

Draft

NGMN Copyright 2011

Any publication outside NGMN has to be pre-approved in writing by NGMN



NGCOR CONSOLIDATED REQUIREMENTS

BY NGMN ALLIANCE

DATE: 18-JULY-2011

VERSION 0.92

**(APPROVED
<BY GREMIUM (OC/BOARD)>**

For all Confidential documents (CN, CL, CR):

This document contains information that is confidential and proprietary to NGMN Ltd. The information may not be used, disclosed or reproduced without the prior written authorisation of NGMN Ltd., and those so authorised may only use this information for the purpose consistent with the authorisation.

For Public documents (P):

© 2011 Next Generation Mobile Networks Ltd. All rights reserved. No part of this document may be reproduced or transmitted in any form or by any means without prior written permission from NGMN Ltd.

The information contained in this document represents the current view held by NGMN Ltd. on the issues discussed as of the date of publication. This document is provided "as is" with no warranties whatsoever including any warranty of merchantability, non-infringement, or fitness for any particular purpose. All liability (including liability for infringement of any property rights) relating to the use of information in this document is disclaimed. No license, express or implied, to any intellectual property rights are granted herein. This document is distributed for informational purposes only and is subject to change without notice. Readers should not design products based on this document.



PUBLICATION	<Confidential & Restricted / NGMN CONFIDENTIAL/ NGMN Confidential with Limited Dissemination/ PUBLIC>
PROJECT	NGMN NGCOR
EDITOR IN CHARGE	KLAUS MARTINY
EDITING TEAM	AXEL HECK
DOCUMENT STATUS	<DRAFT VERSION>
APPROVED BY	

Abstract: Consolidate all requirement documentation by the sub tasks GEN; FM; IM; CON and MT

DOCUMENT HISTORY

DATE	VERSION	AUTHOR	CHANGES
13/06/2011	V 0.1	AXEL HECK, DEUTSCHE TELEKOM/SMC	INITIAL DOCUMENT
14/06/2011	V 0.2	AXEL HECK, DEUTSCHE TELEKOM/SMC	STREAM: ADDED TO CONSOLIDATED VERSION AFTER EDITORIAL CHANGES: <ul style="list-style-type: none"> • IM • CON • GEN
14/06/2011	V 0.3	AXEL HECK, DEUTSCHE TELEKOM/SMC	STREAM: ADDED TO CONSOLIDATED VERSION AFTER EDITORIAL CHANGES: <ul style="list-style-type: none"> • FM
15/06/2011	V 0.4	AXEL HECK, DEUTSCHE TELEKOM/SMC	STREAM: ADDED TO CONSOLIDATED VERSION AFTER EDITORIAL CHANGES: <ul style="list-style-type: none"> • MT • ABBREVS • REFS/CITATIONS • → NO FORMAT CHANGES • → INDICATES WHERE TEMPLATE OF SUBTASK IS CORRUPTED
16/07.2011	v 0.8	AXEL HECK, DEUTSCHE TELEKOM/SMC	RECOVERY AFTER DOC CORUPTION (v0.5-v0.8)
17/07/2011	v 0.9	AXEL HECK, DEUTSCHE TELEKOM/SMC	EDITORIAL CHANGES ON IM, UPDATED OVERALL STRUCTURE
18/07/2011	v 0.92	KLAUS MARTINY, DEUTSCHE TELEKOM AXEL HECK, DEUTSCHE TELEKOM/SMC	INTRODUCTIONS



Contents

1	INTRODUCTION TO NGMN NGCOR	10
1.1	Introduction Sub Task Converged Operation	11
1.2	Introduction Modelling and Tooling	12
1.3	Introduction Fault Management.....	12
1.4	Introduction Inventory Management	12
2	HIGH LEVEL REQUIREMENTS For CONVERGED NETWORK OPERATIONS (GEN).....	13
2.1	Rationale for Sub Task “Converged Network Operations”	13
2.2	Scope of Recommendations for Converged Operations.....	15
2.3	Main NGCOR Use Cases.....	16
2.4	Basic Converged Operations Use Cases.....	17
2.4.1	Use Case Architecture “No Convergence” (Today).....	17
2.4.2	Use Case Architecture “Convergence at Network Management Layer”	18
2.4.3	Use Case Architecture “Convergence at Element Management Layer”	18
2.4.4	Use Case Architecture “Convergence at Northbound Interface”	19
2.5	Generic Converged Operations Use Cases (“Combinations”).....	20
2.5.1	C1 - Converged Element Management Layer together with Converged Northbound Interface.....	20
2.5.2	C2 - Converged Network Management Layer together with Converged Northbound Interface.....	21
2.5.3	C3 - Converged Element Management Layer together with Converged Northbound Interface and Converged Network Management Layer	22
2.5.4	Basic Converged Operations Use Cases vs Generic Converged Operations use cases.....	23
2.6	Requirements wrt. Converged Operations	24
2.6.1	Converged Element Management System	24
2.6.2	Harmonized EMS northbound Interfaces	25
2.6.3	Multi-domain Network Management applications	25
2.6.4	Multi-Operator Network Management.....	26
2.6.5	Mono-operator Management via a third party	27
2.6.6	Converged Service Management applications	28
2.6.7	Open architecture for EMS / NMS.....	29
2.6.8	To which players the requirements are addressed	29
2.7	Conclusion on Requirements for Converged Network Operations.....	30
3	GENERIC NEXT GENERATION CONVERGED OPERATIONAL REQUIREMENTS (CON)	32
3.1	Introduction for Generic Converged Operational Requirements.....	32
3.2	Non-Functional Interface Requirements	32
4	REQUIREMENTS FOR MODELLUNG AND TOOLING (MT).....	39
4.1	Background for Modelling and Tooling	39
4.2	Objectives	39
4.3	Definitions	39
4.3.1	Federated Model	39
4.3.2	Interface	40
4.4	Project Scope, Deliverables and Methodology	41
4.4.1	Requirements Definition.....	42
4.5	Requirements Definition.....	43
4.5.1	Modelling Requirements	44
4.5.2	Tooling Requirements	67
4.6	Appendix For Modelling and Tooling	72
4.7	References used in Modelling and Tooling	73
5	REQUIREMENT SPECIFICATION FOR FAULT MANAGEMENT INTERFACE	74
5.1	Introduction for Fault Management	74
5.1.1	Objective	74



5.1.2	Approach.....	75
5.1.3	Benefit and Drivers.....	75
5.1.4	Scope.....	75
5.2	Non-Functional Requirements.....	76
5.3	Functional Requirements for Fault Management Interface.....	77
5.3.1	X.733 Event/Alarm Attributes.....	77
5.3.2	Event/Alarm Transport.....	78
5.3.3	Clear – Event/Alarm Transport.....	78
5.3.4	Unambiguous Notification ID.....	79
5.3.5	Event/Alarm Query.....	79
5.3.6	Heartbeat.....	80
5.4	EMS Specific Functional Requirements for Interface Support.....	80
5.4.1	Reliable Event/Alarm Communication (supported by EMS).....	80
5.4.2	Configurable EMS Heartbeat Message.....	80
5.4.3	Alarm Suppression.....	80
5.4.4	Summary Alarms.....	81
5.5	NMS Specific Functional Requirements for Interface Support.....	81
5.5.1	Re-Synchronization.....	81
6	HIGH LEVEL OSS REQUIREMENTS FOR INVENTORY MANAGEMENT.....	82
6.1	Introduction and Scope of Inventory Management Sub Task.....	82
6.2	Forming a Common View on Inventory Management.....	83
6.2.1	Resource Inventory Management.....	85
6.2.2	Service Inventory Management.....	87
6.2.3	Product Inventory Management.....	89
6.2.4	Definition of the Scope of Work of Inventory Management Sub Task and Limitations.....	90
6.3	High Level Inventory Management Requirements.....	90
6.3.1	Functional Requirements.....	90
6.3.2	Information / Operations Model Requirements.....	91
6.3.3	Interfacing / Integrations Requirements.....	92
6.4	Appendix Inventory Management.....	95
6.4.1	Source Material, Scope of Analysis and Context.....	95
6.4.2	TMF.....	103
6.4.3	3GPP.....	125
6.4.4	ITIL.....	129
7	ALL References.....	131

Tables

Figure 1: Scope of NGCOR within the eTOM framework.....	16
Figure 2: Use case architecture „No convergence”.....	17
Figure 3: Use case architecture „Converged Network Management Layer”.....	18
Figure 4: Use case architecture „Converged Element Management Layer”.....	19
Figure 5: Use case architecture „Converged Northbound Interface”.....	20
Figure 6: Combination “Converged Northbound Interface and Converged EMS”.....	21
Figure 7: Combination “Converged Northbound Interface and Converged NMS”.....	22
Figure 8: Combination “Converged Northbound Interface and Converged EMS & NMS”.....	23
Figure 9: Single EMS for multiple affiliates.....	24
Figure 10: Single NMS for multiple affiliates.....	26
Figure 11: RAN Sharing with EMS shared amongst operators.....	27
Figure 12: Operations architecture within outsourcing.....	28
Figure 13: Requirements for recommended API / interface.....	32



Figure 14: Managed objects in the context of service model and inventory	37
Figure 15: Federated model	40
Figure 16: Converged Interface peers	41
Figure 17: Interface Harmonisation Levels	45
Figure 18: Operator's Harmonized OSS, End-to-End Network Multi-Domain, Multi-Technology Management View	46
Figure 19: Example Mobile and Fixed Network – detailed layered view	47
Figure 20: Event / Inventory relation	48
Figure 21: Example OSS receives the alarms from different EMS and different models.....	49
Figure 22: Example OSS receives the alarms of one EMS and a one data model FNM – model.....	50
Figure 23: Model Artefacts.....	52
Figure 24: Meta Model	53
Figure 25: Meta Model: Object Class.....	55
Figure 26: Meta Model: Service Interface.....	56
Figure 27: Meta Model: Operation	58
Figure 28: Meta Model: Operation Parameter.....	59
Figure 29: Meta Model: Notification.....	60
Figure 30: Meta Model: Notification Parameter.....	60
Figure 31: Meta Model: Data Type.....	62
Figure 32: Meta Model: Association.....	63
Figure 33: Meta Model: Association End.....	65
Figure 34: Modelling/Tooling Architecture	68
Figure 35: Modelling differences between 3GPP and TM Forum.....	72
Figure 1: Key scope of IM sub task in eTOM framework.....	84
Figure 2: Resource inventory as part of OSS architecture	87
Figure 3: Service inventory as part of OSS architecture.....	89
Figure 4: Product management domain from TAM 4.5	105
Figure 5: Service management domain from TAM 4.5.....	107
Figure 6: Resource management domain from TAM 4.5.....	112
Figure 7: Information framework domains & level1 ABEs	118
Figure 8: Instance and specification relationships for resources, services and products.....	118

Files

File 1: Product, service and resource inventory retrieval for cloud and IT.....	125
File 2: Product, Service and resource inventory update for cloud and IT.....	125



Abbreviations

Abbreviation	Meaning & Terms
2G/3G/LTE	standards for mobile communication network and devices capabilities
3GPP	(SDO)
AAA	authentication and authorization
ABE	aggregate business entity?
ABR	asynchronous batch response
ADSL	asynchronous digital subscriber line
AFB	asynchronous (file) bulk
AFI	autonomic future internet
AKA	also known as
AN	asynchronous notification
API	application programming interface
ARPU	average revenue per user
ARR	asynchronous request/reply
ASCII	American Standard Code of Information Interchange
ASP	application service provider
ATM	asynchronous transfer mode
B2B	business-to-business
BA	Business Agreement (TM Forum)
BBF	Broadband Forum (SDO)
BER	bit error ratio
BSS	business support system
CAPEX	capital expenditures (costs to set up / change a network)
CBE	core business entity
CDR	call details records
CFS	customer Facing Service
CFSS	customer Facing Service Specification
CI	configuration item
close loop	autonomous operated SON function
CM	configuration management
CMDB	configuration management data bases
CMDB	configuration management data bases
CMS	configuration management system
CN	core net
CORBA	common object request broker architecture
COTS	commercial of the shelf
COTS	connection oriented transport services OR commercial of the shelf
CRUD	create, read, update, delete
csv	comma seperated value
CTK	compliance test kit
DSLAM	digital subscriber line access multiplexer
DT	Deutsche Telekom (Operator)
e2e	end-to-end
EM	element management OR manager (--> Emgr)??
EMS	element management system
EPC	evolved packet core ???
eTOM	enhanced telecommunication operations map, TMF



ETSI	European Telecommunications Standards Institute (SDO)
FAB	fulfillment, assurance and billing
FCAP	fault, configuration, assurance, performance
FDD	feature description document
FIM	federated information model
FM	fault management
FMC	fixed-mobile convergence
FMC	fixed-mobile convergence
FMH	Fixed-mobile harmonization
FOM	federated operations model
FRU	field replaceable unit
FT	France Telecom (operator)
GEN	generic next generation operational requirements
HLR	home location register
HO	handover
HTTP	hyper text transfer protocol
HW	hardware
IA	Information Agreement (TM Forum)
IDL	interface definition language
IDL	interface definition language
IETF	Internet engineering task force (SDO)
IETF	Internet Engineering Task Force (SDO)
IIS	interface implementation specification (TM Forum)
IM	information management
IM	inventory management
InvM	inventory management
InvM	inventory management
IP	internet protocol
IPR	intellectual property rights
IRP	integration reference point (3GPP term)
ISG	Industry Specification Group
itf-N	northbound interface
ITIL	information technology infrastructure language
ITU-T	International Telecommunications Union - Telecommunication Standardization Sector (SDO)
JVT	Java Value Types ??? Java Vision Toolkit???
LCC	lower camel case
LTE	long term evolution
MEF	Metro Ethernet Forum (SDO)
MEP	Message exchange pattern
MO	managed object
MPLS	multi protocol label switching
MRI	manage resource interface
MRI	manage resource inventory
MSI	manage service inventory
MSI	manage service inventory
MT	modelling and tooling
MTNM	multi-technology network management
MTOSI	multi technology OS interface



MVNE	mobile virtual network environment
MVNO	mobile virtual network operator
MW	TM Forum Management World
NE	network element
NEM	network element manager
NGCOR	Next Generation Converged Operations Requirements
NGMN	Next Generation Mobile Network
NGN	Next Generation Network
NGOSS	Next Generation Operation Support System
NM	network management OR manager (--> NMgr)??
NMS	network management system
NOC	network operation centre
NRM	network resource management ODER manager ?????
NRM	network resource model (3GPP)
NT	network technology
O&M	operation & maintenance
OA&M	operation, administration & maintenance
OC	Operating Committee (NGMN)
OMC	radio and core network infrastructure network element manager
OOTB	out-of-the-box
OPE	operational efficiency
OPEX	operational expenditures
OS	OSS oder Operating system (UNIX ,...) ?
OS&R	operation, support and readiness (TMF term?)
OSS	operating support system
PCC	policy charging and control
PDF	portable document format
PM	performance management
PT	Portugal Telecom (operator)
QoS	quality of service
RAM	resource alarm management
RAN	radio access network
RFS	resource facing service
RFSS	resource facing service specification
RI	reference implementation
RM	ressource management
RM&O	resource management and operations
SACM	service asset and configuration management
SDH	synchronous digital hierarchy
SDO	standards developing organisation
SFB	synchronous (file) bulk
SI&P	strategy, (see func architcture NGSSM) ???
SI&P ??	supporting, planning and implementation
SID	Shared Information and Data Model (TMF)
SIT	synchronous iterator
SLA	service level agreement
SLA	service level agreement
SM	service management
SM	security management



SN	synchronous notification
SOA	service oriented architecture
SOAP	simple object access protocol
SON	self organizing network
SONET	synchronous optical network
SP	service provider
SQM	service quality management
SRR	synchronous request/reply
SuM	subscription management ???
SW	software
TA	tracking area
TAM	Telecom Applications Map (TMF term)
TMF	Tele Management Forum (SDO)
TWG SC	?
UDC	user data convergence
UDC	universal data connector?????
UMC	upper camel case
UML	unified modelling language
UMTS	universal mobile telecommunication system
VF	Vodafone (operator)
WDM	wavelength division multiplexing
WDM	wavelength division multiplexing
WiMax	worldwide interoperability for microwave access
WLAN	wireless local area network
WS	web service
WSDL	web service description language
XML	extended markup language
XSD	XML scheme



1 INTRODUCTION TO NGMN NGCOR

The Next Generation Converged Operations Requirement (NGCOR) project is approved by the board of NGMN. The project is a continuation of the projects SON and NGMN Top OPE Recommendation from 2010. SON focused on radio capabilities of a mobile network, OPR specified operations requirement for mobile networks. The result of both activities are considered in the NGCOR project since the results are valid and essential for the management of an mobile network.

NGCOR is an enhancement of OPE, because NGCOR details specifications of operations requirements for both wireline and wireless networks. It is obviously that both networks will be merged in the near future. NGCOR is describing requirements for converged operations. It is not the intention to specify the conversions of wireline and wireless networks.

Converged operation is one of the key issues for each operator and service provider, because the services will be delivered via a network, which does not distinguish between the capabilities of the network themselves. The current situation is caused by the fact that O&M capabilities for wireline and wireless network elements are implemented by various standards if they there standardized at all. The impacts are high operational cost and slow time to market. The expected results from a common standardization are reduced OPEX and CAPEX and significantly shortened time to market. Without a higher grade of standardization the optimization of commercial figures isn't possible.

There is a need for the definition of converged O&M requirements to ensure that the operational activities within the converged networks perform optimally.

The project has the claim to give guidance to SDOs and industry bodies (e.g. 3GPP or TM Forum) in order to prioritize the work. Develop the solutions for most important requirements first and specify the recommendations for the best solutions.

“An increasing number of service providers (SP) has to operate a variety of network and service production infrastructures, from mobile and fixed network environments up to converged networks and services across many countries. The increasing demand to maintain and improve customer experience requires full end-to-end service management and hence, multi-technology and multi-vendor network management capabilities. On the other hand, financial downturn has put even more pressure on operational efficiency improvement.”

Source: Deutsche Telekom (DT), France Telecom (FT), Vodafone (VF), BT, Portugal Telecom (PT).

The key assumptions are:

- Existing standards shall be considered.
- Operations requirements has to be specified for
 - Fault Management
 - Inventory Management
 - Configuration Management
 - Performance Management

NGCOR will be focused in the first phase on

- Converged operations requirements in general based on uses cases
- Fault Management
- Modelling and Tooling of the north bound Interfaces
- Inventory Management
- Will be focused on definition of operations requirements for EMS north bound interfaces



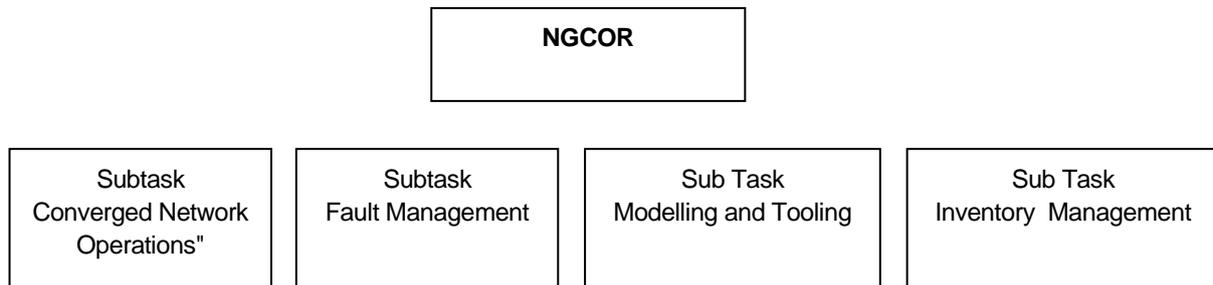
Project lead Co Lead	Klaus Martiny, DT Wille Siebert, VF	
Sub task	Sub task leader / main author	Contributors & Review (Operator)
GEN	Andreas Buschmann; VF	Vodafone O2, Telecom Italia, China Mobile, KPN/ E-PLUS, Bell Mobility, Telefonica France Telecom Orange Deutsche Telekom Vodafone O2 TeliaSonera
CON	Tayeb Benmeriem, FT	
MT	Bernd Zeuner, DT	
FM	Andreas Buschmann, VF	
IM	Pekka Olli TeliaSonera	

Table 1_ NGCOR sub task, main authors and contributors

Out of scope are

- Customer Care and Billing
- Internal processes and organization of the operators & service providers.

The structure of the project is depicted in the following figure:

**Note:**

Please consider that each of the sub tasks have a different timeline. The result is the maturity of the chapters is on various levels. We as a project team convinced that each of the chapters is mature enough to distribute the entire document in order to stimulate the discussion and to get feedback from the Partners of NGMN.

The documents will be updated continuously based on the feedback which we got because not all of the comments could be consider due by the time. The appendix of the charters will be updated as well. Of course partner feedback will discussed and consider as well.

1.1 Introduction Sub Task Converged Operation

Converged operations are facing a strong need for harmonised and standardized EMS northbound interfaces, i.e. the interface between the operators' OSS / NMSs and the vendor's EMSs. This includes the management capabilities (or IT / OSS applications), information / data models and protocols as described by the NGCOR "Federated Model" in sub task "Modelling & Tooling" (see MT section). The



expect benefits of the converged operations are proportional to the level of compliancy of vendors' solutions with respect to standards, with a direct impact on the integration cost of these northbound interfaces.

Another type of operations costs are due to that some EMSs are still not today multi-technology / multi-domain. Indeed, an EMS is also a cost element and operations costs can be reduced thanks to multi-technology / domain capabilities.

Similarly, costs savings can be achieved by operators through extending their NMSs scope to multiple technologies / domains, e.g. a single fault management NMS for 2G/3G/LTE (i.e. multiple technologies) and EPC networks (i.e. multiple domains).

1.2 Introduction Modelling and Tooling

The project has a strong believe that a clear description on modelling and tolling capabilities is needed, in order to give guidance to the SDOs and the industry. The sub task Modelling and Tooling will describe on a detail level the capabilities from an operations point of view. The purpose of the subtask is to give much more clarity as in the other sub task about the operators expectations, because the implementation of possible solution will be impacting the cost structure dramatically.

1.3 Introduction Fault Management

Today's fault management interfaces between element management systems (EMS) and network management systems (NMS) are based on a large variety of different technologies and standards. Each EMS which has been delivered to service providers (SP) in the past uses his own specific interface type and implements element-specific extensions and behaviour, which evolve over time, leading to a continuous need for upgrades on EMS side and to related adaptations/upgrades on NMS side. SPs estimate of one major upgrade project per EMS per two to three years. The cost and effort for the EMS upgrades are often covered by the budgets for the related network element upgrades. But there are additional costs and effort for the related upgrade adapters/access-modules in the NMS-FM system, although the main requirements on such an interface are almost the same for the last ~ 15 years. So SPs are driven by vendors to start interface upgrade projects, perform complex and time consuming type acceptance to ensure the needed quality, train administrators and project managers, etc. ... to get at least no additional value.

1.4 Introduction Inventory Management

The inventory sub task of NGCOR places the inventory management in the focal point of view as it is understood that inventories are the key and core parts of OSS architecture of operators. The main role of inventories is to provide comprehensive and reliable data supporting efficiently different operational, planning and deployment processes when managing the infrastructure and the services.



2 HIGH LEVEL REQUIREMENTS FOR CONVERGED NETWORK OPERATIONS (GEN)

2.1 Introduction for Sub Task “Converged Network Operations”

The CON section aims at capturing high level requirements for converged operations. The methodology chosen is to:

1. Describe basic converged operations use cases highlighting the problems behind the concept of converged operations and the challenges to be overcome
2. Describe three generic converged operations use cases as relevant combinations of the basic converged operations use cases from which requirements can be derived and which can be used as a framework to pave the way towards the converged operations
3. Select real use cases of high interest to operators and map them with the relevant generic converged operations use case
4. Derive high-level requirements from the generic converged operations use cases and detailed requirements from the real use cases
5. Identify the expected benefits in terms of CAPEX / OPEX reduction.

To whom these requirements are addressed from cost saving perspective

In the CON section we are focusing on the three following cost elements and thereby, three key players are targeted:

- Northbound management interfaces level: related requirements are addressed to SDOs
- Operators' OSS-NMS (Operations Support System - Network Management System) level: related requirements are addressed to OSS vendors and integrators
- Vendors' EMS (Element Management System) level: related requirements are addressed to the telecom equipment vendors and network equipment providers.

An additional cost saving lever is to make converged operations architectures autonomic-aware, hence agile and flexible. This can be achieved by introducing autonomics and self-management concepts and design principles in operators' OA&M / OSS solutions. It's an ongoing work within the ETSI Industry Specification Group (ISG) for Autonomic Future Internet (AFI).

The timeframe of this sub task

This subtask is 2-phase structured.

Scope of Phase 1: the use cases covered in phase 1 constitutes the current CON section. It is composed of the three following categories of use cases:

- Basic Converged Operations Use Cases
 - Converged network management layer
 - Converged element management layer
 - Converged northbound interface
- Generic Converged Operations Use Cases
 - Converged element management layer together with converged northbound interface
 - Converged network management layer together with converged northbound interface
 - Converged element management layer & network management layer together with converged northbound interface



- Real Use Cases in a single operator environment and in a multi-operator environment
 - Mobile network operator with multi-technology access networks (2G, 3G, 4G)
 - Operator under operations outsourcing agreement
 - Various mobile operators sharing their RAN (under peering agreement, regulation constraint)
 - Various affiliates sharing an EMS (Element Management System)
 - Various affiliates sharing an NMS (Network Management System)

Scope of Phase 2: the use case covered in phase 2 constitutes the content of the next release of this section

This phase is composed of three steps:

- New real use cases of interest for operators
- Use cases within fixed – mobile network convergence (FMC) or harmonization (FMH) most of them are imported from joint 3GPP-BBF (FMC) activity
- How to make NGCOR converged operations use cases autonomic-aware.

Step 1: New real use cases in a single operator environment and in a multi-operator environment

In this step, we will address only new real use cases based on the generic converged operations use cases as pre-requisite according the description performed in phase 1. This list below presents an overview of potential use cases. A subset of them relevant from operators' business perspective will be subject to converged operations studies in this phase and related requirements will be derived accordingly.

- Mobile network operator with a mono-technology (e.g. 3G) vertically integrated (Home, Access, Backhaul, Core) network
- Mobile network operator: RAN, Backhaul, Core
- Mobile network operator with MVNO/MVNE agreements
- Operator managing a partitioned network (Home, Access, Backhaul, Core)
- Fixed-mobile operator with control plane interworking (IMS)
- Mobile 3GPP network-non 3GPP mobile network (Trusted or Untrusted)
- Mobile 3GPP network-non 3GPP fixed network (Trusted or Untrusted)
- Mobile network operator within offload models

- Various fixed/mobile operators with interworking at the service level
- Various fixed/mobile operators with control plane interworking (IMS)
- Various access network operators (mobile, fixed) with common EPC network
- Various access network operators (mobile, fixed) with common EPC network + common databases)
- Various affiliates sharing an EMS (Element Management System) with common core Network + common HSS/HLR
- Various affiliates sharing an NMS (Network Management System) with common core Network + common HSS/HLR
- Various mobile 3GPP network-Non 3GPP mobile network (trusted or untrusted)
- Various mobile 3GPP network-Non 3GPP fixed network (trusted or untrusted)

Step 2: Use Cases within fixed – mobile network convergence (FMC) from joint 3GPP-BBF activity

Other use cases from "converged network" so-called FMC (Fixed Mobile Convergence) or FMH (Fixed-Mobile Harmonization) including cloud & virtualization of these FMC and FMH will be addressed in this section from operations requirements perspective.

Indeed, the willingness of the operators to move towards the FMC and FMH is not new. Back to end of 90s and early 2000, some operators did intensive work aiming at harmonizing and converging some functions in both 2G/3G architecture and ADSL architecture mainly GGSN in one hand and BRAS in the



other hand or P-GW and BNG now. But we can go further in the FMH and FMC concept as 3,5 and 4G are all IP based architectures therefore, other transport functions and "network service functions" such as Policy Charging & Control (PCC/PCRF), Authentication & Authorization (AAA), mobility, user profile now within the concept of UDC (User Data Convergence) can be addressed from FMC/FMH perspective and their operations requirements accordingly.

The good news is that operators' early work on FMC and FMH is now shifting to the standardization community. In this context, use cases coming from the joint 3GPP-BBF (FMC) work will be taken into account as input and will be analyzed in order to design and describe related operations architectures in the framework of converged operations of NGCOR (Federated Information / Data Model & Tooling, converged northbound interface, and management functions: FM, PM, CM, IM) but connection to BSS (provisioning, billing) part can be done as well.

Step 3: How to make NGCOR Converged Operations use cases Autonomic-Aware

An additional cost saving lever is to make converged operations architectures autonomic-aware, hence agile and flexible. This can be achieved by introducing autonomics and self-management concepts and design principles in operators' OA&M / OSS solutions. It's an ongoing work within a liaison to be established with the ETSI Industry Specification Group (ISG) for Autonomic Future Internet (AFI).

2.2 Scope of Recommendations for Converged Operations

Referring to the **eTOM Business Process Framework**, both use cases and requirements identified in this section focus on the process area named "Operations", which covers the core of operational management. Within the operations process area, the recommendations made in the current section (Figure 1) focus on the following functional process groupings:

- Horizontally:
 - Resource management & operations
 - Service management & operations
- Vertically:
 - Fulfilment
 - Assurance

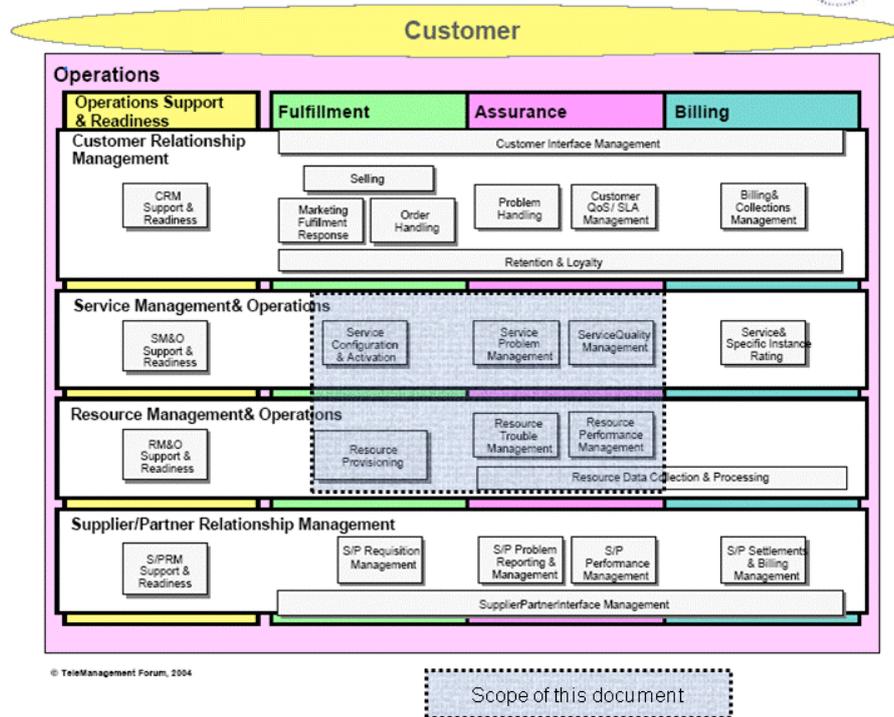


Figure 1: Scope of NGCOR within the eTOM framework

2.3 Main NGCOR Use Cases

This chapter gathers main operators' use cases as illustrations of real scenarios from operators' perspective, from which requirements can be derived. This list is composed of two categories of use cases, the first one involving a single operator facing convergence challenges, the second one involving various operators within convergence policy or agreement:

Use cases covered in phase 1: current section

- Basic Converged Operations Use Cases
 - Converged network management layer
 - Converged element management layer
 - Converged northbound interface
- Generic Converged Operations Use Cases
 - Converged element management layer together with converged northbound interface
 - Converged network management layer together with converged northbound interface
 - Converged element management layer & network management layer together with converged northbound interface
- Real Use Cases in a single operator environment and in a multi-operator environment
 - Mobile network operator with multi-technology access networks (2G, 3G, 4G)
 - Operator under operations outsourcing agreement
 - Various mobile operators sharing their RAN (under peering agreement, regulation constraint)

- Various affiliates sharing an EMS (Element Management System)
- Various affiliates sharing an NMS (Network Management System)

2.4 Basic Converged Operations Use Cases

This chapter firstly describes "basic converged operations use cases" as building blocks. For each of them, we identify the level of the operations convergence. Secondly, in chapter 2.5, we propose three possible combinations of these "basic converged use cases" within an operator's environment, leading to three generic converged operations use cases.

2.4.1 Use Case Architecture "No Convergence" (Today)

Use Case Architecture description

This use case architecture is considered as the bottom line, i.e. it reflects the case where convergence exists neither at the element management layer, nor at the northbound interface nor at the network management layer, as illustrated in Figure 2. It is characterized by:

1. Element management systems are dedicated to specific network domains, e.g. LTE EMS is different from 3G EMS, EPC core network EMS is different from IP backhaul EMS.
2. EMS northbound Interfaces are specific to network technologies. Typically, they can be based on 3GPP IRPs for mobile network domain EMSs and on TMF interface programs for wireline domain EMSs.
3. Operator's OSS applications are dedicated to network technologies and OA&M functional domains. For example, for legacy reasons, it may happen that operator has got one OSS application for fault management of their 2G network, another OSS application for fault management of their 3G network and yet another OSS application for fault management of their IP backhaul network.

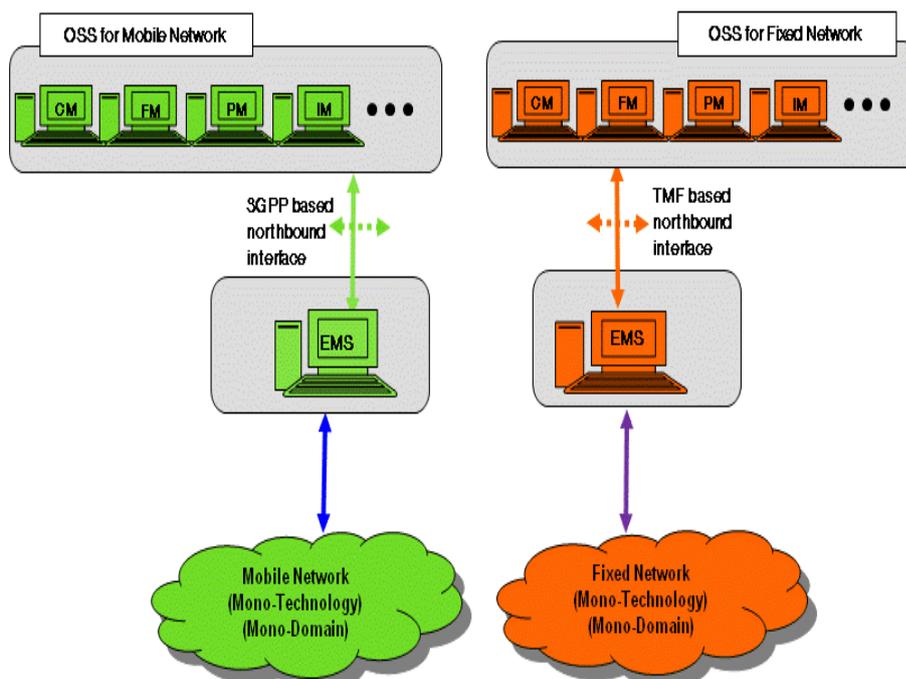


Figure 2: Use case architecture „No convergence”

2.4.2 Use Case Architecture “Convergence at Network Management Layer”

Use Case Architecture description

1. Element Management Systems are dedicated to network domains, e.g. LTE EMS is different from 3G EMS.
2. EMS northbound Interfaces are specific to a given network technology. Typically, they are based on 3GPP IRPs for mobile network domain EMSs or on TMF interface programs for wire line domain EMSs.
3. Operator has got one single OSS application for multiple network domains, for a specific OA&M functional domain, e.g.:
 - a. One single OSS application for fault management, covering all network domains / technologies;
 - b. One single OSS application for performance management covering all network domains / technologies;
 - c. Etc.

Figure 3 depicts this architecture.

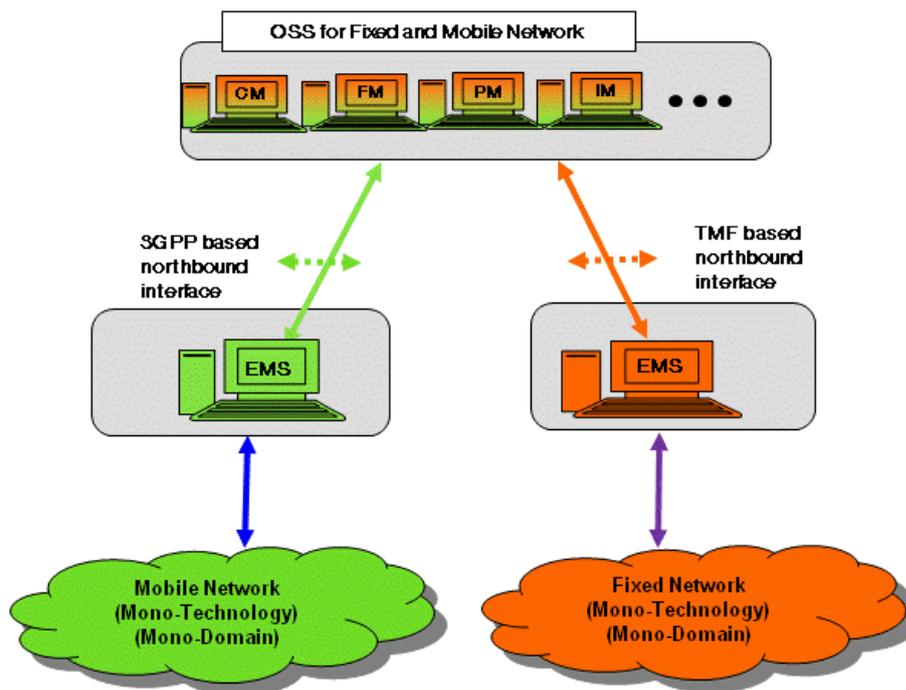


Figure 3: Use case architecture „Converged Network Management Layer”

2.4.3 Use Case Architecture “Convergence at Element Management Layer”

Use Case Architecture description

1. Vendors offer one single element management system common to multiple network domains / technologies, e.g. vendor X EMS is the same for 2G / 3G / LTE / IMS / etc. (though software / hardware upgrades may be needed to cope with cross-domain requirements).

