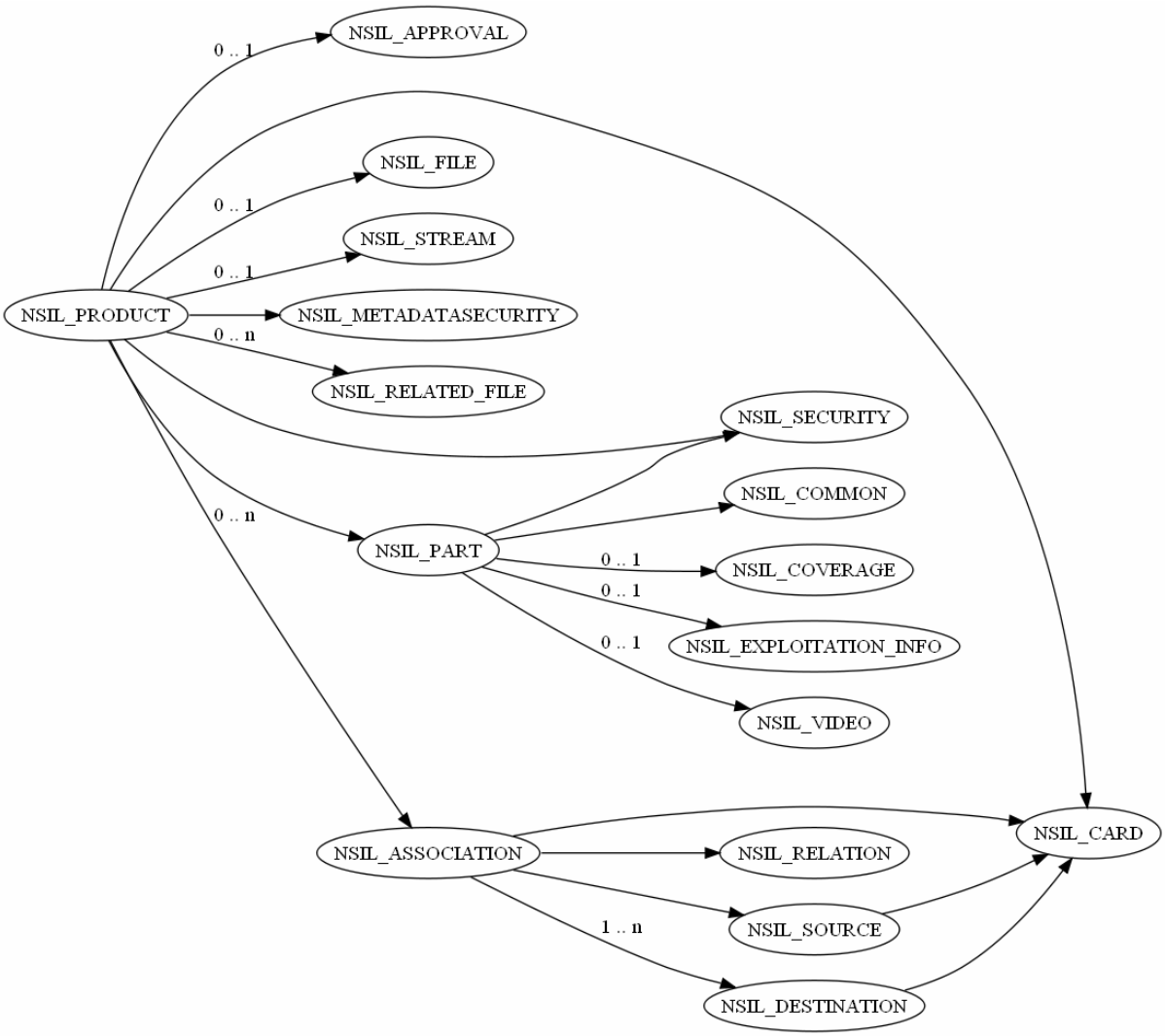

Figure G- 5 NSIL_VIDEO_VIEW

2.4.6.1 Products with the following types of parts are present in 'NSIL_VIDEO_VIEW'.

NSIL_PART:NSIL_COMMON.type = 'VIDEO'

2.4.6.2 This view is orderable through the OrderMgr. The layout of the full entity graph, showing all entity relationships for 'NSIL_VIDEO_VIEW', is shown in the table below. Note that the cardinality marked by grey cell and italic text is different from the cardinality between the same two entities in the NSIL_ALL view.

Table G- 90 NSIL_VIDEO_VIEW

Node list	NSIL_PRODUCT, NSIL_APPROVAL, NSIL_CARD, NSIL_FILE, NSIL_STREAM, NSIL_METADATASECURITY, NSIL_RELATED_FILE, NSIL_SECURITY, NSIL_PART, NSIL_COMMON, NSIL_COVERAGE, NSIL_EXPLOITATION_INFO, NSIL_VIDEO, NSIL_ASSOCIATION, NSIL_RELATION, NSIL_SOURCE, NSIL_DESTINATION		
Start node	**End node**	**Forward cardinality**	**Reverse cardinality**
NSIL_PRODUCT	NSIL_ASSOCIATION	ONE_TO_ZERO_OR_MORE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_APPROVAL	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_CARD	ONE_TO_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_FILE	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_STREAM	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_METADATASECURITY	ONE_TO_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_RELATED_FILE	ONE_TO_ZERO_OR_MORE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_SECURITY	ONE_TO_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_PART	*ONE_TO_ONE*	ONE_TO_ONE
NSIL_PART	NSIL_COMMON	ONE_TO_ONE	ONE_TO_ONE
NSIL_PART	NSIL_COVERAGE	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PART	NSIL_SECURITY	ONE_TO_ONE	ONE_TO_ZERO_OR_ONE
NSIL_PART	NSIL_EXPLOITATION_INFO	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PART	NSIL_VIDEO	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_ASSOCIATION	NSIL_CARD	ONE_TO_ONE	ONE_TO_ZERO_OR_MORE
NSIL_ASSOCIATION	NSIL_SOURCE	ONE_TO_ONE	ONE_TO_ZERO_OR_MORE
NSIL_ASSOCIATION	NSIL_DESTINATION	ONE_TO_ONE_OR_MORE	ONE_TO_ZERO_OR_MORE
NSIL_ASSOCIATION	NSIL_RELATION	ONE_TO_ONE	ONE_TO_ONE
NSIL_SOURCE	NSIL_CARD	ONE_TO_ONE	ONE_TO_ZERO_OR_ONE
NSIL_DESTINATION	NSIL_CARD	ONE_TO_ONE	ONE_TO_ZERO_OR_ONE

Table G- 91 Conceptual attributes present in 'NSIL_VIDEO_VIEW'

Concept type	**Attribute**
CLASSIFICATION	NSIL_SECURITY.classification
DATASETTYPE	NSIL_PART:NSIL_COMMON.type
DATASIZE	NSIL_FILE.extent
DIRECTACCESS	NSIL_FILE.productURL
FOOTPRINT	NSIL_PART:NSIL_COVERAGE.spatialGeographicReferenceBox
MODIFICATIONDATE	NSIL_CARD.dateTimeModified
PRODUCTTITLE	NSIL_FILE.title
UNIQUEIDENTIFIER	NSIL_CARD.identifier

2.4.7 NSIL_ASSOCIATION_VIEW

2.4.7.1 Associations are handled as if they were products. The main reason for this is that the GIAS specification allows creation of associations, but does not define a way to access these associations. As Associations can have attributes, modeling

them as Products seemed to be a logical fit.