2. Vendors' EMSs support various kinds of northbound interfaces, e.g. one set for mobile networks (based on 3GPP IRPs), another set for wire line networks (based on TMF Interface Programs).
3. Operator's OSS applications are dedicated to specific network domains / technologies and OA&M functional domains.

Figure 4 depicts this architecture.

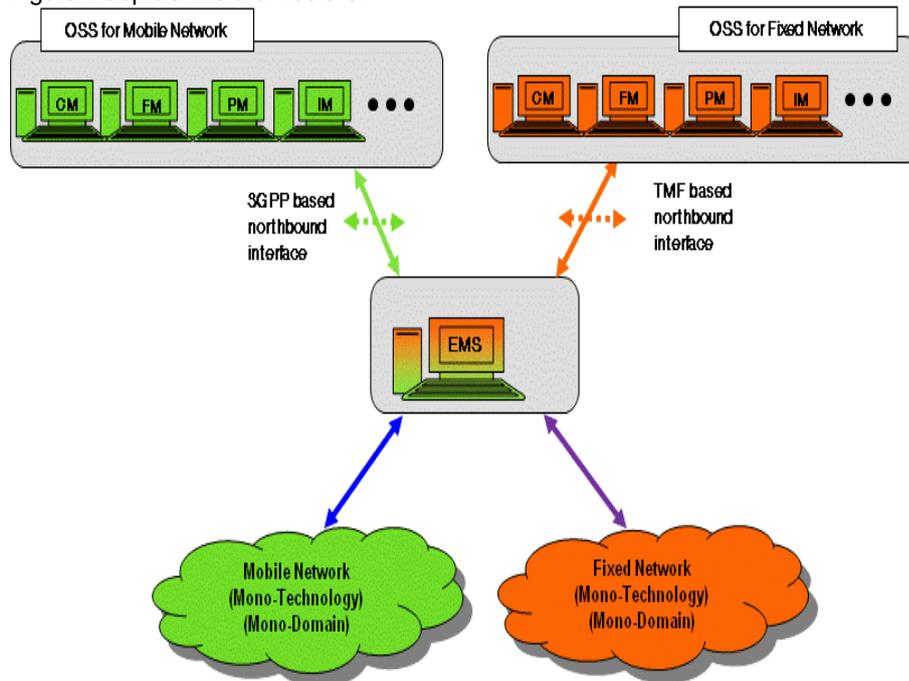


Figure 4: Use case architecture „Converged Element Management Layer“

2.4.4 Use Case Architecture “Convergence at Northbound Interface”

Use Case Architecture description

1. Vendors offer either multiple element management systems per network domain / technology or one single element management system common to multiple network domains / technologies
2. Vendors' EMS(s) support one single northbound interface:
 - a. Based on a federated network information model, for both wireless and wire line network domains
 - b. Based on an harmonized functional interface per functional area, e.g. one single harmonized functional interface for fault management, for both wireless and wire line network domains, one other single harmonized functional interface for configuration management, etc.;
3. Operator has got either multiple OSS applications for specific network domains / technologies or one single OSS application for all network domains / technologies, for a specific OA&M functional domain, e.g.:
 - a. Fault management
 - b. Performance management, etc.

Figure 5 depicts this architecture.

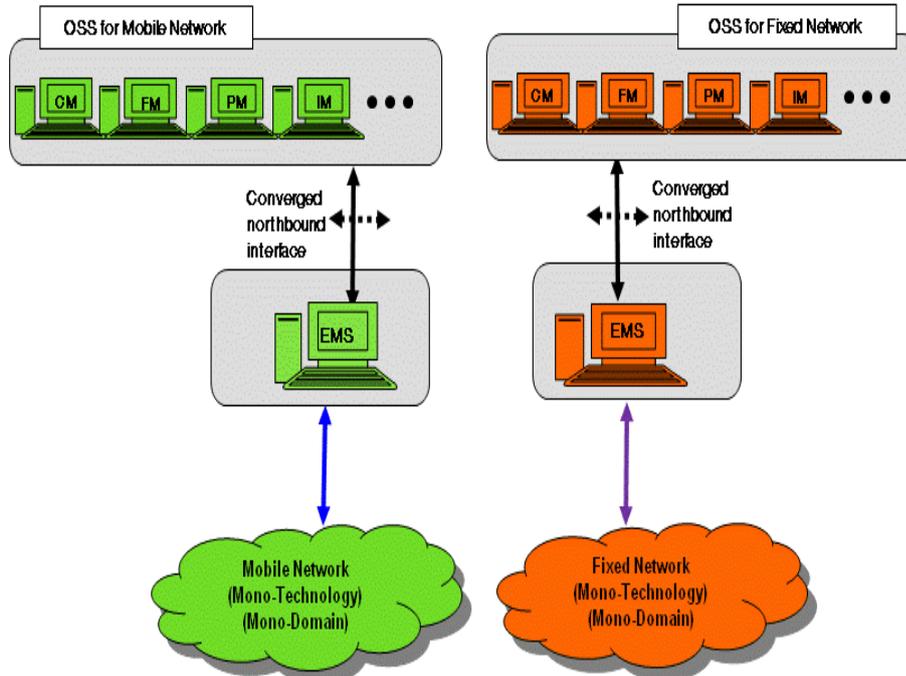


Figure 5: Use case architecture „Converged Northbound Interface”

2.5 Generic Converged Operations Use Cases (“Combinations”)

In this chapter, we propose three possible combinations of the aforementioned architectures (basic converged operations use cases) within an operator’s environment, leading to three generic converged operations use cases :

- C1: Converged element management layer together with converged northbound interface
- C2: Converged network management layer together with converged northbound interface
- C3: Converged element management layer together with converged northbound interface and converged network management layer

2.5.1 C1 - Converged Element Management Layer together with Converged Northbound Interface

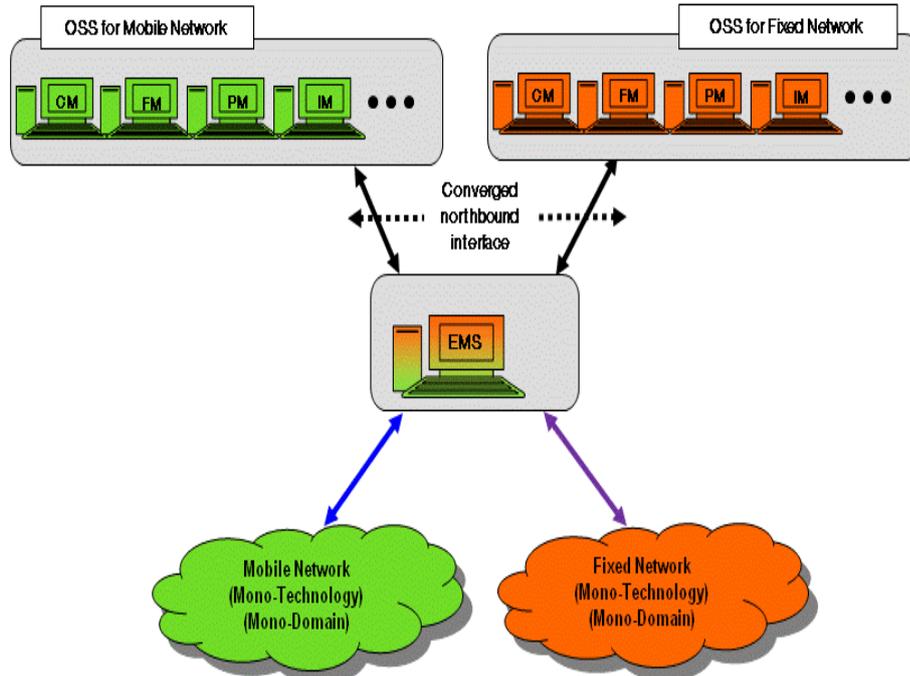


Figure 6: Combination "Converged Northbound Interface and Converged EMS"

This generic converged operations use case combines the basic converged operations use cases depicted in Figure 4 and Figure 5.

2.5.2 C2 - Converged Network Management Layer together with Converged Northbound Interface

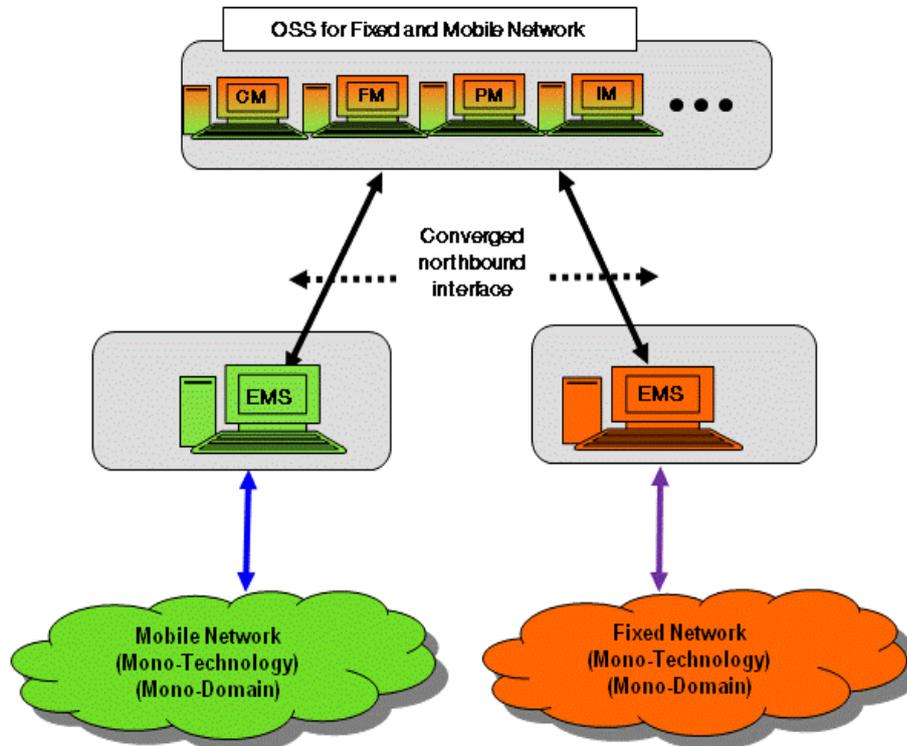


Figure 7: Combination “Converged Northbound Interface and Converged NMS”

This generic converged operations use case combines the basic converged operations use cases depicted in Figure 3 and Figure 5.

2.5.3 C3 - Converged Element Management Layer together with Converged Northbound Interface and Converged Network Management Layer

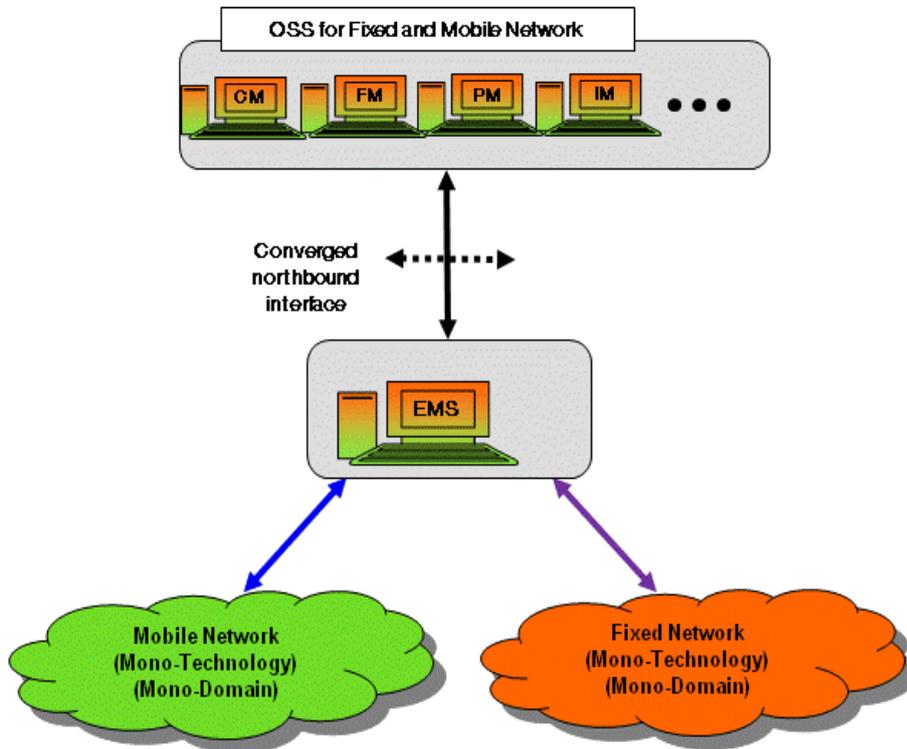


Figure 8: Combination “Converged Northbound Interface and Converged EMS & NMS”

This generic converged operations use case combines the basic converged operations use cases depicted in Figure 3, Figure 4 and Figure 5.

2.5.4 Basic Converged Operations Use Cases vs Generic Converged Operations use cases

Table 2 summarizes the relationship between basic converged operations use cases wrt. generic converged operations use cases in order to provide a global overview of relevant combinations.

		Generic Converged Operations use case		
		Figure 6 Converged itf-N + converged EMS	Figure 7 Converged itf-N + converged NMS	Figure 8 Converged itf-N + converged EMS + converged NMS
Basic converged operations use case	Figure 4 converged EMS			
	Figure 3 converged NMS			
	Figure 5 converged itf-N			

Table 2: Basic Converged Operations use cases vs Generic Converged Operations use cases

2.6 Requirements wrt. Converged Operations

2.6.1 Converged Element Management System

Real Use Case – EMS shared amongst Operator's affiliates

Large service providers have footprints in many countries. Though, in some of these countries, they are incumbents, it also happens that, in some other countries, they are challengers, have limited footprints and have to lower their CAPEX and OPEX to be competitive. In some cases, they deploy a relatively limited number of network elements in each country and put in place a unique organization responsible for operating these domestic networks. The resulting 24/7 shared **Network Operation Centre (NOC)** uses a single EMS for all the nation-wide networks it is in charge of. NOC staff is responsible for daily operation of the various networks. However, in some countries, local staff, thanks to their local OSS applications, keep limited capabilities for managing their network, e.g.:

- Alarms coming from operator Affiliate X domestic network up to the shared EMS is treated by shared NOC staff. However, it might be required to filter such alarms and forward them to operator Affiliate X OSS FM application, either for information only or for action (acknowledge, clear, etc.).
- Operator affiliate X might want to collect performance management counters / KPIs related to its own network. He may want to trigger, from its own OSS PM application, performance measurement campaigns for its own purpose, and collect related PM measures at its OSS.

Figure 9 depicts this real use case.

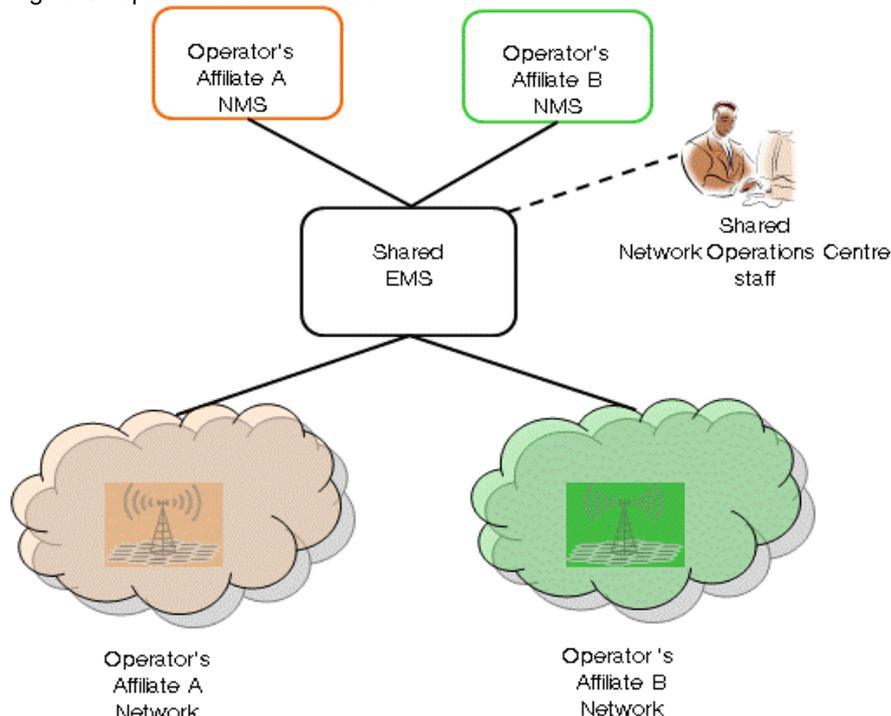


Figure 9: Single EMS for multiple affiliates

Instantiation and relevance

This use case is an instantiation or an implementable scenario of generic operations use case depicted in Figure 6 which requires a converged EMS and converged northbound interface.

In this use case, all affiliates deploy networks from a single vendor (mono-vendor environment)

High-level requirements



R1: Vendors' EMS shall be common to multiple network domains / technologies, e.g. it shall be able to cover not only multiple radio access technologies but shall also enable network operators to manage their wireless and wireline network domains in a unified way.

Expected benefits

- CAPEX savings (Only one EMS HW platform but limited savings). CAPEX / N (N number of affiliates)
- OPEX savings (can be very important, depending on the number of affiliates that are involved). OPEX / N (N number of affiliates)

2.6.2 Harmonized EMS northbound Interfaces

Real Use Case:

To be completed.

High-level requirements

R2: Vendors' EMS shall offer a unique set of management capabilities at its northbound interfaces. It is expected that EMS northbound interfaces are implemented according to the following rules:

- Network resource models for various network domains are built on a federated network resource model, i.e. network resource model for wire line network domains shall not be 100% different from network resource models for wireless network domains.
- Functional interfaces for wireline and wireless networks shall be similar for at least configuration management, fault management, performance management, inventory management, software management. EMS northbound Interface shall offer common management capabilities to the operator, regardless of the network domain.
- It is of primary importance that EMS northbound interface fully implements:
 - standardized northbound interfaces firstly and
 - clearly identifiable, vendor-specific extensions to capture vendors' own set of parameters and/or value added management capabilities. Vendor's specific capabilities shall be implemented as extensions
- EMS northbound interface shall be based on web services.

Expected benefits:

- CAPEX savings
- OPEX savings

2.6.3 Multi-domain Network Management applications

Real Use Case – NMS shared amongst operator's affiliates

Large network operators have deployed their networks in many countries. Instead of developing a dedicated OSS application in each country for e.g. fault management, it is common that they develop a single OSS application for multiple countries and/or multiple domains and and/or multiple technologies. Such operator-wide OSS applications are based on a kernel and possible adaptations due to local and/or domain-specific and/or technology-specific requirements. Such operator-wide OSS applications can be deployed either on a per-country roll-out basis or in an ASP mode. Figure 10 depicts this use case.

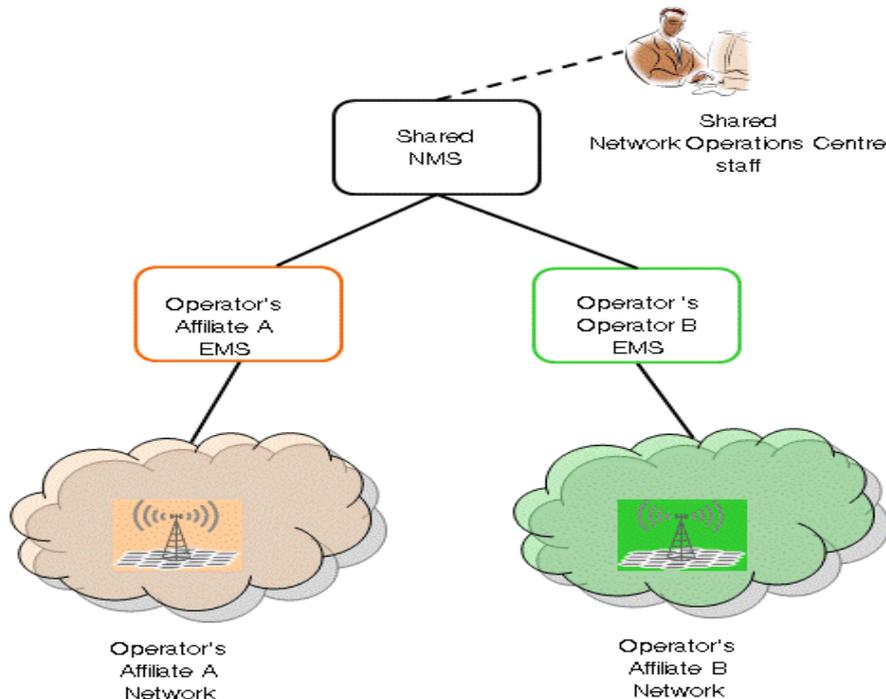


Figure 10: Single NMS for multiple affiliates

Instantiation and relevance

This use case is an instantiation or an implementable scenario of generic operations use case depicted in Figure 7 which requires a converged NMS and converged northbound interface.

This use case is relevant when operator's affiliates have networks from different vendors (multi-vendor environment).

High-level requirements

R3: Network management applications shall be, up to the maximum, common to multiple network domains / technologies. They shall be based on a kernel, common to multiple network domains / technologies, and possibly technology-specific management capabilities.

Expected benefits

- CAPEX savings (only one NMS HW and SW platform): CAPEX / N (N number of affiliates)
- OPEX savings (can be very important, depending on the number of affiliates that are involved): OPEX / N (N number of affiliates)

2.6.4 Multi-Operator Network Management

Real Use Case – RAN Sharing with EMS shared amongst operators

Several RAN sharing scenarios have been described within 3GPP, including MOCN, GWCN, MORAN, national Roaming, etc. In all these scenarios, a **“Master Operator”** is given the responsibility to operate the shared network on behalf of the other operators. This master operator will have its own EMS to manage the shared network, while the other operators will have their own OSS applications. However, the aforementioned “other operators”, i.e. those who are sharing the network, can either sub-contract all network operations to the master operator (in this case, they don't claim any rights on the shared network), or they require to be able to have operational capabilities on the shared network.

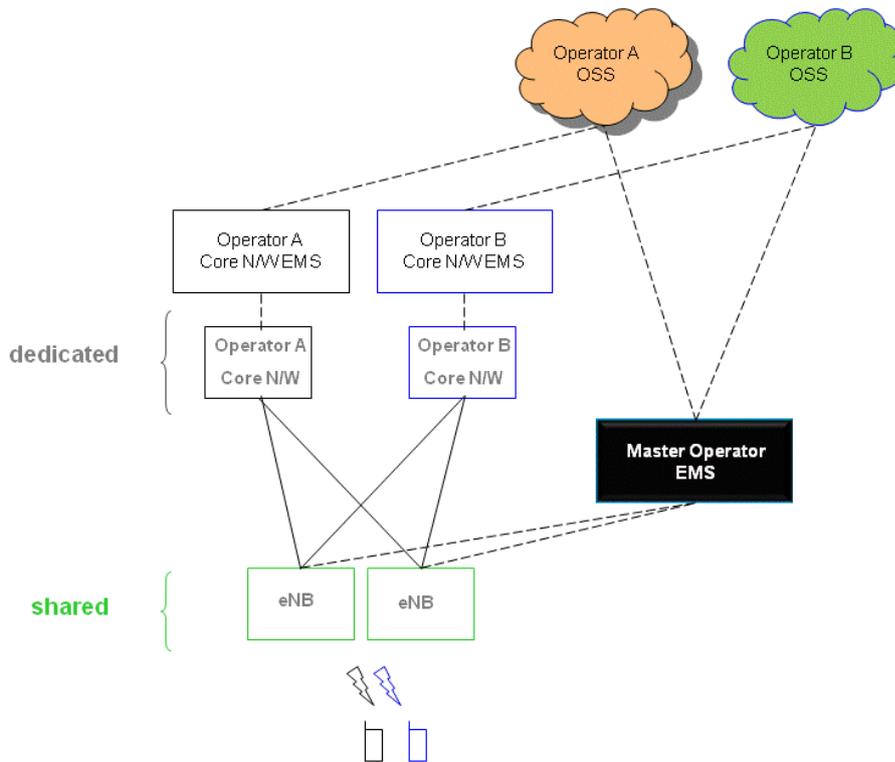


Figure 11: RAN Sharing with EMS shared amongst operators

Instantiation and relevance

This use case is an instantiation or an implementable scenario of generic operations use case depicted in Figure 6 which requires a converged EMS and Converged Northbound interface.

High-level requirements

For a number of reasons or regulation obligations, operators' expectations to reduce their CAPEX and OPEX, radio access networks may be shared among operators. From operations perspective, management cost must be reduced accordingly within an operations agreement structured through a master operator role and secondary operators roles.

R4: It shall be possible that master operator EMS and secondary operators NMS communicate with each others through a standardized northbound interface.

Expected benefits

- OPEX savings: 50% (if there are 2 operators)

2.6.5 Mono-operator Management via a third party

Real Use Case- Outsourcing of operator's Network management

The operator is deploying a network in a country or region characterized by a very low ARPU while the goal is to make a profitable business. He identified from his cost modelling study that the most significant cost component is network operations, hence he decides to outsource this activity to a 3rd party through an operations agreement.

The operator strategy is only focusing on marketing and sale activities and no longer on operations with expectations of reducing significantly its OPEX.

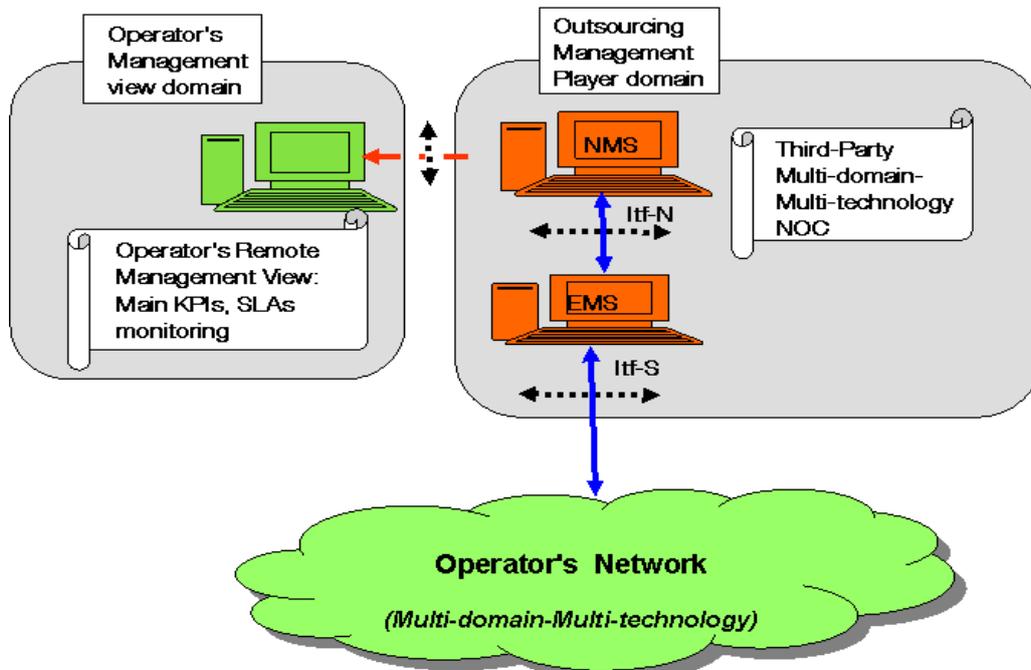


Figure 12: Operations architecture within outsourcing

Instantiation and relevance

This use case is an instantiation or an implementable scenario of generic operations use case depicted in Figure 8 which requires a converged EMS, converged NMS and converged northbound interface.

High-level requirements

R5: Operator outsources the operations of his network to a highly qualified 3rd party.

R6: The 3rd-party NOC (Network Operation Centre) must be composed of an EMS and NMS applications connected through a standardized converged northbound interface.

R7: The 3rd-party NOC shall expose to operator a view on the way the management agreement is fulfilled through a remote screen with relevant KPIs, indicators, SLAs, ...

Expected benefits

- CAPEX savings: 100%
- OPEX savings : 100%

2.6.6 Converged Service Management applications

Real Use Case



End-to-end service configuration and activation from a unique OSS application is key for service providers. In the future, when a new fixed and mobile IMS VoIP subscriber is to be provisioned, the following list of NEs will have to be provisioned:

- Home Gateway
- IMS HSS
- HLR
- EPC HSS
- SPR/PCRF
- Possibly FemtoCell.

In order to enable end-to-end provisioning in a timely manner and error-freely, having a single service configuration and activation application capable of orchestrating provisioning requests to various underlying domain specific provisioning applications will help in reducing OPEX and improve customer satisfaction.

High-level requirement

R8: Operators expect common service management applications for the following functional processes, belonging to service operation and management:

- Service configuration and activation
- Service problem management
- Service quality management

Expected benefits

- CAPEX savings
- OPEX savings

2.6.7 Open architecture for EMS / NMS

Real Use Case

Operators want more modularity in EMSs architecture. For example, they want to get rid of some EMS modules in charge of collecting and analysing statistics based on performance measurements and, in some cases, prefer to collect and analyse these statistics directly within their in-house OSS applications. This implies that the format of the files containing the statistics and the protocol used to convey these files from the network elements up to the OSS application must be open.

High-level requirement

R9: EMSs shall have open and modular architectures, with open APIs between modules. It shall be possible for operators to decide whether they want to get all modules or only a part of them.

Expected benefits

- CAPEX savings

2.6.8 To which players the requirements are addressed

As indicated in chapter 1.1 Introduction Sub Task Converged Operation, the requirements formulated in this document are addressed to three key players in order to be translated into standards or implementable solutions meeting the operators' needs in terms of CAPEX and OPEX reduction:

- SDOs



- OSS vendors and integrators
- Telecom equipment vendors and network equipment providers

In Table 3 we classify the requirements per player.

	Players		
	SDOs	OSS Vendors and Integrators	Telecom Equipment Vendors and Network Equipment Providers
Converged Operations Requirements	R1		
	R2		
	R3		
	R4		
	R5		If he is operator
	R6		
	R7		
	R8		
	R9		

Table 3: Requirements vs Players

2.7 Conclusion on Requirements for Converged Network Operations

The CON section describes technology and domain independent converged operations use case architectures. They are then instantiated to some operators' real use cases in order to make them converged operations aware. As illustrated, these generic operations use case architectures can be combined up to the ultimate level of operations convergence that meets operators' needs.

In order to illustrate the way these converged operations use cases architectures are applicable in the real operations world from operators' perspective, we instantiated them to the four real use cases extracted from the list we proposed in chapter 2.3:

- RAN Sharing with converged EMS layer
- Converged EMS layer for multiple affiliates
- Converged NMS layer for multiple affiliates
- Outsourcing of operator's network management

This constitutes the outcome of phase 1 of this NGCOR sub task which objective is the validation of the three generic converged operations use cases, the instantiations to three real use cases of high interest to the operators and related requirements.

The follow up of the CON section is planned for phase 2: the use cases covered in phase 2 will constitute the content of the next release of this section and will contain:

- New real use cases of interest for operators
- Use Cases within fixed – mobile network convergence (FMC) or harmonization (FMH), most of them being imported from joint 3GPP-BBF (FMC) activity
- How to make NGCOR converged operations use cases autonomic-aware. Indeed, an additional cost saving lever is to make converged operations architectures autonomic-aware, hence agile and flexible. This can be achieved by introducing autonomics and self-management concepts and design principles in operators' OA&M /

Draft

NGMN Copyright 2011

Any publication outside NGMN has to be pre-approved in writing by NGMN



OSS solutions. It's an ongoing work within the ETSI Industry Specification Group (ISG) for Autonomic Future Internet (AFI).

3 GENERIC NEXT GENERATION CONVERGED OPERATIONAL REQUIREMENTS (CON)

3.1 Introduction for Generic Converged Operational Requirements

The GEN section contains the generic part of the Next Generation Converged Operational Requirements (NGCOR), which are valid for all other specific NGMN NGCOR sections Converged Network Operations (CON), Modelling and Tooling (MT), and Inventory Management (IM). The intention of the GEN section is to avoid redundant requirement descriptions in different NGMN NGCOR sections, e.g. the high-level, generic requirements from the FM interface requirement section has been identified to be generic for all other types of interface requirement specifications as well, not only for FM.

3.2 Non-Functional Interface Requirements

The following topics describe core business driven requirements for interfaces in the OSS domain. The following figure provides the overview.

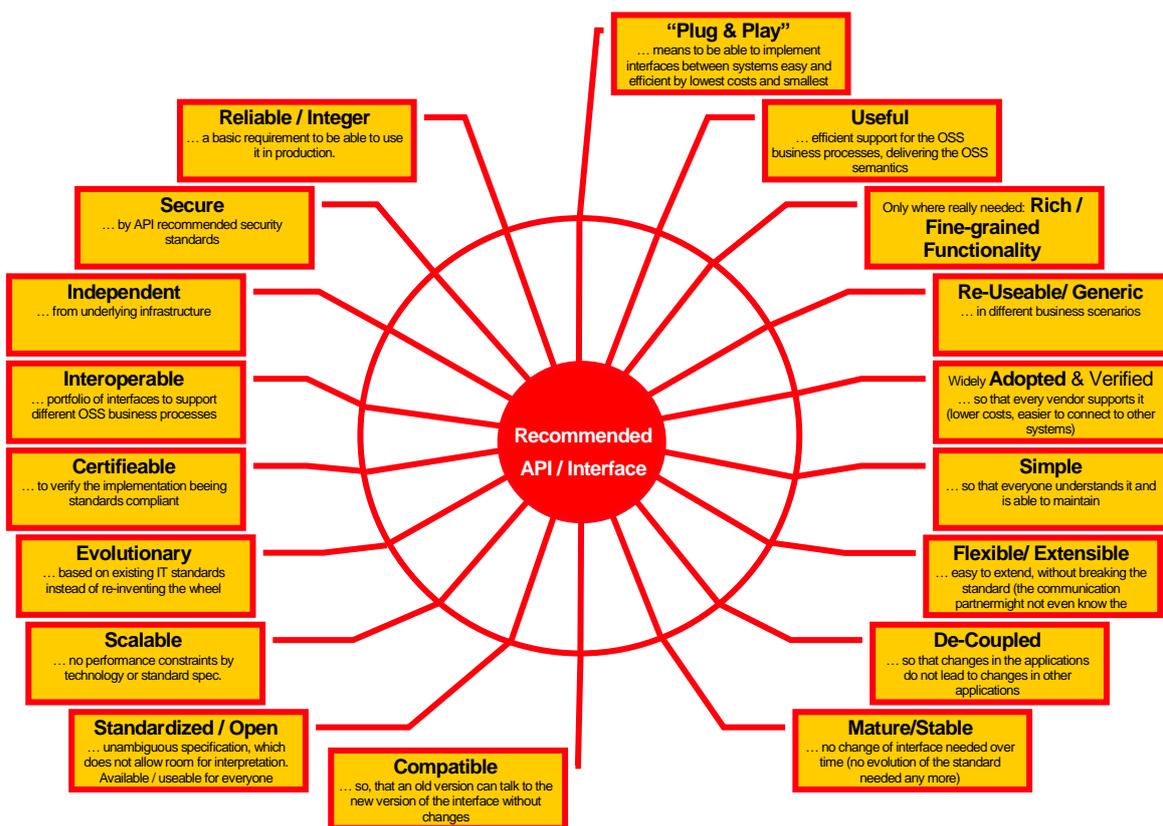


Figure 13: Requirements for recommended API / interface



R1: “Plug & Play”

It must be possible to implement the interfaces between the OSS easy and efficient by lowest costs and smallest effort (ideally without any development and/or configuration).

- Backward compatibility (see related topic) is a prerequisite to support this characteristics during the whole life-cycle of the standard interface (e.g. plug & play must be still possible, if the client uses version 1.0 and the server uses version 1.2 of the same interface specification)

R2: Useful

It must deliver efficient support for the OSS business processes. The interface must deliver the needed OSS semantics to support the process.

- Implementable (not academic) support of business process frameworks (e.g. eTOM and ITIL) and common information models (e.g. SID semantic)
- Clear and unambiguous scope of the interface (e.g. to differentiate from Service Inventory), without mixing different business scenarios (e.g. an interface which supports resource configuration management should not be mixed with a resource fault management interface, because this might lead to complex interface specifications and expensive implementations)

R3: Re-Useable / Generic

The interface must be generic enough, to enable the re-use in different integration scenarios.

(e.g. NMS - FM offers a standard interface for communication with other NMS such as trouble ticketing)

- This is a prerequisite to support M : N integrations and to reduce cost and effort for integrations
- Extensions in future versions will not hinder to implement it in a generic way and will not hinder to re-use

R4: Simple

The interface must be simple enough, so that people which have not been involved in the specification are able to understand it (or even do not need to understand the details), so that they are able to implement, maintain and use the interface.