2.4.7.2 Associations, being meta-Products, are modeled as metadata-only Products. This means that they cannot be ordered through the OrderMgr, as they do not have a file associated with them. Like their normal Product counterparts, the attributes can be updated and deleted with the UpdateMgr. Associations within the GIAS model are one-to-zero-or-more relationships. In order to model this, the metadata model needed is shown in the entity graph below. The association itself is described by the attributes in the NSIL_RELATION entity. Like regular products, an association also has a NSIL_CARD entity which is used to find the association. In order to represent the Products involved in the association, two pseudo-entities called 'NSIL_SOURCE' and 'NSIL_DESTINATION' are introduced. NSIL_SOURCE is used to identify the originator of the relationship, while NSIL_DESTINATION is used to identify the receiving ends of the relationship. NSIL_SOURCE and NSIL_DESTINATION do not have any attributes themselves, but serve as an identifying step into the products they represent. The actual product is identified by its NSIL_CARD entity, which carries the 'identifier' attribute.

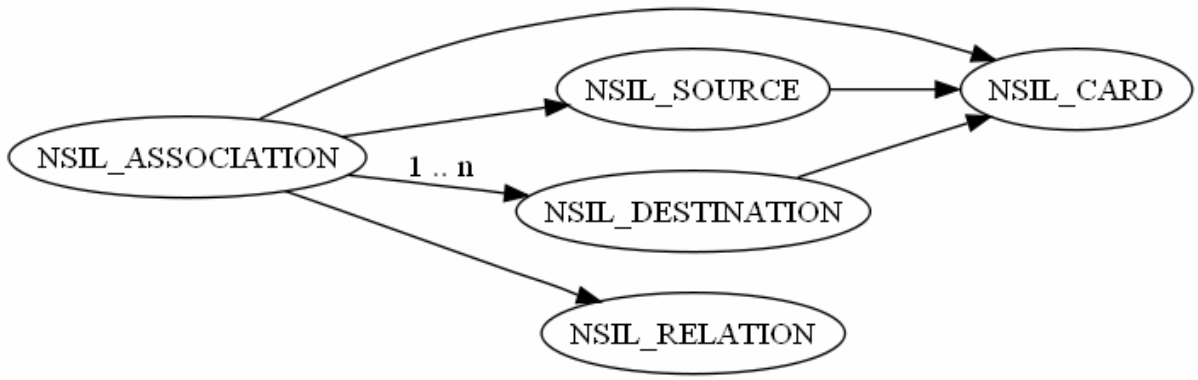

Figure G- 6 NSIL_ASSOCIATION_VIEW

2.4.7.3 It is important to note that this View is not orderable through the OrderMgr.

2.4.7.4 The layout of the full entity graph, showing all entity relationships for 'NSIL_ASSOCIATION_VIEW', is shown in the table below.

Table G- 92 NSIL_ASSOCIATION_VIEW

Node list	NSIL_ASSOCIATION, NSIL_CARD, NSIL_SOURCE, NSIL_DESTINATION, NSIL_RELATION		
Start node	**End node**	**Forward cardinality**	**Reverse cardinality**
NSIL_ASSOCIATION	NSIL_CARD	ONE_TO_ONE	ONE_TO_ZERO_OR_MORE
NSIL_ASSOCIATION	NSIL_SOURCE	ONE_TO_ONE	ONE_TO_ZERO_OR_MORE
NSIL_ASSOCIATION	NSIL_DESTINATION	ONE_TO_ONE_OR_MORE	ONE_TO_ZERO_OR_MORE
NSIL_ASSOCIATION	NSIL_RELATION	ONE_TO_ONE	ONE_TO_ONE
NSIL_SOURCE	NSIL_CARD	ONE_TO_ONE	ONE_TO_ZERO_OR_ONE
NSIL_DESTINATION	NSIL_CARD	ONE_TO_ONE	ONE_TO_ZERO_OR_ONE

Table G- 93 Conceptual attributes present in 'NSIL_ASSOCIATION_VIEW'

Concept type	Attribute
MODIFICATIONDATE	NSIL_CARD.dateTimeModified
UNIQUEIDENTIFIER	NSIL_CARD.identifier

3 REPORTING LAYER

3.1 REPORTING LAYER ENTITIES

3.1.1 NSIL_REPORT

The following section lists attributes of the 'NSIL_REPORT' entity.

Table G- 94 NSIL_REPORT.originatorsRequestSerialNumber

Description	Based on the originators request serial number STANAG 3277
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	originatorsRequestSerialNumber
Attribute type	TEXT
Domain	TEXT_VALUE
Max field length	10
Data type	CORBA: string
Units	None (empty string)
Sortable	NO
Queryable	YES

Table G- 95 NSIL_REPORT.priority

Description	A priority marking of the report
Requirement mode	MANDATORY
Updateable	YES
IPL Assigned	NO
Attribute name	priority
Attribute type	TEXT
Domain	List: [ROUTINE, PRIORITY, IMMEDIATE, FLASH]
Data type	CORBA: string
Units	None (empty string)
Sortable	NO

Queryable	YES

Table G- 96 NSIL_REPORT.type

Description	The specific type of report.
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	type
Attribute type	TEXT
Domain	List:[IQREP, ISRSPOTREP, MIEXREP, MTIEXREP, RECCEXREP, WLEXREP]
Data type	CORBA: string
Units	None (empty string)
Sortable	NO
Queryable	YES

3.2 REPORTING LAYER VIEWS

3.2.1 All the defined views, shown in the bullet list below, are individually described in the following sections.

NSIL_REPORT_VIEW

3.2.2 NSIL_REPORT_VIEW

The NSIL_REPORT_VIEW is included to enable the server to better support cataloguing of exploitation products, where exploitation products are defined as products that have been generated manually, semi automatically or automatically by processing a raw sensor product and enriching it with information. Within the NATO there are several standardized report types available depending on the tasking, the types of sensors and the information needs.