- This will help to reduce cost and effort for the implementation and the operation/maintenance of the interface.

R5: Flexible / Extendible

The interface can be extended and refined, from basic setup to more complex implementations without impact on the other communication partners. It must be possible to extend the interface capabilities (methods and attributes), without breaking the standard.

- It must be possible to use a very simple, basic setup of the API in one side of the communication partners, and a more complex API on the other side of the communication partners (which contains the “simple” API as the basic core) without disturbing the communication. That means, that there is a stable basic core, which can be extended and optionally used, but there is no dependency on all communication partners to use the extensions (as long as it is not part of the common standard itself).
- (The communication partner might not even know the extension, e.g. the server uses extended attributes, while only a small number of clients is aware about the extension → The interface still works as specified, without any impact on the clients which do not know the extension.)

Rationale



- Avoid dependencies between server and client. But, at the same time, enable complex interactions, to support complex NE behaviour.
- This capability can be used to implement new versions with extended capabilities without losing backward compatibility.

R6: Rich (as far as needed)

Fine grained, rich functionality ONLY where really needed and absolutely necessary to support the common basic process. Adding more and more capabilities into the interface specification will lead to complex and expensive implementations (which often hinders the adoption of the interface) and might lead to a dilution of the scope of the interface and overlapping functionality with other interfaces.

- Fine grained / rich functionality must be delivered in specific areas to address e.g. technology specific requirements (e.g. in case of Resource Configuration Management)
- BUT: consideration of the richness to support the business process in an appropriate way vs. business benefit for all standard interface implementers.

R7: Standardized / Open

The interface has to be based on unambiguously standardized specification, which does not allow room for interpretation. The specification and related artefacts must be freely available and useable for everyone.

- This is a prerequisite to enable compatibility between interface implementations of different vendors.

R8: Mature / Stable

The interface must be stable and mature, to avoid expensive changes on implemented interfaces.

(Ideally there is no requirement for change any more).

- Prerequisite: the interface specification has to be faulty – free before it is released to the market.
- Prerequisite: the managed OSS domain does not change very often.
- This helps also to avoid backward incompatibility by avoiding continuously changing interface specifications.

R9: De-Coupled

Changes in the application or in the interface implementation at one of the communication partners may not lead to the need for changes in the application or in the interface implementation of the other communication partners. (Please consider, that this requirement does not assume any specific type of implementation technology.)

- This is a prerequisite to ensure, that changes in one OSS will not impact other OSS, to avoid dependencies between OSS applications which might lead to high costs for the impacted communication partners and to enable M : N integrations.

R10: Evolutionary

OSS interface shall re-use already existing, widely adopted and mature IT standards (e.g. transport protocols) to avoid “reinventing the wheel”.

- This will reduce cost and effort to create and to implement new technologies.



R11: Independent

The interface specification must be independent from underlying infrastructure.

- This will allow to re-use the same interface implementation in different environments, without dependencies on vendor specific capabilities, (e.g. the specification has to be independent from hardware, operating system, bus environment, etc.) to avoid costs for the customization of interface implementations due to environmental dependencies of the specification.

R12: Certifiable

It must be possible to certify the standard compliancy of the interface implementation.

- This will allow the verification, that the interface implementation is compliant with the standardized interface specification to avoid compatibility problems between interface implementations of different communication partners.

R13: Compatible

It must be possible to implement a new version of an interface specification at one of the communication partners, while the other communication partners still use an old version of the interface specification. This “mixed versions” of interface implementations can be used without any impact on the communication partners or the interface implementations of the communication partners.

- The implementation of the new interface version at one of the communication partners must ensure the mapping according to the interface specification.
- This will allow to implement new interface versions in a productive environment without the cost and effort to upgrade all other communication partners (a real business need might lead to the upgrade sooner or later, but this can be decided by the owner of the “old” communication partner itself. Immediate upgrades are often difficult or simply impossible).

R14: Interoperable

The interface implementation shall be based on an interoperable portfolio of interfaces / interface specifications to support different OSS business processes using a common architecture and a common information model.

- This will allow the implementation of complex business scenarios, spanning different integrated OSS components, using a common, well known interface environment without complex mapping of information models.

R15: Scalable

No performance constraints caused by the interface specification or the implementation.

- The specification or the selected implementation technology may not result in performance issues.

R16: Secure



The interface has to be able to ensure confidentiality, integrity and availability of the data, which is transferred by the interface.

R17: Reliable

The interface implementation has to ensure the reliability of the data, which is transferred by the interface.

- This is a basic requirement to be able to use an interface in a productive environment.

R18: Interface Robustness

No interface dependencies on availability between NMS and EMS, if one of the EMSs (Server) communication partners is not available.

Description

- An outage of one or more EMSs (source) may not lead to any impact on the connectivity between NMS and other EMSs.

Rationale:

- Avoid complex behaviour of the interfaces. The interface to the remaining EMSs must be still available during the time then one or more EMSs are down.

R19: Adopted and Verified

Widely adopted and verified, so that every vendor supports it.

- This means, that it is possible to implement the interface efficiently (low cost and effort) because there is know how in the market, the interface specification is widely accepted and proven to be useful and it is already available as part of OSS COTS packages.

R20: Simple Trace and Logging

The interface must deliver a simple “trace and Logging” functionality (in readable text format).

Description

- The interface must allow logging of all commands (send, receive, query, etc. ...), including the content in simple, human readable text format (no hex or binary, etc. ...) to support the error-analysis of the interface itself. The logging/tracing functionality is configurable.
 - The level of details can be configured
 - All attributes of the content can be used as to configure trace – masks
 - Masking of attributes
 - Masking of attribute- content
 - Logging of interface problems / errors

Rationale:

- The goal is to enable the operator/administrator to restore a connection problem on the interface very quickly.

R21: 1:1 Relation between Event MO Instances and Inventory MO Instances

Description

- If MO identifiers used/provided by the inventory component of an element manager need to be mapped to meet naming requirements of the inventory database, the same mapping must be applied to the MO identifiers in the event. The corresponding is true if mapping is driven by event naming requirements.

- If MO identifiers of events and inventory within an element manager are different, the difference must be eliminated before the above mapping can be applied.

Rationale

- An event must be unambiguously related to a known object instance (in the inventory).

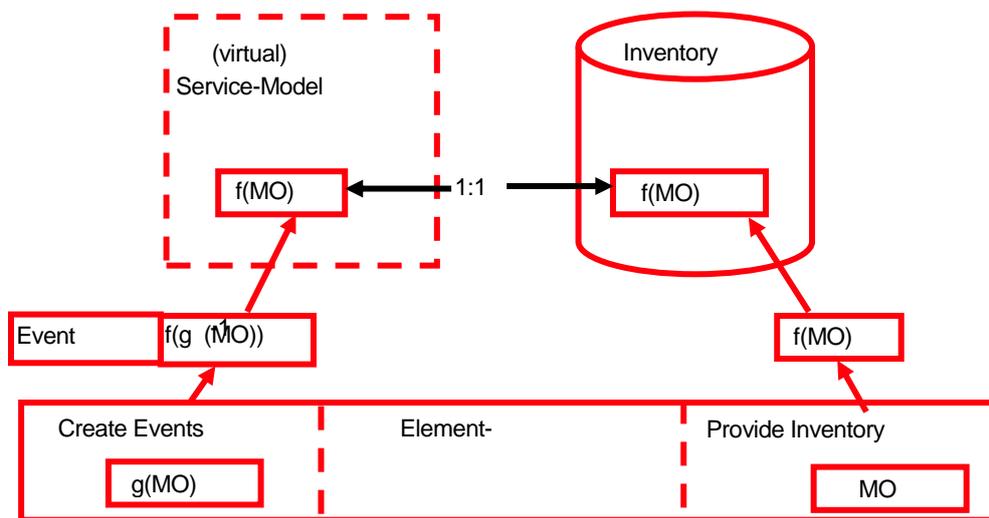


Figure 14: Managed objects in the context of service model and inventory

R22: “MO Instance” Attribute Information Structure for EMS ↔ NMS Event Interfaces

Description

The information in the “managed object” attribute of the interface must allow a clear and unambiguous identification of the component (HW or SW), which is the originator of the event.

- The managed object, as an attribute of the basic generic event – object, shall not contain any detailed topology information. The assumption is that the NMS will use an inventory database (internal or external) to map between managed object instance and inventory topology tree if needed.
- The basic assumption for this is that there is a one-to-one mapping between managed object instance and the inventory information, so that the instance can be unambiguously identified. If this is not the case, the instance must contain a very simple and standardized methodology to describe the relationship between the first unambiguously identifiable object and the related not-unambiguously identifiable object, which is the originator of the event.
- NMs requirement (specific for the NMS layer): As soon as the event information leaves the area of the local network and the managed object attribute value does not deliver unambiguously any more, the network manager will add additional information, the “NameSpace” - string to the Managed_Object_Identifier attribute (Proposal: Company_Name + Technology-Domain → “Access”), so that it is unambiguous in the larger context again. (Remark: the name of the EMS should be part of the “additional information” attribute, and not part of the MO_ID)
- Here the general proposed structure of the “Managed Object Instance” attribute:
 - Managed Object Instance ::= <NameSpace.>* <MO_Name> <MO_Detail>*
 - NameSpace ::= <Global IdentifierString> (see NMS Requirement above)
 - MO_Name ::= <Ressource_Name>|<Inventory_Name>
 - The Ressource_Name is delivered by the Ressource or the EMS itself. This name might be enriched or normalized on EMS or NMS layer with some information from Inventory systems, e.g. topological information.

Example:



- Inventory_Name ::= <Hostname>|<Service>|<Serviceelement>|<ResourceGroup>|<UseCase>|<UseCaseSubtype>| ...
- MO_Detail ::= <Blocknn>|<Racknn>|<Slotnn>|<Portnn>|<IP_address>|...
 - (The MO_Detail information is delivered by the resource or the EMS itself. It adds information about the detailed origin of the alarm as far as this is known by the resource or the EMS. There is no limit on the number of topological elements, but it should be limited to an absolute minimum, just to the number which is really necessary to unambiguously identify the defective component.
- A semicolon is used as a delimiter between the structural components of the managed object instance.

R23: M : N Connectivity

Multiple NMS applications might be connected (logically) to several EMS applications (M : N)

Description

- The API specification allows connecting one NMS to multiple EMS. (This might have an impact on addressing – mechanisms in the API).
- Furthermore the API specification allows splitting the incoming Event/Alarm traffic between different instances of the same API implementations to avoid overload situations in one API instance.

Rationale

- This capability allows reducing the effort for the maintenance of several different client- side interfaces



4 REQUIREMENTS FOR MODELLUNG AND TOOLING (MT)

4.1 Background for Modelling and Tooling

The main important future O&M requirements are specified and defined in the NGMN Top OPE Recommendations. Those requirements will need further enhancement with more details for guiding towards well standardized interfaces and interworking solutions throughout O&M/OSS. Resolving misalignments and open questions in the standardization of the area needs immediate actions already in the short term.

The specification of common usable network data and operations (information model) for these networks allow reducing CAPEX (harmonised networks) and OPEX (seamless operation processes). Reducing integration cost by harmonising the data model, reducing maintenance cost by unifying the operations model.

4.2 Objectives

The objective of the project is to produce detailed requirements from operator's point of view for an infrastructure that allow an efficient specification of management interfaces for converged networks. These requirements are based on the operator's expectations on a converged modelling and tooling infrastructure which need to be taken into account by the Standards Developing Organisations (SDOs).

Already existing modelling and tooling specifications in 3GPP and TM Forum are taken into account and will be used as input to produce the requirements for the converged interface specification infrastructure.

4.3 Definitions

The MT section defines or specializes the following terms:

4.3.1 Federated Model

The federated model contains the overall harmonised model elements and is sometimes also called umbrella model. It should enable the implementation of convergent network management functions and processes (for example alarm correlation) which need to operate on objects belonging to different network domains (for example wireless and wireline). The **Federated model** is composed of the **Federated Information Model** (FIM) containing the data part of the model (object classes and their attributes) and the **Federated Operations Model** (FOM) containing the dynamic part of the model (notifications and operations grouped in interfaces).

The model covers resource and service management layers and all their management functions like **Configuration Management** (CM), **Fault Management** (FM), **Performance Management** (PM) , **Inventory Management** (Inv. M) or **Provisioning and Assurance**.

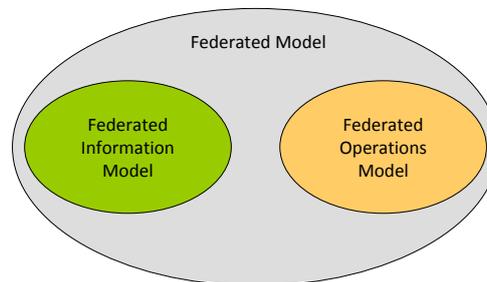


Figure 15: Federated model

Note:

The FIM is similar to the NRM IRPs (Network Resource model Integration Reference Points) in 3GPP and the Shared Information & Data Model (SID) in TM Forum.

The FOM is similar to the Interface IRPs (Integration Reference Points) in 3GPP and the Business Services in TM Forum.

4.3.2 Interface

The term “interface” used in the MT section is a network level management interface between various kinds of operation systems. Consequently, interfaces between element management system (EMS) and the network elements (NE) are out of scope.

In 3GPP terms:

The interface in scope is the northbound interface (Itf-N) between the EMS and the NMS (or operator's OSS for Operations Support System). This is clearly depicted in Figure 18. This Itf-N could be evolving towards a communication infrastructure as depicted in Figure 16 such as SOA (Service Oriented Architecture).

The southbound interface, between EMS and network elements is out of scope.

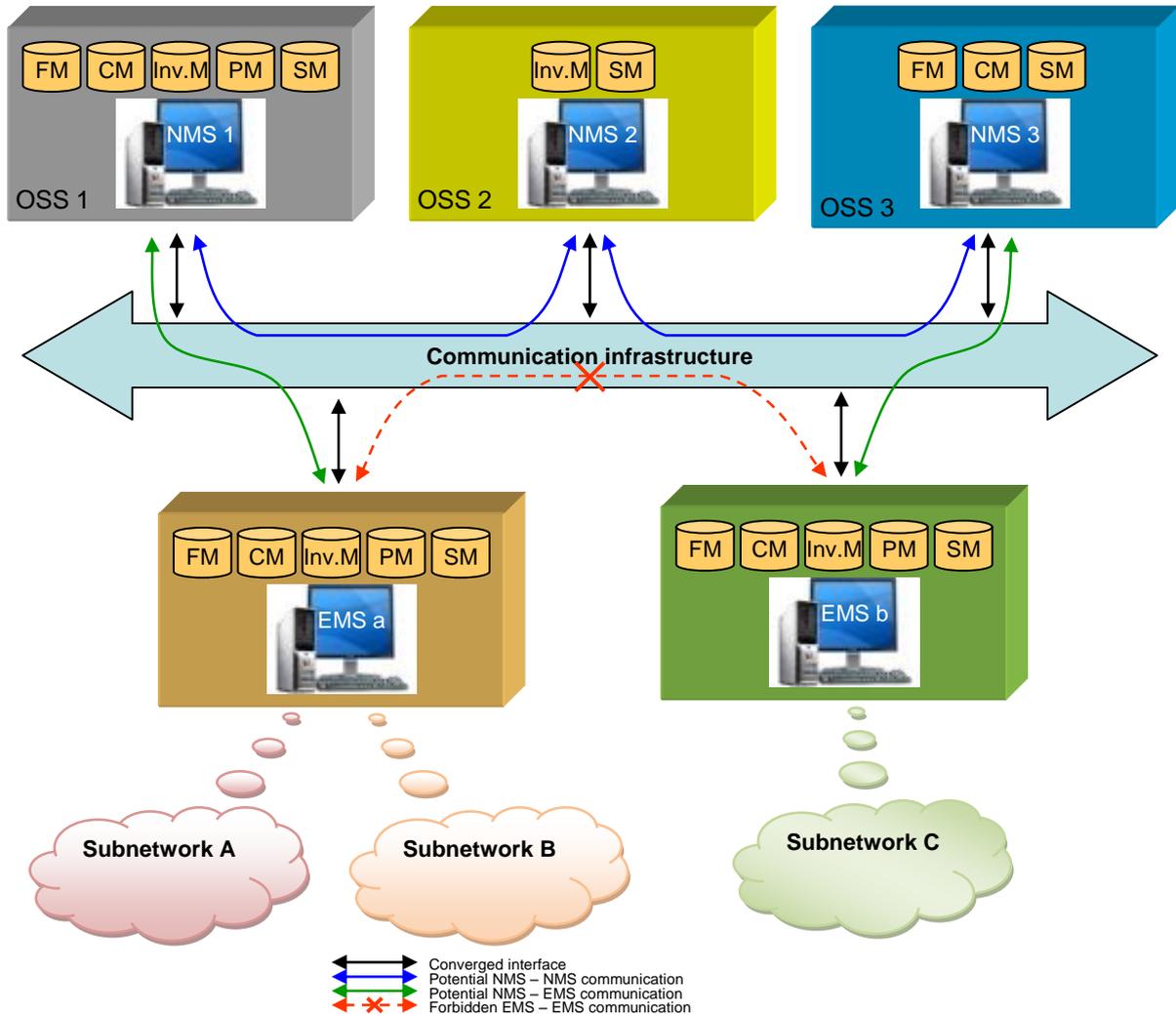


Figure 16: Converged Interface peers

4.4 Project Scope, Deliverables and Methodology

Main scope of this sub task:

- Define requirements for the modelling environment
- Define requirements for the Federated Information Model
- Define requirements for the Federated Operations Model
- Define requirements for the tooling infrastructure
- Define requirements for general operations used at the interface
- Analyse gaps between requirements and existing solutions.

Out of scope for this sub-task:

- Define requirements for specific operations used at the interface

Deliverables of this sub task:

- Modelling environment requirements (e.g., specification structure, general design principals and modelling patterns)



- Tooling infrastructure requirements (e.g., interchange file formats)
- Recommendations regarding implementation

Methodology of this sub task:

- Definition of the level of details
- Audit of the data models, design principles and guidelines from 3GPP SA5 and TM Forum
- Definition of design principals and patterns
- Definition of interface modelling requirements
- Analyse requirements and existing solutions

4.4.1 Requirements Definition

For Requirements definition the following tasks have been identified:

- Modelling environment requirements
- Tooling infrastructure requirements.

“Modelling environment” includes the following requirements:

- General requirements to SDOs like a common terminology and the harmonisation of operations processes, etc.
- Modelling related requirements to SDOs for creating a shared umbrella model → federated model
- Data and operation models like all wireless/wireline agnostic objects and operations have to be used whenever possible and must not be specialised in the individual domains.

“Tooling infrastructure” includes the following requirements:

- Basic requirements for developing management interfaces, for example, an open source tool shall be supported
- Converged framework model requirements, like wireline and wireless management models have to be consolidated in one harmonized model.



4.5 Requirements Definition

Abstract:

3GPP WG SA5 has specified detailed **Network Resource Models** (NRMs) for the management of mobile networks, plus a generic network resource model.

TM Forum has done the same for the management of various kinds of fixed networks, as well as a **Shared Information & Data** (SID) model providing a "common reference model for enterprise information that service providers, software providers, and integrators use to describe management information", i.e., also generic definitions for network and service management aspects.

It shall be noted that the 3GPP generic network resource model and TM Forum SID have different scopes and have been developed independently from one another. As a consequence the resulting models are different. Though there will always be a part in the generic NRM and the SID which is different due to the different network technologies modelled, there are numerous model elements which do not have to be different between the two models for the different network technologies.

Examples of these aspects are the generic NRM and the SID, modelling of resource inventory information, modelling of security aspects, modelling techniques and how vendor specific resource model extensions are managed using NRMs and SID.

Parallel to 3GPP und TM Forum are even more other standards development organisations (SDOs) such as the Internet Engineering Task Force (IETF), International Telecommunications Union – Telecommunication Standardization Sector (ITU-T), Broadband Forum (BBF), Metro Ethernet Forum (MEF), etc., which have defined different management standards / recommendations for mobile and fixed networks. In addition to the SDOs many vendors deliver element management systems (EMS) with their own proprietary solutions for specific technologies / networks.

Because all sets of specifications have been specified independently, the management of the mobile part and the fixed part is currently structured along silos with different management interfaces, resource models, management architectures, and management workflows.

An additional problem is that even within mobile or fixed networks, there exist different specifications (Modelling/Tooling) which are developed by different SDOs or vendors.

All these different Standards (from SDOs) and proprietary solutions (from vendors) use different modelling/tooling, therefore the CAPEX and OPEX for network operators and integrators to integrate all these interfaces have increased dramatically.

I think that we shall make it clearer that there are two things:

This heterogeneous modelling/tooling (1/ different models for different network domains / technologies and 2/ different modelling frameworks (e.g. Stage 1-3 for 3GPP, BA, IA, IIS for TMF; UML for TMF with an inter-exchangeable format versus picture in 3GPP) also has a massive influence to scalability, time to market, complexity and applicability of these standards in OSS.

In the future the mobile and fixed networks will no longer be managed as separate networks. The convergence of mobile and fixed networks requires the convergence of the mobile and fixed OSSs.

The network operators and the telecommunication industry would greatly benefit from aligned management interfaces, management models, management architectures, and management workflows.



4.5.1 Modelling Requirements

Fixed and mobile networks are growing together → FMC. The specification of common usable network data and converged operations for these networks allow reducing CAPEX and OPEX.

We will be able to reduce integration cost by harmonising the data model and reduce the maintenance cost by unifying the operations model.

4.5.1.1 General Requirements

1. The following SDOs (at least 3GPP, TM Forum, ITU-T, BBF, MEF, and others) shall strengthen their joint activities regarding the Management topic.
2. It shall be possible to add other SDOs in the future.
3. The resulting harmonised data model shall be openly available.
4. The harmonised data model shall allow SDO-specific enhancements based on the common modelling patterns.
5. SDO specific enhancements should be realised in a way that enables a drill down process: from the federated model to SDO-specific one or vice versa. The drill down process means ability to identify a more generic class (concept) in the federated model which is enhanced in the SDO-specific model. This requirement is to assure that SDO-specific extensions can be clearly identified as detailed version of the commonly agreed classes and concepts.
6. The interfaces which use the SDO-specific data model should be compliant with the interfaces defined in the federated model. The compliance must mean that objects being passed as arguments to or returned as a result from methods of the interface can be treated as objects of classes defined in the federated model if the SDO-specific functionality is not required by a client using the interface.
7. The proposed mechanism of SDO-specific extension is via inheritance and composition (decomposition) object modelling design patterns.
8. The other SDOs shall be informed of SDO-specific enhancements.
9. The number of SDO-specific enhancements shall be reduced to the absolute necessary minimum.
10. The common management operations for fixed and mobile networks shall be unified.
11. SDOs shall agree on a common terminology.
12. The functional coverage of the converged specifications shall continuously grow; i.e., shall replace the functions in the individual specifications.
13. The harmonisation shall begin with high level business use cases, requirements and usage scenarios. Followed by the model harmonisation and finished by the protocol harmonisation. See Figure 17.
14. The Modelling shall be able to **comprehensively** describe the functions in a protocol-neutral way. "Comprehensively" means that the modelling shall be detailed enough to be used as the basis for another

protocol-specific specification.

Reason for this is because operators are mainly interested in functions which stay the same even when the protocol changes.

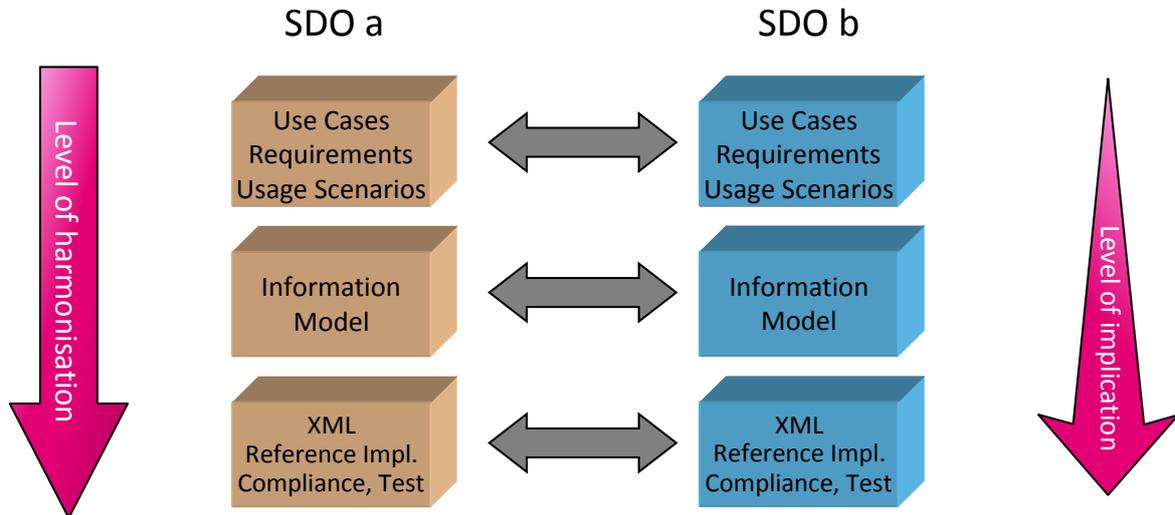


Figure 17: Interface Harmonisation Levels

Notes:

The uses cases are the basis for the requirements and usage scenarios are defined for each required operation.

The "usage scenarios" are called "use cases" in TM Forum.

The level of impact is increasing because of the backward compatibility constraints appearing on the XML level.

- Harmonisation should include all network layers at vertical and horizontal view, in order to achieve Operator's Harmonized OSS from multi-domain, multi-technology perspective, see example in Figure 18.

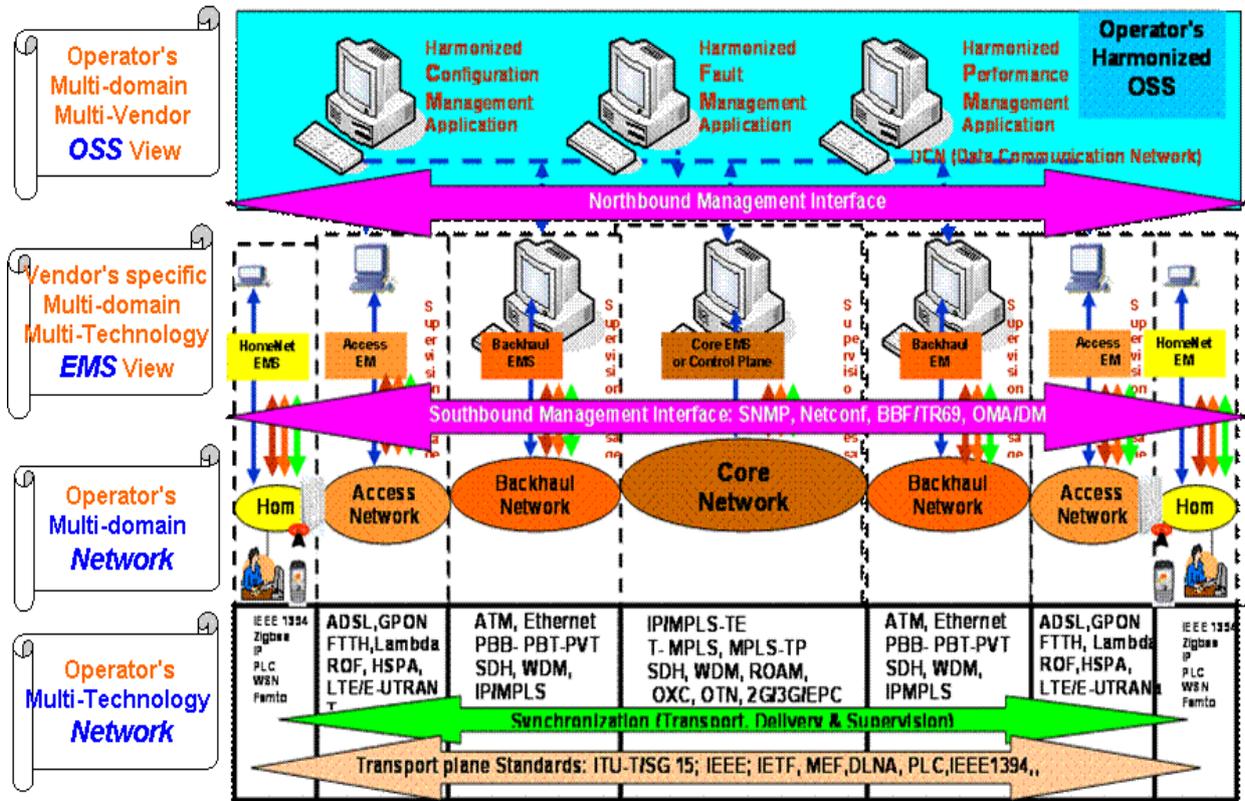


Figure 18: Operator's Harmonized OSS, End-to-End Network Multi-Domain, Multi-Technology Management View

4.5.1.2 federated model Requirements

16. The SDOs should define a common model for mobile and fixed networks as a shared umbrella model → federated model.

17. The FIM shall enable the modelling of all components of the mobile and fixed networks; see example in Figure 19.

Note:

The figure at left side is showing the EPC (Core Network) sharing between various access networks 3GPP and Non-3GPP.

The figure at right side is showing the LTE/EPC layering architecture.

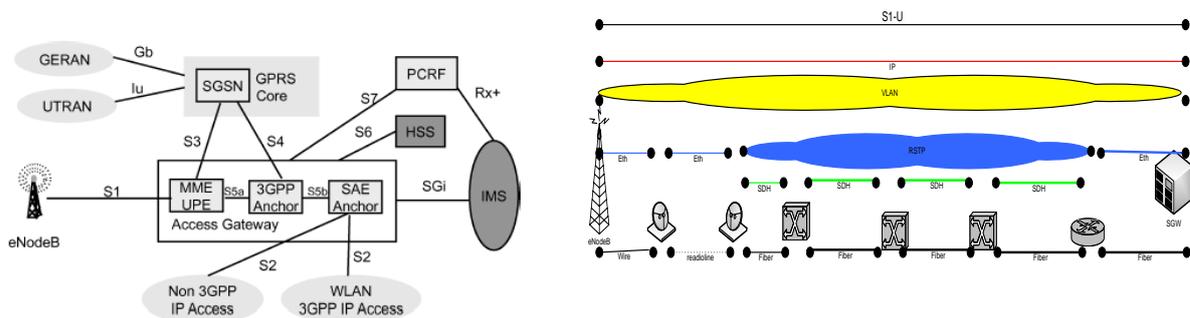


Figure 19: Example Mobile and Fixed Network – detailed layered view

18. The federated model shall contain over 80% of the data and operations which are necessary for managing mobile and fixed networks. When the amount of SDO / vendor specific data or operations is high, the costs for the operators (OPEX and CAPEX) increase significantly.
For example: If an operator has two suppliers whose products are based on the federated model but the data models have a high number of specific data (vendor specific, mobile/fixed specific), an additional network abstraction layer (mediation Layer) is necessary to build a uniform management view of the network.
19. The FIM shall enable the modelling of both the connection oriented technologies and connectionless technologies.
20. All functionalities in the areas of fault management, performance management, configuration management (incl. resource provisioning and service configuration & activation) and inventory management which are common to wireline and wireless management interfaces have to be consolidated in one harmonised federated model.
21. The static data models from wireline (e.g. MTOSI) and wireless (e.g. 3GPP) technologies have to be harmonised.
It is acceptable to have wireline and wireless specific parts but these parts shall as much as possible be based on a common overarching model.
22. The federated model shall offer the necessary data and operations for all domains such as **Operations Support & Readiness (OS&R)** which includes inventory management, fulfilment and assurance.
23. The FIM shall contain general attributes for network elements (eg. id, userLabel, hwVersion, swVersion, gpsInfo, userDefinedState, etc).
24. Network resources (managed objects) shall be named using a harmonised naming convention. The naming convention must uniquely identify the network resources.
25. 1:1 Relation between **Event Managed Object Instances** and **Inventory Managed Object Instances**
If **Managed Object (MO)** Identifiers used/provided by the inventory component of an element manager need to be mapped to meet naming requirements of the inventory database, the same mapping must be applied to the MO identifiers in the event. The corresponding is true if mapping is driven by event naming requirements.
If MO identifiers of events and inventory within an element manager are different, the difference must be eliminated before the above mapping can be applied.
Rationale:
An event must be unambiguously related to a known object instance (in the inventory).

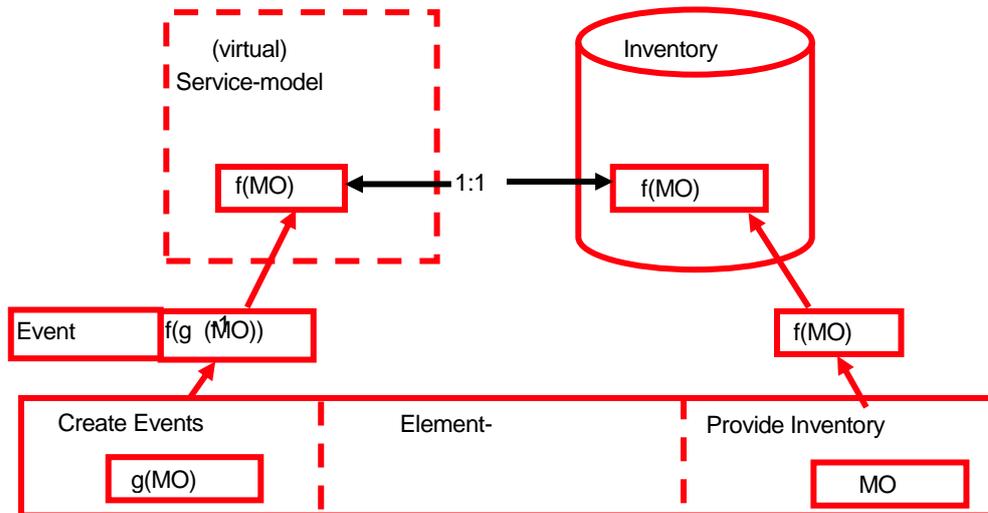


Figure 20: Event / Inventory relation

26. “Managed Object Instance” attribute information structure for EMS \leftrightarrow NMS event interfaces

The information in the “managed object” attribute of the Interface must allow a clear and unambiguous identification of the component (HW or SW), which is the originator of the Event.

- The managed object, as an attribute of the basic generic event object, shall not contain any detailed topology information. The assumption is, that the NMS will use an inventory database (internal or external) to map between managed object instance and inventory topology tree if needed.

- The basic assumption for this is, that there is a one-to-one mapping between managed object instance and the inventory information, so that the instance can be unambiguously identified. If this is not the case, the instance must contain a very simple and standardized methodology to describe the relationship between the first unambiguously identifiable object and the related not-unambiguously identifiable object, which is the originator of the event.

- NMs requirement (specific for the NMS layer): As soon as the event information leaves the area of the local network and the managed object attribute value does not deliver unambiguously any more, the network manager will add additional information, the “NameSpace” - string to the Managed_Object_Identifier attribute (Proposal: Company_Name + Technology-Domain \rightarrow “Access”), so that it is unambiguous in the larger context again. (Remark: the name of the EMS should be part of the “additional information” attribute, and not part of the MO_ID).

- Here the general proposed structure of the “Managed Object Instance” attribute:

Managed Object Instance ::= <NameSpace.>*<MO_Name> <;MO_Detail>*

- NameSpace::=<Global IdentifierString> (see NMS Requirement above)
- MO_Name ::= <Ressource_Name>|<Inventory_Name>
- The Ressource_Name is delivered by the Ressource or the EMS itself. This name might be enriched or normalized on EMS or NMS layer with some information from Inventory systems, e.g. topological Information.

Example:

Inventory_Name::=<Hostname>|<Service>|<Serviceelement>|<ResourceGroup>|<UseCase>|<UseCaseS ubtype>| ...

• MO_Detail ::=<Blocknn>|<Racknn>|<Slotnn>|<Portnn>|<IP_address>|...

The MO_Detail information is delivered by the Ressource or the EMS itself. It adds information about the

detailed origin of the alarm as far as this is known by the resource or the EMS. There is no limit on the number of topological elements, but it should be limited to an absolute minimum, just to the number which is really necessary to unambiguously identify the defective component.

A semicolon is used as a delimiter between the structural components of the managed object instance.

27. The federated model shall provide the static (read only attributes) and dynamic (create/delete/modify objects; modify attributes).
28. The federated model shall provide a common identification mechanism (format) of entities.
29. The federated model shall enable the correlation of the management information
 - between different layers and technologies in fixed networks (eg, WDM, SDH/SONET, ATM, IP/MPLS)
 - in fixed and mobile networks (eg, IP/MPLS <-> RAN, WDM <-> core network)
 - from different components in mobile networks (RAN, core network, etc.)
 - from different mobile network technologies (eg, WiMAX, WLAN, LTE, UMTS, etc.).