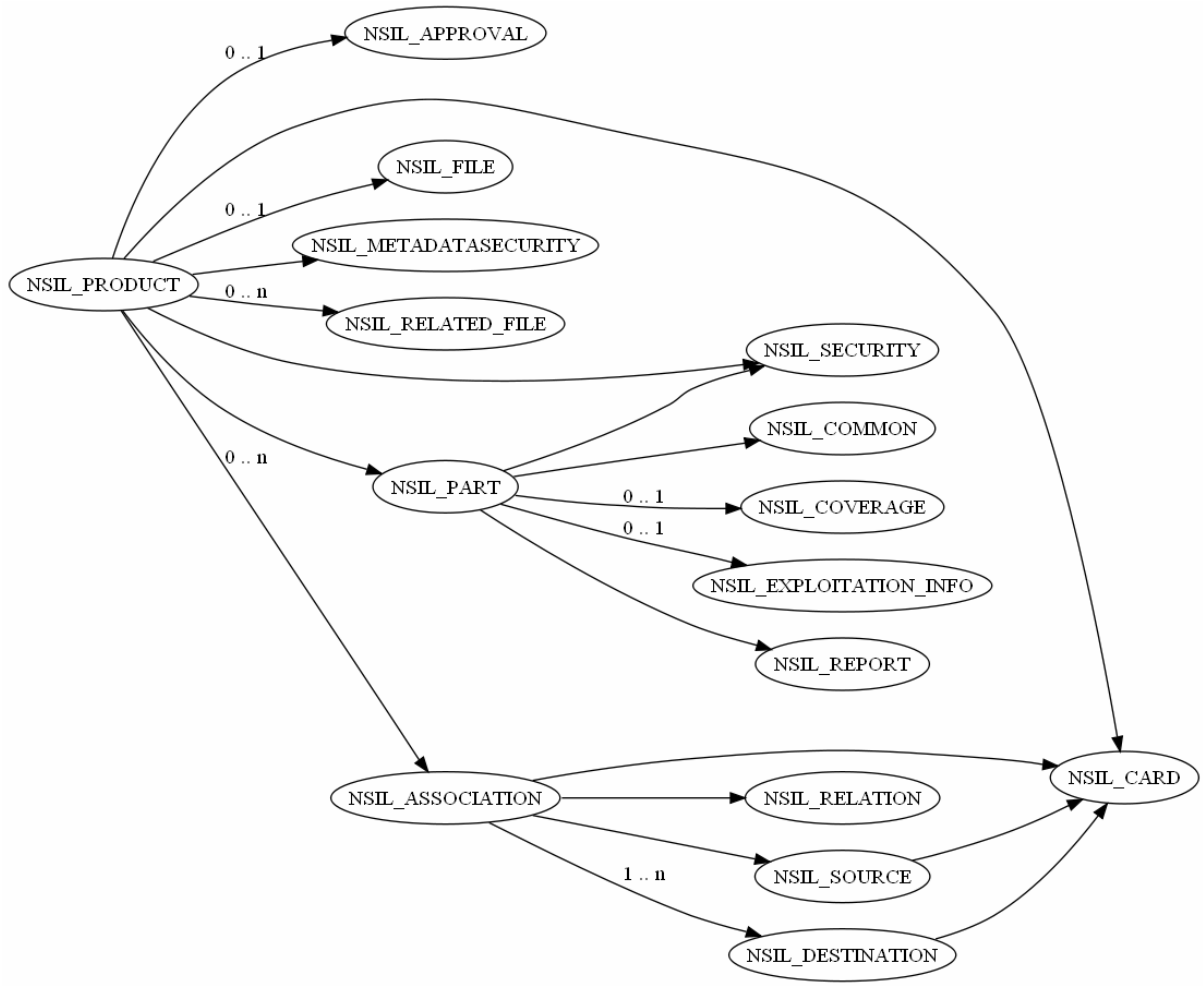

Figure G- 7 NSIL_REPORT_VIEW

3.2.2.1 Products with the following types of parts are present in 'NSIL_REPORT_VIEW'.

NSIL_PART:NSIL_COMMON.type = 'REPORT'

3.2.2.2 This view is orderable through the OrderMgr. The layout of the full entity graph, showing all entity relationships for 'NSIL_REPORT_VIEW', is shown in the table below. Note that the cardinality marked by grey cell and italic text is different from the cardinality between the same two entities in the NSIL_ALL view.

Table G- 97 NSIL_REPORT_VIEW

Node list	NSIL_PRODUCT, NSIL_APPROVAL, NSIL_CARD, NSIL_FILE, NSIL_METADATASECURITY, NSIL_RELATED_FILE, NSIL_SECURITY, NSIL_PART, NSIL_COMMON, NSIL_COVERAGE, NSIL_EXPLOITATION_INFO, NSIL_REPORT, NSIL_ASSOCIATION, NSIL_RELATION, NSIL_SOURCE, NSIL_DESTINATION		
Start node	**End node**	**Forward cardinality**	**Reverse cardinality**
NSIL_PRODUCT	NSIL_ASSOCIATION	ONE_TO_ZERO_OR_MORE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_APPROVAL	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_CARD	ONE_TO_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_FILE	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_METADATASECURITY	ONE_TO_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_RELATED_FILE	ONE_TO_ZERO_OR_MORE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_SECURITY	ONE_TO_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_PART	*ONE_TO_ONE*	ONE_TO_ONE
NSIL_PART	NSIL_COMMON	ONE_TO_ONE	ONE_TO_ONE
NSIL_PART	NSIL_COVERAGE	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PART	NSIL_SECURITY	ONE_TO_ONE	ONE_TO_ZERO_OR_ONE
NSIL_PART	NSIL_EXPLOITATION_INFO	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PART	NSIL_REPORT	*ONE_TO_ONE*	ONE_TO_ONE
NSIL_ASSOCIATION	NSIL_CARD	ONE_TO_ONE	ONE_TO_ZERO_OR_MORE
NSIL_ASSOCIATION	NSIL_SOURCE	ONE_TO_ONE	ONE_TO_ZERO_OR_MORE
NSIL_ASSOCIATION	NSIL_DESTINATION	ONE_TO_ONE_OR_MORE	ONE_TO_ZERO_OR_MORE
NSIL_ASSOCIATION	NSIL_RELATION	ONE_TO_ONE	ONE_TO_ONE
NSIL_SOURCE	NSIL_CARD	ONE_TO_ONE	ONE_TO_ZERO_OR_ONE
NSIL_DESTINATION	NSIL_CARD	ONE_TO_ONE	ONE_TO_ZERO_OR_ONE

Table G- 98 Conceptual attributes present in 'NSIL_REPORT_VIEW'

Concept type	**Attribute**
CLASSIFICATION	NSIL_SECURITY.classification
DATASETTYPE	NSIL_PART:NSIL_COMMON.type
DATASIZE	NSIL_FILE.extent
DIRECTACCESS	NSIL_FILE.productURL
FOOTPRINT	NSIL_PART:NSIL_COVERAGE.spatialGeographicReferenceBox
MODIFICATIONDATE	NSIL_CARD.dateTimeModified
PRODUCTTITLE	NSIL_FILE.title
UNIQUEIDENTIFIER	NSIL_CARD.identifier

4 CCIRM LAYER

4.1 CCIRM LAYER ENTITIES

4.1.1 NSIL_CXP

The following section lists attributes of the 'NSIL_CXP' entity.

Table G- 99 NSIL_CXP.status

Description	A status field describing whether the Collection and Exploitation Plan is active, planned or expired.
Requirement mode	MANDATORY
Updateable	YES
IPL Assigned	NO
Attribute name	status
Attribute type	TEXT
Domain	List: [DRAFT, COORDINATEDDRAFT, CURRENT, EXPIRED, REJECTED]
Data type	CORBA: string
Units	None (empty string)
Sortable	NO
Queryable	YES

4.1.2 NSIL_RFI

The following section lists attributes of the 'NSIL_RFI' entity.

Table G- 100 NSIL_RFI.forAction

Description	A nation, command, agency, organization or unit requested to provide a response. Ref. STANAG 2149 (edition 6). Note that the forAction value is singular and only identifies one entity.
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	forAction
Attribute type	TEXT
Domain	TEXT_VALUE
Max field length	50
Data type	CORBA: string
Units	None (empty string)
Sortable	NO
Queryable	YES

Table G- 101 NSIL_RFI.forInformation

Description	A comma-separated list of nations, commands, agencies, organizations and units which may have an interest in the response. Ref. STANAG 2149 (edition 6)
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	forInformation
Attribute type	TEXT
Domain	TEXT_VALUE
Max field length	200
Data type	CORBA: string
Units	None (empty string)
Sortable	NO
Queryable	YES

Table G- 102 NSIL_RFI.serialNumber

Description	Unique human readable string identifying the RFI instance.
Requirement mode	MANDATORY
Updateable	YES
IPL Assigned	NO
Attribute name	serialNumber
Attribute type	TEXT
Domain	TEXT_VALUE
Max field length	30
Data type	CORBA: string
Units	None (empty string)
Sortable	NO
Queryable	YES

Table G- 103 NSIL_RFI.status

Description	Indicates the status of the RFI.s
Requirement mode	MANDATORY
Updateable	YES
IPL Assigned	NO
Attribute name	status
Attribute type	TEXT
Domain	List:[APPROVED, INACTION, STOPPED, FULFILLED]
Data type	CORBA: string
Units	None (empty string)
Sortable	NO
Queryable	YES

Table G- 104 NSIL_RFI.workflowStatus

Description	Indicates the workflow status of the RFI.
Requirement mode	MANDATORY
Updateable	YES
IPL Assigned	NO
Attribute name	workflowStatus
Attribute type	TEXT
Domain	List: [NEW, ACCEPTED, DENIED, CANCELLED, COMPLETED]
Data type	CORBA: string
Units	None (empty string)
Sortable	NO
Queryable	YES

4.1.3 NSIL_TASK

The following section lists attributes of the 'NSIL_TASK' entity.

Table G- 105 NSIL_TASK.comments

Description	Status of the task
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	comments
Attribute type	TEXT
Domain	TEXT_VALUE
Max field length	255
Data type	CORBA: string
Units	None (empty string)
Sortable	NO
Queryable	YES

Table G- 106 NSIL_TASK.status

Description	Status of the task
Requirement mode	MANDATORY
Updateable	YES
IPL Assigned	NO
Attribute name	status
Attribute type	TEXT
Domain	List:[PLANNED, ACKNOWLEDGED, ONGOING, ACCOMPLISHED, INTERRUPTED, INFEASIBLE, CANCELLED]
Data type	CORBA: string
Units	None (empty string)
Sortable	NO
Queryable	YES

4.2 CCIRM LAYER VIEWS

All the defined views, shown in the bullet list below, are individually described in the following sections.

- NSIL_CCIRM_VIEW

4.2.1 NSIL_CCIRM_VIEW

The 'NSIL_CCIRM_VIEW' is included to enable a 4559 Edition 3 to support the Collection Coordination Information Requirements Management (CCIRM) process.

Notes:
This view may return four different and mutually exclusive products, a CXP, an IR, a RFI, a TASK and a Geographic AOI. In the case of Geographic AOI being returned there will be no "part specific" parts. It is assumed that the RFI and the Geographic AOI are stored as XML files accessible through the NSIL_FILE.productURL. The IR is an artificial metadata-only product with no specialized attributes, it is only there as placeholder to allow associations to be linked to a uniquely identified IR entry. For details about an IR the client will have to look at the content of a CXP file. The TASK is a metadata only product.

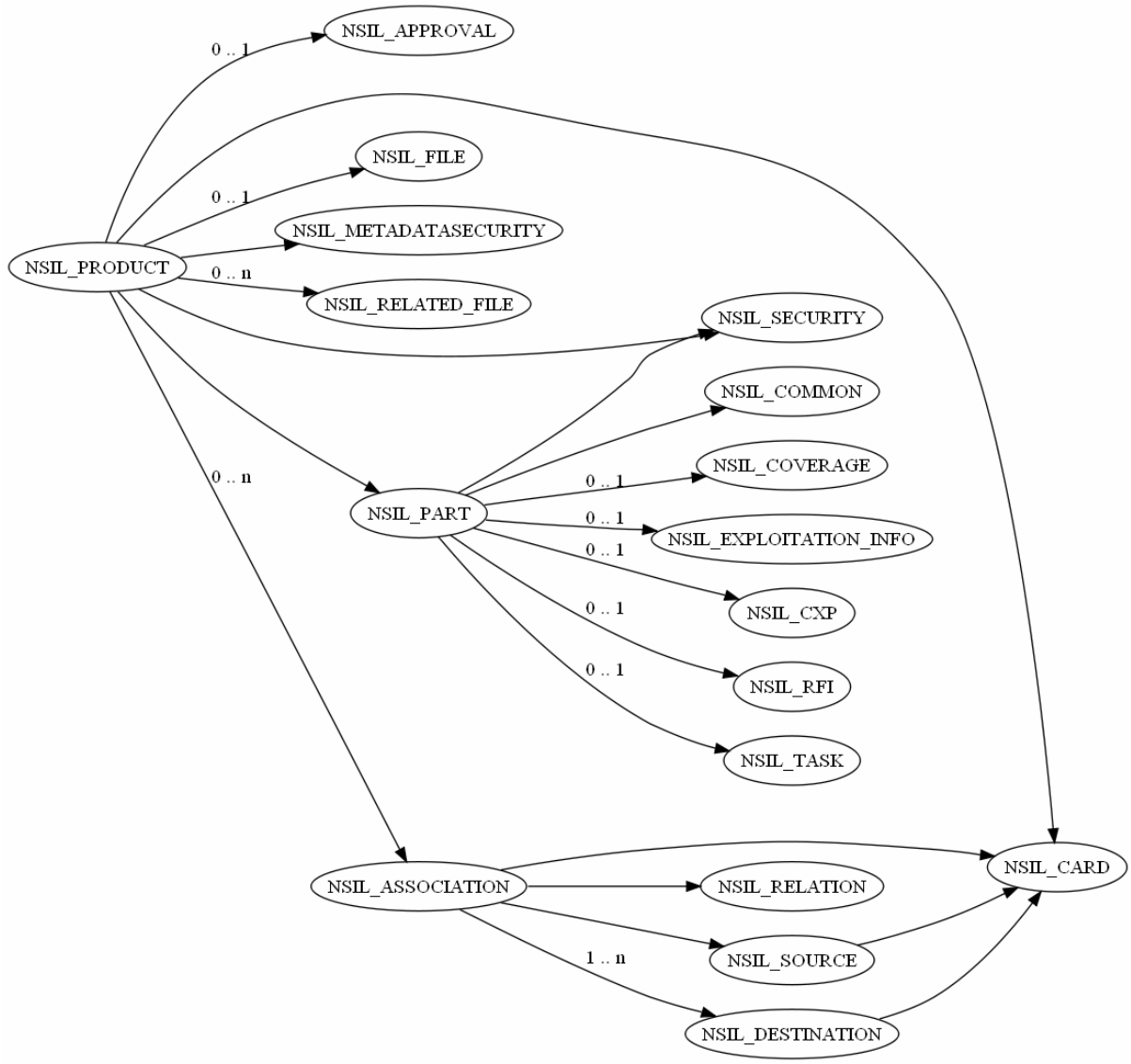

Figure G- 8 NSIL_CCIRM_VIEW

4.2.1.1 Products with the following types of parts are present in 'NSIL_CCIRM_VIEW'.

NSIL_PART:NSIL_COMMON.type = 'COLLECTION/EXPLOITATION PLAN'

NSIL_PART:NSIL_COMMON.type = 'INTELLIGENCE REQUIREMENT'

NSIL_PART:NSIL_COMMON.type = 'RFI'

NSIL_PART:NSIL_COMMON.type = 'TASK'

NSIL_PART:NSIL_COMMON.type = 'GEOGRAPHIC AREA OF INTEREST'

4.2.1.2 This view is orderable through the OrderMgr. The layout of the full entity graph, showing all entity relationships for 'NSIL_CCIRM_VIEW', is shown in the table below. Note that the cardinality marked by grey cell and italic text is different from the cardinality between the same two entities in the NSIL_ALL view.