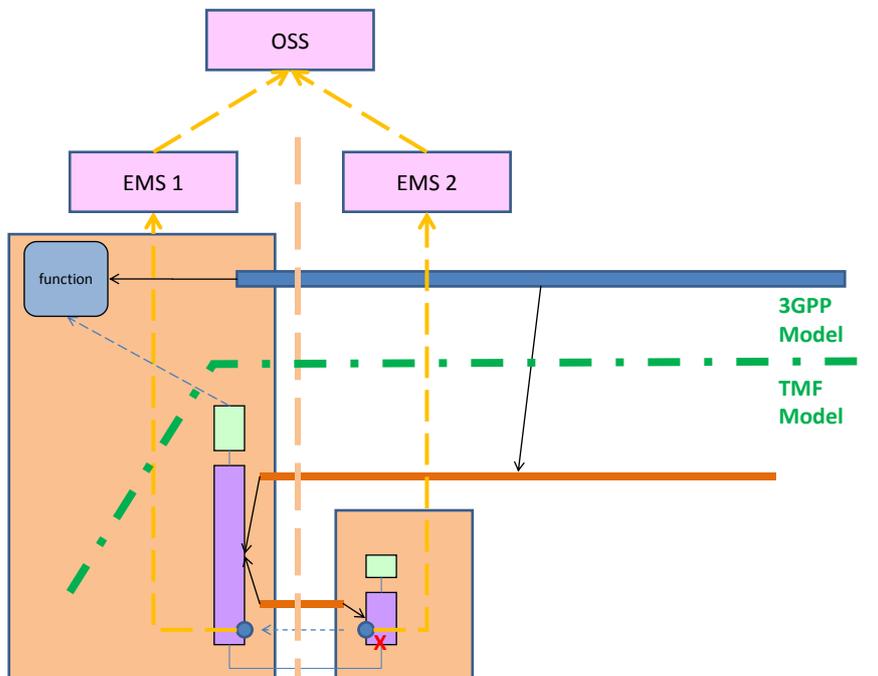


Figure 21: Example OSS receives the alarms from different EMS and different models (Mobile Network model from 3GPP model and Fix Network model from TMF model)

(Figure extracted from [7])

Editor's note:

Based on the result of the RAM catalyst during TM Forum MW in Dublin (May 2011) we will update this requirement.

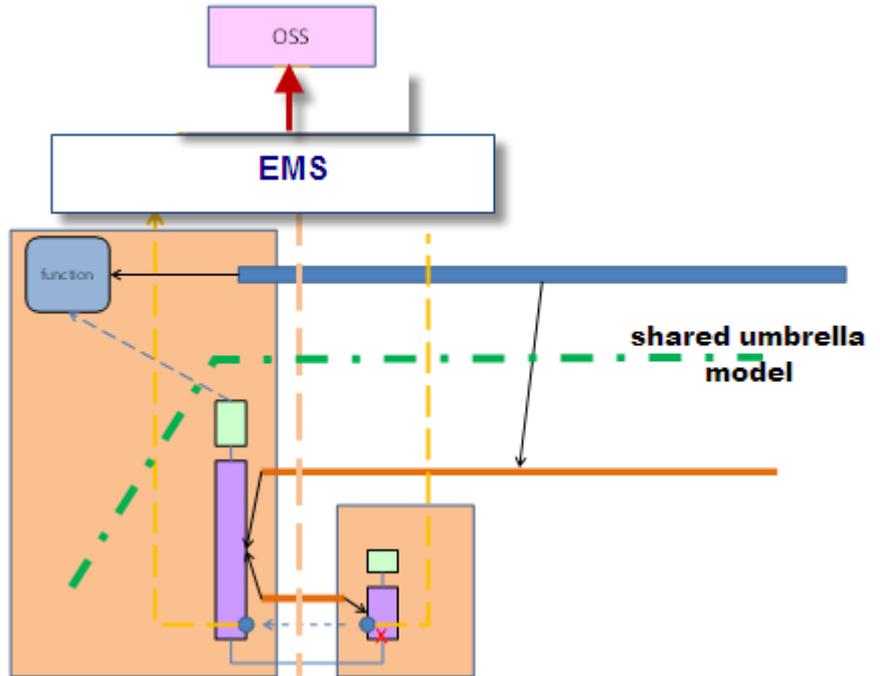


Figure 22: Example OSS receives the alarms of one EMS and a one data model FNM – model

30. The federated model shall enable the management of control plane (wrt resource provisioning).
31. The federated model shall enable the management of physical and logical resources (object / attribute) for fixed and mobile networks.
32. The federated model shall enable the management of physical and logical links.
33. The federated model shall allow the management of left and right side (or side A and side B) of links together and separated.
34. The federated model shall allow the realisation of important capabilities in performance management systems (PM) for fixed and mobile networks eg, BER (bit error ratio), QoS, busy hour, average value, max value, min value etc.
35. The federated model shall allow the realisation of major capabilities in Configuration Management Systems (CM), such as
 - export/import of the configuration to external systems (OSS)
 - download of the target configuration to the EMS
 - upload of the live configuration from EMS
 - data processing e.g., create/delete/read/modify of management objects / attributes
 - data quality e.g., consistency check, verification/check of resource availability
 - mass upload.
36. The SDOs shall specify the federated model in a protocol neutral way using UML.
37. The federated model shall be governed by all participating SDOs via a dedicated cross-SDOs structure.



38. The federated model shall be machine readable.
39. The federated model shall be delivered "run-time implementation technology neutral" (GDMO, UML, XML/XSD, WS/WSDL, CORBA/IDL etc).
40. The federated model shall also be delivered in portable document format (PDF).
41. The modelling of the SDO-specific enhancements shall be based on the federated model and should not exceed 20% of the total data model.
42. Traceability between model and requirements/use cases shall be provided in two ways:
 1. Where appropriate, a UML artefact should reference the corresponding requirement and/or use case identifier in the documentation field.
 2. Traceability matrices shall be provided for:
 - mapping from object classes to requirements
 - mapping from object class attributes to requirements
 - mapping from object class operations to requirements
 - mapping from object class operations to use cases
 - mapping from use cases to requirements.
43. M : N Connectivity

Multiple NMS applications might be connected (logically) to several EMS applications (M : N)
The API specification must allow to connect one NMS to multiple EMS. (This might have an impact on addressing – mechanisms in the API).
Furthermore the API specification must allow splitting the incoming event/alarm traffic between different instances of the same API implementations to avoid overload situations in one API instance.

Rationale:
This capability allows reducing the effort for the maintenance of several different client-side interfaces.
44. The federated model shall cover network resources with dimensions of "physical resources", "logical resources" and "compound resources".
45. The federated model shall provide the relationship of network resources from different networks (e.g., wireless network, core network, transmission network, IP network, switching network, etc.), such as correlation of wireless network resource and transmission network resource can be easily learned.
46. The federated model shall support to provide the uniform view of resources from different networks, such as end-to-end topology of network resources.
47. The federated model shall be used as an equipment information template, since it is useful to implement large quantities of network equipment instances. An equipment information template can provide information rules of verification and constraints for card/bay/slot/rack, thereby it shall improve the data accuracy and quality of the stock of equipment resources to support network resource lifecycle management.

4.5.1.3 Model Artefact Property Requirements

This chapter defines the requirements for the properties of the model artefacts:

- managed object classes

- attributes
- service interfaces
- operations
- parameters
- notifications
- data types
- relationships between managed object classes
- UML diagrams.

Editor's notes:

Requirements for mandatory, optional and conditional qualifiers (as defined in ITU-T M.3020) may need to be added to object classes, attributes, association ends, interfaces, operations, operation parameters and notifications. Requirements for extension mechanisms may need to be added. The definition of the multiplicity in the meta model may be too restrictive.

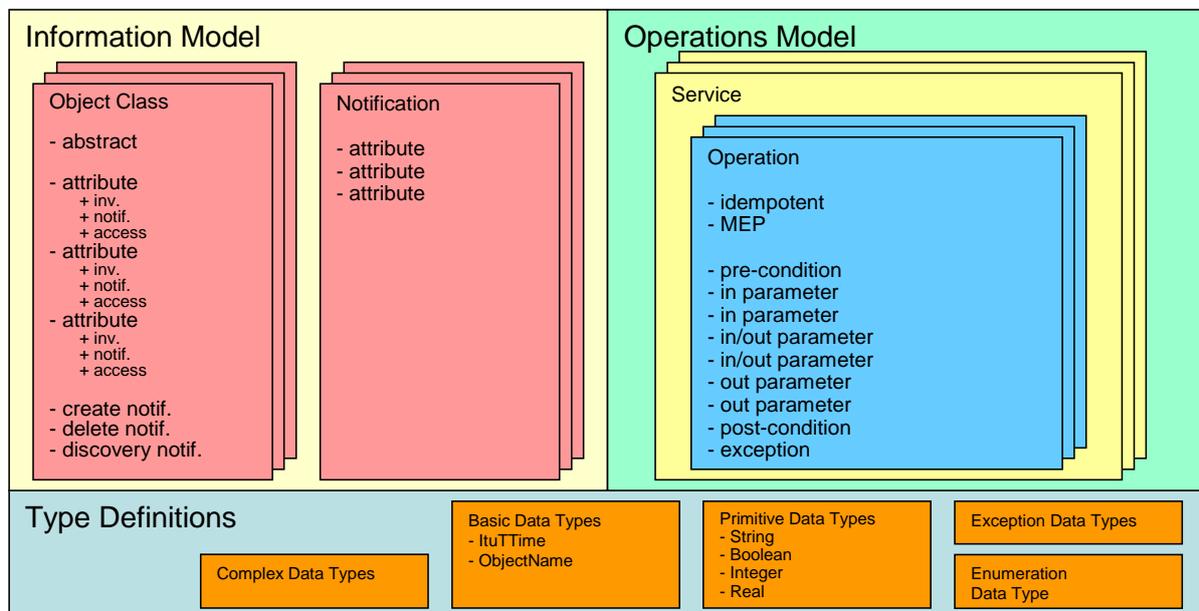


Figure 23: Model Artefacts

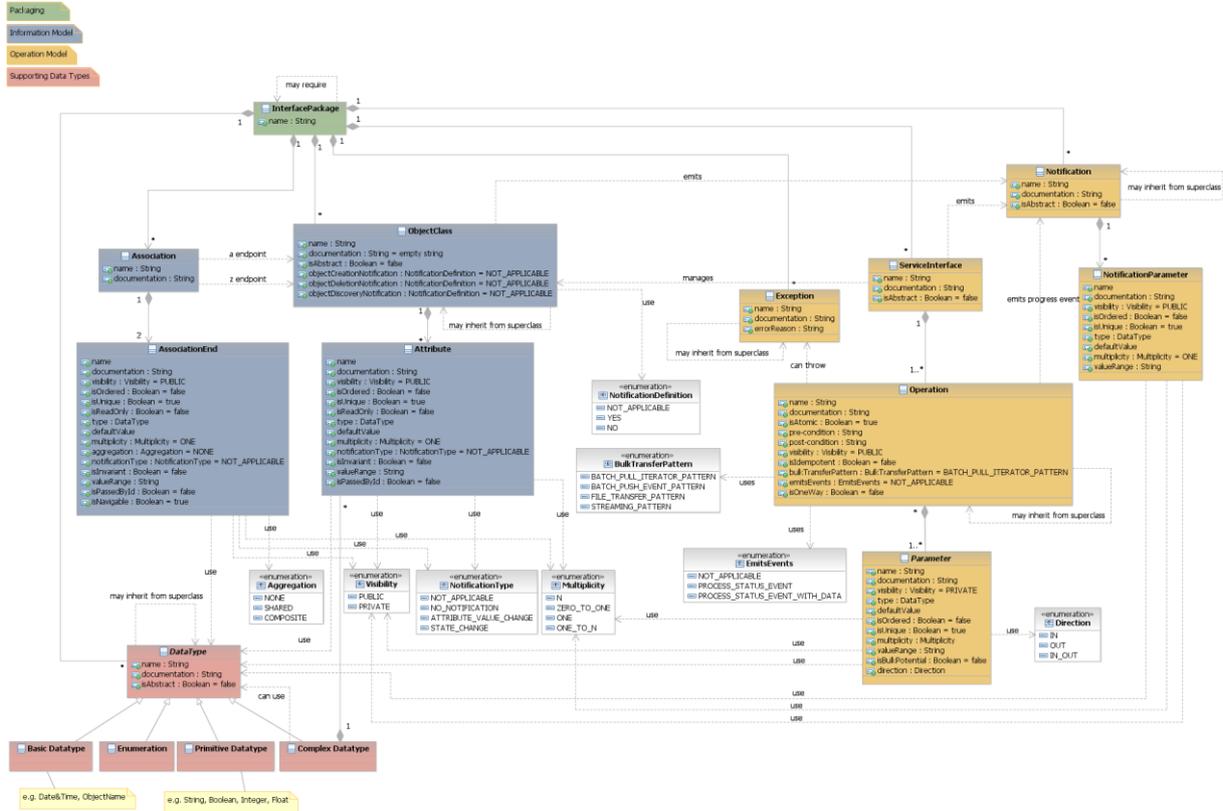


Figure 24: Meta Model

4.5.1.3.1 Object Class Requirements

Object classes are used to model data entities in the Information model and shall be derived from the static requirements.

48. An object class shall have the following properties:

- Object Class name
Shall follow Upper CamelCase (UCC).
The complete Distinguished Name (DN) having this name as a component must be unique across an interface instance.
- Object Class description
Shall contain a short summary of usage.
Shall refer to the appropriate requirement.
- Superclass(es)
Inheritance and multiple inheritance may be used.
- Abstract Object Class
Indicates if the object class can be instantiated or is just used for inheritance.
- Required Object Notifications
Shall identify if creation/deletion notifications are to be send
"objectCreationNotification" <NO | YES | NOT_APPLICABLE>
"objectDeletionNotification" <NO | YES | NOT_APPLICABLE>
"objectDiscoveryNotification" <NO | YES | NOT_APPLICABLE>.



49. An attribute within an object class shall have the following properties:

- **Attribute name**
Shall follow Lower CamelCase (LCC).
 - Boolean typed attribute names shall always start with a verb like 'is', 'must', etc. (e.g., 'isAbstract') and the whole attribute name must be composed in a way that it is possible to answer it by "true" or "false".
 - Enumeration typed attributes always end with "Kind" (e.g., 'aggregationKind').
 - List typed attributes shall end with the word "List".
 - Attributes referencing an instance identifier shall contain the word "Ref".
- **Attribute description**
Shall contain a short summary of usage.
Shall refer to the specific requirement (if defined).
- **Visibility**
Public: if the attribute shall be inherited to subclasses; public should be used in general within the model.
Private: if the attribute shall not be inherited to subclasses.
- **Qualifiers**
 - **Ordered**
For a multi-valued multiplicity; this specifies whether the values in an instantiation of this attribute are sequentially ordered; default is false.
 - **Unique**
For a multi-valued multiplicity, this specifies whether the values in an instantiation of this attribute are unique (i.e., no duplicate attribute values are allowed); default is true.

Excerpt from UML Superstructure Specification, v2.1.1: *When isUnique is true (the default) the collection of values may not contain duplicates. When isOrdered is true (false being the default) the collection of values is ordered. In combination these two allow the type of a property to represent a collection in the following way:*

Ordered	Unique	Collection type
false	true	Set
true	true	OrderedSet
false	false	Bag
true	false	Sequence

Table 4: Table 7.1 - Collection types for properties
from UML Superstructure Specification, v2.1.1

- **Read Only**
If true, the attribute may only be read, and not written by the requesting OS. The default value is false.
- **Type**
Refers to a basic or complex data type.
- **Default Value**
Provides the value that the attribute has to start with in case the value is not provided during creation or already defined because of a system state.
- **Multiplicity**
Defines the number of values the attribute can simultaneously have.
- **Aggregation**
An association may represent by a composite aggregation (i.e., a whole/part relationship). Only binary associations can be aggregations. Composite aggregation is a strong form of aggregation that requires a part instance be included in at most one composite at a time. If a composite is deleted, all of its parts are normally deleted with it. Note that a part can (where allowed) be removed from a composite before the composite is deleted, and thus not be deleted as part of the composite.



Compositions may be linked in a directed acyclic graph with transitive deletion characteristics; that is, deleting an element in one part of the graph will also result in the deletion of all elements of the sub graph below that element. Composition is represented by the isComposite attribute on the part end of the association being set to true.

- None
The affect on the attribute is unspecified when the parent is deleted.
- Shared
The attribute is not deleted when its parent is deleted.
- Composite
The attribute is deleted when its parent is deleted.
- Invariant
Identifies if the value of the attribute can be changed after it has been created; default value is "False".
- Value Range
Identifies the allowed values the attribute can have.
- Attribute Notifications
Identifies if a notification has to be sent in case of a value change.

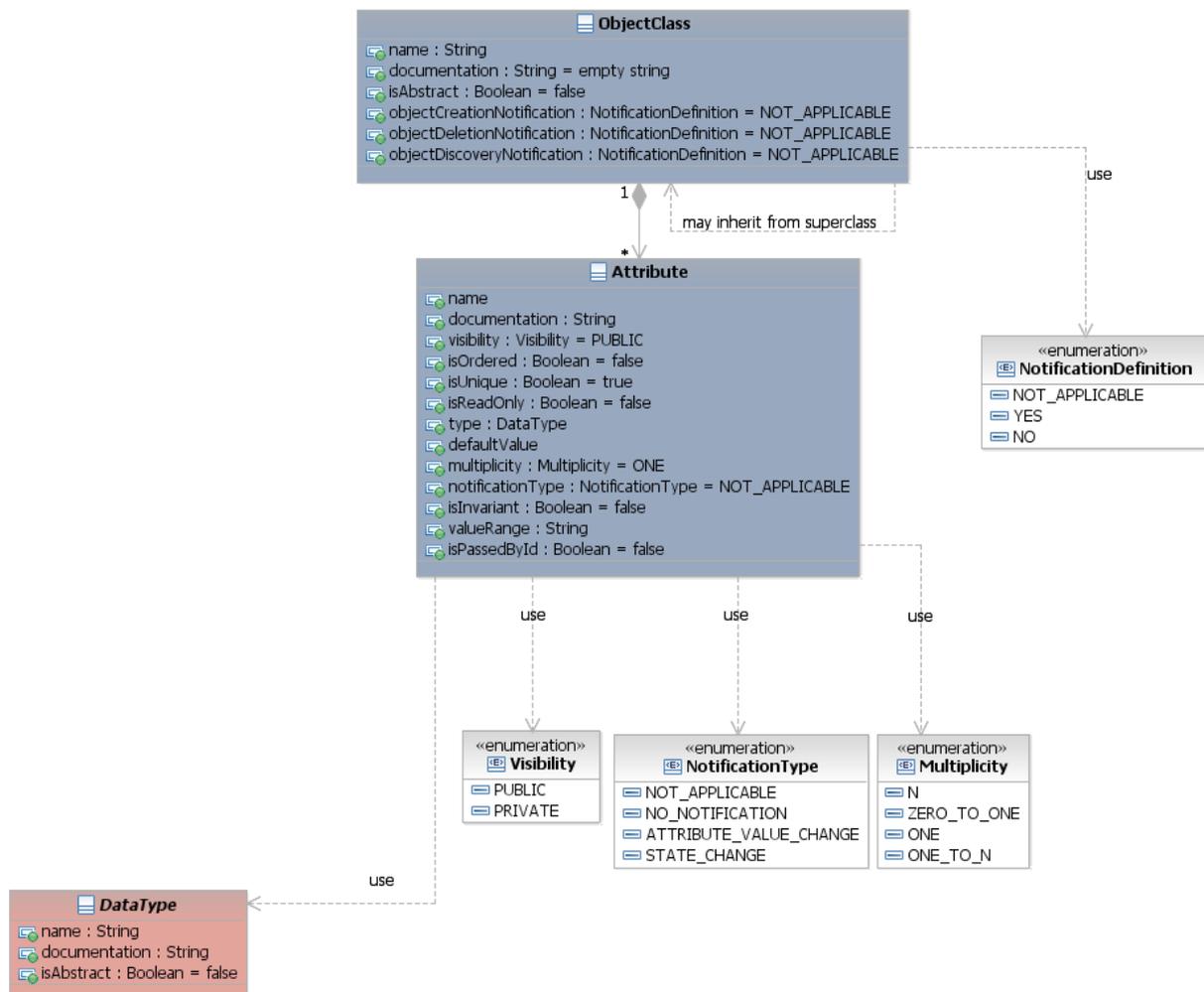


Figure 25: Meta Model: Object Class

4.5.1.3.2 Service Interface Requirements

50. Interface object classes shall be used to model the interfaces in the operations model and shall be derived from the dynamic requirements.
51. A service interface shall have the following properties:
- Service interface name
Shall follow Upper CamelCase (LCC).
Shall be expanded by the word "Service"
 - Service interface description
Shall contain a short summary of usage.
Shall refer to the specific requirement (if defined).



Figure 26: Meta Model: Service Interface

4.5.1.3.3 Operation Requirements

52. Operations shall be grouped in interface object classes and shall be derived from the dynamic requirements and use cases.
53. An operation shall have the following properties:
- Operation name
Shall follow Lower CamelCase (LCC).
 - Atomic
Identifies if the operation is best effort or is successful/not successful as a whole.
 - Visibility
Public: if the operation shall be inherited to subclasses.
Private: if the operation shall not be inherited to subclasses = default value.
 - Return Type
Shall be fixed to "void".
 - Operation description
Shall contain a short summary of usage.
 - Pre-condition(s)
Shall list the conditions that have to be true before the operation can be started (i.e., if not true, the operation will not start at all).
Note: It is also possible to define the pre-condition in OCL.
 - Parameter(s)
Refer to specific requirement [below](#).
 - Post-condition(s)
Shall describe the state of the system after the operation has been successfully executed.
 - Idempotency
Defines if the operation is idempotent or not.
 - Message Exchange Pattern (MEP)
The MEP fully identifies the messages and the choreography (sequencing and cardinality) of the



messages independently from a business activity.

The following distinct communication patterns are required:

- Simple Response
- Multiple Batch Response
- Bulk Response (e.g. file transfer)
- Notification.
- The following MEPs are required:
 - Synchronous Request/Reply (SRR) and Asynchronous Request/Reply (ARR) – Message (SRM) are used for requests that have a single response.
 - Synchronous Iterator (SIT) – this MEP allows for a synchronous (i.e., RPC style) request for an iterator.
 - Asynchronous Batch Response (ABR) – this MEP allows for an asynchronous (i.e., message style) request for a multiple batch response.
 - Synchronous (File) Bulk (SFB) – this MEP allows for a synchronous (i.e., RPC style) request for inventory to be returned in a file. The file is delivered via an out-of-band method (i.e., not using the CCV).
 - Asynchronous (File) Bulk (AFB) - this MEP allows for an asynchronous (i.e., message style) request for inventory to be returned in a file. The file is delivered via an out-of-band method.
 - Synchronous Notification (SN) and Asynchronous Notification (AN) – these MEPs facilitate the dissemination of notifications.
- Operation Exceptions
The allowed exceptions together with a failure reason shall be defined for each operation.

54. The following list of common exceptions shall be supported by the operations:

- AlreadyInPostCondition
This exception can be used by operations which are not defined as idempotent. It is used to indicate that the target OS is already in the post-condition.
- AtomicTransactionFailure
This exception shall be raised when an atomic operation is not successful due to a failure of one of its sub-parts. The failure reason shall indicate which object/part failed.
- CapacityExceeded
This exception shall be raised when the request will result in resources being created or activated beyond the capacity supported by the NE or target OS.
- Duplicate
This exception shall be raised if an entity cannot be created because an object with the same identifier/name already exists.
- EntityNotFound
This exception shall be raised when the specified object does not exist.
- FilterNotSupported
This exception shall be raised when a filter definition is not supported by the implemented filter. The failure reason shall indicate the more precise reason.
- InventoryOutOfSync
This exception shall be raised when the operation fails because the inventory data bases from the target and requesting OS are out of sync.
- NotInValidState
This exception shall be raised when the state of the specified object is such that the target OS cannot perform the operation.
- ObjectInUse
This exception shall be raised when the object identified in the request is currently in use.
- UnableToNotify
This exception shall be raised when the target OS is unable to connect to the Notification Service.



- **CommunicationLoss**
This exception shall be raised when the target OS is unable to communicate with the subordinate OS.
- **InternalError**
This exception shall be raised when the request has resulted in an OS internal error.
- **NotImplemented**
This exception shall be raised when the target OS does not support this operation.
- **UnableToComply**
This exception shall be raised when the target OS cannot respond to the request.
- **AccessDenied**
This exception shall be raised when the requesting OS is not permitted to perform the operation.
- **InvalidInput**
This exception shall be raised when the operation contains an input parameter that is syntactically incorrect or identifies an object of the wrong type or is out of range.

55. The following common exceptions shall be supported by all operations:

- AccessDenied
- CommunicationLoss
- InternalError
- InvalidInput
- NotImplemented
- UnableToComply.



Figure 27: Meta Model: Operation

4.5.1.3.4 Operation Parameter Requirements

56. Each parameter within an operation shall have the following properties:

- **Parameter name**
Shall follow Lower CamelCase (LCC).
- **Visibility**
Public: if the parameter shall be inherited to subclasses.
Private: if the parameter shall not be inherited to subclasses = default value.
- **Type**
Shall refer to a basic or complex data type.
Note: A list of input (in a few cases also output) parameters could also be combined in a data type.
- **Direction**
In | InOut | Out.
- **Default Value**
Provides the value that the parameter has to start with in case the value is not provided.
- **Ordered**
For a multi-valued parameter; the order of the values is important.
- **Unique**
For a multi-valued parameter, no duplicate values are allowed.
- **Multiplicity**
Defines the number of values the parameter can simultaneously have.
- **Value Range**
Identifies the allowed values the attribute can have.
- **Parameter description**
Contains a short summary of usage.
- **Bulk Potential**
Indicates that this parameter can potentially carry a very large amount of data which will require a bulk data transfer pattern.

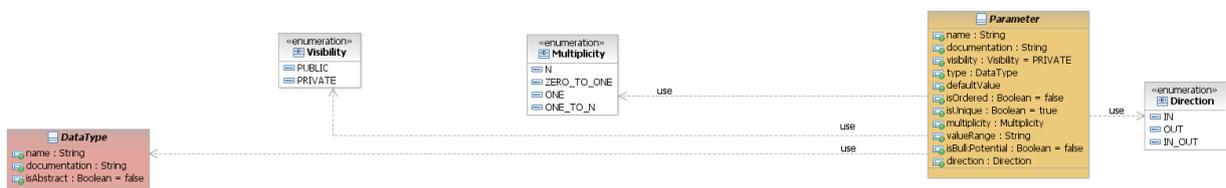


Figure 28: Meta Model: Operation Parameter

4.5.1.3.5 Notification Requirements

57. Object classes shall be used to model the notifications in the Information model.

58. Notifications shall have the following properties:

- **Notification name**
Shall follow Upper CamelCase (UCC).
Shall end with the word "Notification" (e.g., EquipmentProtectionSwitchNotification).
- **Notification description**
Contains a short summary of usage.
Refers to the appropriate requirement.
- **Superclass(es)**
Inheritance and multiple inheritance may be used.
- **Abstract Object Class**
Indicates if the notification can be instantiated or is just used for inheritance.

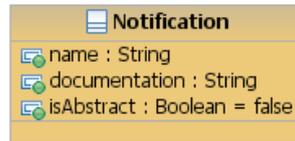


Figure 29: Meta Model: Notification

4.5.1.3.6 Notification Parameter Requirements

The information which has to be provided by a notification is contained in the notification parameters which are modelled as attributes of Notification object classes.

59. Notification Parameters shall have the following properties:

- Parameter name
 Shall follow Lower CamelCase (LCC).
 Shall follow the naming conventions defined for the object class attribute names defined in chapter 4.5.1.3.1.
- Parameter description
 Contains a short summary of usage.
 Refers to the specific requirement; if defined.
- Type
 Refers to a basic or complex data type.

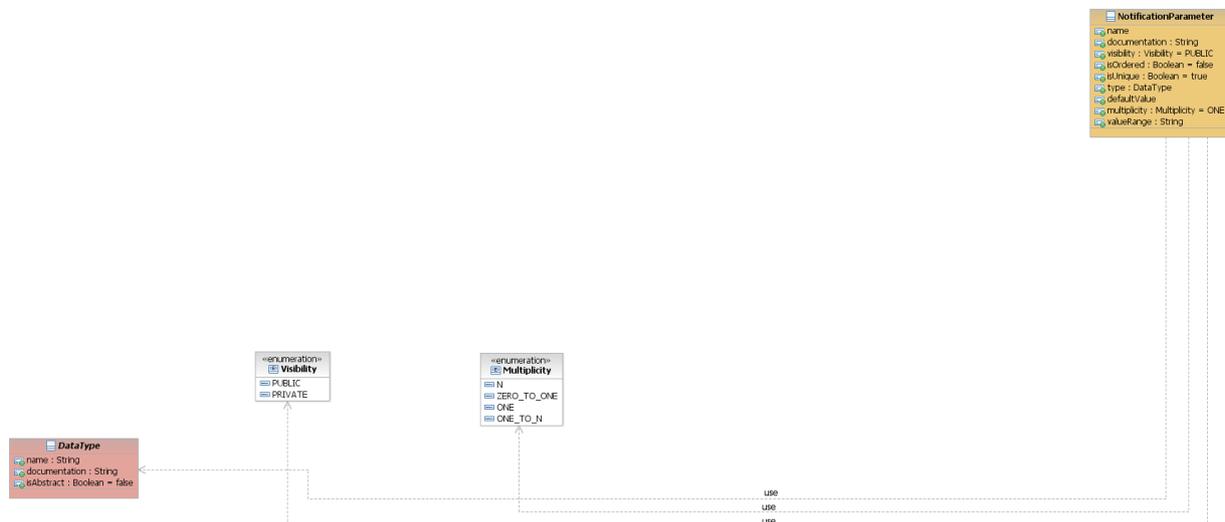


Figure 30: Meta Model: Notification Parameter

4.5.1.3.7 Data Type Requirements

Data Types are distinguish between "basic" and "complex" data types.

60. The following Basic Data Types shall be used:



- Boolean
- Integer
- Float
- Double
- String
- ObjectName

The ObjectName has to be used for the unique, read-only name of an object. The exact type is protocol specific.

- DateAndTime

"yyyyMMddhhmmss.s[Z{+|-}HHMm]" where:

yyyy	"0000".."9999"	year
MM	"01".."12"	month
dd	"01".."31"	day
hh	"00".."23"	hour
mm	"00".."59"	minute
ss	"00".."59"	second
s	".0".."9"	tenth of second (set to ".0" if EMS or ME cannot support this granularity)
Z	"Z"	indicates UTC (rather than local time)
{+ -}	"+" or "-"	delta from UTC
HH	"00".."23"	time zone difference in hours
Mm	"00".."59"	time zone difference in minutes.

61. Complex Data Types shall have the following properties:

- Data type name
Shall follow Upper CamelCase (UCC).
- Data type description
Shall contain a short summary of usage.
Shall refer to the appropriate requirement.
- Attributes within data types
 - Name
Shall follow Lower CamelCase (LCC).
 - Type
Shall refer to a basic or complex data type.
 - Default Value
 - Multiplicity

62. Enumeration "value" names of data types shall have only upper case characters; words are separated by "_".

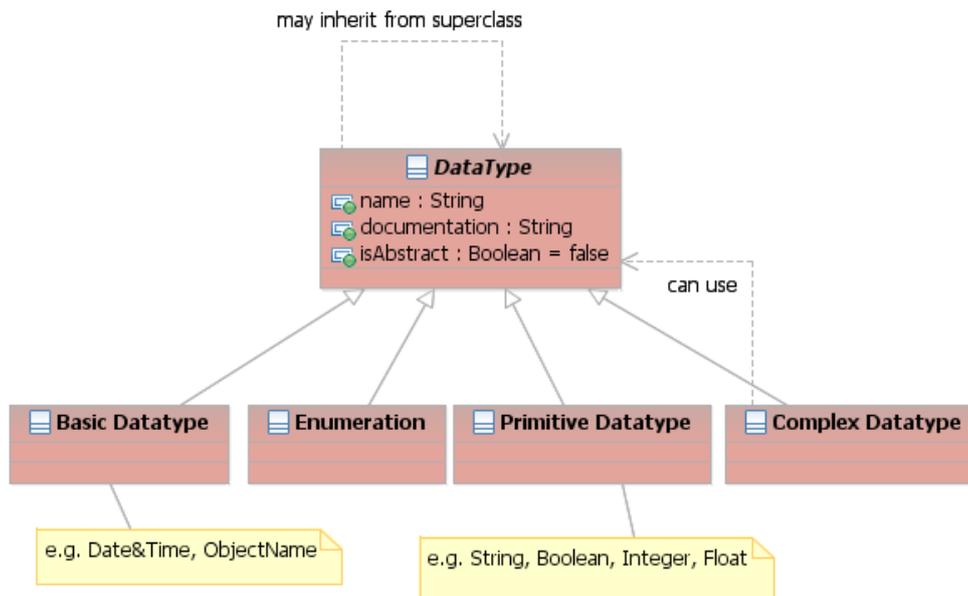


Figure 31: Meta Model: Data Type

4.5.1.3.8 Association Requirements

63. Associations shall have the following properties:

- Association name
Shall follow Upper CamelCase (UCC).
Format: "<ClassName><VerbPhrase><ClassName>" where the verb phrase creates a sequence that is readable and meaningful (e.g., SubnetworkConnectionIsTerminatedByTerminationPoint).
Must be unique across all association names defined in the whole model.
- Association description
Shall contain a short summary of usage.
Shall refer to the appropriate requirement.
- Stereotype
E.g., <<naming>> shall be used if the association defines the object naming tree.
- Association Type
E.g., inheritance, association (composition, aggregation, and association class), dependency, and realisation.
- Role names
Identifies the role that the object plays at this end of the relationship.
Shall follow Lower CamelCase (LCC).
Shall follow the naming conventions defined for the object class attribute names defined in chapter 4.5.1.3.1.
Note: Only navigable relationships have role names.
- Constraint(s)
List the constraint(s) under which the association can exist.
- Abstract
It is recommended to create associations which are just for explanation to the reader of the model.
These associations should be defined as "abstract", they are not navigable and have no role names.
They shall not be taken into account in the protocol specific specification. This can for example be used to show the association to the object which is retrieved by a get-operation.

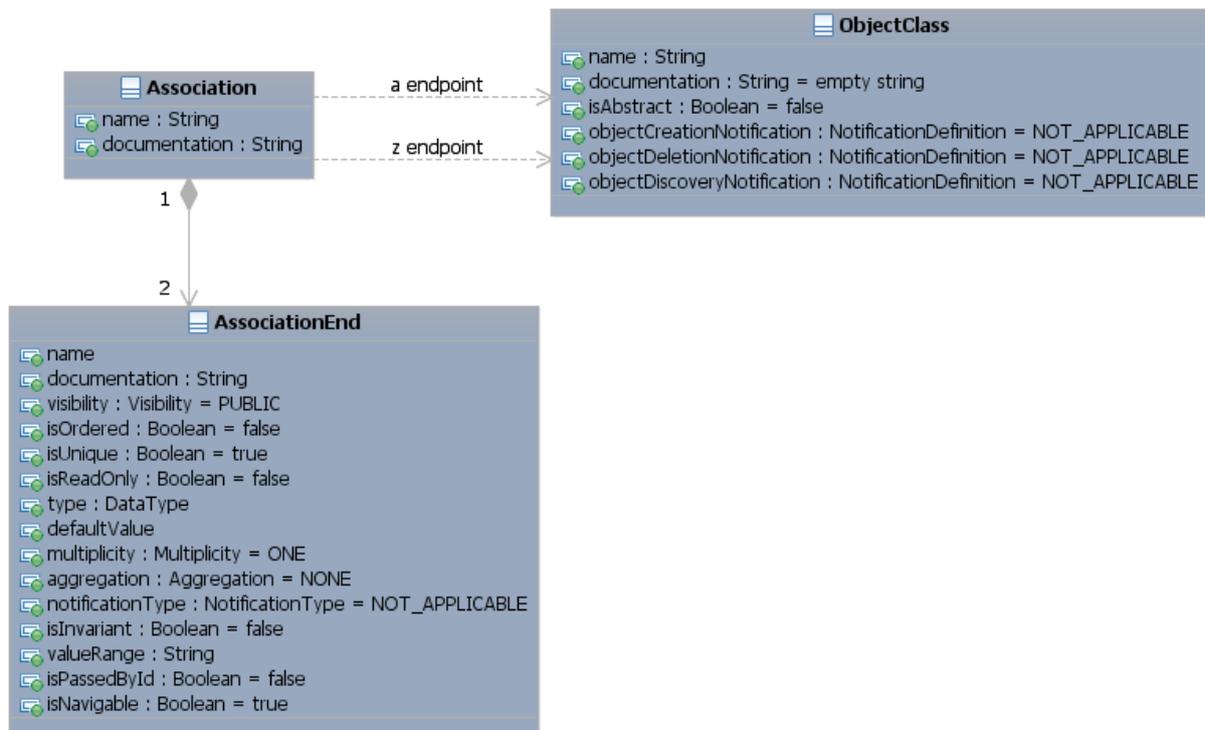


Figure 32: Meta Model: Association

64. An association end shall have the following properties:

- Name
 - Shall follow Lower CamelCase (LCC).
 - Boolean typed association end names shall always start with a verb like 'is', 'must', etc. (e.g., 'isAbstract') and the whole association end name must be composed in a way that it is possible to answer it by "true" or "false".
 - Enumeration typed association end always end with "Kind" (e.g., 'aggregationKind').
 - List typed association ends shall end with the word "List".
 - Association ends referencing an instance identifier shall contain the word "Ref".
- Description
 - Shall contain a short summary of usage.
 - Shall refer to the specific requirement (if defined).
- Navigable
 - Navigable association ends will lead to an attribute in the remote object class. At least one end of an association should be navigable.
- Visibility
 - Public: If the association end shall be inherited to subclasses; public should be used in general within the model.
 - Private: if the association end shall not be inherited to subclasses.
- Qualifiers
 - Ordered
 - For a multi-valued multiplicity; this specifies whether the values in an instantiation of this association end are sequentially ordered; default is false.
 - Unique
 - For a multi-valued multiplicity, this specifies whether the values in an instantiation of this



association end are unique (i.e., no duplicate association end values are allowed); default is true.

Excerpt from UML Superstructure Specification, v2.1.1: *When isUnique is true (the default) the collection of values may not contain duplicates. When isOrdered is true (false being the default) the collection of values is ordered. In combination these two allow the type of a property to represent a collection in the following way:*

Ordered	Unique	Collection type
false	true	Set
true	true	OrderedSet
false	false	Bag
true	false	Sequence

Table 5: Table 7.1 - Collection types for properties
from UML Superstructure Specification, v2.1.1

- Read Only
If true, the association end may only be read, and not written by the Requesting OS. The default value is false.
- Type
Refers to a basic or complex data type.
- Default Value
Provides the value that the association end has to start with in case the value is not provided during creation or already defined because of a system state.
- Multiplicity
Defines the number of values the association end can simultaneously have.
- Aggregation
An association may represent by a composite aggregation (i.e., a whole/part relationship). Only binary associations can be aggregations. Composite aggregation is a strong form of aggregation that requires a part instance be included in at most one composite at a time. If a composite is deleted, all of its parts are normally deleted with it. Note that a part can (where allowed) be removed from a composite before the composite is deleted, and thus not be deleted as part of the composite. Compositions may be linked in a directed acyclic graph with transitive deletion characteristics; that is, deleting an element in one part of the graph will also result in the deletion of all elements of the subgraph below that element. Composition is represented by the isComposite attribute on the part end of the association being set to true.
 - None
The affect on the association end is unspecified when the parent is deleted.
 - Shared
The association end is not deleted when its parent is deleted.
 - Composite
The association end is deleted when its parent is deleted.
- Invariant
Identifies if the value of the association end can be changed after it has been created; default value is "False".
- Value Range
Identifies the allowed values the association end can have.
- Notifications
Identifies if a notification has to be sent in case of a value change.

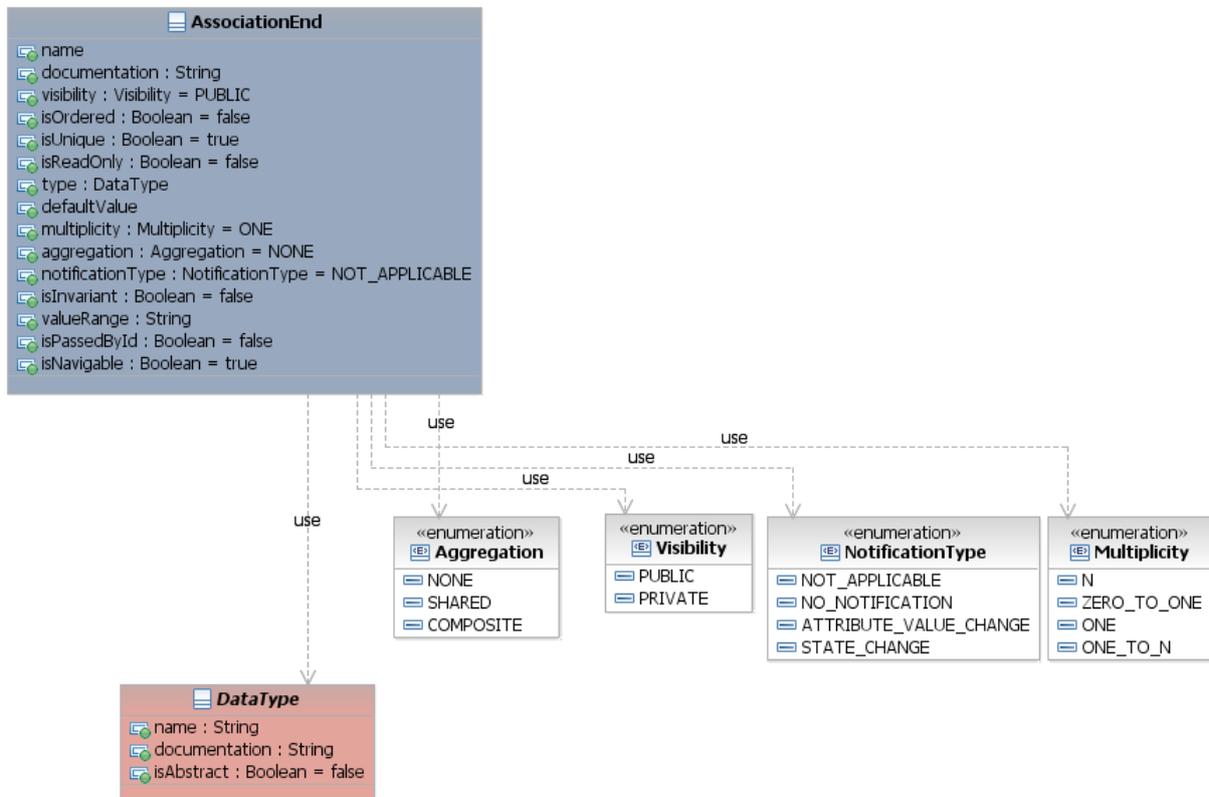


Figure 33: Meta Model: Association End

4.5.1.3.9 UML Diagram Requirements

65. Objects and their relationships shall be presented in class diagrams.
66. It is recommended to create
 - An overview class diagram containing all object classes related to a specific management area (Class Diagram).
 - An overview interface diagram containing all interfaces related to a specific management area (Interface Diagram).
 - A separate inheritance class diagram in case the overview diagram would be overloaded when showing the inheritance structure (Inheritance Class Diagram).
 - A class diagram containing the defined notifications (Notifications Diagram).
 - A class diagram containing the defined data types (Type Definitions Diagram).
 - Additional class diagrams shall be established to show specific parts of the DDP in detail.
 - State diagrams shall be created for complex state attributes.
 - Activity diagrams/Sequence Diagrams (possibly use cases) shall be created for complex operations.
 - The class name compartment shall contain the "Qualified Name"
 - The class attributes and operation shall show the "Signature".



4.5.1.4 Infrastructural Requirements

67. The SDOs shall agree on a list of common modelling patterns defined in a kind of meta-model.
68. The SDOs shall define a migration path which allows bringing the present individual models into the common federated model.
69. It shall be possible to use the federated model (and its SDO specific enhancements) as input to a tool based Interface development process.
70. The SDOs shall agree on a common UML version (e.g., 2.1).
71. The SDOs shall use open source modelling tools.



4.5.2 Tooling Requirements

4.5.2.1 General Requirements

72. The interfaces shall be based on high level business requirements.
73. Requirements shall be created for the static and dynamic parts of the interface.
74. The dynamic high level business requirements shall be converted into specific use cases.
75. The dynamic operation models from wireline and wireless technologies have to be harmonised. The harmonisation shall concentrate on:
 - Common Operations (basic operations for create/delete, modification and retrieval)
 - Common Exceptions
 - Common Notifications
 - Common Extendibility patterns
 - Common Message Exchange patterns
 - Common Scheduling mechanisms
 - Common Filter mechanisms.
76. The complete data and operation models shall be part of standardized specifications and made available in a machine readable format.
77. The interface specification shall be tool supported to significantly reduce the time to market for those who are specifying and implementing the interfaces.
78. The interface protocol specification shall be created automatically supported by a single software tool to ensure the usage of common design guidelines.
Using a single tool increases also the interoperability of the specified interfaces.
79. The tool shall be able to provide:
 - an XML based interface protocol specification (Web Services)
 - interface documentation
 - input for a reference implementation
 - input for a compliance and test tool kits
 - traceability mechanisms, e.g. between requirements and protocol neutral information model and between protocol neutral information model to protocol-specific parts.
80. The tool shall be developed outside of any specific standardisation body in an open source environment.
This allows the usage of the tool by other standardisation bodies.

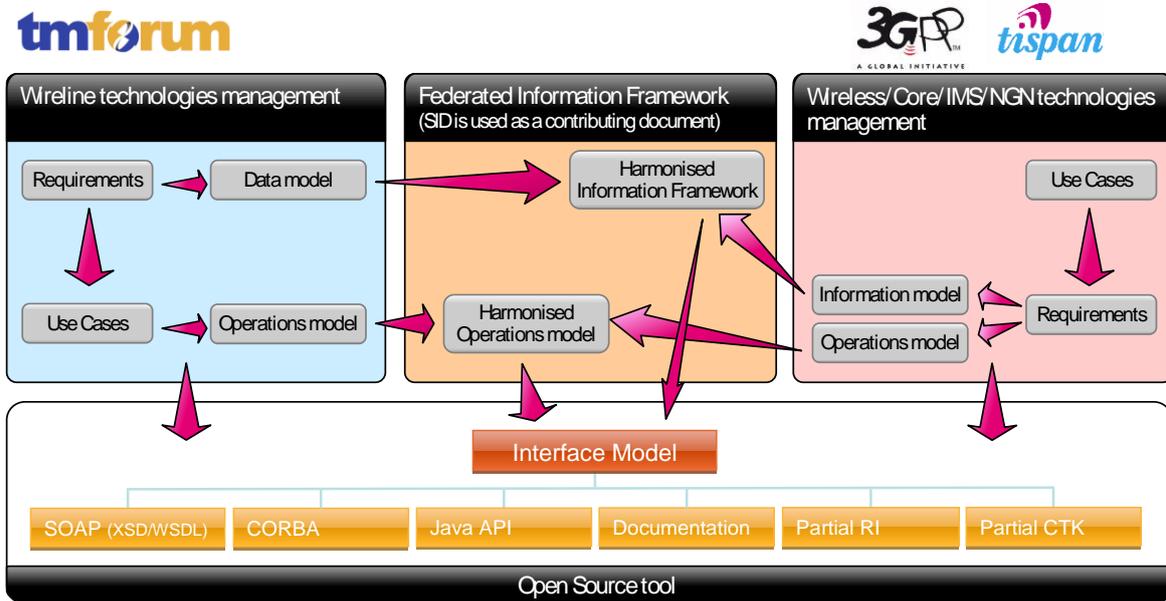


Figure 34: Modelling/Tooling Architecture

4.5.2.2 General Pattern Requirements

81. The tool shall provide general patterns to ensure a common basis for all interfaces.

4.5.2.2.1 Object Identifier Pattern

82. The tool shall add a globally unique object identifier to every object to uniquely identify the object across an interface.

83. The object identifier shall contain a context, a distinguished name and a type.

4.5.2.2.2 Common Exceptions Pattern

84. The tool shall provide two types of common exceptions: predefined common exceptions and optional common exceptions.

The predefined common exceptions shall be automatically inserted into all operations by the tool. The optional common exceptions shall be inserted into the operations by the tool on request.

85. All exceptions shall be able to provide a reason and a details description.

86. The following list of predefined common exceptions shall be automatically inserted into all operations by the tool:

- InternalException (default exception)
- AccessDenied
- CommunicationLoss



- InternalError
- InvalidInput
- NotImplemented
- UnableToComply

For a description of the exceptions see chapter 4.5.1.3.3.

87. The following list of predefined common exceptions shall be automatically inserted into all operations by the tool:

- AlreadyInPostCondition
- AtomicTransactionFailure
- CapacityExceeded
- Duplicate
- EntityNotFound
- FilterNotSupported
- InventoryOutOfSync
- NotInValidState
- ObjectInUse
- UnableToNotify

For a description of the exceptions see chapter 4.5.1.3.3.

4.5.2.2.3 Iterator Pattern

88. The tool shall support a common iterator pattern for bulk data transfer.

89. The iterator pattern shall contain the following functionality:

- IteratorInfo
This is the Info contained in the first reponse to a bulk based request.
- GetNextResponse
This is the response object to a getNextRequest.
- GetNextRequest
This is the Iterator getNextRequest to retrieve the next batch of replies.
- ReleaseRequest
This is the Iterator release request to release all the associated resources and invalidate the iterator.
- HasNext
Returns a Boolean; True meaning that additional data is available; false meaning that this is the last information.
- Remove
Deletes the information contained in the iterator.
- IsEmpty
Returns a Boolean; True meaning that iterator has no information; false meaning that the iterator contains still information.
- ReleaseResponse
- IteratorNotFound
- InvalidIteratorContext.

4.5.2.2.4 Notification Pattern

90. The tool shall support common notifications.



91. The following types of notifications shall be provided:

- AttributeValueChangeNotification
- ObjectCreationNotification
- ObjectDeletionNotification
- ObjectDiscoveryNotification.

92. All notifications shall at least provide:

- Object identifier
- Object type
- Source time.

4.5.2.2.5 Common Operations Pattern

93. The tool shall support common operations covering create, delete, set and get associated to a single interface class.

94. It shall be possible for the common **create** operation to define a reference object (existing instance of a managed object). The attribute values associated with the reference object instance shall become the default values for those not specified by the also provided create data attribute values.

95. The tool shall support the following types of **get** operations:

- Single object get
Getting the values of a single instance
- Multiple entities get
Get all entities matching a filter; returning the attributes and values of the entities
- Multiple entities get by ids
Get all entities matching a filter; returning only the identifiers of the entities.

96. The created object instances shall be returned.

97. It shall be possible to have all three types of get operations associated to the same interface class.

98. It shall be possible for the common **delete** operation to provide a list of object instances (object identifiers) to be deleted

99. The delete operation shall return the list of object instances that could not be deleted.

100. The tool shall support the following types of **set** operations:

- Single object set
Setting a single object; all attributes should be set in an atomic way.
- Multiple entities set, best effort
Setting all entities matching a filter in a best effort way.
- Multiple entities set, atomic
Setting all entities matching a filter in an atomic way.

4.5.2.2.6 Filter Pattern

101. The tool shall support a common filter construct (based on attribute values) for operations requiring the selection of object instances.



102. The filter construct shall be a template or a combination of a template and a query filter.

103. A query filter shall be mapped to a string which is implementation technology specific. For example in XML it is filled by the implementation with an XPATH expression. In Java it is filled by a JPA query expression.

104. A template filter shall be mapped to a sequence of attribute matching filters.



4.6 Appendix For Modelling and Tooling

The following figures show the containment/naming hierarchy and the associations of the classes defined in the Joint 3GPP/TMF model alignment project.

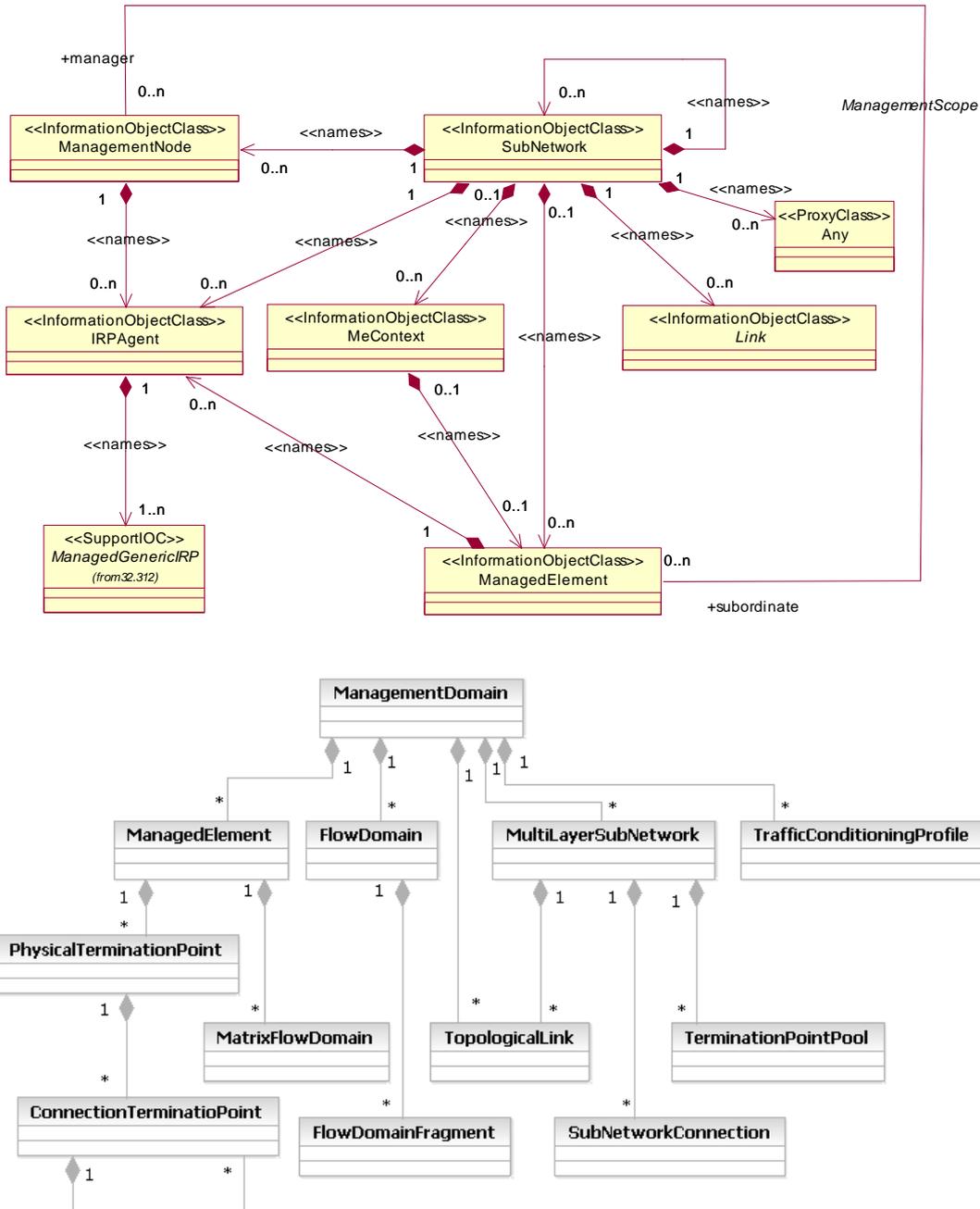


Figure 35: Modelling differences between 3GPP and TM Forum

(Top figure extracted from Figure 6.1: Generic NRM Containment/Naming and Association diagram (3GPP TS 32.622 [6])
 Bottom figure extracted from Figure LR.35 - MTOSI/MTNM Containment (TM Forum SID Rel. 9.5 [12])



4.7 References used in Modelling and Tooling

- [1] MTOSI 2.0: Network Resource Fulfillment DDP IA, TMF612_NRF, Version 1.0
- [2] ATM Forum, Technical Committee, Network Management, M4 Network View CMIP MIB Specification, CMIP Specification for the M4 Interface, Sep, 1995
- [3] 3GPP TS 32.xyz series on NRM
- [4] 3GPP TS 32.300 Telecommunication management; Configuration Management; Name convention for Managed Objects
- [5] S5-102610 S5vTMFa033 E NSN Proposed enhancement of Generic NRM IOCs v3
- [6] TS 32.622 Generic network resources IRP: NRM
- [7] 3GPP SA5-TM Forum model alignment JWG meeting in Budapest, April 4-6, 2011
- [8] TR166 - Federated Information Framework - Concepts and Principles - v0.1.docx
- [9] Fixed Mobile Convergence (FMC) Network Management – Federated Network model (FNM) Umbrella, Version 1.2 JWG meeting in Budapest, April 4-6, 2011
- [10] GB922, Information Framework (SID) Suite, Release 9.0
(<http://www.tmforum.org/browse.aspx?catID=9285&artf=artf2048>)
- [11] MTOSI 2.0 (<http://www.tmforum.org/MTOSIRelease20/MTOSISolutionSuite/35252/article.html>)
- [12] TM Forum Information Framework (SID) Suite; Release 9.5
(<http://www.tmforum.org/Guidebooks/GB922InformationFramework/45046/article.html>)
- [13] TM Forum MTOSI Rel. 2.1 supporting document SD2-5_Communication_Styles
- [14] JOSIF Guidebook
(http://sourceforge.net/apps/mediawiki/openoss/index.php?title=JOSIF_Guidebook)



5 REQUIREMENT SPECIFICATION FOR FAULT MANAGEMENT INTERFACE

5.1 Introduction for Fault Management

The authors of the FM section strongly believe, that there is potentially huge business benefit in using a common officially standardized technical approach, enabling the re-use of the same interface for different EMSs, enabling the planned exchange/upgrade of the NMS-FM system to become a **Next Generation Service Assurance** system, and enables us to stop vendor driven upgrades of interfaces which deliver no or small additional value. So, the FM interface “plug & play” concept, described in the FM section, will be used as a goal for next generation service assurance.

In today's market, service providers aim to ever increase the time-to-market of new and enhanced services in a cost-conscious manner. As a consequence, the need arises for existing OSS/BSS infrastructure components to adapt in an ever increasing pace. This affects not only OSS applications themselves, but also increasingly their integration. Furthermore, there is a growing demand for automation of business processes at service providers, especially in the area of network/service operations to improve operational efficiency. This leads to the need for improved integration of OSS as a common demand from service providers. An integration strategy using SOA concepts, commonly adopted interface standards and NGOSS concepts like eTOM and SID might have the potential to deliver the needed technical basis for real life, standardized OSS integrations.

In the past, Service providers often over-specified the tenders for FM interfaces and, on the other hand, opened to many degrees of freedom for the implementation of the interface. So they missed the opportunity to describe a simple, useable, maintainable interface, with a clear responsibility assignment between EM and NM.

Most of the existing integrations between EM systems and NM systems are based on proprietary point-to-point interfaces although vendors offer “standard” interfaces such as SNMP, CORBA, etc., which are adapted to their applications. In a real integration scenario these interfaces need a lot of customization to fulfil the business requirements and to allow the communication between different proprietary OSSs because each of these applications follow its own business process, internal logic and semantic. Usually application needs to know a part of the business logic of system B (and vice versa) to be able to implement the interface. This situation ends with the implementation of very specific interfaces with dependencies on the integrated OSS. This means, re-use of interfaces or dedicated parts of the interfaces in other integration scenarios is not possible. So, there is a need for a standardized interface, which delivers the semantic connectivity and not only the underlying transport mechanisms, which helps to provide out-of-the-box interoperability and more flexible integration.

See also chapter 8.1 “Abstract” from NGMN Top OPE Requirements Version 1.0:

“Although it is not the intention of the current document to specify implementation details, the operators expect the industry to jointly develop and use common standards, which deliver the semantic connectivity and not only the underlying transport mechanisms. The goal is to achieve out-of-the-box interoperability and more flexible integration, as well as the re-use of the same interfaces between OSS/BSS and the Network or EMS. Based on existing frameworks, provided by the standardization bodies, solutions should be implemented that support plug & play behaviour of network and OSS/BSS infrastructure. This will lead to more open interfaces to allow for 3rd party software integration. Amongst others this implies usage of common data models, e.g. based on SID, interface standards, such as SNMP and XML (if appropriate), and state-of-the-art technologies as SOA, web services, etc. As those standards are evolving over time, the operators resign from specifying exact software versions and implementation details. Our aim is to ensure upwards and downwards compatibility to ease integration of multi-vendor, multi-technology systems for all management areas.”

5.1.1 Objective

The objective of the FM section is to deliver the specification of the major requirements for a unified, re-useable fault management API for the alarm interface between EMS → NMS. The FM section will serve as an input for standardization activities which address the FM interface standard.



The FM interface requirements are generic for FMC. They are completely independent from the network/service type which will be monitored by the EMS. So the FM interface requirements are valid for wireless and wireline networks, as well as for IT systems or service platforms.

Please consider that the FM section contains only mandatory requirements to deliver a basic, simple and cost efficient FM Interface. Additional requirements might be added later on as “optional”. (All requirements are “mandatory”, as long as they are not explicitly marked as “optional”). These requirements may not harm the business goals of the basic, mandatory requirements to achieve a simple, cost efficient and easy to integrate FM interface. It must be possible to implement an interface, which contains functionalities in line with the optional requirements, in a mixed mode with the simple/basic interfaces, which contain the mandatory requirements in this section, without any change for simple/basic interface (e.g. the EMS delivers just the mandatory interface functionality and the NMS delivers also the optional part of the interface. In that case, the interface will use only the mandatory functionality, without any change on the EMS or NMS interface functionality).

5.1.2 Approach

It's the intention to describe the interface capabilities from business point of view, without technology specific requirements. That means, that these requirements reside on the semantically layer and not on protocol specifications. Nevertheless, there are some assumptions which might have an impact on the selected technology, e.g. the de-coupling of the interface specification (which is a basic requirement to support re-usability, exchange of SW versions, etc. ...) might have an impact of the technology. Furthermore the requirements have to be independent from the tool selection, so that they may not depend on specific tool capabilities.

5.1.3 Benefit and Drivers

The main benefit is achieved, as soon as the specification can be re-used to implement similar interfaces for different integration scenarios, to connect different EMS to NMS applications without creating a complete new implementation of the interface. The goal is to improve efficiency (in terms of cost and effort) for the integration of new EMS and to reduce cost and effort to maintain each single interface in a different way. Another benefit comes from the fact, that a real decoupled approach will reduce the effort to adapt both communication partners, in case there is a need to upgrade just one of the partners.

Saving potential

- The support for a better level of standardization of the itf-N will reduce the integration effort between EMS and NMS (OSS) during the implementation and the life cycle of network technologies and related EMS.

Possible issues for guidance:

- Plug & Play” integration of EMS into the OSS environment (no additional cost and effort during the implementation and the life cycle of network technologies and related EMS)
- De-coupling of EMS – OSS layers (changes on EMS or on NE may not lead to changes on OSS layer)
- Re-use of OSS client interfaces

5.1.4 Scope

The main scope is the specification of the business requirements and related semantics, which describes the interaction of element management systems to network management (umbrella fault management) systems to exchange event/alarm information. The interface requirements support converged networks, that means that wireless and wireline networks are in scope.

In addition to this, there are specific requirements for the EM systems and NM systems to use the capabilities of the specification in order to support the business requirements.



Please consider that different application topologies have to be supported by the interface:

- Several NMSs can be connected to the EMSs, e.g. operational NMS and test NMS
- An NMS can serve as an EMS (e.g. a technology domain specific NMS, which acts like an EMS to upper

5.2 Non-Functional Requirements

The following topics describe some core business driven requirements for the EMS → “alarm” → NMS interface, independent from functional requirements. These requirements are not specific for the FM Use Cases and can be used as core “non-functional” requirements for other types of interfaces as well.

Note: The detailed descriptions of these “Non-Functional Requirements” have been shifted into the Generic-Next-Generation-Converged-Operational-Requirements (GEN) section, because they are valid for most types of OSS interfaces.



5.3 Functional Requirements for Fault Management Interface

The functional requirements for the FM interface describe the mandatory and some optional requirements for the Fault Management API between EMS and NMS from an FM business point of view. The optional requirements are not intended to be complete, but mention some of the most likely needed optional features for the API. It does not define the functional capabilities needed on EMS or the NMS itself, although there are some requirements in this area's mentioned to serve as a "basic" information to understand the needed capabilities on system level (they can be used for EMS/NMS vendor selection processes).

Please consider: several functional requirements have been shifted into the generic requirements section, because they are valid for most types of OSS interfaces.

Examples listed here are:

- Trace and Logging
- "Managed Object Instance" Attribute Information Structure
- M : N Connectivity
- 1:1 Relation between Event Managed Object Instances and Inventory Managed Object Instances

5.3.1 X.733 Event/Alarm Attributes

The event/alarm must contain structured information according to the X.733 specification

Description:

- The attributes of the event/alarm object shall follow the X.733 standard definition (for details see X.733 specification in chapter Appendix)

Short overview of attributes:

- The yellow marked attributes are mandatory or the interface. So they have to contain a useable value. (The other attributes are optional in this specification. The interface and the connected systems must work in a proper way, if the optional attributes do not contain any value).

Parameter name	Req/Ind	Rsp/Conf
Invoke identifier	P	P
Mode	P	—
Managed object class	P	P
Managed object instance	P	P
Event type	M	C(=)
Event Subtype	M	—
Event time	P	—
Event information	—	—
Probable cause	M	—
Specific problems	U	—
Perceived severity	M	—
Backed-up status	U	—
Back-up object	C	—
Trend indication	U	—
Threshold information	C	—
Notification identifier	U	—
Correlated notifications	U	—
State change definition	U	—
Monitored attributes	U	—
Proposed repair actions	U	—
Additional text	U	—
Additional information	U	—
Current time	—	P
Event reply	—	—
Errors	—	P

→ Event Subtype, which is a substructure, is requested additional attribute to the X.733 specification. It depends on event type, and provides more detail information than event type. For example, event type is equipment alarm type, event subtype should be input and/or output equipment, processors etc. It is very useful for operators to locate the alarms and decide which professional team to do trouble shooting.

→ The Notification ID must be unambiguous to resolve the clear-problem and the synchronization problem (see specific requirements later on)

→ Additional information from Service Quality Management (SQM) oriented data sources (e.g. KPI, DATASOURCE, STIME, etc. ...) will be part of the „Additional Text“ attribute.

Table 6: Event/Alarm Attributes



Special remarks:

- * The event/alarm has to be encoded in ASCII
- * The "Date" attributes will have the following format: DD.MM.YYYY
- * The "Time" attributes will have the following format: hh:ss

Rationale:

- X.733 is widely used as a standard for the specification of a generic event/alarm. The attributes, as well as the state model and the behaviour of the model are quite stable since more than 15 years now. So that this seems to be a commonly accepted definition for the FM interface, which can be adopted to create an "implementation-ready" standardized API.

The abbreviations and conventions used here are part of the CCITT Rec. X.733 specification. See document: T-REC-X[1].733-199202-III-PDF-E.pdf , quoted here:

Chapter 4 Abbreviations

Conf	Confirm
Ind	Indication
Req	Request
Rsp	Response

...

Chapter 5 Conventions

This Recommendation | International Standard defines services following the descriptive conventions defined in CCITT Rec. X.210 | ISO/TR 8509. In clause 9, the definition of each service includes a table that lists the parameters of its primitives. For a given primitive, the presence of each parameter is described by one of the following values

M	the parameter is mandatory
(=)	the value of the parameter is equal to the value of the parameter in the column to the left
U	the use of the parameter is a service-user option.
–	the parameter is not present in the interaction described by the primitive concerned.
C	the parameter is conditional. The condition(s) are defined by the text which describes the parameter.
P	subject to the constraints imposed on the parameter by CCITT Rec. X.710 ISO/IEC 9595.

...

5.3.2 Event/Alarm Transport

It must be possible to send (Server) [and receive/listen to (Client) event/alarms]

The data transport must be reliable (see "non-functional" requirements)

Description:

- * EMSs (FM servers) can distribute (send) event/alarms according to X.733 event/alarm structure specification to NMS (OSS)
- [* NMSs (FM clients) can receive/listen to event/alarms according to X.733 event/alarm structure specification. ("NM send" is not required)]

Rationale:

- This is a basic and generic requirement for an FM interface.

(Remark: the NMS can also query for alarms, beside "Send" and "Receive". This requirement is covered under chapter "5.3.5 Event/Alarm Query".)

5.3.3 Clear – Event/Alarm Transport

It must be possible to send [and receive/listen to] "clear" event/alarm eventsDescription:



- The interface specification has to support “clear” events, according to the X.733 specification. EM systems (servers) should be able to deliver “clear-event/alarm” events, which can be unambiguously mapped on related event/alarms events (See “clear correlation” requirement later on). The NM system (client) must be able to handle the clear-event/alarms. The interface specification has to support this capability. The EMS must support clear-event/alarm handling. (But the NMS must be able to handle situations, if there are missing clear-events/alarms.)

Rationale:

- Support for clear–event/alarms improve the ability of network operators to understand the actual status of NEs -> do they deliver the NE service, or are there still open faults in the NE which might impact the NE service and eventually other subsequent end user services. Clear-event/alarms reduce the costs for operational processes, because they reduce the effort to identify the status of NEs. Without clear-event/alarms, the operator has to perform additional tests to verify the actual NE status.

5.3.4 Unambiguous Notification ID

It must be possible to correlate between clear–event/alarm and the original event/alarm, by using an unambiguous notification ID (which shall be a combination of the numerical notification ID and the “Managed Object”

[Details on requirements for the managed object see later on]

Description:

- A unique and unambiguous Notification ID is a prerequisite to enable the NMS to correlate between “clear” – event/alarms and original event/alarms. It is not allowed to use a combination of different attributes to create unambiguousness.
- The EM will send a “clear” – event/alarm, as soon as the incident, which caused the original event/alarm, does not exist any more. The NMS needs to be able to correlate between the Clear–event/alarm and the original event/alarm. So the EM system must be able to deliver “clear-event/alarm” events, which can be unambiguously mapped on related event/alarms events. The interface specification has to support this capability. Although this is a general requirement for EM systems and out of scope for this requirement specification for the interface itself, there must be an interface specification which describes the usage of the event/alarm attributes, so that the relation between event/alarm and clear-event/alarm can be uniquely identified.
- Remark: the requirement is different to the correlation mechanism described in the document “ITU-T X.733 Correction”.

Rationale:

- The actual X.733 mechanisms used to correlate between “clear”–event/alarms and the original event/alarms are inefficient and complex. They lead to complex and expensive implementations of FM interfaces, especially to be able to deliver NM support for **Event/Alarm Correlation (Clearing) and Re-Synchronization**.

5.3.5 Event/Alarm Query

It must be possible for the server (NM) to query all active event/alarms.

Description:

- The interface has to support the “Synchronization” functionality of the NM system. That means, the NM system can use a “query” functionality of the interface to get all event/alarms, which are known by the EM system (during the time of the “query” command) and which do not have the perceived severity = “cleared”.
- Remark: this capability requires the “unambiguous Notification ID” (see related requirement)

Rationale:

- This functionality allows the implementation of a synchronization mechanism in the NM – system. In case of an undefined state of the event/alarm data in the NMS (e.g. caused by a restore of the NMS database), the NM system can send a query to the EMS to synchronize between EMS event/alarm data and NMS event/alarm data.



5.3.6 Heartbeat

The interface has to support a heartbeat capability which allows EMS to send heartbeats (configurable) and NMS to receive/listen to heartbeats.

Description:

- The interface has to support the EMS heartbeat signals to the NMS. This functionality allows to indicate, that the EMS and the connection between EMS and NMS and is up and running.

Rationale:

- The heartbeat functionality ensures, that the NMS is able to inform the operator about a connection loss between EMS and NMS (alarming of connection-loss and clearing if connection is back).

5.4 EMS Specific Functional Requirements for Interface Support

5.4.1 Reliable Event/Alarm Communication (supported by EMS)

*** EMS buffers event/alarms if they cannot be sent to the NMS**

*** EMS sends event/alarms immediately as soon as the connectivity to the NMS is up again**

Description:

- The main intention of this requirement is, to ensure that no event/alarm is lost, when NMS goes down (caused by NMS problems or by maintenance work). X.733 (relates to X.710 for events) requests a logging mechanism for events on the originator site. This enables the NMS to synchronize with its data sources as soon as the NMS is back again → this is a requirement for the EMS.

Another problem might occur, when the transport mechanism between EMS and NMS is not available. To ensure, that the operator is aware about the malfunction of the interface, which will stop the ability to retrieve and to monitor event/alarms. This situation cannot be handled by the interface itself, but it can be handled either on EMS site (X.733 specifies a confirmation event which has to be delivered by the NMS, as soon as the NMS receives the event/alarm.) and/or by the NMS (e.g. via regular queries to the EMS [heartbeat]). → These requirements have to be supported by EMS and NMS. The interface itself has to support the confirmation of "send – events" and it has to support "queries".

Rationale:

- Ensure, that no event/alarm is gets lost, if the NMS or the interface to the NMS goes down.

5.4.2 Configurable EMS Heartbeat Message

EMS will send heartbeats in regular (configurable) intervals to NMS.

Description:

- The EMS will send heartbeat signals to the NMS in regular intervals (configurable intervals) to indicate, that the EMS and the connection between EMS and NMS and is up and running.

Rationale:

- The heartbeat functionality ensures, that the NMS is able to inform the operator about a connection loss between EMS and NMS (event/alarming of connection loss and clearing if connection is back).

5.4.3 Alarm Suppression

The northbound interface of EMS - Fault Management should enable the alarm suppression.

Description:

- The EMS interface offers the possibility to suppress the alarm of physical and logical objects when NMS will not receive any alarm from EMS. After alarm suppression all alarms will be cleared on the NMS and a warning will be generated on the NMS which indicate the alarm suppression. After re-enable of the alarms all alarms will be sent from EMS to NMS. This capability has to be configurable (manual / automatically).

Rationale:



- This functionality is very important for maintenance of equipment, hardware / software upgrade, testing etc.

5.4.4 Summary Alarms

EMS interface summary should provide summary alarm functionality.

Description:

- For minor alarm is sometimes not practicable to send every alarm from EMS to NMS. EMS generates a summary alarm and sends it to SMS when an alarm occurs several times within a certain window-time. This capability should be configurable. E.g. if a alarm occurs and clear more than 50 times per minute, then EMS will send a summary alarm to NMS. If this alarm occurs and clear less than 50 times per minute, then EMS will sent clear alarm to NMS.

Rationale:

- This feature protects the NMS from alarms flood.

5.5 NMS Specific Functional Requirements for Interface Support

5.5.1 Re-Synchronization

The NMs must be able to synchronize the own event/alarm list with the EMs event/alarm lists

Description:

- The NMs will use the query functionality of the FM interface to synchronize the own event/alarm list with all EMs event/alarms with a perceived severity \neq "cleared". This functionality will be invoked automatically by re-connection of the NMs with the EMs after startup of the NMs or the interface

Rationale:

This capability has to ensure, that the event/alarm lists of the EMs and the NMs are always synchronized.