Table G- 107 NSIL_CCIRM_VIEW

Node list	NSIL_PRODUCT, NSIL_APPROVAL, NSIL_CARD, NSIL_FILE, NSIL_METADATASECURITY, NSIL_RELATED_FILE, NSIL_SECURITY, NSIL_PART, NSIL_COMMON, NSIL_COVERAGE, NSIL_EXPLOITATION_INFO, NSIL_CXP, NSIL_RFI, NSIL_TASK, NSIL_ASSOCIATION, NSIL_RELATION, NSIL_SOURCE, NSIL_DESTINATION		
Start node	**End node**	**Forward cardinality**	**Reverse cardinality**
NSIL_PRODUCT	NSIL_ASSOCIATION	ONE_TO_ZERO_OR_MORE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_APPROVAL	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_CARD	ONE_TO_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_FILE	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_METADATASECURITY	ONE_TO_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_RELATED_FILE	ONE_TO_ZERO_OR_MORE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_SECURITY	ONE_TO_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_PART	*ONE_TO_ONE*	ONE_TO_ONE
NSIL_PART	NSIL_COMMON	ONE_TO_ONE	ONE_TO_ONE
NSIL_PART	NSIL_COVERAGE	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PART	NSIL_SECURITY	ONE_TO_ONE	ONE_TO_ZERO_OR_ONE
NSIL_PART	NSIL_EXPLOITATION_INFO	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PART	NSIL_CXP	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PART	NSIL_RFI	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PART	NSIL_TASK	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_ASSOCIATION	NSIL_CARD	ONE_TO_ONE	ONE_TO_ZERO_OR_MORE
NSIL_ASSOCIATION	NSIL_SOURCE	ONE_TO_ONE	ONE_TO_ZERO_OR_MORE
NSIL_ASSOCIATION	NSIL_DESTINATION	ONE_TO_ONE_OR_MORE	ONE_TO_ZERO_OR_MORE
NSIL_ASSOCIATION	NSIL_RELATION	ONE_TO_ONE	ONE_TO_ONE
NSIL_SOURCE	NSIL_CARD	ONE_TO_ONE	ONE_TO_ZERO_OR_ONE
NSIL_DESTINATION	NSIL_CARD	ONE_TO_ONE	ONE_TO_ZERO_OR_ONE

Table G- 108 Conceptual attributes present in 'NSIL_CCIRM_VIEW'

Concept type	**Attribute**
CLASSIFICATION	NSIL_SECURITY.classification
DATASETTYPE	NSIL_PART:NSIL_COMMON.type
DATASIZE	NSIL_FILE.extent
DIRECTACCESS	NSIL_FILE.productURL
FOOTPRINT	NSIL_PART:NSIL_COVERAGE.spatialGeographicReferenceBox
MODIFICATIONDATE	NSIL_CARD.dateTimeModified
PRODUCTTITLE	NSIL_FILE.title
UNIQUEIDENTIFIER	NSIL_CARD.identifier

5 EXPERIMENTAL LAYER

5.1 EXPERIMENTAL LAYER ENTITIES

5.1.1 NSIL_APPROVAL

The following section lists attributes of the 'NSIL_APPROVAL' entity.

Table G- 109 NSIL_APPROVAL.approvedBy

Description	The name of the organization responsible for Quality Assurance of the resource (product) available in an IPL.
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	approvedBy
Attribute type	TEXT
Domain	TEXT_VALUE
Max field length	30
Data type	CORBA: string
Units	None (empty string)
Sortable	YES
Queryable	YES

Table G- 110 NSIL_APPROVAL.dateTimeModified

Description	The date and time upon which the approval information on the product was last stored or changed in the IPL.
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	dateTimeModified
Attribute type	UCOS_ABS_TIME
Domain	DATE_VALUE (Implementation dependent.)
Data type	UCO: AbsTime
Units	UTC
Sortable	YES
Queryable	YES

Table G- 111 NSIL_APPROVAL.status

Description	A status field reporting on whether this product has been approved by the party responsible for Quality Control of the product.
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	status
Attribute type	TEXT
Domain	List: [APPROVED, REJECTED, NOT APPLICABLE]
Data type	CORBA: string
Units	None (empty string)
Sortable	YES
Queryable	YES

5.1.2 NSIL_EXPLOITATION_INFO

The following section lists attributes of the 'NSIL_EXPLOITATION_INFO' entity.

Table G- 112 NSIL_EXPLOITATION_INFO.description

Description	A textual description of the performed exploitation.
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	description
Attribute type	TEXT
Domain	TEXT_VALUE
Max field length	400
Data type	CORBA: string
Units	None (empty string)
Sortable	NO
Queryable	YES

Table G- 113 NSIL_EXPLOITATION_INFO.level

Description	The degree of exploitation performed on the original data. A value of '0' means that the product is not exploited. The interpretation of other attribute values is currently undefined.
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	level
Attribute type	INTEGER
Domain	INTEGER_RANGE: 0 .. 9
Data type	CORBA: short
Units	None
Sortable	NO
Queryable	YES

Table G- 114 NSIL_EXPLOITATION_INFO.autoGenerated

Description	A flag indicating if the exploitation was automatically generated
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	autoGenerated
Attribute type	BOOLEAN_DATA
Domain	BOOLEAN_VALUE (YES/NO default=NO)
Data type	CORBA: boolean
Units	None
Sortable	NO
Queryable	YES

Table G- 115 NSIL_EXPLOITATION_INFO.subjectiveQualityCode

Description	A code that indicates a subjective rating of the quality of the image or video.
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	subjectiveQualityCode
Attribute type	TEXT
Domain	List: [EXCELLENT, FAIR, GOOD, POOR]
Data type	CORBA: string
Units	None (empty string)
Sortable	NO
Queryable	YES

5.1.3 NSIL_SDS

The following section lists attributes of the 'NSIL_SDS' entity.

Table G- 116 NSIL_SDS.operationalStatus

Description	SDS operational status
Requirement mode	MANDATORY
Updateable	YES
IPL Assigned	NO
Attribute name	operationalStatus
Attribute type	TEXT
Domain	List: [UNKNOWN, LIMITED OPERATIONAL, OPERATIONAL, NOT OPERATIONAL]
Data type	CORBA: string
Units	None (empty string)
Sortable	NO
Queryable	YES

5.1.4 NSIL_TDL

The following section lists attributes of the 'NSIL_TDL' entity.

Table G- 117 NSIL_TDL.activity

Description	A number that together with the 'platform' number defines the identity of a track
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	activity
Attribute type	INTEGER
Domain	INTEGER_RANGE (STANAG 5516): 0 .. 127
Data type	CORBA: short
Units	None
Sortable	NO
Queryable	YES

Table G- 118 NSIL_TDL.messageNumber

Description	The Link 16 J Series message number.	
Requirement mode	MANDATORY	
Updateable	YES	
IPL Assigned	NO	
Attribute name	messageNumber	
Attribute type	TEXT	
Domain	List:[J2.2, J2.5, J3.0, J3.2, J3.3, J3.5, J3.7, J7.0, J7.1, J7.2, J14.0, J14.2]	
Data type	CORBA: string	
Units	None (empty string)	
Sortable	NO	
Queryable	YES	
Aliases	Category: Name:	NACT_L16 /HJ1/MessageNumber

Table G- 119 NSIL_TDL.platform

Description	A number that together with the 'activity' number defines the identity of a track
Requirement mode	OPTIONAL
Updateable	YES
IPL Assigned	NO
Attribute name	platform
Attribute type	INTEGER
Domain	INTEGER_RANGE (STANAG 5516): 0 .. 63
Data type	CORBA: short
Units	None
Sortable	NO
Queryable	YES

Table G- 120 NSIL_TDL.trackNumber

Description	Link 16 J Series track number for the track found in the product. The track number shall be in the decoded 5-character format (e.g. EK627).	
Requirement mode	OPTIONAL	
Updateable	YES	
IPL Assigned	NO	
Attribute name	trackNumber	
Attribute type	TEXT	
Domain	TEXT_VALUE	
Max field length	10	
Data type	CORBA: string	
Units	None (empty string)	
Sortable	NO	
Queryable	YES	
Aliases	Category: Name:	STANAG 5516 TRACK NUMBER REFERENCE

5.2 EXPERIMENTAL LAYER VIEWS

All the defined views shown below, are individually described in the following sections.