6 HIGH LEVEL OSS REQUIREMENTS FOR INVENTORY MANAGEMENT

6.1 Introduction and Scope of Inventory Management Sub Task

The baseline of important future Operation and Maintenance (O&M) requirements are defined in the NGMN Top OPE Recommendations. Those requirements are being further enhanced with more details by NGMN NGCOR project for guiding towards well standardized interfaces and interworking solutions throughout O&M/OSS. The enhanced requirements are targeting to give guidance to Standards Developing Organizations (SDOs), industry bodies (e.g. 3GPP or TM Forum) as well as OSS industry in order to prioritize the work, to develop the standards and implement the solutions for operational use.

The NGCOR project extends the original NGMN Top OPE Recommendations which are dealing with requirements of wireless environment to cover also converged (wireline and wireless networks) operations area. It is foreseen that wireline and wireless networks will be merged in the near future within many operators. There is a need for the definition of converged O&M requirements to ensure that the operational activities within the converged networks perform optimally. Already existing specifications and standards are taken into account and will be used as input to produce the requirements for the converged operations.

The inventory sub-task of NGCOR places the inventory management in the focal point of view as it is understood that inventories are the key and core parts of OSS architecture of operators. The main role of inventories is to provide comprehensive and reliable data supporting efficiently different operational, planning and deployment processes when managing the infrastructure and the services. A direction to harmonized inventory interfaces and information/data models is a must when having a growing complexity of OSS support needs. Operators still do have a lot of old legacy inventory systems; the information of which is not flexible to use, where the information is split to many pieces and many data stores. When implementing next generation networks and services increasing amount of new network and service information/data has to be managed in conjunction with the older. At the same time customer focused information management accelerates integration needs between BSS layer and OSS layer and requires inventory support. Generally inventory development projects are perceived as expensive and a question how to make migration paths cost effectively and secure way to new generation commercial-of-the shelf (COTS) inventories is of high importance for operators. NGCOR inventory sub-task has so far paid specific attention to specifications and activities within 3GPP and TMF as well as the joint work of them, the work in both in both organizations will be taken into account. Existing NGMN Top OPE recommendations concerning inventory management are naturally in scope for enhancements.

The IM section is prepared within NGMN member community (operators) during March - June 2011 and thereafter to be sent to NGMN partners to be discussed and potentially elaborated with clarifications and enhancements. The focus of NGCOR inventory sub-task in the first phase has been to get a common view on inventory management area in broad sense; the main inventory management concepts, the main roles and characteristics of inventories within OSS/BSS environment of operators. This is presented as high level inventory management requirements. In later phases of the NGCOR inventory sub-task during 2011 selected prioritized areas of high level requirement are planned to be worked out as more detailed requirements. The next steps of the work will address for example details for information modelling objects and attributes as well as interfacing/integration features and functionalities.

The IM section is structured in the following way:

First in order to create consistent set of Inventory management requirements NGCOR project has performed an extensive analysis of existing inventory management definitions, specifications and standards. The analysis is based on

- considerations of inventory roles on different management layers; market/customer/product management, service management, resource management



- inventory definitions, specifications and standards, mainly from TMF and 3GPP, as well as some comparisons conducted between those
- viewpoints related to aligning TMF frameworks and ITIL framework
- The analysis part of report is included as an appendix of requirements (chapter 1.1 in the IM section)

In chapter 1.1 a common and consolidated view of fundamental roles and concepts of inventories as a part of OSS architecture for operators is described. This is done based on analysis mentioned above, [Appendix Inventory Management](#).

[Chapter 1.1](#) also suggests scoping of the NGCOR work and prioritized focus areas of inventory management requirements.

[Chapter 0](#) contains the prioritized high level inventory management requirements.

6.2 Forming a Common View on Inventory Management

This chapter summarizes the findings regarding operators' common views on inventory management area. The summary is presented in order to create a solid basis for inventory management requirements further work overall in the context of NGMN NGCOR project. The main inventory management concepts, the main roles and characteristics of inventories are described briefly. Also the summary addresses 'the full picture' of inventories spanning from BSS-layer product inventory management to OSS with service and resource/network layer inventory management. In the later work the scope and priority of inventory requirements are focused on service and resource layer inventories.

The terminology, concepts and descriptions are based on analysis and reference definitions presented in [Appendix Inventory Management](#).

As a high level characterization of the role and direction towards enhanced inventory management can be stated:

- Inventories are the key OSS components/systems and central points of managed and structured way of information handling throughout different management layers.
- Inventories support different OSS applications with accurate data.
- In a converged fixed-mobile environment distributed (logical or physical) inventories have to provide merged/federated data together to support e2e management.
- Open and standard information/data model and easy integration is required.
- Creation, maintaining access of information has to be seamless and user friendly.
- It is understood that physical implementation of inventories consists of different databases throughout S/BSS and it is important that total architecture is designed in consistent way to support different separated or consolidated views considering inter-dependencies between management layers and covering full lifecycle of the logical and physical entities managed in the inventories.

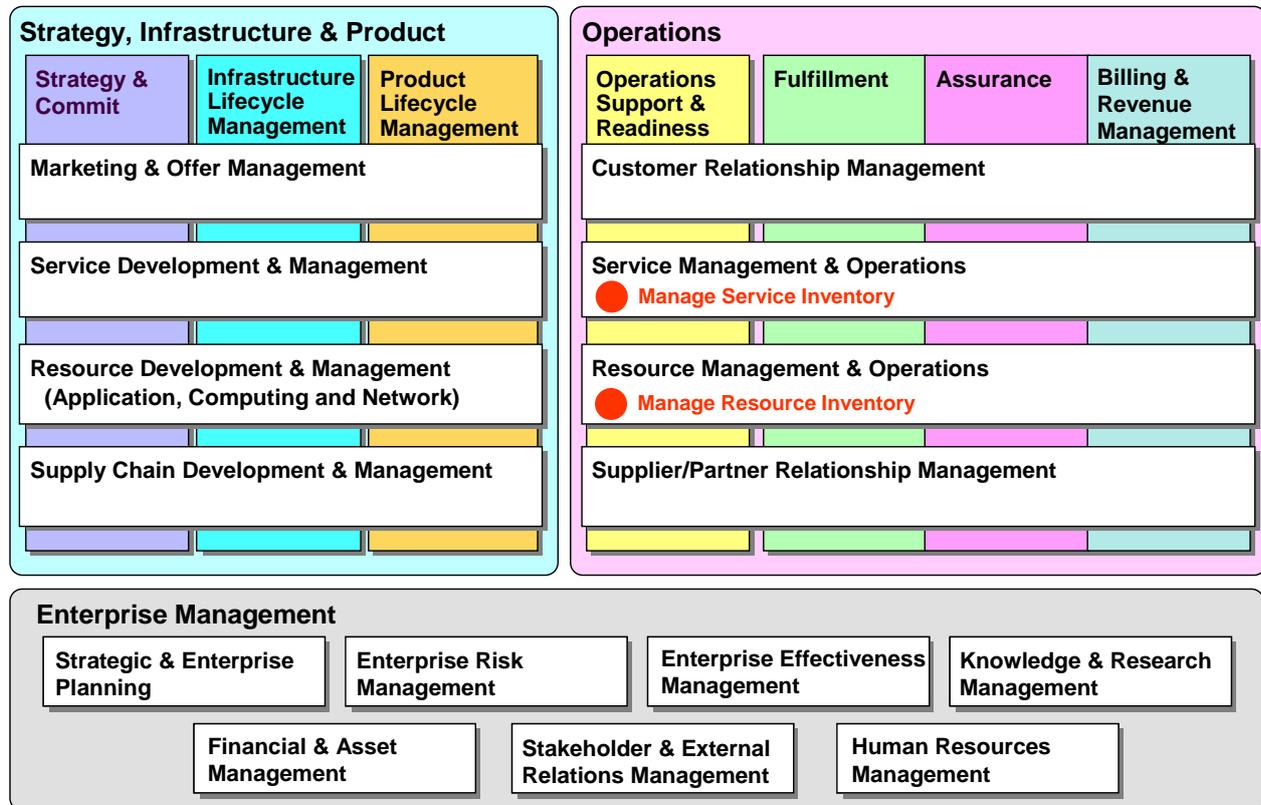


Figure 36: Key scope of IM sub task in eTOM framework

The figure shows the key scope of the NGCOR inventory sub task in terms of the **TMF Business Process Framework** (eTOM). The relevant level 3 processes concerned by the project's work are "**Manage Service Inventory**" (MSI) and **Manage Resource Inventory** (MRI) (see chapter 1.4.4.1 **Closely Related ITIL Concepts**). Both MRI and MSI are part of the operations process area. Vertically they are included in the **Operations Support & Readiness** processes. Horizontally they are covered by the **Service Management & Operations** processes (for MSI), and by the **Resource Management & Operations** processes (for MRI). Both MSI and MRI are defined to have wide interaction horizontally and vertically.

In relation to ITIL framework it is considered generally that ITIL practices and related system solutions share an analogue problem with telecom inventories on how information about IT infrastructure components and services can be managed. A respective key concept in ITIL framework is the **Configuration Management System (CMS)**; a coherent logical model of the IT organization's infrastructure, typically made up of several **Configuration Management Databases (CMDBs)** as physical sub-systems. It is used to store information on all configuration items (CIs) under the control of Configuration Management. CIs are mainly hardware or software items and are characterized by their attributes (recorded in the CI's Configuration Record) and their relationships to other CIs. Similarly like telecom inventory information is used e.g. by other operations process the ITIL CI information is utilized by e.g. ITIL incident management, problem management and change management processes.

It is notable that TMF and itSMF have done a joint technical work for converging TMF and ITIL concepts – the report is: TR143 Building bridges ITIL and eTOM.

Considering scoping with regards to 3GPP specifications it is to be noted that it is not possible to show direct match. 3GPP specifications do not model distinct management layers and structures for upper layer NMS/OSS (service management, resource management). Both resource infrastructure information and service related information is defined via NRM IRPs and interface IRPs. One exception is in the area of **Subscription Management (SuM)** defining a conceptual subscription management architecture (TS 32 140) where mapping of subscription management with eTOM fulfilment/provisioning is specified. Generally it is expressed that SuM will



need to manage subscription information in e.g. the OSSs, HSS, UE, OSA, MMS and IMS sub systems, but it is left as an open question in which extent and in which way SuM information is handled with OSS.

6.2.1 Resource Inventory Management

NGCOR inventory management uses TMF originated concepts (eTOM, TAM) for structuring the management of the different kinds of inventories. The scope is to address widely both service management layer and resource management layer needs for setting requirements and seeing relations from inventory management perspective in a comprehensive OSS architecture context.

Main Functionality

This chapter addresses **Resource Inventory Management** as a holistic concept without any major attempt to consider possible approaches for implementations. Main function of resource inventory management is to manage information of all resources used to implement services and products. Fundamental principle is manage resource information in a uniform and organized way as a key part of OSS architecture. The resource information to be managed covers all physical and logical resources including spare parts and extending to external plant and passive customer premises equipment.

As main logical capabilities resource inventory management needs to include

- Capabilities to manage, create, maintain and provide access to information of resource specifications/resource catalogs. The resource specifications are deployed by SI&P process functions, and the resource catalog is initially populated by OS&R process functions.
- Capabilities for providing and maintaining on-line resource instance information to automated and manual operation process functions. Resource instances are created based on resource specifications during fulfilment process and updated according usage assignment status of resources. All physical and logical configuration of the infrastructure including network elements and service systems (full e2e view: access, core, transport, control layer, application layer etc.) and their components as well as IT systems (SW and HW) are kept track on.

In the following a typical approach and some clarification is made how configuration information management and functionality may be addressed as a specific portion of resource inventory management concepts.

- The **Resource Inventory** contains physical, logical and network related information about resources, resource specifications and associations between these resources. It is populated by resource planning functions. All resources which are deployed in network infrastructure are maintained in the resource inventory. Reconciliation happens with resources discovered by the **Resource Discovery** processes. It lists all relevant resources and their topology, e.g. physical devices, device models, logical resources (numbers, ID, etc.), and licenses. Examples are ports of a DSLAM, cables, cable interconnections, etc.
- The **Configuration Data** is typically stored centrally (CMDB in ITIL) for the configuration management purposes. It contains the (initial and life) configuration information and parameters for resources (devices and applications) of the network infrastructure and the OSS. It is populated by configuration management which is triggered by e.g. resource life cycle, activation, assurance and discovery processes. Versions and validity ranges for configurations are stored there, which are closely linked to resource management life cycles. Only the configuration management is allowed to update the configuration data. The configuration data are used to setup a device, e.g. in case of initial configuration, update or restore. Furthermore, internal consistency checks are done on the configuration data. Examples for configuration data are data for router/server configuration.

Both the configuration data and the resource inventory together fully describe the resources with all parameters and capabilities that are needed for operational purposes.

- The **Configuration Management** function performs the device configuration to bring resources into operation. It performs initial device configurations triggered by SI&P processes, and keeps the configuration data up to



date. Furthermore, it manages all changes to configurations including update, restore and retirement of a device. It also handles resource and configuration changes detected by discovery processes to ensure the consistency of the configuration data. Moreover, the configuration management provides a complete audit trail (i.e. when, by whom and why configurations have been changed) and ensures compliance to company internal configuration standards and policies.

Considering the role of resource inventory management in management of dynamic information in the network – such as functioning of **Self Organized Network** (SON) features in the network elements, it can be generally characterized OSS and management environment needs (ref. NGMN Top 10 recommendations)

- OSS with SON needs to support of centralized, distributed and hybrid solution.
- An NE can operate with SON function or without SON function and can easily be transferred between these two modes. The ability to suspend/ resume/ enable/ disable the SON function shall be determined on a case by case basis.
- Degree of automation to be configurable by the operator
- Support completely automated optimization cycle
- Support automated import of optimized settings
- OSS should provide a general SON monitoring & control application covering policy control, history log and switch on/off functionality. OSS shall be synchronized in real time with SON initiated network changes. Capability to monitor the specific results of each particular SON function needs to exist.

As regards to various optimization features enabled by SON (ANR, Cell Phy ID management, Cell outage compensation, load balancing, etc) it is needed that

- OSS should provide analysis, alarms and user friendly visualization of the optimization feature in question
- OSS should provide the operator with resolution scenarios as suggestions for each specific optimization case which the operator can choose and select to solve the conflict resolution. Optionally these suggestions can be enabled automatically following operator policies.

As a conclusion the dynamic and automatic behaviour of the network sets new requirements for both new types of OSS applications as well as keeping up-to-date information of the dynamic status of resources for resource inventory management.

Resource Inventory Interfacing/Integration with other OSS Components and with Resource Infrastructure

This chapter addresses various interfacing and integrations needs of resource inventory management. TMF TAM is here used as a generic model to present various applications/application areas of OSS environment; more specifically NGCOR has used the latest framework model from TAM v4.5.

Interactions of resource inventory management with other OSS applications/functions and with resource infrastructure

- The **Resource Inventory** stores information on available capacity of logical and physical resources to be accessible for service inventory management in order to design a service. service inventory management also uses information stored in the resource inventory to understand the infrastructure layer components and relations.
- The **Resource Order Management** retrieves equipment and connectivity details from the resource inventory in order to create requests to provision the network. It also stores intended and scheduled changes to the infrastructure in the resource inventory. Resource Activation can also create in the resource inventory logical resources (e.g. connections) in support of services.
- **Fault Management** retrieves information from resource inventory in order to correlate resource faults with resource topology information to be used in various functionalities e.g. displaying operational status of resources, root cause analysis, fault correction and fault reporting.

- **Service Problem Management/Trouble Ticketing** retrieves information from resource inventory to correlate service problems with resource topology information.
- **Service Quality Management** retrieves information from resource inventory to correlate service quality with resource topology information.
- **Performance Management** accesses the resource inventory for having topology information to identify the appropriate performance data collection points in order to accurately represent the performance of the resource.
- **Resource Discovery** function provides means to upload, synchronize and reconcile the resource inventory information with the actual resource element information. The interface is either via element management systems or directly to network elements.
- **Resource Inventory Synchronizing** function provides a common inventory view across the applications in resource management and ensures OSS inventory data generated in each application is available to other applications as required.
- **Configuration Management** performs the device configuration to bring resources into operation. It performs initial device configurations triggered by SI&P processes, and keeps the configuration data up to date.
- Billing data collection and mediation accesses the resource Inventory in order to retrieve topology information to identify the appropriate usage data collection points.
- **Resource Lifecycle Management** applications/functions such as resource planning, resource catalog management produce and consume resource inventory data.
- **Resource Test Management** accesses resource inventory for obtaining the resource information under testing.

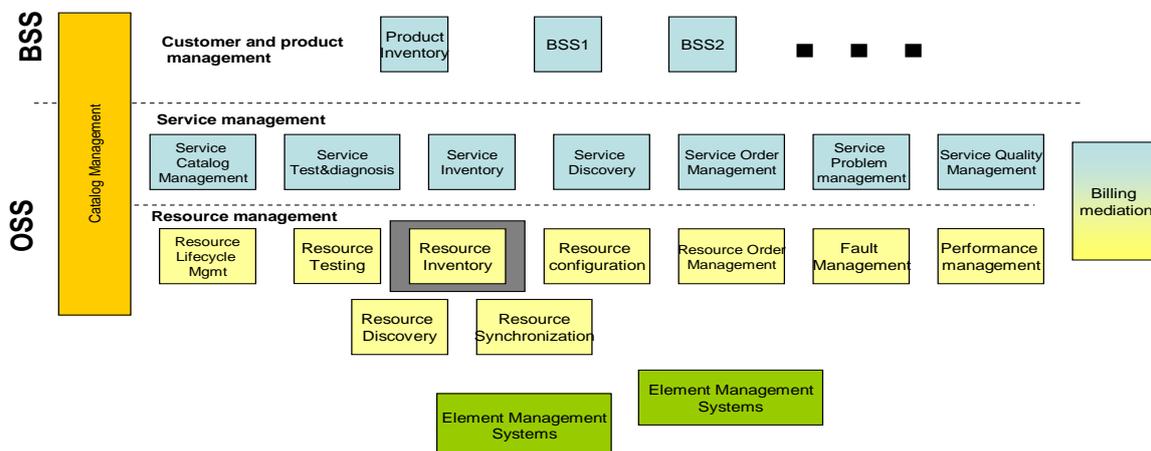


Figure 37: Resource inventory as part of OSS architecture

6.2.2 Service Inventory Management

NGCOR inventory management uses TMF originated concepts (eTOM, TAM) for structuring the management of the different kinds of inventories. The scope is to address widely both Service Management layer and Resource



management layer needs for setting requirements and seeing relations from Inventory Management perspective in a comprehensive OSS architecture context.

This chapter addresses service inventory management as a holistic concept without any attempt to consider possible approaches for implementations.

Main Functionality

The main function of **Service Inventory Management** is to manage and store information of all service specifications (service catalogs) and service instances. The **Service Inventory** implements an abstraction layer between products (owned & managed by BSS) and resources (owned & managed by OSS).

To enable collaboration between different domains, service inventories need to be harmonized. An agreement on a common service model (service specifications) for all involved domains is essential in that case.

As main logical capabilities of service inventory management needs to include:

- **Service Catalog:** Captures the engineering view of the service offering and consists of collections of service descriptions as **Customer Facing Service Specifications (CFSS)** and **Resource Facing Service Specifications (RFSS)** including their relationships.
RFSS are associated with resource specifications, stored in the resource catalog, thus capturing the relationship between a service and the set of resources supporting this service.
CFSS are associated with product specifications, stored in the product catalogue, thus capturing the relationship between a service and the product that is supported by this service.
Furthermore, related engineering processes/properties for provisioning and monitoring can be included, e.g. a production plan that covers the activation sequence and timing considerations, which have to be ensured during instantiation.
The service specifications are deployed by SI&P process functions, and the service catalogue is initially populated by OS&R process functions.
- **Service Instances** are created from service specifications during fulfilment processes, as Customer Facing Services (CFS) and Resource Facing Services (RFS), including their relationships among each other as well as with resource instances (RFS concerned) and product instances (CFS concerned).

Service Inventory Interfacing/Integration with other OSS Applications / Functions

This chapter addresses various interfacing and integrations needs of service inventory management. TMF TAM is here used as a generic model to present various applications/application areas of OSS environment; more specifically NGCOR has used the latest framework model from TAM v4.5.

- Operations Support & Readiness
 - **Service Discovery** checks services (service instances) which have been discovered against the service inventory to validate data quality, and to trigger the reconciliation process in case of discrepancy.
 - Resource inventory implements - together with the service inventory - the complete linkage between resources and services needed for the fulfilment, assurance, and mediation functions.
 - Service catalog is a subset of general cross-domain **Catalog Management**. A service catalogue deploys and stores service specifications as basis for service inventory data model definitions.
- Fulfilment (Order Management, Provisioning, Activation)
 - Creates the service instances based on BSS requests.
 - Creates, updates and stores specific engineering properties, e.g. a production plan that covers the activation sequence and timing considerations, which have to be ensured during instantiation of services.
 - Implements information brokering towards BSS on service related matters.
- Assurance
 - **Service Problem Management/Trouble Ticketing** retrieves service instance information, and navigates the service inventory for impact analysis.
 - **Service Quality Management** reads the service specification and service tree, and uses the information to set the desired monitoring thresholds.

- **Test & Diagnostics** retrieves service instance information, reads the test plans and stores test results.
- Billing Mediation
 - Uses information from the service inventory for proper grouping of the Call Detail Records (CDR) as they are forwarded to BSS.
- Catalog Management
 - Catalog management provides general, full lifecycle entity management capabilities cross domains, multilayer and acting as master repository for componentized entities of products, services and/or resources within one or more domains of a service provider's environment. Catalog management includes the abilities to create and design new entities, map entity definitions, manage complex rules, support componentization of entities and manage their relationships and dependencies. In service management layer context the consistency of service specifications mastered has to be ensured within the SM layer and SM with other layers in the catalog. For example, how product definition translate to different services provisioning rules, and so on.

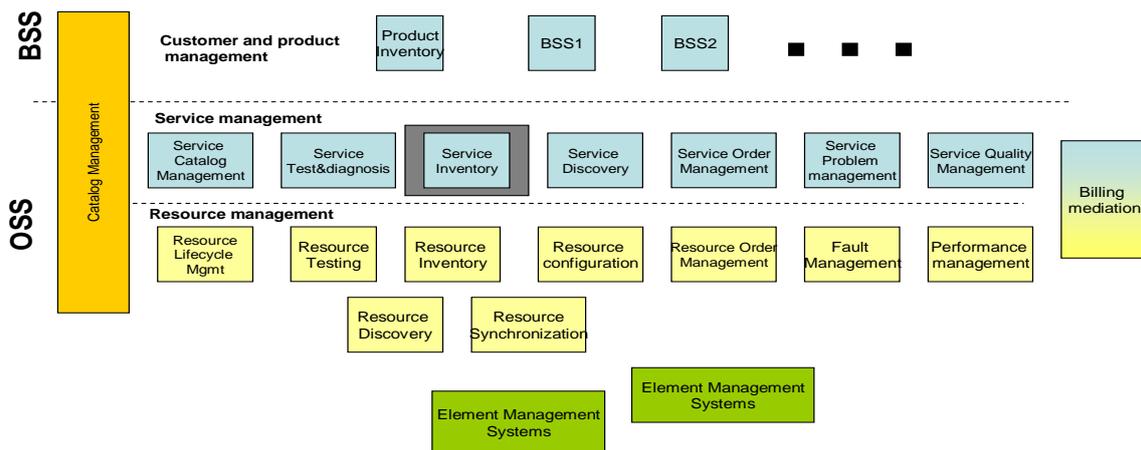


Figure 38: Service inventory as part of OSS architecture

6.2.3 Product Inventory Management

This chapter addresses product inventory management as a holistic concept without any attempt to consider possible approaches for implementations.

Main Functionality

The main responsibility of the **Product Inventory** is to manage the **Product Catalog** and keep track of the product subscriptions. The product catalog defines the product offering from marketing perspective and consists of a collection of **Product Specifications**. Each product specification describes a **Product Type**. Several product specifications may be defined for the same product type. Product specifications are associated with service specifications, stored in the service catalogue, thus capturing the relationship between a product and the set of services bundled by this product.



Each subscription is captured in the product inventory through a product instance associated with the corresponding specification in the catalog. The Product Instance is also associated with the subscriber of the product and the related subscriber account information.

Product Inventory Interfacing/Integration with other BSS/OSS Applications / Functions

- Customer SLA Management, service problem management/trouble ticketing, and billing and Customer Order Management use the information stored in the product inventory.
 - **Customer Order Management** function stores in the product inventory customer details, order and product detail, and account information acquired when a new order is created. Customer order management also retrieves product specifications from the product catalogue in order to create product instances and to decompose the product orders.
 - **Service Problem Management/Trouble Ticketing** function may access the Product Inventory to correlate a subscriber to a service, and to retrieve details about the subscriber, when creating a trouble ticket.
 - **Customer SLA Management** retrieves subscribers for given products and the subscriber contact information, using the Product Inventory.
- Service inventory management retrieves product inventory information for capturing the relationship between a service and the product that is supported by this service.

6.2.4 Definition of the Scope of Work of Inventory Management Sub Task and Limitations

Based on discussions it has been concluded that the scope of inventory management sub task within NGCOR project is limited to

- Resource management level inventory aspects / resource inventory as the first priority
- Service management layer inventory aspects / service inventory as the second priority/if time left
- In terms of vertical (eTOM) processes the working assumption is that the focus in later work is in operations process area and less in SI&P process area support

6.3 High Level Inventory Management Requirements

This chapter outlines the high level inventory management requirements identified based on the analysis about the roles and functions of resource inventory management and service inventory management within the OSS architecture.

To be noted that the focus of NGCOR inventory sub task in the first phase has been to get a common view on inventory management area in broad sense; the main inventory management concepts, the main roles and characteristics of inventories within OSS/BSS environment of operators. This is presented as high level inventory management requirements. In later phases of the NGCOR inventory sub-task during 2011 selected prioritized areas of high level requirement are planned to be worked out as more detailed requirements. The next steps of the work will address for example details for information modelling objects and attributes as well as interfacing/integration operations features and functionalities.

6.3.1 Functional Requirements

6.3.1.1 Resource Inventory

R1: Capability to manage resource models of variety of technology infrastructure domains and areas of converged fixed-mobile environment



In order to be able to act in a centric role in managing and storing resource data in a converged fixed-mobile environment all different resources models from e2e management perspective shall be possible to manage.

R2: Capability to offer and maintain resource data to/with the different applications supporting planning & implementation, fulfilment, assurance and billing (generally SI&P, OSR, FAB), and with resource infrastructure

Resource inventory shall store and manage common data for other OSS applications and synchronized with actual resource data.

R3: Capability to organize and offer ownership of resource information/data among applications, functions and processes.

Mechanisms to have organized data master ships and ways for CRUD (Creation/Reading/Updating/Deleting) of Resource Inventory data.

R4: Capability to model and document the horizontal relationship (on physical and logical level) between resources, spanning all types of resource – technologies.

Mechanisms to organize the horizontal relationship between resources. It must be possible to analyse the interworking of resources which delivers the E2E network service. The logical layer is needed to understand the ability of the network which delivers the E2E network service as a prerequisite for the Impact Analysis function in service management capabilities. The physical layer (including the documentation of redundancy) is a prerequisite for the impact analysis as well (e.g. to understand the impact of an outage on the E2E network service, and it is a prerequisite for Root Cause Analysis in NMS

6.3.1.2 Service Inventory

R5: Capability to manage service models of different domains and areas for converged fixed-mobile services

In order to be able to act in a centric role in managing and storing service data in a converged fixed-mobile environment all different services shall be possible to model and manage.

R6: Capability to offer and maintain service data to/with the different applications supporting planning & implementation, fulfilment, assurance and billing (generally SI&P, OSR, FAB)

Service inventory shall store and manage common data for other OSS applications.

R7: Capability to organize and offer ownership of service information/data among organization functions and processes

Mechanisms to have organized data master ships and ways for CRUD (Creation / Reading / Updating / Deleting) of service inventory data.

6.3.2 Information / Operations Model Requirements

6.3.2.1 Resource Inventory

The requirements will be enhanced with more details in the later phases of NGCOR project considering e.g. the model content and attributes from resource inventory perspective.

R8: A common harmonized and consistent resource data model covering different infrastructure domains of converged fixed-mobile environment



It is crucial that the data managed centrally in the resource inventory is comprehensive covering all different resources of a converged fixed-mobile environment and modelled in consistent way. Resource modelling characteristics and extensive details are presented in the section from “Modelling and Tooling” sub task of NGCOR.

R9: A common, harmonized and consistent resource data model agreed between interworking OSS applications/areas for resource management.

Resource inventory manages and stores centrally common information for various other OSS applications. The other OSS applications producing or consuming resource inventory data shall have a common data model with resource inventory. Resource modelling characteristics and extensive details are presented in the section from “Modelling and Tooling” sub task of NGCOR.

6.3.2.2 Service Inventory

Note: The requirements will be enhanced with more details in the later phases of NGCOR project considering e.g. the model content and attributes from service inventory perspective.

R10: A common harmonized and consistent service data model covering different services of converged fixed – mobile environment

It is crucial that the data managed centrally in the service inventory is comprehensive covering all different services of a converged fixed-mobile environment and modelled in consistent way. Service modelling characteristics and extensive details are presented in the section from “Modelling and Tooling” sub task of NGCOR.

R11: A common, harmonized and consistent service data model agreed between interworking OSS/BSS applications/areas for service management.

Service inventory manages and stores centrally common information for various other OSS applications. The other OSS applications producing or consuming service inventory data shall have a common data model with Service Inventory. Service modelling characteristics and extensive details are presented in the section from Modelling and Tooling sub-task of NGCOR.

R12: Vertical service data model, which contains the relationship of Services to Resource/Product/Customer – Layers.

Service inventory manages and stores the relationship of services downwards to the resources they are build upon and upwards to the products and customer which make use of these services. This is a prerequisite for the impact analysis capability of the service management functions.

6.3.3 Interfacing / Integrations Requirements

6.3.3.1 Resource Inventory

Abstract

In the following requirements the purpose of interfacing/integration of resource inventory with different other OSS applications are explained. TMF TAM is used as a generic model to present various applications/application areas of OSS environment.

Note: The requirements will be enhanced with more details in the later phases of NGCOR project considering e.g. the operation model content, attributes and integration standards and protocols from resource inventory perspective.

R13: Resource inventory integration/interfacing with service inventory management



The resource inventory stores information on available capacity of logical and physical resources which needs to be accessible for service inventory management in order to design a service.

R14: Resource inventory integration/interfacing with resource order management

The resource order management retrieves equipment and connectivity details from the resource inventory in order to create requests to provision the network.

R15: Resource inventory integration/interfacing with fault management

Fault management retrieves information from resource inventory in order to correlate resource faults with resource topology information.

R16: Resource inventory integration/interfacing with service problem management

Service problem management retrieves information from resource inventory to correlate service problems with resource topology information

R17: Resource inventory integration/interfacing with service quality management

Service quality management retrieves information from resource inventory to correlate service quality with resource topology information.

R18: Resource inventory integration/interfacing with performance management

Performance management accesses the resource inventory for having topology information to identify the appropriate performance data collection points.

R19: Resource inventory integration/interfacing with resource discovery

Resource discovery function provides means to upload, synchronize and reconcile the resource inventory information with the actual resource element information. The interface is either via element management systems or in some cases directly to network elements.

R20: Resource inventory synchronization

Resource inventory synchronizing function provides a common inventory view across the applications in Resource management and ensures OSS inventory data generated in each application is available to other applications as required.

R21: Resource inventory integration/interfacing with billing mediation

Billing mediation accesses the resource inventory in order to retrieve topology information to identify the appropriate usage data collection points using standardized formats and protocols.

R22: Resource inventory integration/interfacing with configuration management

Configuration management performs the device configuration to bring resources into operation. It performs initial device configurations triggered by SI&P processes, and keeps the configuration data up to date.

R23: Resource inventory integration/interfacing with resource testing

Resource test management accesses resource inventory for obtaining the resource information under testing.

R24: Resource inventory integration/interfacing with other resource lifecycle management

Resource lifecycle management applications/functions such as resource planning, resource change management and resource catalog management produce and consume resource inventory data.

6.3.3.2 Service Inventory

Abstract



In the following requirements the purpose of interfacing/integration of service inventory with different other OSS applications are explained. TMF TAM is used as a generic model to present various applications/application areas of OSS environment.

Note: The requirements will be enhanced with more details in the later phases of NGCOR project considering e.g. the operation model content, attributes and integration standards and protocols from service inventory perspective.

R25: Service inventory integration/interfacing with fulfilment

Fulfilment creates the service instances based on BSS requests. It creates, updates and stores specific engineering properties, e.g. a production plan that covers the activation sequence and timing considerations, which have to be ensured during instantiation of services. It implements information brokering towards BSS on service related matters.

R26: Service inventory integration/interfacing with service problem management / trouble ticketing

Service problem management (including service monitoring functions) / trouble ticketing retrieves service instance information, and navigates the service inventory for impact analysis.

R27: Service inventory integration/interfacing with service quality management

Service quality management reads the service specification and service tree, and uses the information to set the desired monitoring thresholds.

R28: Service inventory integration/interfacing with SLA management

SLA management reads the service specification and service tree, and uses the information to set the desired SLA thresholds.

R29: Service inventory integration/interfacing with test & diagnostics

Test & diagnostics retrieves service instance information, reads the test plans, and stores test results.

R30: Service inventory integration/interfacing with billing mediation

Billing mediation accesses to information in the service inventory for proper grouping of the CDR as they are forwarded to BSS, using standardized formats and protocols.

R31: Service inventory integration/interfacing with service discovery

Service discovery checks services (service instances), which have been discovered, against the service inventory to validate data quality, and to trigger the reconciliation process in case of discrepancy.

R32: Service inventory integration/interfacing with resource inventory

Resource inventory implements, together with the service inventory, the complete linkage between resources and services needed for the fulfilment, assurance, and mediation functions (OSS).

R33: Service inventory integration/interfacing with product / customer inventory

Service inventory implements, together with the product / customer inventory, the complete linkage between services and products and customers, needed for the fulfilment and assurance functions (OSS).

R34: Service inventory integration/interfacing with catalog management

Service catalog is a subset of general cross-domain catalog management. Service catalog deploys and stores service specifications as basis for service inventory data model definitions supporting full lifecycle of services including e.g. test plans.



6.4 Appendix Inventory Management

6.4.1 Source Material, Scope of Analysis and Context

6.4.1.1 Identification of Essential Source Material

In order to create consistent set of Inventory management requirements NGCOR project has first performed an extensive analysis of existing inventory management definitions, specifications and standards. The analysis is based on

- considerations of inventory roles on different management layers; customer / product management, service management, resource management
- inventory definitions, specifications and standards, mainly from TMF and 3GPP, as well as some comparisons conducted between those
- viewpoints related to aligning TMF frameworks and ITIL framework

6.4.1.2 Clarifying the Scope of Analysis in Terms of Management Layers: Market, Product and Customer Management, Service management, Resource management

Some analysis based on TMF definitions of different management layers (processes view).

The **Market, Product and Customer** layer processes include those dealing with sales and channel management, marketing management, and product and offer management, as well as operational processes such as managing the customer interface, ordering, problem handling, SLA management and billing.

The **Service Management** layer processes include those dealing with service development and delivery of service capability, service configuration, service problem management, quality analysis, and rating

The **Resource Management** layer processes include those dealing with development and delivery of resource (network and IT) infrastructure, and its operational management including aspects such as provisioning, trouble management and performance management. Resource infrastructure supports products and services, as well as supporting the enterprise itself.

Operator's OSS involves typically support and applications / systems addressing service management and resource management layers whereas market, product and customer layers are regarded belonging to BSS. It is understood however that there is no exact defined borderline between what is included in the OSS and what in BSS.

Some more specific analysis on Service management and Resource management

Service Management & Operations

Brief Description

This horizontal functional process grouping focuses on the knowledge of services (access, connectivity, content, etc.) and includes all functionalities necessary for the management and operations of communications and information services required by or proposed to customers.



Extended Description

This horizontal functional process grouping focuses on the knowledge of services (access, connectivity, content, etc.) and includes all functionalities necessary for the management and operations of communications and information services required by or proposed to customers. The focus is on service delivery and management as opposed to the management of the underlying network and information technology. Some of the functions involve short-term service capacity planning for a service instance, the application of a service design to specific customers or managing service improvement initiatives. These functions are closely connected with the day-to-day customer experience.

The processes in this horizontal functional process grouping are accountable to meet, at a minimum, targets set for service quality, including process performance and customer satisfaction at a service level, as well as service cost.

Resource Management & Operations

Brief Description

Maintains knowledge of resources (application, computing and network infrastructures) and is responsible for managing all these resources (e.g. networks, IT systems, servers, routers, etc.) utilized to deliver and support services required by or proposed to customers.

Extended Description

This horizontal functional process grouping maintains knowledge of resources (application, computing and network infrastructures) and is responsible for managing all these resources (e.g. networks, IT systems, servers, routers, etc.) utilized to deliver and support services required by or proposed to customers. It also includes all functionalities responsible for the direct management of all such resources (network elements, computers, servers, etc.) utilized within the enterprise. These processes are responsible for ensuring that the network and information technologies infrastructure supports the end-to-end delivery of the required services. The purpose of these processes is to ensure that infrastructure runs smoothly, is accessible to services and employees, is maintained and is responsive to the needs, whether directly or indirectly, of services, customers and employees. RM&O also has the basic function to assemble information about the resources (e.g. from network elements and/or element management systems), and then integrate, correlate, and in many cases, summarize that data to pass on the relevant information to Service Management systems, or to take action in the appropriate resource.

In an ebusiness world, application and computing management are as important as management of the network resources. Moreover, network, computing and applications resources must increasingly be managed in a joint and integrated fashion. To cope with these needs, the eTOM framework includes the **Resource Management & Operations** process grouping (together with the corresponding **Resource Development & Management** grouping within SIP), to provide integrated management across these three sets of resources: applications, computing and network. These areas also encompass processes involved with traditional Network Element Management, since these processes are actually critical components of any resource management process, as opposed to a separate process layer.

The RM&O processes thus manage the complete service provider network and sub-network and information technology infrastructures.

To be noted still possible confusion points of using the term service, from TR143:

“So, on this basis, ITIL provides IT Services – this is reflected in the ITIL publications and provides a distinction from the services discussed in eTOM. IT Services are often focused internally within the enterprise but increasingly can play a part in supporting external customers and users. Since this takes us very much into eTOM territory, we should not regard IT Services as directly accessible by external customers/users, but instead see these (where needed) as bound into “Products”– see next paragraph. It is noted that within ITIL documentation, “IT Service” is contracted to just “service”, but this should be read as “IT Service” in all cases.



On the same basis, eTOM can be considered to provide services oriented to the communications and content needs of external businesses and individual customer/users/consumers. There is not, unfortunately, an established term that encompasses the full scope of these services, so we will use here "Communication and Content Services". This is a little cumbersome, but at least covers the intended scope.

Thus, on this basis, eTOM provides Communication and Content Services, as indicated, but note that these Communication and Content Services are not delivered to customers as is. Within eTOM (and NGOSS generally), we portray that Products are offered to customers, where a product may incorporate one of more services, and typically binds around the service(s) other, non-engineering, aspects such as tariffs, SLAs, support agreements, etc. To be consistent, we should thus say that eTOM offers Communication and Content Products to the external customers "

6.4.1.3 From NGMN Top OPE recommendations

Recommendation 9: OSS Tool Support for Optimisation & Operation

Abstract

Tools and application in the north from the OMC plays a significant role in an operator organization to coordinate operator processes: e.g. workflow management tools, optimization tools and data bases. In this chapter the focus is on recommendations regarding these tools as complementary area of the chapter OSS Standard Interface covering the standard northbound interface which is the bridge between the so called OSS tools in the north and the radio and core network infrastructure network element managers (or OMC). The main idea of the coming chapters is to formulate main use cases and recommendations on related tools in OSS to support standardization.

Recommendations

The recommendations on optimization in NMS layer can be described as in the following. Generally the recommendations as given in 4.1.3 are applicable and repeated to underline their importance for OSS tool support.

- SON functionality / capability shall have controlled implementation in order to build trust and confidence in automation and avoid massive operational impact
- SON solutions shall provide an easy transition from operator controlled (open loop) to autonomous (closed loop) operation, as the network operator gains more trust in the reliability of the SON.
- For operator controlled (open loop) SON function, the implementation of any update proposed by the SON function shall take effect only after a response by the operator. OSS should provide the possibility to configure certain break points for SON operations, allowing the operator for manual intervention to proceed with the logic, or to halt / abort it. The vendor shall provide a prediction of the expected results prior to executing SON logic. The operator shall be able to proceed with the logic after having previewed the expected results.
- For closed loop SON function, the implementation of any update proposed by the SON function shall take effect without the need for response by the operator.
- An NE can operate with SON function or without SON function and can easily be transferred between these two modes. The ability to suspend/ resume/ enable/ disable the SON function shall be determined on a case by case basis.
- The IRP manager shall be able to monitor the specific results of each particular SON function OSS should provide a general SON monitoring & control Application covering policy control, History log and switch on/off functionality. OSS shall be synchronized in real time with SON initiated network changes.
- During open loop operation, network operations staff manually reviews the results of the SON function at intermediate steps in the particular SON process. The network operations staff decides upon and manually initiates the appropriate next step in the SON process.
- The vendor shall provide for each SON feature a methodology to demonstrate the robustness & quality of the SON feature related algorithms (e.g. through simulations under various conditions).
- The vendor shall provide for each SON feature a methodology for acceptance of the feature.



- Network and Management System should provide a general SON monitoring & control application covering policy control, history log and switch on/off functionality per site and cell.
- If SON is not functioning as expected, it shall be possible to disable individual portions and perform the operation manually.
- SON centralized, distributed and hybrid approach must be supported (depending on the SON use case).
- Network and Management System should provide possibility to configure certain break points for SON Operations, allowing the operator for manual intervention to proceed with the logic, or to halt / abort it.
- Network and Management System shall be synchronised in real time with SON initiated network changes. Notifications shall also be available real-time via the CM Northbound Interfaces to NMS
- Network and Management System should provide a valuable Reporting Suite for SON activities
- Network and Management System shall fully support SON as defined in 3GPP standards, inclusive CM Northbound Interface 3GPP BulkCM IRP (CORBA or SOAP based)
- Provide an open northbound interface for all SON related parameters for interoperability with 3rd party vendors
- Network and Management System should be able to request or report the SON related changes for statistical analysis and historical view.
- It shall be possible to customise SON policies. On the one hand, there shall be flexibility to adjust the SON functionality to the operator's recommendations. On the other hand, customisation shall be a simple process to minimise the manual effort required.
- Optimisation for identified parameters shall be done within a value range, defined by the operator.
- Optimisation shall be done with respect to KPIs and parameters not directly related to the use-case KPI (i.e. other KPIs shall not become worse than defined thresholds (e.g. Handover-Optimisation shall be done with respect to capacity related parameters resp. KPIs).
- Dependency between KPIs resp. definition which KPIs shall be considered in addition to use-case KPI(s) shall be configurable by the operator.
- Thresholds for start and end point of parameter optimisation shall be configurable by the operator.
- Optimisation cycle should be configurable (periodically, event-based).
- Support of centralized, distributed and hybrid solution
- Degree of automation configurable by the operator.
- Optimization cycle completely automated: yes / no
- Automated import of optimized settings: yes / no

From the above recommendations, the open and close Loop architecture should support the following functionalities. It is highlighted that these recommendations for the following functionalities are addressed to all relevant standardisation bodies (3GPP as e.g. SA5 or TMF or others). It is the task of these bodies to decide and to agree on work split and definite body specific areas.

ANR

- EMS shall fully support ANR as defined in 3GPP standards, inclusive CM northbound interface 3GPP BulkCM IRP (CORBA based). ANR based changes in the eNB shall be "online" synchronised with EMS.
- The ANR functionality supports real time behaviour of relationship configuration to ensure that HO is possible a few seconds after neighbour detection.
- OSS shall be able to configure / manage "no X2 flag", "no remove flag" and "no HO flag" (as opposed to eNB only per 3GPP).
 - OSS shall be able to support monitoring of the main ANR steps:
 - Neighbour cell detection
 - X2 Set-up
 - Neighbour cell configuration adaptation
 - ANR Optimization

Cell Phy_ID allocation & configuration shall be automated

- OSS should provide analysis, alarms and user friendly visualization for Phy_Cell_ID collision and confusion detection



- OSS should provide the operator with resolution scenarios as suggestions for Phy_Cell_ID collision and confusion, which the operator can choose and select to solve the conflict resolution. Optionally these suggestions can be enabled automatically following operator policies.

Cell Outage Detection and Compensation

- OSS should provide analysis, alarms and user friendly visualization for cell/service outage detection
- OSS should provide the operator with resolution scenarios as suggestions for specific cell / service outage situation, which the operator can choose and select to solve the conflict resolution. Optionally these suggestions can be enabled automatically following operator policies.

Load Balancing

- OSS should provide analysis, alarms and user friendly visualization of load situations in different RATs.
- OSS should provide the operator with resolution scenarios as suggestions for overload situations, which the operator can choose and select to solve the conflict resolution. Optionally these suggestions can be enabled automatically following operator policies.

HO (Mobility) Optimisation

- OSS should provide analysis, alarms and user friendly visualization of HO related statistics as HO failure rate per neighbour combination or call drop rates etc.
- OSS should provide the operator with resolution scenarios as suggestions for HO mobility related problem, which the operator can choose and select to solve the conflict resolution. Optionally these suggestions can be enabled automatically following operator policies.

Trace Management for Optimisation Purpose

- OSS should provide analysis, alarms and user friendly visualization for general optimisation purpose as available based on trace data. It is possible to correlate trace data with other information as PM, alarms etc.
- OSS should provide the operator with resolution scenarios as suggestions for problem scenarios identified by trace data and other correlated data, which the operator can choose and select to solve the conflict resolution. Optionally these suggestions can be enabled automatically following operator policies.

QoS Optimisation

- OSS should provide analysis, alarms and user friendly visualization for QoS related problems as low threshold per user, higher delays or blocking rates.
- OSS should provide the operator with resolution scenarios as suggestions for QoS problems, which the operator can choose and select to solve the conflict resolution. Optionally these suggestions can be enabled automatically following operator policies.

Tracking Area Optimisation

- OSS should provide analysis, alarms and user friendly visualization for tracking area related issues as high paging load or high tracking area update load in a certain cluster.
- OSS should provide the operator with resolution scenarios as suggestions for specific TA area problem scenarios, which the operator can choose and select to solve the conflict resolution. Optionally these suggestions can be enabled automatically following operator policies.

SON in Core net

- Strong focus is on use cases in the RAN area to define optimisation use cases and their SON solutions. It is highlighted that also in CN interesting use cases can be beneficially be covered by SON functionality as e.g. load balancing among core network nodes (MME, S-GW)

10. Energy Saving

- OSS should provide analysis, alarms and user friendly visualization to understand the energy consumption within a network.



- OSS should provide the operator with resolution scenarios as suggestions for finding scenarios with minimised energy consumption in a cluster, which the operator can choose and select to solve the conflict resolution. Optionally these suggestions can be enabled automatically following operator policies.

Common channel optimisation

- OSS should provide analysis, alarms and user friendly visualization related to common channel optimisation as e.g. the load on common channels or specific errors.
- OSS should provide the operator with resolution scenarios as suggestions for solving common channel related problems, which the operator can choose and select to solve the conflict resolution. Optionally these suggestions can be enabled automatically following operator policies.

Optimisation reg. Interactions between Macro and Home eNB

- OSS should provide analysis, alarms and user friendly visualization related to home and macro eNB interworking scenarios as e.g. interference situation in macro and home eNB layer.
- OSS should provide the operator with resolution scenarios as suggestions to solve negative impact of one layer onto the other one, which the operator can choose and select to solve the conflict resolution. Optionally these suggestions can be enabled automatically following operator policies.

Note: The above listed use cases are analysed in dedicated NGMN projects and in a next version of the IM section the results should be transferred to this section to provide detailed examples of indicators, algorithms or configuration having to be handled by OSS functionality.

In the following also operation related use cases should be mentioned to be handled by OSS functionality:

Automatic Inventory

- OSS supports the automatic inventory by a configuration management system based on standardised and proprietary infrastructure input.
- Vendor infrastructure (RAN & core elements)
- Standardised interface for signalling information about changes performed in the Network
- Standardised interface to poll the information about Network Element Configuration and Components.
- All changes are available via a push or pull mechanism, e.g. following the final self test the eNodeB delivers
 - A state change notification
 - Details on its resource configuration (resource inventory)
 - Details on its parameter configuration (configuration data)
 - The [pictures on the next page](#) is meant to illustrate the High Level architecture of CMS integration in Operators OSS environment
- There should be a standardized network (resource) inventory model which will enable to create centralized cross-domain multi-vendor Inventory which can be filled with data provided by domain specific NEMs. Standard model is expected to eliminate costs for translating vendor specific network resource models.
- The recommended approach is to leverage TMF SID model as e.g.:
 - Describe radio lines
 - Describe dependency between logical connections and physical layer
- The standardized network inventory should be extendable, for example
 - For FM – there should be a dictionary of common problems referring to the appropriate types of resources. This dictionary should be used by NEMs when reporting alarms to indicate the type of problem.
 - For performance management – there should be a set of KPIs defined per resource to which the standardized KPIs refer to. This would enable to create standardized dictionary of KPIs
- NEM should assure consistency between inventory data which it provides up to central Inventory and data exposed by other functional interfaces, for example on FM interface or performance management interface .

Information correlation for fault management and automated fault correction



- NEM should provide standard itf-N interface for delivering FM functionality. The interface should provide information about alarms according to the standard format.
- To assure semantic consistency between NEMs provided by different vendors, FM interface should leverage a standard reference resource model. This is a necessity to assure that incoming alarms can be correctly interpreted by identifying resource type (NE type) an alarm refers to. This is even more important for inter-NEM correlation. Standard reference resource model should include topology relations and dependency between NEs and thus enable proper interpretation of alarms
- One of the essential responsibilities of NEM should be unique identification of NE (Managed Object) to which the Alarm (Event) refers. The aim is to enable precise identification of the MO in the OSS_NetworkInventory and thus to enable correlation of multi-vendor alarms.
- NEM or deeper level should perform initial root cause analysis and correlation in order to be able to provide the most precise information about a fault as it is possible to infer within the NEMs domain.
- The correlation of alarms done by a NEM should be described in the standard way leveraging the standardized network model. For example the alarm informing about radio line failure, when indicating that the root cause is a transceiver problem, should leverage a standard model for describing a radio line and its transceivers.
- The cause of a fault identified by a NEM should be contained in an alarm in standardized manner to avoid the need for vendor specific alarm processing.
- There should be standardized dictionary of problems, causes of failure defined together with the network model. This recommendation is meant to avoid vendors using their own vendor specific codes to inform about the common problems.
- There should be a standardized interface between OSS_NetworkInventory and OSS_ServiceInventory which would enable to identify services implemented over the resources and thus enabling to calculate the service impact of a resource fault.
- There recommended interface between OSS_NetworkInventory and OSS_ServiceInventory should be based on SID model taking as a skeleton for integration the “Customer Facing Service-Resource Facing Service-Resource” model to glue network resource domain with the service one.
- There should be a standard interface between OSS_FaultManagement and OSS_CellOutageCompensation enabling OSS_FaultManagement to initiate Cell Outage Compensation process.

Real time Performance Management

- Free configurable measurement and delivery periods for each counter or counter group.
- Simply structured and compact raw data format with a maximum net data rate, e.g. csv (current XML-based 3GPP standard has large overhead)
- NEM Internal post-processing of raw data without significant delay (near real-time)
- Automated counter or counter group administration (incl. activation).
- Automated quality management of performance data, e.g. automatic counter restart after outage
- Function for simple threshold based on counters and KPIs
- Function for simple KPI calculation based on counters
- Automatic identification of network problems and error correction.

SW Management:

- **“NE health check”**: OSS system has to be able to verify automatically that network elements are ready for software upgrade. The health check (e.g. faulty HW modules, critical alarms, free disk space) has to be executed during the dayshift to ensure the correct behaviour and preconditions of the NE itself.
- **Automated software download**: the software download to the NEs should work in parallel with a minimum of unavoidable manual steps. A result overview list must be provided.
- One-click NE software activation: software activation should also work in parallel with a minimum of unavoidable manual steps. The NE health check should support also the wrap-up activities for urgent issues.
- **Automatic rollback**: Only if the software activations fail completely an automatic rollback should be initiated.
- Long Term Vision:
 - SW package is made available on OSS and NEs are tagged on OMC for upgrade. Policies for software activation are set.



- All necessary activities (NE health check, SW download, SW activation, corrective actions) are carried out policy controlled by the software management application.
- A final upgrade report is provided that will be used as basis for the final wrap up phase.
- It is understood that with the long term approach the operator loses detailed control of each single step necessary for a software upgrade. A policy controlled bulk software upgrade is expected to be less error prone than today's solutions

Recommendation 10: Automatic Inventory

An automatic inventory function shall synchronize in real time with the configuration management system (CMS). Notification of any change to a passive or active element or its configuration relevant to a business process must be possible: consumer of that might be directly the configuration management system (CMS) / network inventory or other OSSs. The same information shall also be available in addition via batch load or polling mechanisms. The configuration management system (CMS) is the grouping of all relevant inventory systems to provide information required by the planning, deployment and operations processes.

The introduction of standardized functions and protocols to support automatic inventory will ensure:

- more efficient management of configuration data in the CMS
- availability of accurate and real-time information, as a basis for planning, deployment and operation

Automatic Inventory

- OSS supports the automatic inventory by a configuration management system based on standardised and proprietary infrastructure input.
- Vendor infrastructure (RAN & core elements)
- Standardised interface for signalling information about changes performed in the network.
- Standardised interface to poll the information about Network Element Configuration and Components.
- All changes are available via a push or pull mechanism, e.g. following the final self test the eNodeB delivers
 - A state change notification
 - Details on its resource configuration (resource inventory)
 - Details on its parameter configuration (configuration data)
 - The **pictures on the next page** is meant to illustrate the high level architecture of CMS integration in operators OSS environment.
 - There should be a standardized network (resource) inventory model which will enable to create centralized cross-domain multi-vendor Inventory which can be filled with data provided by domain specific NEMs. Standard model is expected to eliminate costs for translating vendor specific network resource models.
- The recommended approach is to leverage TMF SID model as e.g.:
 - Describe radio lines
 - Describe dependency between logical connections and physical layer.
- The standardized network inventory should be extendable, for example
- For FM – there should be a dictionary of common problems referring to the appropriate types of resources. This dictionary should be used by NEMs when reporting alarms to indicate the type of problem
- For performance management – there should be a set of KPIs defined per resource to which the standardized KPIs refer to. This would enable to create standardized dictionary of KPIs.
- NEM should assure consistency between inventory data which it provides up to central Inventory and data exposed by other functional interfaces, for example on FM interface or performance management interface .
- The **picture below** aims to illustrate the high level architecture of a configuration management system integration in operator's OSS environment **(to be found)**.



6.4.2 TMF

6.4.2.1 TMF Frameworks / eTOM process view

6.4.2.1.1 Manage Product Offering Inventory

Process Context

This process element represents part of the overall enterprise, modelled in business process terms, and can be applied (i.e. “instantiated”) with other similar process elements for application within a specific organization or domain.

Brief Description

Establish, manage and administer the enterprise's product offering inventory, as embodied in the **Product Offering Inventory Database**, and monitor and report on the usage and access to the product offering inventory, and the quality of the data maintained in it.

Extended Description

The purpose of the manage product offering inventory processes are twofold - establish, manage and administer the enterprise's product offering inventory, as embodied in the product offering inventory database, and monitor and report on the usage and access to the product offering inventory, and the quality of the data maintained in it. The product offering inventory maintains records of all product offerings, their interactions with the enterprise, and any other product offering related- information, required to support CRM and other processes.

The product offering inventory is also responsible for maintaining the association between customers and purchased product offering instances, created as a result of the order handling processes.

Responsibilities of these processes include, but are not limited to:

- Identifying the inventory-relevant information requirements to be captured for product offerings;
- Identifying, establishing and maintaining product offering inventory repository facilities;
- Establishing and managing the product offering inventory management and information capture processes;
- Managing the registration and access control processes that enable processes to create, modify, update, delete and/or download product offering data to and from the product offering inventory;
- Ensuring the product offering inventory repository accurately captures and records all identified product offering details, through use of automated or manual audits;
- Tracking and monitoring of the usage of, and access to, the product offering inventory repository and associated costs, and reporting on the findings; and
- Identifying any technical driven shortcomings of the product offering inventory repository, and providing input to resource development & management processes to rectify these issues.

6.4.2.1.2 Manage Service Inventory

Process Context

This process element represents part of the overall enterprise, modelled in business process terms, and can be applied (i.e. “instantiated”) with other similar process elements for application within a specific organization or domain.

Brief Description

Establish, manage and administer the enterprise's service inventory, as embodied in the **Service Inventory Database**, and monitor and report on the usage and access to the service inventory, and the quality of the data maintained in it.

Extended Description



- The responsibilities of the manage service inventory processes are twofold - establish, manage and administer the enterprise's service inventory, as embodied in the service inventory database, and monitor and report on the usage and access to the service inventory, and the quality of the data maintained in it.
- The service inventory maintains records of all service infrastructure and service instance configuration, version, and status details. It also records test and performance results and any other service related- information, required to support SM&O and other processes.
- The service inventory is also responsible for maintaining the association between customer purchased product offering instances and service instances, created as a result of the service configuration & activation processes.
- Responsibilities of these processes include, but are not limited to:
- Identifying the inventory-relevant information requirements to be captured for service infrastructure and service instances;
- Identifying, establishing and maintaining service inventory repository facilities;
- Establishing and managing the service inventory management and information capture processes;
- Managing the registration and access control processes that enable processes to create, modify, update, delete and/or download service data to and from the service inventory;
- Ensuring the service inventory repository accurately captures and records all identified service infrastructure and service instance details, through use of automated or manual audits;
- Tracking and monitoring of the usage of, and access to, the service inventory repository and associated costs, and reporting on the findings; and
- Identifying any technical driven shortcomings of the service inventory repository, and providing input to resource development & management processes to rectify these issues.

6.4.2.1.3 Manage Resource Inventory

Process Context

This process element represents part of the overall enterprise, modelled in business process terms, and can be applied (i.e. "instantiated") with other similar process elements for application within a specific organization or domain.

Brief Description

Establish, manage and administer the enterprise's resource inventory, as embodied in the resource inventory database, and monitor and report on the usage and access to the resource inventory, and the quality of the data maintained in it

Extended Description

- The responsibilities of the manage resource inventory processes are twofold - establish, manage and administer the enterprise's resource inventory, as embodied in the **Resource Inventory Database**, and monitor and report on the usage and access to the resource inventory, and the quality of the data maintained in it.
- The resource inventory maintains records of all resource infrastructure and resource instance configuration, version, and status details. It also records test and performance results and any other resource related- information, required to support RM&O and other processes.
- The resource inventory is also responsible for maintaining the association between service instances and resource instances, created as a result of the **Resource Provisioning Management** processes.
- Responsibilities of these processes include, but are not limited to:
 - Identifying the inventory-relevant information requirements to be captured for resource infrastructure and resource instances;
 - Identifying, establishing and maintaining resource inventory repository facilities;
 - Establishing and managing the resource inventory management and information capture processes;

- Managing the registration and access control processes that enable processes to create, modify, update, delete and/or download resource data to and from the resource inventory;
- Ensuring the resource inventory repository accurately captures and records all identified resource infrastructure and resource instance details, through use of automated or manual audits;
- Tracking and monitoring of the usage of, and access to, the resource inventory repository and associated costs, and reporting on the findings; and
- Identifying any technical driven shortcomings of the resource inventory repository, and providing input to resource development & management processes to rectify these issues.

6.4.2.1.4 Further Processes

6.4.2.1.5 1.4.2.1.4 Further Processes

Further process descriptions from SI&P, OS&R and FAB areas can be considered here to identify usage scenarios on how other OSS functions use the inventories.

Example for SI&P:

- Product & Offer Development & Retirement
- Service Development & Management
- Service Development & Retirement
- Resource Development & Management
- Resource Development & Retirement

6.4.2.2 1.4.2.2 TMF Frameworks / TAM view

Product management domain



Figure 39: Product management domain from TAM 4.5

In TAM 4.5 improved definitions are developed for catalog management concept. The concept of a catalog containing specifications/offers is complementary to the concept of an Inventory containing the instances (as delivered data) based on catalog contents, such as products, services and resources.

6.4.2.2.1 1.4.2.2.1 Product Catalog Management

Overview

Product catalog management is a realization of the cross-domain catalog management application in the customer domain. The applications are repositories of product listing within a service provider and include the ability to



design, create, augment and map new entities and supporting data. The type of catalog management application is an implementation choice of the enterprise.

See: [Cross-domain catalog management for more information.](#)

Functionality

Supported Business Services

- Get product offering/component effective duration: retrieves product effective date information from the catalog based on input unique ID or other search criteria for product offering or product component.
- Get product offering/component sales availability duration: retrieves product sales availability date information from the catalog based on input unique ID or other search criteria for product offering or product component.
- Get product offering/component characteristics: retrieves product characteristic content information from the catalog based on input unique ID or other search criteria for product offering or product component, in addition to criteria to identify a characteristic.
- Get product offering/component characteristic duration: retrieves product offering/component characteristic duration information from the catalog based on input unique ID or other search criteria for product offering or product component, in addition to criteria to identify a characteristic.
- Get product offering/component characteristic version: retrieves product offering/component characteristic version information based on input unique ID or other search criteria for product offering or product component. Can be applied against prior or future versions of product offering/component characteristics.
- Get product offering/component pricing: retrieves product offering/component pricing information based on input unique ID or other search criteria for product offering or product component.
- Get product offering/component costing: retrieves product offering/component cost information based on input unique ID or other search criteria for product offering or product component.
- Get product offering/component description: retrieves product offering/component descriptive information based on input unique ID or other search criteria for product offering or product component.
- Get product offering/component structure: retrieves product offering/component structural information (such as related/child product offering/components) based on input unique ID or other search criteria for product offering or product component.
- Get entities where product/component used: retrieves other entities within the catalog (i.e. Tariffs, Discounts) based on input unique ID or other search criteria for product offering or product component.
- Get master product offering/component ID: retrieves product catalog master ID based on input unique related ID or other search criteria for product offering or product component. This service is used to maintain product offering/component synchronization between other systems.
- Get campaigns which relate to product offering/component offering: retrieves campaigns within the catalog based on input unique ID or other search criteria for product offering or product component.
- Get discounts which relate to product offering/component offering: retrieves Discounts within the catalog based on input unique ID or other search criteria for product offering or product component.
- Check operational compatibility (between product offering/component): determines whether two product offering/components are compatible from an operational standpoint, based on input of multiple input unique IDs or other search criteria for product offerings or product components.
- Check customer compatibility (between product offerings/components and customer): determines whether a customer and a product offering/component are compatible based on input of input unique ID or other search criteria for product offerings or product components and customer attributes.
- Get product offering/component SLA: retrieves SLA from the catalog based on input unique ID or other search criteria for product offering or product component.
- Get product offering/component BOM: retrieves bill of materials list from the catalog based on input unique ID or other search criteria for product offering or product component.
- Get available product offering/component business services: retrieves associated business services from the catalog based on input unique ID or other search criteria for product offering or product component.

Service Management Domain

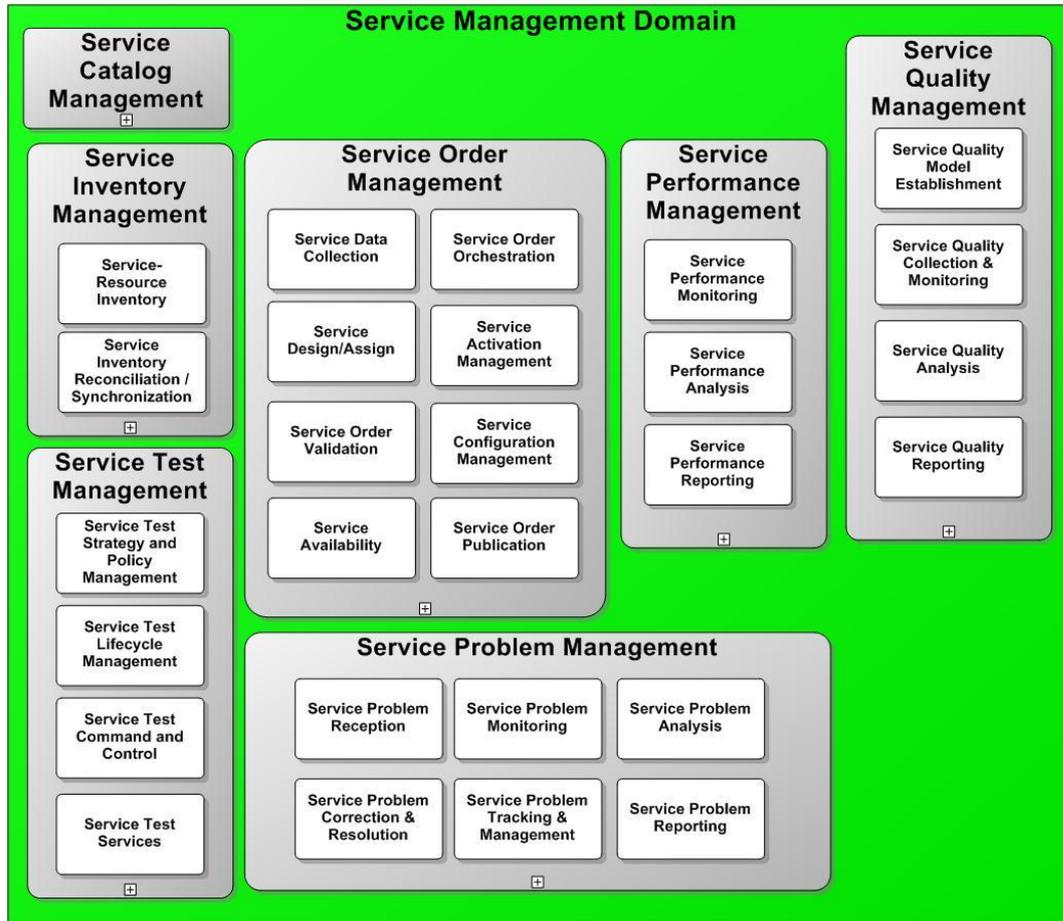


Figure 40: Service management domain from TAM 4.5

To be noted the changes from earlier TAM versions

- Service Catalog Management
- Service Quality Management
- Service Level Agreement Management only on Customer management layer

6.4.2.2.2 Service Inventory Management

Overview

Service inventory management represents the applications which contain and maintain information about the instances of services in a telecom organization.

A service inventory application may store and manage any or all of the following entities:

- Customer facing service (CFS) instances, and their attributes
- Resource facing service (RFS) instances, and their attributes

The service inventory may also store and manage service relationships:

- The mapping of services (RFSes or CFSes) to other services and/or service components, the components being either:



- Other child services
- Resources and the resource domain managers used to implement the service, or
- Services and resources in supplier/partner systems used to implement the service
- This mapping is stored either intrinsically in the core service inventory, or discretely via service-supporting resource inventory applications.

Service inventory may include the following relationship types between entity instances:

- **Realization by Composition** – A mapping from a service to the child services and/or resources which specifically compose that service (e.g. the RFS instance or instances whose whole purpose is to implement a CFS, the assignable resources which realize an RFS). If a parent service is torn down, child objects with a composition relationship are typically removed or reallocated (e.g. transitioned to spares inventory).
- **Realization by Aggregation** – A mapping from a service to the services and/or resources which support this service in addition to other services. (e.g. a network access RFS which supports a number of different network CFSes). If a parent service is torn down, child objects with an aggregation relationship are typically maintained as long as at least one other parent service still exists.
- **Dependency** – A link between services and/or resources which is not strong enough to qualify as composition or Aggregation, but where various fulfilment, assurance, and change management processes need to be aware of the relationship. Dependency relationships support the ability for change management processes to evaluate if a dependent service or resource may be impacted by changes to a specific service or resource.

Functionality

- Service Inventory Information Model
- Service Inventory Retrieval
- Service Inventory Update Notifications
- Service Inventory Update
- Service-Supporting Resource Inventory
- Service Inventory Reconciliation / Synchronization

Service Inventory Information Model

This function is the underlying information model for the service instances to be managed. The model serves as the foundation for the data itself and a guiding force for the definition and modelling of new services.

The service inventory information model should evolve in close coordination with the service specification data model, since the service inventory model must be able to store instances designed in accordance with all the service specifications defined via the service specification management system.

Typically, the service provider would need to add a lot of detail concerning the services to be managed. The suggested approach for the service provider is to start with the TMF information framework service model and then specialize the model for the specific services to be managed. The service model should indicate or point to the supporting component services and resources for each service (the information framework, in fact, does do this).

Service Inventory Retrieval

This function allows for client system to retrieve a part or all of the service inventory known to the service inventory management system.

This feature may support the following selection criteria:

- retrieval based on attribute matching
- retrieval of only the object instances that have been modified after a provided date and time
- retrieval based on relationship to a specific entity (e.g. all CFS instances supported by a specific RFS instance)

For the selected objects, this feature may allow the client OSS to specify what specific attributes and relationships shall be returned.

Service Inventory Update Notifications



This function entails the generation of inventory update notifications based on changes to the inventory known to the Service Inventory Management system. The notification types typically include object creation, object deletion, attribute value changes, and object relationship changes.

Single Entity Notifications – in this variation of the feature, each notification pertains to only one entity, e.g., an IP VPN service instance

Multi-entity Notifications – in this variation of the feature, a single notification may report on inventory changes for multiple entities (e.g. changes in any component services of a specific CFS).

Service Inventory Update

This function entails an external system requesting that the Service Inventory Management system update its inventory based on a provided collection of updates. The expectation is that the Service Inventory Management system updates its inventory as requested, but no other side-effects are expected (e.g., creating a service in the network). This is a key point concerning this capability. The inventory update request can involve creation of an object, deletion of an object, or modification of an object's attributes, or creation or deletion of an object's relationships to other objects.

Supported Business Services

Consumed Business Services

- Service Specification
- Resource Inventory Management Systems

Exposed Business Services

- Customer Order Management
- Service Order Management
- Service Problem Management
- Service Performance Management
- Service Level Agreement Management
- Service Quality Monitoring
- Revenue Assurance

6.4.2.2.3 Service-Resource Inventory

Overview

Service resource inventory is a shared function between service inventory and resource inventory, and, depending on the needs of an individual organization, may be implemented in a service inventory management system, a resource inventory management system, some combination of both, or even in a standalone application which bridges the gap between service and resource inventory management.

Service resource inventory entails managing the relationship between RFSes and the resources and resource domain managers which implement the services on the network. Resources may all be directly managed by the carrier's resource inventory systems, or may also include references to resources from a supplier / partner asset management system.

Typically, this inventory does not track all possible network resources involved in delivery of the service (this is the realm of resource inventory management systems themselves), but rather:

- Any stand-alone physical or logical resources whose assignment is critical to service fulfilment, and whose tracking is critical to service operations, assurance, and billing. Examples may include: modem or other special CPE equipment which may not be tracked directly as part of the provider network, static IP addresses and other network identifiers, etc.
- Assignment-level resources which represent a larger resource structure supporting the service, often referred to as an access point. Examples include: the ADSL DSLAM port assigned to a service, a data circuit service's assigned customer facing router interface or sub interface, etc.



- In some cases, the service supporting resource Inventory may also track the domain manager applications (e.g. resource inventory and/or activation systems) which manage the resource in question, although in a mature SOA implementation, the service supporting resource inventory can often be agnostic of which resource layer systems actually master the resource data.

Functionality

- Service-Resource Relationship Creation
- Service-Resource Relationship Update
- Service-Resource Relationship Update Notifications
- Service-Resource Relationship Deletion
- Service-Resource Relationship Retrieval
- Service-Resource Relationship Reconciliation / Synchronization

Supported Business Services

Consumed Business Services

- Service Specification
- Resource Inventory Management Systems

Exposed Business Services

- Customer Order Management
- Service Order Management
- Service Problem Management
- Service Performance Management
- Service Level Agreement Management
- Service Quality Monitoring
- Revenue Assurance

6.4.2.2.4 Service Inventory Reconciliation / Synchronization

Overview

This function entails reconciliation of the data in a service inventory management system with inventory discovered from another source and/or synchronization of mismatched service inventory records.

When new service inventory information is discovered, the service inventory reconciliation / synchronization system will try to match the newly discovered information with an entity or entities already existing in the Service Inventory. If no match is found, the service inventory reconciliation / synchronization system will typically assume that a new entity has been discovered and add the entity to the inventory. Alternately, as decided by the service provider as part of their procedures, the service inventory reconciliation system may record this event as an exception, implicitly or explicitly triggering a workflow to resolve the exception. For example, this may happen if the service provider always expects to have the planned service inventory in their service inventory management system systems before the actual services are activated.

If a match is found and there are no unexpected discrepancies, the service inventory reconciliation / synchronization system will update the inventory as needed. For example, records may be updated to fill in missing attributes or update attribute values which have changed. If a match is found and there are unexpected discrepancies, the service inventory reconciliation system will typically raise an exception so that service provider personnel can correct the problem. Exceptions may be managed within the application itself, via a report, or via a generalized worklist tool.

Functionality

- Service instance comparison



- Service reconciliation exception management

Supported Business Services

Consumed Business Services

- Service Specification
- Resource Inventory Management Systems

Exposed Business Services

- Customer Order Management
- Service Order Management
- Service Problem Management
- Service Performance Management
- Service Level Agreement Management
- Service Quality Monitoring
- Revenue Assurance

In TAM 4.5 improved definitions are developed for catalog management concept. The concept of a catalog containing specifications/offerings is complementary to the concept of an Inventory containing the instances (as delivered data) based on catalog contents, such as products, services and resources.

6.4.2.2.5 1.4.2.2.5 Service Catalog Management

Overview

Service catalog management is a realization of the cross domain catalog management application in the Service Domain. The applications are repositories of service listing within a service provider and include the ability to design, create, augment and map new entities and supporting data. The type of catalog management application is an implementation choice of the enterprise.

See: cross domain catalog management for more information.