- NSIL_TDL_VIEW

NSIL_TDL_VIEW

Notes:

In the case of streaming TDL the NSIL_TDL entity will not be used since this entity describes data particular to a limited message and not for a stream of messages.

In the case of a file-product the NSIL_TDL entity is mandatory.

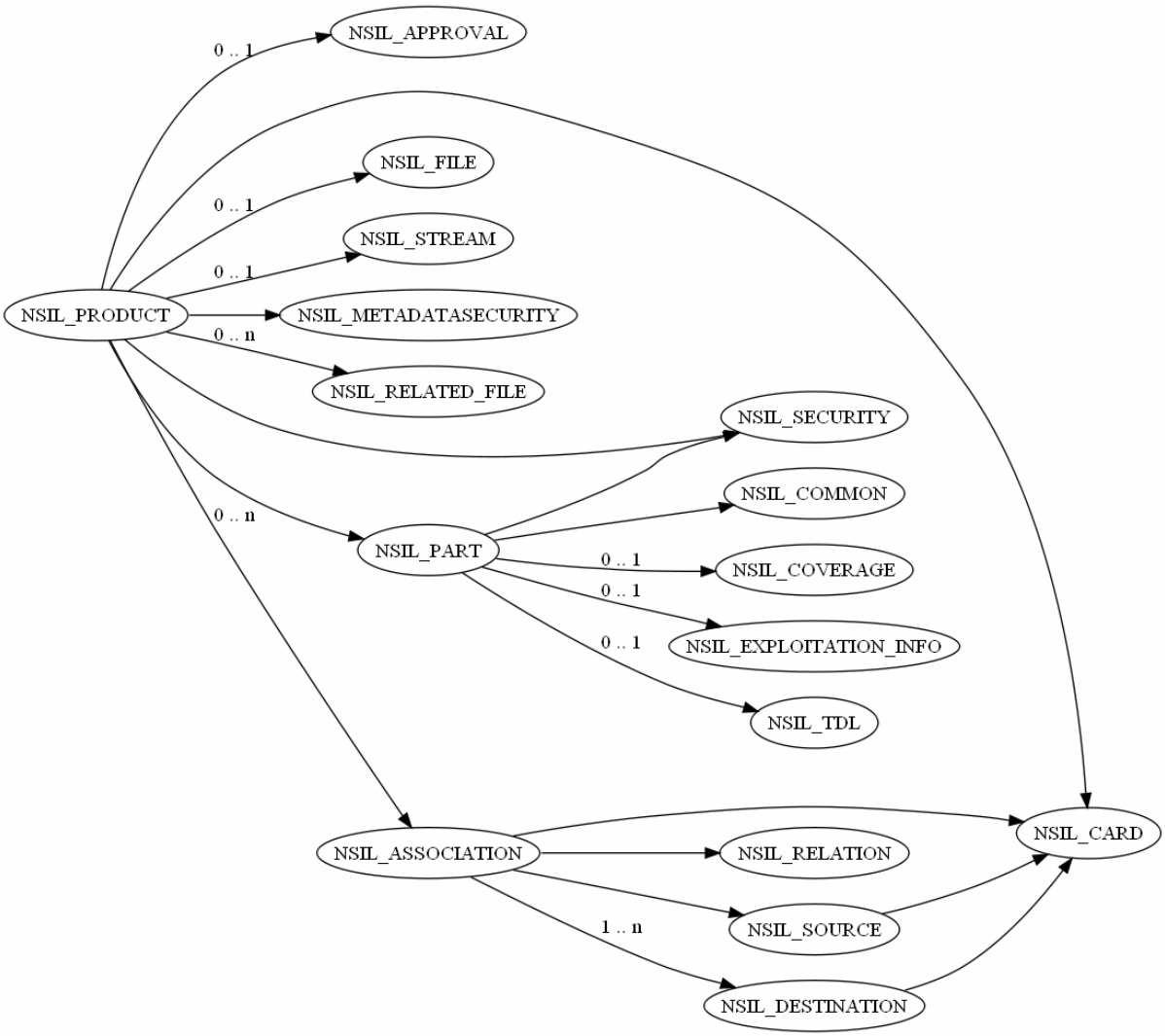

Figure G- 9 NSIL_TDL_VIEW

5.2.1 Products with the following types of parts are present in 'NSIL_TDL_VIEW'.
NSIL_PART:NSIL_COMMON.type = 'TDL DATA'

This view is orderable through the OrderMgr.

5.2.2 The layout of the full entity graph, showing all entity relationships for 'NSIL_TDL_VIEW', is shown in the table below. Note that the cardinality marked by grey cell and italic text is different from the cardinality between the same two entities in the NSIL_ALL view.

Table G- 121 NSIL_TDL_VIEW

Node list	NSIL_PRODUCT, NSIL_APPROVAL, NSIL_CARD, NSIL_FILE, NSIL_STREAM, NSIL_METADATASECURITY, NSIL_RELATED_FILE, NSIL_SECURITY, NSIL_PART, NSIL_COMMON, NSIL_COVERAGE, NSIL_EXPLOITATION_INFO, NSIL_TDL, NSIL_ASSOCIATION, NSIL_RELATION, NSIL_SOURCE, NSIL_DESTINATION		
Start node	**End node**	**Forward cardinality**	**Reverse cardinality**
NSIL_PRODUCT	NSIL_ASSOCIATION	ONE_TO_ZERO_OR_MORE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_APPROVAL	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_CARD	ONE_TO_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_FILE	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_STREAM	*ONE_TO_ZERO_OR_ONE*	ONE_TO_ONE
NSIL_PRODUCT	NSIL_METADATASECURITY	ONE_TO_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_RELATED_FILE	ONE_TO_ZERO_OR_MORE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_SECURITY	ONE_TO_ONE	ONE_TO_ONE
NSIL_PRODUCT	NSIL_PART	*ONE_TO_ONE*	ONE_TO_ONE
NSIL_PART	NSIL_COMMON	ONE_TO_ONE	ONE_TO_ONE
NSIL_PART	NSIL_COVERAGE	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PART	NSIL_SECURITY	ONE_TO_ONE	ONE_TO_ZERO_OR_ONE
NSIL_PART	NSIL_EXPLOITATION_INFO	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_PART	NSIL_TDL	ONE_TO_ZERO_OR_ONE	ONE_TO_ONE
NSIL_ASSOCIATION	NSIL_CARD	ONE_TO_ONE	ONE_TO_ZERO_OR_MORE
NSIL_ASSOCIATION	NSIL_SOURCE	ONE_TO_ONE	ONE_TO_ZERO_OR_MORE
NSIL_ASSOCIATION	NSIL_DESTINATION	ONE_TO_ONE_OR_MORE	ONE_TO_ZERO_OR_MORE
NSIL_ASSOCIATION	NSIL_RELATION	ONE_TO_ONE	ONE_TO_ONE
NSIL_SOURCE	NSIL_CARD	ONE_TO_ONE	ONE_TO_ZERO_OR_ONE
NSIL_DESTINATION	NSIL_CARD	ONE_TO_ONE	ONE_TO_ZERO_OR_ONE

Table G- 122 Conceptual attributes present in 'NSIL_TDL_VIEW'

Concept type	**Attribute**
CLASSIFICATION	NSIL_SECURITY.classification
DATASETTYPE	NSIL_PART:NSIL_COMMON.type
DATASIZE	NSIL_FILE.extent
DIRECTACCESS	NSIL_FILE.productURL
FOOTPRINT	NSIL_PART:NSIL_COVERAGE.spatialGeographicReferenceBox
MODIFICATIONDATE	NSIL_CARD.dateTimeModified
PRODUCTTITLE	NSIL_FILE.title
UNIQUEIDENTIFIER	NSIL_CARD.identifier

6 ASSOCIATIONS

This section identifies the supported associations (discoverable through the DataModel Manager).

Table G- 123 Supported associations

Relationship	View-a	View-b	Cardinality	Attributes	Description
HAS PART	NSIL_ALL_VIEW	NSIL_ALL_VIEW	MANY_TO_MANY	NSIL_CARD.* NSIL_RELATION.*	Described resource (e.g. 'Target Folder') includes the referenced resource either physically or logically.
IS VERSION OF	NSIL_ALL_VIEW	NSIL_ALL_VIEW	MANY_TO_ONE	NSIL_CARD.* NSIL_RELATION.*	Described resource (source) is a version edition or adaptation of the referenced resource (destination). A change in version implies substantive changes in content rather than differences in format.
REPLACES	NSIL_ALL_VIEW	NSIL_ALL_VIEW	ONE_TO_MANY	NSIL_CARD.* NSIL_RELATION.*	Described resource (source) supplants, displaces or supersedes the referenced resource (destination).
IS SUPPORT DATA TO	NSIL_ALL_VIEW	NSIL_ALL_VIEW	ONE_TO_MANY	NSIL_CARD.* NSIL_RELATION.*	Described resource (source) supplements information to the referenced RFI and IR (destination)
ORIGINATING FROM	NSIL_ALL_VIEW	NSIL_ALL_VIEW	MANY_TO_ONE	NSIL_CARD.* NSIL_RELATION.*	Described resource (source) originates from referenced resource (destination) and thereby shows the hierarchical relationship between the two products.
FOLLOWS	NSIL_ALL_VIEW	NSIL_ALL_VIEW	ONE_TO_ONE	NSIL_CARD.* NSIL_RELATION.*	Described resource (source in association) is the next in the chronological sequence after the referenced resource (destination in the association).

7 RELATED FILES

A 4559 Edition 3 server shall support the 'related file' mechanism defined in STANAG 4559 through the 'GetRelatedFilesRequest' even if these are directly accessible through metadata attributes. The table below lists the related file types, their file format and a description of the related file type.

Table G- 124 Related files

Related File Type	File Format	Description
OVERVIEW	1024x1024 NSIF file	A lower resolution overview of an NSIF file. The overview file shall be in the STANAG 4545/NSIF format and have a CCS of at most 1024*1024 pixels. The OVERVIEW file has special meaning for some chipping parameters in the OrderMgr. Refer to the AEDP-5 of STANAG 4559 for details.
THUMBNAIL	128x128 JPEG	A thumbnail of the image. The thumbnail file shall be a 128*128 pixel JPEG file.
THUMBNAIL_FIRST_FRAME	128x128 JPEG	A thumbnail of the first frame in the video sequence. The thumbnail file shall be a 128*128 pixel JPEG file.
THUMBNAIL_LAST_FRAME	128x128 JPEG	A thumbnail of the last frame in the video sequence. The thumbnail file shall be a 128*128 pixel JPEG file.
VIDEO_DESCRIPTOR	STANAG 4609 (MPEG-2 TS/PS with KLV private data stream)	A file containing all the KLV metadata separated out in a separate file. The file format shall be as defined by STANAG 4609 (MPEG2-TS), but with the entire video elementary stream data omitted.
VIDEO_INTRO	STANAG 4609	A file containing a short movie (10 seconds or less) that consists of initial scenes from the product main movie. The client can then use this video intro-file in making the decision of downloading the main video file. The introfile should be in full-quality - frame rate, frame size and encoding as the main video - to allow operator to make a decision on the useabilty of the main video
VIDEO_PREVIEW	STANAG 4609	A file containing a short movie that consists of representative scenes from the product main movie. The client can then use this video preview file in making the decision to download the main video file. The preview could be a version of the video where only a short duration of each scene is being played (or a few seconds played every 10-20 second of the video)
AGGREGATED_CXP	XML	A XML file where all the elements of the CXP have been aggregated. The file will include the IRs, tasks and geographic areas of interest.

8 BACKWARD COMPATIBILITY WITH STANAG 4559 EDITION 1 AND 2

Backward compatibility with STANAG 4559 edition 1 and 2 will be ensured by maintaining the NSIL_CORE view defined by the STANAG. Please note that since NSIL_CORE only supports description of single image NSIF files, only single-image NSIF files will be visible in the NSIL_CORE view. Table G- 125 shows how to map the new model to the STANAG 4559 edition 1 and 2 NSIL_CORE model.

Table G- 125 Mapping metadata to STANAG 4559 edition 1 and 2 NSIL_CORE model

NSIL_CORE	New model	Additional comments
MISNID	NSIL_COMMON.identifierMission	
ISORCE	NSIL_COMMON.source	
TGTID	NSIL_COMMON.targetNumber	
CNTRYCODE	NSIL_COVERAGE.spatialCountryCode	FIPS 10-4/STANAG 1059 mapping required
IGEOLO	NSIL_COVERAGE.spatialGeographicReferenceBox	
IDATIM	NSIL_COVERAGE.temporalStart	
OSTAID	NSIL_FILE.creator	
FDT	NSIL_FILE.dateTimeDeclared	
FL	NSIL_FILE.extent	
DAID	NSIL_FILE.productURL	
FTITLE	NSIL_FILE.title	
ICAT	NSIL_IMAGERY.category	
ICOM	NSIL_IMAGERY.comments	
IID1	NSIL_IMAGERY.identifier	
IID2	NSIL_IMAGERY.title	
OSTAID	NSIL_RELATED_FILE.creator	
FL	NSIL_RELATED_FILE.extent	
PSCLAS	NSIL_SECURITY.classification	Map PSCLAS to NSIL_PRODUCT:NSIL_SECURITY
PSCLSY	NSIL_SECURITY.policy	Map PSCLSY to NSIL_PRODUCT:NSIL_SECURITY. FIPS 10-4/STANAG 1059 mapping required.
PSREL	NSIL_SECURITY.releasability	Map PSREL to NSIL_PRODUCT:NSIL_SECURITY. PSREL separates the entities in the list with a single BCS Space character (code 0x20) while 'releasability' separates the entities with BCS Comma (code 0x2C). FIPS 10-4/ STANAG 1059 mapping required.
OSTAID	NSIL_STREAM.creator	
FDT	NSIL_STREAM.dateTimeDeclared	

9 DATA DEFINITIONS – 4559 Edition 3 CONTENT

This section describes the various types of data that may be retrieved from the NSIL Interface. Some of the data are files formatted according to NATO STANAGs and some data are files formatted as XML files in accordance with XML schemas developed for the MAJIIC project, and some data are fully captured by the 4559 Edition 3 metadata. 4559 Edition 3 contains the data types as listed in the table below.

Table G- 126 Supported data types

NSIL_COMMON.type	MIMEtype (NSIL_FILE.format)	Comment
COLLECTION/EXPLOITATION PLAN	Metadata only	The CXP is an (incomplete) metadata only product where the full CXP is available through the Related Files as an XML file.
DOCUMENT	application/pdf application/msword application/vnd.ms-powerpoint application/vnd.ms-excel text/plain	
GEOGRAPHIC AREA OF INTEREST	text/xml	
GMTI	application/x-cgmti	Files holding GMTIF data shall be binary according to STANAG 4607. A file shall contain all Dwell Segments of one complete revisit. The first segment in each file shall be a Mission Segment, followed by a Job Definition Segment. All GMTIF Segments may be contained in a single GMTIF Packet, which is referred. In case multiple GMTIF Packets are used, there shall be no padding bytes between those packets.
IMAGERY	application/x-nsif	application/x-nitf
INTELLIGENCE REQUIREMENT		Metadata only
MESSAGE	text/xml text/html text/plain	
OPERATIONAL ROLES	text/xml	
ORBAT	text/xml	

REPORT	text/xml	A variety of different report types can be stored in the Edition 3 server. They are all expected to be in XML format. RFI text/xml It is assumed that the RFI is stored as an XML formatted file.
SYSTEM ASSIGNMENTS	text/xml	
SYSTEM SPECIFICATIONS	text/xml	
SYSTEM DEPLOYMENT STATUS	text/xml	
TACTICAL SYMBOL	text/xml	
TASK		Metadata only
TDL DATA	application/x-nact-link16	Link 16 data modified for edtion3 will be made available as binary data files in accordance with STANAG 5516 with the addition of the NACT header. These files will only report one type of information per file e.g. Air PPLI.
VIDEO	video/mp2t video/mpeg	

ANNEX H.

METAMODEL OF EDITION 3 METADATA MODEL

ANNEX H.

METAMODEL OF NSILI DATAMODEL

TABLE OF CONTENTS

PARAGRAPH	TITLE	PAGE
1.	Introduction……..	H-1
2.	Naming Conventions..…….	H-1
3.	Model Rules Compliance……	H-2
4.	Errata……..……	H-2

TABLES	TITLE	PAGE
H- 1	UML Diagram of the Required Information Structure of a Model.....	H-2

ANNEX H - METAMODEL OF NSILI DATAMODEL

1. INTRODUCTION

This Annex provides a metamodel and naming conventions to support the STANAG 4559 Edition 3 Metadata Model.

2. NAMING CONVENTIONS

- Attribute names may only contain alpha-numeric values (a-z, A-Z, 0-9)
- Entity names may only contain alpha-numeric values (a-z, A-Z, 0-9) and the underscore symbol (_)
- In each entity, all attributes must have distinct names (case insensitive).
- In each view, all entities must have distinct names (case insensitive).
- In each model, all views must have distinct names (case insensitive).
- If there are entities with the same name, those entities must have exactly the same content (case insensitive)
- The directed graph formed by the entity graph of each view must be free of recursion (no loop may be formed by the directed graph)

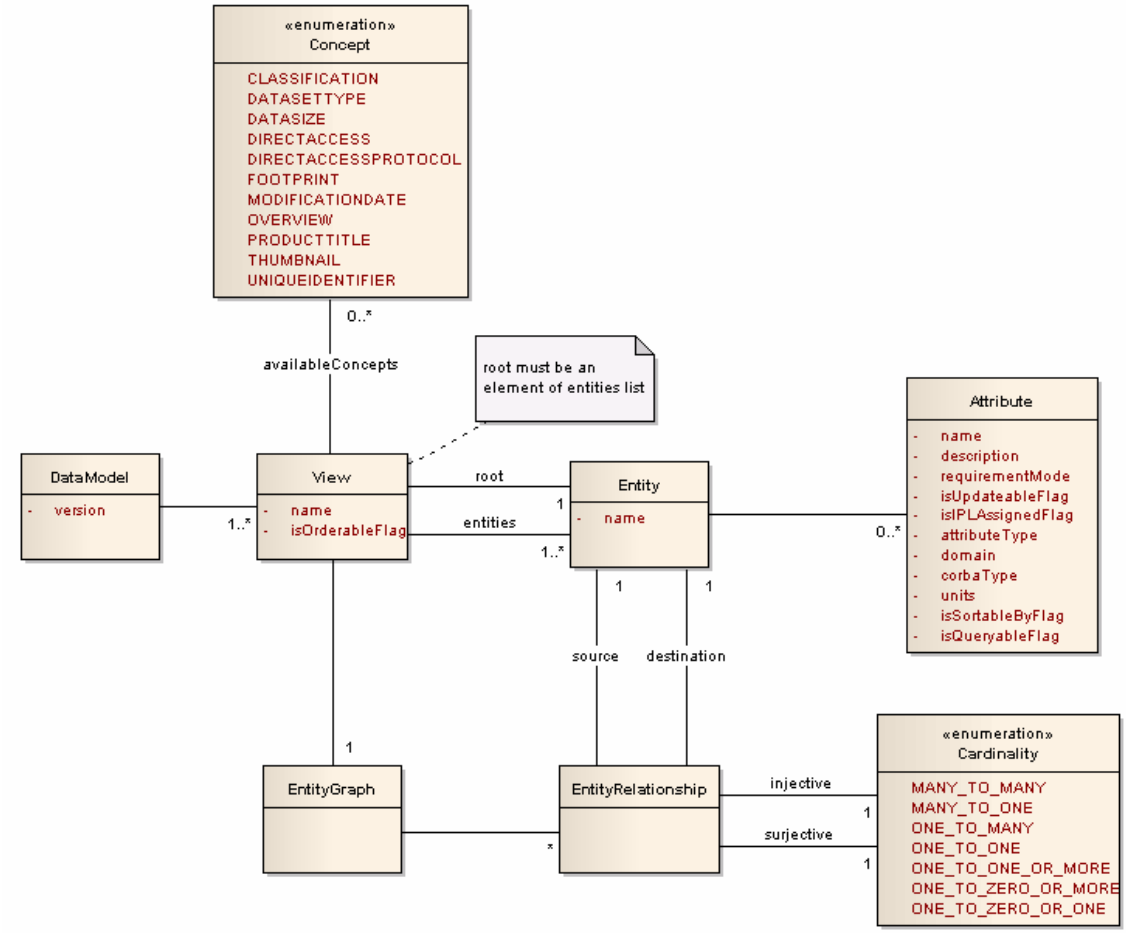

Figure H- 1 UML Diagram of the Required Information Structure of a Model

3. MODEL RULES COMPLIANCE

If the model being used by any software does not fulfil any of the rules above, the software (server/client/service/…) is not expected to be able to handle the model. The software should check those rules when a model is loaded from a file or retrieved from somewhere else, but this is not mandated.

4. ERRATA TO THE METAMODEL

ERRATA: View.root must be contained within the list View.entities
EntityRelationship: from → source to→destination
fromToTo→injective toToFrom→surjective