Resource Management Domain

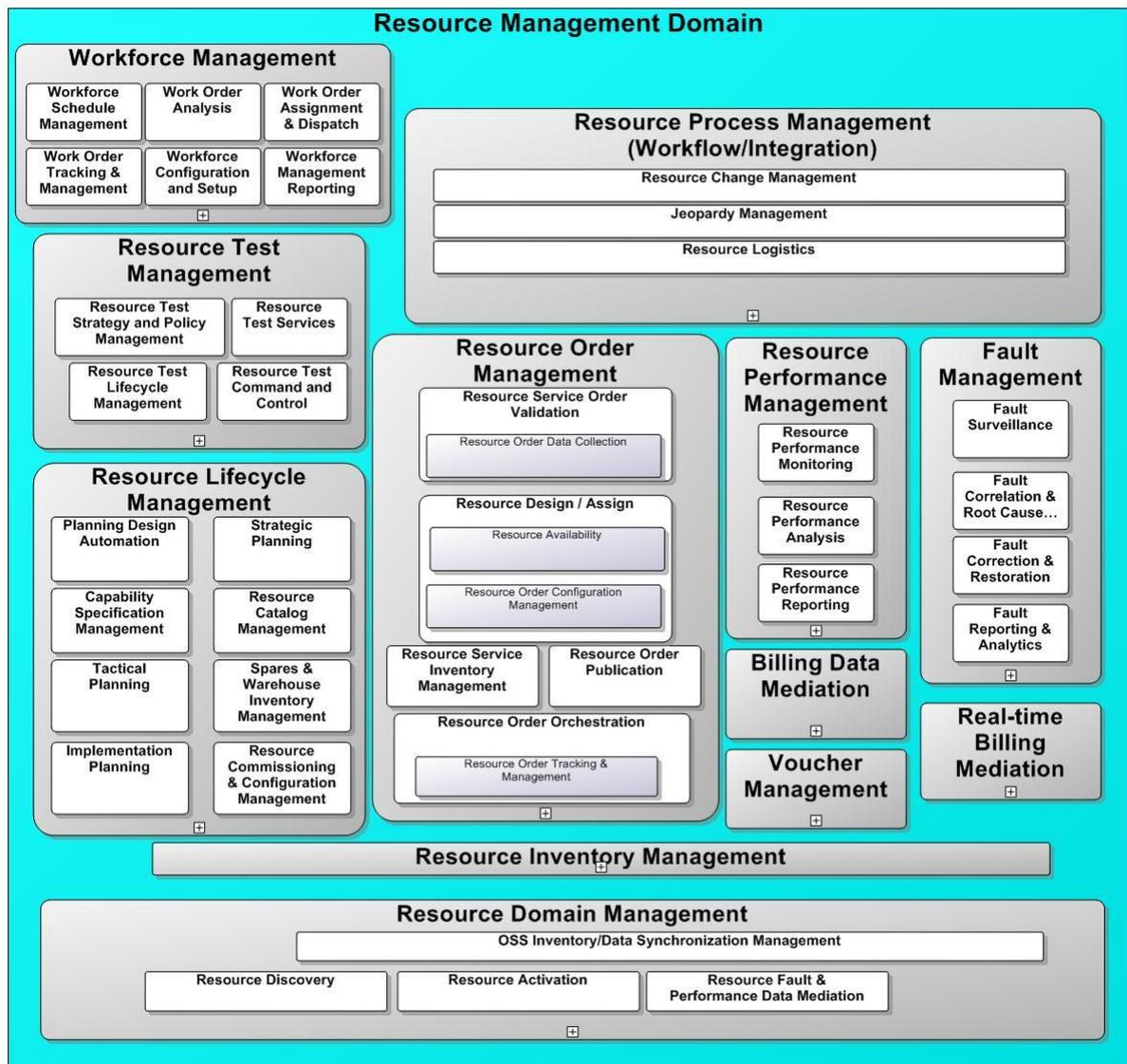


Figure 41: Resource management domain from TAM 4.5

6.4.2.2.6 Resource Inventory Management

Overview

Resource Inventory applications manage information of all resources used to implement services and products. This application area is typically linked to various element management systems (i.e. building inventory for actual server, applications, and network and resource assets) and resource inventory database systems which may or may not be combined with service inventory application(s) or database(s). In addition, resource management applications have a major role to play managing spare parts; passive resources including cable pairs and external plant and passive customer premises equipment.

In addition, resource Inventory applications are used to discover and manage underutilized or 'stranded' resources.

Functionality



- Resource inventory information model – the assumption is that this feature implements the standardized information model for the resources to be managed. Typically, the service provider would need to add a lot of detail concerning the resource attributes that are to be managed. The specific details will depend on the particular resources (e.g., particular types of managed elements and equipment) and associated technologies (e.g., SONET/SDH, ATM and ethernet) to be managed. The suggested approach for the service provider is to start with the TM Forum information framework and then define or make use of an existing model that specializes the information framework for the specific technologies that need to be managed.

Key Functions:

- Accurately describes the state of resources (network elements and their components, IT systems and applications, resources defined within systems etc.). A critical aspect of this is the recording of what resources are consumed by service instances at the physical and technology layers.
- Track status all resources
- Database of all spares (capacity management and optionally interface to asset tracking)
- Barcode/RFID tracking of all resources including spares
- Resource site information
- Resource history tracking for all problems and returns
- Interacts with resource activation and resource provisioning
- Manages under-utilized or 'stranded' assets
- Resource inventory retrieval – this feature allows for client operations support (service assurance and billing systems) to retrieve part or all of the resource inventory known to the target OSS.

This feature may allow the following selection criteria:

- Retrieval of a specified set of one or more sub-trees
- Exclusion or inclusion of specified object types from the selected sub-tree
- Further filtering based on attribute matching
- Retrieval of only the object instances that have been modified after a provided date and time
- For the selected objects, this feature may allow the client operations support (service assurance and billing systems) to specify what specific attributes and relationships shall be returned. This (the attributes and relationships to be returned) would be the same for all objects of the same type.
- Resource inventory update notifications – this feature entails the generation of inventory update notifications based on changes to the inventory known to a given OSS. The notifications concerning object creation, object deletion and attribute value changes to other systems.
- Single entity notifications – in this variation of the feature, each notification pertains to only one entity, e.g., an equipment instance
- Multi-entity notifications – in this variation of the feature, a single notification may report on inventory changes for multiple entities.
- Notification suppression – in this variation of the feature, each notification pertains to only one entity. However, in cases where a container object is created (e.g., a managed element) that has many contained objects, the sending OSS may only report on the container object creation. The expectation is that the receiving OSS will use a retrieval operation to obtain the contained object. This concept is explained further in TM Forum document SD2-1, MTOSI Implementation Statement (see section 2.5.1, Publisher Notification Suppression).
- Resource inventory update – this feature entails an OSS requesting that another OSS (referred to as the target OSS) update its inventory based on a provided collection of updates. The expectation is that the target OS update its inventory as requested, but no other side-effects are expected (e.g., creating an SNC in the network). This is a key point concerning this capability. The inventory update request can involve addition (new object), modification (change to an existing object) or deletion (removal of an object).
- Resource inventory reconciliation – this feature entails an OSS reconciling its own inventory with inventory discovered from another source (typically, the network). When new inventory information is discovered, the OSS will try to match the newly discovered information with an entity or entities already known to the OSS
- If no match is found, the OSS will typically assume that a new entity has been discovered and add this to its inventory. Alternately, as decided by the service provider as part of their procedures, the OSS may record this



event as an exception. For example, this may happen if the service provider always expects to have the planned inventory in their inventory OSS before actual resources are installed.

- If a match is found and there are no unexpected discrepancies, the OSS will update its inventory as needed.
- If a match is found and there are unexpected discrepancies, the OSS will typically raise an exception so that service provider personnel can correct the problem.

Supported Business Services

To Be Added

6.4.2.2.7 Resource Discovery

Overview

The resource Discovery applications are responsible for automatically discovering the resources and their details through an management channel. These applications may either directly communicate with the network resources or communicate through a **Resource Domain Manager**.

Functionality

The resource discovery applications are one of the core applications of resource management and provide a feedback loop from the resource. In many cases where the accurate topology is not available in OS systems, they provide the only source of topology for management.

These applications will either communicate directly or through a domain manager to retrieve the resource information details. The applications should be able to support either retrieving the overall resource information or the detailed resource information which can include sub-components.

There are a number of users for the discovered information. For example, the resource inventory system will use the discovered information to reconcile its data against as-is information while a resource root cause analysis application will use the discovered topology to enrich the event and pinpoint the true root cause.

Supported Business Services

To Be Added

6.4.2.2.8 OSS Inventory / Data Synchronization Management

Overview

OSS inventory / data synchronization application provides a common inventory view across the applications in Resource Management. This may be a virtual common inventory produced by synchronization of federated inventories, a single inventory system, or some combination of the two.

Functionality

OSS inventory / data synchronization management represents the applications that ensure OSS inventory data generated in each application is available to other applications as required. OSS Inventory will contain at least resource capacity and service utilization modelling. The synchronization may not be as a result of executing other business processes, but can be dedicated synchronization processes, for example a regular re-synchronization scheduled for off-peak periods. For example, it enables **Utilization Management** to be updated with new capacity, and for lifecycle management to be updated with capacity utilization.

Note that this does not predetermine any implementation solution. It is entirely allowable within application framework to have, in this example lifecycle management and utilization management share a common OSS Inventory. The OSS inventory / data synchronization management application would represent the common inventory in this situation. Note also that application framework does not restrict any implementation solution to only one instance of each application, so the need for different common inventories and separate synchronization applications is fully accepted.



Supported Business Services

To Be Added

A specific close relation exists between resource inventory management and resource commissioning & configuration management.

6.4.2.2.9 Resource Commissioning & Configuration Management

Overview

These applications are responsible for managing and tracking the configuration of the resource (AKA CMDB). These applications typically work in a federated environment, where they rely on other applications for the data

Functionality

The functionality provided by these systems includes:

- Resource commissioning process - manage the commissioning process of a resource and ensuring that operational status are configured
- Resource configuration management - database and manage the configuration of the individual resources
- Resource configuration logs - record the history of configuration changes
- Resource configuration verification versus design - work with other applications like discovery application to ensure that the resource configuration matches the designed configuration
- Resource topology verification versus inventory mgmt. systems - work with the inventory management applications to ensure that the topology reflected in its database is in sync with that in the inventory management systems.

Supported Business Services

To Be Added

6.4.2.2.10 Resource Catalog Management

In TAM 4.5 improved definitions are developed for catalog management concept. The concept of a catalog containing specifications/offerings is complementary to the concept of an Inventory containing the instances (as delivered data) based on catalog contents, such as products, services and resources.

Overview

Resource catalog management is a realization of the cross domain catalog management application in the resource domain. The applications are repositories of resource listing within a service provider and include the ability to design, create, augment and map new entities and supporting data. The type of catalog management application is an implementation choice of the enterprise.

Since resource catalogs usually contain a verity of resource types such as directory numbers, addresses, cables and network devices they would typically be a standalone implementations covering the basic functionalities in cross domain, The layering relation between the service and resource catalog will be realized in the other resource lifecycle applications.

See: Cross-Domain catalog management below for more information.

Functionality

To Be Added

Supported Business Services

To Be Added



Catalog Management

In TAM 4.5 improved definitions are developed for catalog management concept. Catalog management is a cross-domain application (cross product / service / resource domains). Catalog Management deals with aspects of handling/administering offer / product / service / resource structures in an organized and efficient way that can be federated across multiple catalogs. The concept of a catalog containing specifications/offerings is complementary to the concept of an Inventory containing the instances (as delivered data) based on catalog contents, such as products, services and resources.

6.4.2.2.11 1.4.2.2.11 Catalog Management

Overview

Catalog management is a cross domain, multilayer application that operates as a master repository for componentized entities of products, services and / or resources within one or more domains of a service provider's environment.

Catalog management includes the abilities to create and design new entities, map entity definitions, manage complex rules, support componentization of entities and manage their relationships and dependencies. Additional aspects include versioning, change management, enhanced viewing, as well as editing and tracking capabilities.

Functionality

The common capabilities of catalog management applications will have the following features:

- Entity handling- A catalog management application should be able to create, modify and delete entities.
- Entity data implementation – The catalog will provide the capability to implement a flexible data model with the ability to structure entities and attributes as desired by the user and to extend the model according to the requirements.
- Integrity rules – Apply integrity rules at the entity level. Rules are required to maintain data integrity in the catalog. Human errors during product and service configuration cause major problems in testing and production phases, and automatic mechanisms that can eliminate such errors in advance are mandatory.
- Compatibility rules- define rules on operational entities which are applied by downstream systems when instantiating the catalog entities in the operational systems. For example when instantiating products out of ProductSpec.
- Componentization – Ability to group entities. A catalog user needs flexibility and openness in configuring data catalog entities. One of the configuration options that enable high re-usability is the possibility to group entities and re-use the group level.
- Component relation management – Ability to manage hierarchical, inheritance and reuse relations between components. Re-usability is a major requirement for management of a catalog. Re-usability is achieved through inheritance and through the re-use of standalone entities as well as entity hierarchies.
- Entity state management: The ability to manage the state of an entity during its lifecycle (e.g. planned, deployed, in operation, replaced by, locked...)
- Inter layer aspects of a catalog management application including
- Inter layer dependency-rules management – Manage rules that governs the relationships between entities in different layers. A catalog that manages different layers needs to maintain the rules within the layer and between the layers. For example, how product definition translate to different services provisioning rules, and so on.
- Inter catalog data integrity management data consistency should be kept not only in the specific layers of products, services and resources but also between layers. A specific product can be provisioned in multiple ways by different services supporting different technologies, and the specific rules and dependencies make it a mandatory requirement to enable management of the inter-layer dependencies.
- Versioning – Manage multi-versions of the same entity is a very important aspect in a catalog. The complexity starts with the ability to manage multiple versions for single entities, however in real life it is required to support



much more complex scenarios where entities relate to other entities that have a different lifecycle and a different versions map, however the validity and maintenance of the versions needs to be maintained.

- Change management – Manage the implications of catalog changes to determine the consequences of any given change. In addition, catalog users should be able to track and locate the history of changes in the catalog in an easy and accessible manner.
- Inquiry handling - Catalog data requires easy storage and retrieval of information. Historical changes should be stored and easily retrieved, including changes done on the entity level or changes done by different users. Retrieval process should return simple queries but also complex queries retrieving data entities that comply with complex conditions, in order to enable easier analyzing and slicing of the catalog data.
- Revision control – A catalog provides a work environment that permits users to work in parallel without interfering with each other's efforts, to manage the relevant permissions on the data or on activity level, and to support the different user interfaces required. The catalog provides the capability to manage access and change control at various levels such as user or group.
- Data driven security – control access to the data by its actual values, so a user may not see offerings in which the customer type (which is an attribute) is business if he (the user) is allowed to see only residential customers information.
- View Management – Generating different views for users that manage different data layers. As a master catalog can support multiple lines of business and multiple layers, it should be dynamic enough to provide different display options for the different cases. For example, a network implementer may require a visual graph that will show relations between services in a graphical manner, while a product manager will require a dedicated view that displays only the product offerings that are under his domain of responsibility. Creating different custom views for different roles is a mandatory requirement in such a catalog.
- Partner integration for both export and import of catalog entities:
 - Export to external partners: A catalog management application should include the ability to allow partners (e.g. content providers, or other SPs) to browse in catalog in real time or batch mode. Not all data shall be exposed; security and access control (as mentioned above) are essential features
 - Import from external partners: External partners may want to populate catalogs with their own (entities) specifications and be able to create associations with existing entities. Access control, validation and testing must be set appropriately.

6.4.2.3 TMF Frameworks / SID view

The information framework business view addresses the information and communication service industry's need for shared information/data definitions and models. The definitions in the business view focus on business entity definitions and associated attribute definitions. A business entity is a thing of interest to the business, while its attributes are facts that further describe the entity. Together the definitions provide a business-oriented perspective of the information and data. When combined with business oriented UML class models, the definitions provide the business view of the information and data.

The content in the Information framework business view is organized using the information framework model. The information framework was developed by the application of data affinity concepts to an enterprise's processes and data to derive a non-redundant view of the enterprise's, shared information and data. The result of this analysis is a layered framework, which partitions the shared information and data.

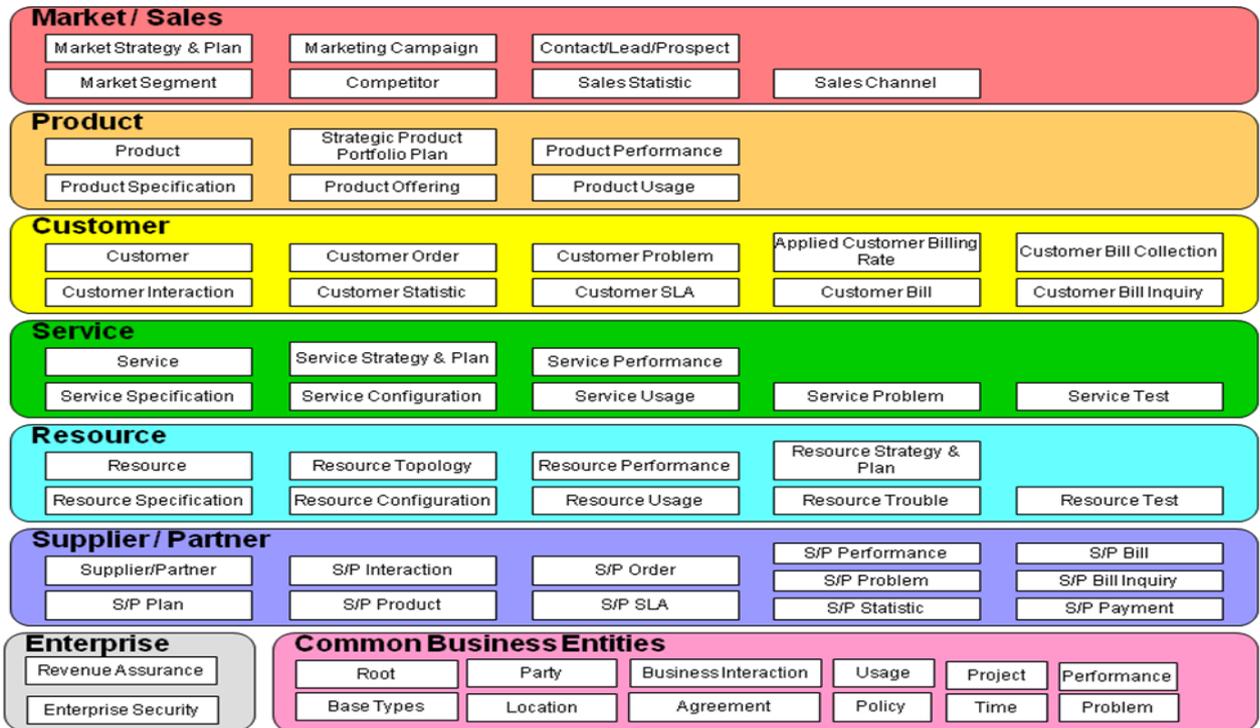


Figure 42: Information framework domains & level1 ABEs

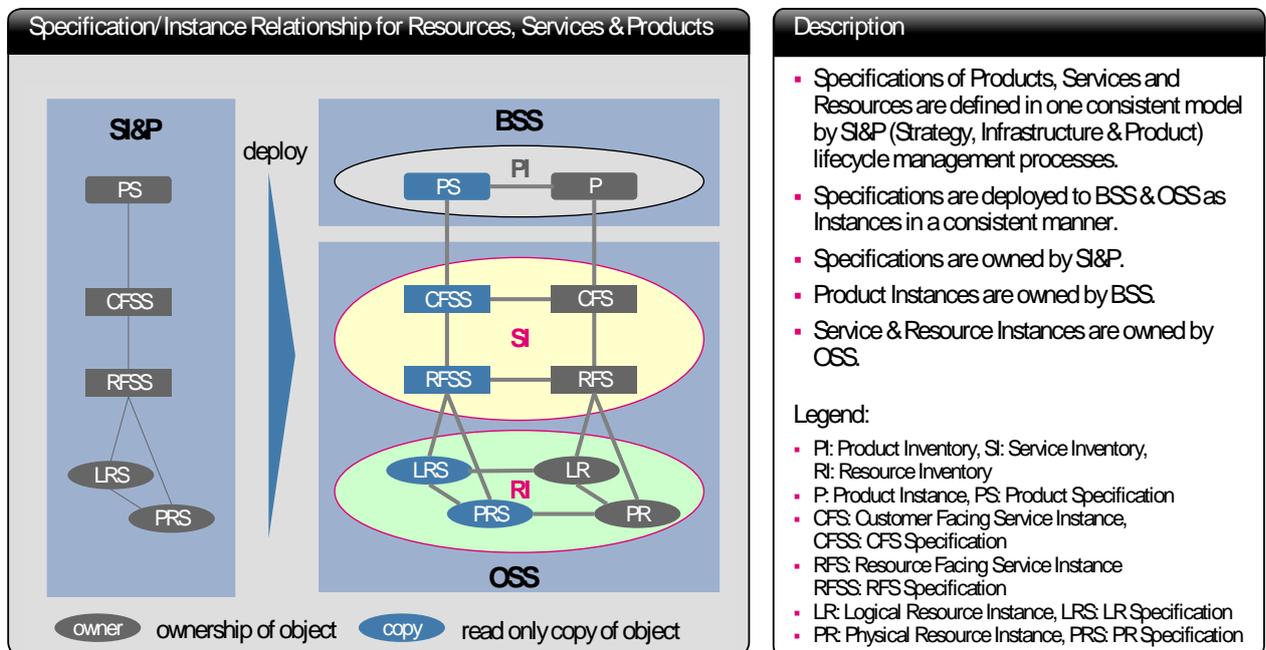


Figure 43: Instance and specification relationships for resources, services and products

Service Domain and Resource Domain modelling with closer focus are including:



The **Service Domain** consists of a set of layered ABEs that are used to manage the definition, development, and operational aspects of Services provided by a framework system. Entities in this domain support various business process framework processes that deal with the definition, development and management of services offered by an enterprise. This includes agreement on service levels to be offered, deployment and configuration of services, management of problems in service installation, deployment, usage, or performance, quality analysis, and rating. Finally, this domain also includes entities to perform planning for future offerings, service enhancement or retirement, and capacity.

The **Resource Domain** consists of a set of layered ABEs that are used to manage the definition, development, and operational aspects of the information computing and processing infrastructure of a framework system. It supports the business process framework processes that deal with the definition, development and management of the infrastructure of an enterprise. This includes the components of the infrastructure as well as products and services that use this infrastructure.

The resource domain has three important objectives. The first is to associate resources to products and services, and provide a detailed enough set of resources entities (organized as ABEs) to facilitate this association. The second is to ensure that resources can support and deliver Services offered by the enterprise. Management of resources involves planning, configuration, and monitoring to capture performance, usage, and security information. This also includes the ability to reconfigure resources in order to fine tune performance, respond to faults, and correct operational deficiencies in the infrastructure. Resources also provide usage information which is subsequently aggregated to the customer level for billing purposes. The final objective of the resource domain is to enable strategy and planning processes to be defined. Entities in the resource domain may be associated with processes that involve planning new and/or enhanced Services, or even the retirement of services, offered by the enterprise.

From Service Inventory and Resource Inventory Management functionality point of view the modelling provides:

Service Inventory Information Model

This function is the underlying information model for the service instances to be managed. The model serves as the foundation for the data itself and a guiding force for the definition and modelling of new services.

The service inventory information model should evolve in close coordination with the service specification data model, since the service inventory model must be able to store instances designed in accordance with all the service specifications defined via the service specification management system.

Typically, the service provider would need to add a lot of detail concerning the services to be managed. The suggested approach for the service provider is to start with the TM Forum SID service model and then specialize the model for the specific services to be managed. The service model should indicate or point to the supporting component services and resources for each service (the SID model, in fact, does do this).

Resource Inventory Information Model

The assumption is that this feature implements the standardized information model for the resources to be managed. Typically, the service provider would need to add a lot of detail concerning the resource attributes that are to be managed. The specific details will depend on the particular resources (e.g., particular types of managed elements and equipment) and associated technologies (e.g., SONET/SDH, ATM and Ethernet) to be managed. The suggested approach for the service provider is to start with the TM Forum SID model and then define or make use of an existing model that specializes the SID model for the specific technologies that need to be managed.

6.4.2.4 TMF Inventory Interfaces / APIs

6.4.2.4.1 MTOSI

Introduction:



Multi-Technology Operations Systems Interface (MTOSI) is an XML-based operations system (OS)-to-OS interface suite. The Network Management System-to-Element Management System communications is a special case and is defined by the **Multi-Technology Network Management (MTNM)** standards.

MTOSI covers both service and resource level interfaces. MTOSI supports the management of these technologies: SONET/SDH, PDH, DWDM, Ethernet, DSL, ATM, and Frame Relay. Support is planned for T-MPLS, PBB-TE, GPON, and control plane management.

MTOSI uses a single interface infrastructure and applies the same patterns across multiple technologies. The interfaces are specified in WSDL, and associated binding to JMS and HTTP are also specified. The resource part of MTOSI is based on the MTNM information model, with some extensions for coarse-grained operations. The service part of MTOSI is based on the SID service model with the addition of specific operations.

Concerning inventories, within MTOSI Release 2.0 there are two interface specifications, for resource and service inventories. While the resource inventory interface was intensively investigated, the service inventory interface was not elaborated with the same level of details.

The **MTOSI Manage Resource Inventory (MRI)** interface allows an OS to retrieve all or part of the resource inventory known to another OS, and also allows an OS to send resource inventory update notifications to a set of interested OSs.

The MRI addresses the following management capabilities:

- General Management such as (among others):
 - Bulk inventory retrieval (retrieving selected information in a single operation)
 - Multi-Object Inventory Update
- Inventory Management of Connection Oriented Technologies
- Inventory Management of Connectionless Technologies
- Inventory Notifications

The source documents relevant to the work in NGMN concerns the MRI **Business Agreement (BA)** that covers use cases and the requirements for management of resource inventories:

- Manage Resource Inventory - DDP BA, TMF518_MRI, Version 1.1, May 2008.

The **MTOSI Manage Service Inventory (MSI)** interface addresses the following management capabilities:

- Service Inventory Retrieval with support of Bulk retrieval (retrieving selected information in a single operation)
- Service Inventory Update

The source document relevant to the work in NGMN concerns the MSI Business Agreement (BA) that covers use cases and the requirements for management of service inventories:

- Manage Service Inventory - DDP BA, TMF518_MSI, Version 1.0, May 2008.

6.4.2.4.2 1.4.2.4.2 OSS/J

Source: OSS/J Inventory API, JSR-142 Overview, Release 1.0, TMF888, TM Forum Approved Version 1.3, January 2010

Scope: The OSS/J Inventory API addresses inventory functions in different areas of service provider operations: customer relationship management, service management and resource management.

The API is specified for **three integration profiles**:

- Java JVT Profile
- JMS Profile or MOM Integration Profile
- Web Services or SOA Profile



OSS/J classifies **inventory information** in three groups focused on products, services, or resources and associates three inventory functions. Each of them has its specific set of inventory entities and relationships, its specific business logic and interacts with different subset of OSS functions. However, all Inventory functions share common abstractions (e.g., entities, associations, entity specifications). These common abstractions are specified in OSS/J as Meta Model entities, called **Core Business Entities** (CBE) which are based on SID.

The OSS/J Inventory API is founded on the following concepts and definitions:

Product Inventory:

- The main responsibility of the product Inventory is to manage the product catalog and keep track of the product subscriptions. The product catalog defines the product offering from marketing perspective and consists of a collection of product specifications. Each product specification describes a product type. Several product specifications may be defined for the same product type. Product specifications are associated with service specifications, stored in the service catalog, thus capturing the relationship between a product and the set of services bundled by this product.
- Each subscription is captured in the product inventory through a product instance associated with the corresponding specification in the catalog. The product instance is also associated with the subscriber of the product and the related subscriber account information.
- SLA management, trouble ticketing, and billing and customer order management use the information stored in the product Inventory.
- Interactions with other OSS functions:
 - The customer order management function stores in the product Inventory customer details, order and product detail, and account information acquired when a new order is created. Customer order management also retrieves product Specifications from the product Catalog in order to create product instances and to decompose the product orders.
 - The trouble ticketing function may access the product Inventory to correlate a subscriber to a service, and to retrieve details about the subscriber, when creating a trouble ticket.
 - SLA management retrieves subscribers for given products and the subscriber contact information, using the Product inventory.
- Information modelling:
 - The Product Inventory portion of the Inventory Core Information Model defines the following entities and specifications:
 - Product
 - Product Specification
 - Party
 - Subscriber Party Role
- All specific products defined by Product Inventory implementations should derive from the Product entity defined in the CBE Core Model.

Service Inventory

- The main responsibility of the Service Inventory is to manage the service catalog and keep track of planned, subscribed and provisioned services. The service catalog captures the engineering view of the service provider's offering and consists of collection of service specifications. Service specifications define the services from an engineering perspective. Several service specifications may be defined for the same service type. Service specifications are associated with resource specifications, stored in the resource catalog, thus capturing the relationship between a service and the set of resources supporting this service.
- Each subscription is captured in the service inventory through a set of service instances associated with the corresponding specifications in the service catalog, the subscribed product, bundling these services and the end-users (recipients) for these services.



- SLA management, trouble ticketing, billing mediation, service activation, service planning, process quality management, service impact analysis, service problem resolution, customer order management, provisioning control, and service quality management, use the information stored in the service inventory.
- Interactions with other OSS functions:
 - The Service Planning function is responsible for defining new service types, which are detailed using service specifications. The service specifications are then stored in the service catalog for use by the provisioning controller when activating a new service. The service specification contains the engineering information required to provision a service. Note that some of this information is not captured by the service specification itself, but by the associated/supporting resource specifications.
 - The service activation function is responsible for updating the service inventory with service instances and their associations with recipients (users), supporting resources, subscribers (customers), etc.
 - The customer order management function queries service specifications from the service catalog in order to decompose a product order to a set of services to be activated. The information needed to be filled in or selected by the customer is exposed at this point.
 - Provisioning control accesses the service inventory in order to retrieve information required for service or resource design or reconfiguration.
 - The trouble ticketing function retrieves services in order to correlate internally generated trouble tickets with each other and with customer generated trouble tickets.
 - The SLA Management function retrieves service details from the service Inventory in order to configure the monitoring of active services.
 - The service problem resolution function retrieves from the service Inventory details on services in trouble in order to understand what should be done, if anything, to restore these services.
 - The service quality management function performs correlation between services retrieved from the service inventory, and resource performance and traffic measurements, in order to determine service failure or degradation.
 - The service impact analysis function uses the information stored in service inventory in order to correlate network alarms and identify the affected services.
 - The process quality management function performs correlation between services retrieved from the service inventory, and process performance metrics.
- Information modelling:
 - The service inventory Information model uses the following CBE service entities and specifications:
 - Service
 - Service Specification
- All specific services defined by service Inventory implementations should derive from the service entity defined in the CBE core model.

Resource Inventory

- The resource inventory stores network and computing equipment, logical resources and topology. This function is responsible for resource assignment. It keeps track of the physical and geographical configuration of the network, equipment inventory (cards, ports, etc.), physical connectivity, and logical connectivity of the different network layers.
- Resource discovery components populate and synchronize the resource inventory.
- Data in the resource inventory is used by provisioning control to design services and understand where capacity is available. It is also used for root-cause alarm analysis and network activation.
- Interactions with other OSS functions:
 - The resource inventory stores information on available capacity and logical resources and provisioning control accesses this information in order to design a service. Provisioning control also uses information stored in the resource inventory to understand the equipment (e.g., ports, devices) available in the network (especially at the boundaries of the network), when designing the network configuration to support the service.
 - The network activation function retrieves equipment and connectivity details from the network inventory in order to create requests to provision the network. It also stores intended and scheduled changes to the



network in the resource Inventory. Network activation can also create in the resource inventory logical resources (e.g., connections) in support of services.

- Network impact analysis retrieves information from resource Inventory in order to correlate resource faults in the network with logical resources supporting services or to compute the state of related resources affected by these faults.
- Performance monitoring accesses the resource Inventory in cases where data is being collected against a complex resource entity such as a network connection. In order to accurately represent the performance of the resource, performance monitoring must have its topology to identify the appropriate performance data collection points.
- The resource information in the resource inventory database is populated and synchronized by the resource discovery function.
- The root cause alarm analysis function retrieves information from resource Inventory in order to filter multiple alarms from a single resource or group of resources into a single root cause alarm or problem.
- Usage monitoring accesses the resource Inventory in cases where data is being collected against a complex resource such as a network connection, in order to accurately represent the usage of the entity. Usage monitoring must have its topology to identify the appropriate usage data collection points.
- Information modelling:
 - The resource inventory portion of the inventory core information model defines the following entities and specifications:
 - Resource
 - Resource Specification
 - Resource Association
- All specific resources defined by resource inventory implementations should derive from the resource entity defined in the CBE core model.

Furthermore, within OSS/J several key **use cases** have been elaborated to derive the functionality of the inventory API and the Inventory CBE core model. The following actors are involved in these use cases:

- Components participating in the fulfilment process (e.g., service and network activation components, provisioning workflow management components, etc.).
- Assurance applications (e.g. service impact analysis, service quality management, SLA management, etc.).
- Planning applications (e.g., service planning, network planning, etc.).
- Discovery applications (e.g., network discovery, discovery management, etc.).
- Billing applications (e.g., billing mediation, etc.)

The detailed use cases can be taken from the specification source document.

The following **requirements** have been identified for the OSS/J Inventory API:

- It should be possible to create, delete, update and query inventory entities, specifications and associations through the Inventory API.
- The API should allow the creation of inventory entities using specifications. The resulting entities should satisfy the constraints defined in the specifications.
- The API should allow queries of available entity specifications and entity specification types (e.g., query of available product types and the specifications for specific product type from the product catalogs).
- Inventory queries should be able to return large collections of inventory entities, associations and specifications of different types.
- Clients of the Inventory API should be able to retrieve attribute, role, cardinality and other constraints applicable to associations and entities for the purpose of data validation. (The definition and retrieval of complex semantic rules are beyond the scope of this specification.)
- It should be possible to invoke queries and update procedures through the Inventory API. A client should be able to query the named update procedures and queries, supported by the API.
- In case the inventory information is distributed over several inventory components, the Inventory API should provide traversal of relationships (in the context of queries and update procedures) across components.
- The API should provide operations for import and export of inventory information through XML files.



- It should be possible to define the scope using various criteria (not necessarily in terms of containment) for the inventory data exported through an XML file.
- It should be possible to define the type of reconciliation expected for the inventory data imported through an XML file.
- Clients should be notified when inventory entities and associations are created, deleted and modified or resource utilization thresholds are crossed. A client should be able to query the event types, supported by the API.
- The inventory API should allow the management of current and planned views. For example information on resource utilization and allocation of resources should be related to specific state, date or a period of time. No assumptions should be made at the interface level with regard to the capabilities of the inventory system to support current and planned views.
- The inventory API should support a mechanism for loosely coupled and asynchronous XML based interactions for integration with workflow systems or B2B integration.
- The inventory API should be strongly typed.
- The inventory API should use the CBE core information model, based on concepts common to models adopted in the industry. The purpose of the core model should be to capture the essential entities and relationships in order to allow traversal across repositories. The core model should be able to coexist with adopted vendor specific or standard information models. Supporting the core model (or the portion of the core model for specific inventory function) should be mandatory for components implementing the inventory API.
- No assumptions should be made at the interface level with regard to the naming conventions used by specific inventory systems.
- It should be possible to extend the data types, queries and update procedures supported by the inventory API. No assumptions should be made at the interface level with regard to the ability of the inventory system to support dynamic (run-time) schema extensions.
- The inventory API should allow federation of Inventory components.
- The inventory API should reuse the OSS/J patterns and should follow the OSS/J design guidelines.

6.4.2.4.3 TIP

6.4.2.4.3.1 Inventory Team Activities

Purpose:

Over the recent past, several important Inventory Interfaces have been made available and deployed in many industrial projects. In particular, the OSSJ Inventory API, the MTOSI Inventory APIs and the 3GPP IRP addressing network synchronization have been well exposed and successfully used in various solutions. While, most of the time, these Interfaces have been integrated separately in specific domains, Service Providers are now looking for more agile Interfaces which can be applied independently of the domain.

For historical reasons, the OSSJ, MTOSI and 3GPP Inventory Interfaces are not compatible and each address a specific portion of the problem space. As a consequence, the cost of integration remains not optimal for Service Providers.

The TIP inventory interface project that has been started at the end of 2010 will:

- analyze the three Inventory Interfaces mentioned above,
- collect and select use cases and requirements from various sources,
- define a new Inventory Interface which will be applied to different domains in a flexible way, thus reducing the integration tax for Service Providers.

Scope and Objectives:

The TIP Inventory team is responsible for assessing the current state and outstanding enhancement status of the varied Inventory interfaces with a view to near term solution consolidation and ongoing evolution to the emerging TM Forum Integration Framework.

- Consider Inventory for:



- Resource
- Service
- Product ? (*note: it is not yet clear if it will be in scope*)
- Include all the corresponding and relevant artefacts aligned with the TM Forum frameworks (business process, information, application, and integration frameworks)
- Consider a flexible approach where different specific information models could be used
 - from the Telecom domain (e.g. incorporating wireless and wireline models), or
 - from other domains (e.g. Information Technology).
- Delineate configuration management versus inventory management, but still include Inventory update.

Status:

The TIP Inventory team released the comparison study of OSS/J, MTOSI and 3GPP inventory interface approaches (see **chapter 1.1.1.8**). The work on a **Feature Description Document (FDD)** that includes use cases and requirements for inventories has started. The team is now waiting for requirements and use cases from the operators, provided by the NGMN NCOR project, to continue its work.

The status of work (01/2011) has been summarized in the following document:

- TMForum Interface Program, Inventory Interface, Progress Status at TAW Paris (Jan 2011).

6.4.2.4.3.2 1.4.2.4.3.2 Business Service Activities for Cloud and IT

In the context of the cloud and IT business service activities in TIP, two documents have been delivered that should be evaluated by NGMN regarding use cases and requirements for resource, service and product inventories:



Inventory
Retrieval

File 1: Product, service and resource inventory retrieval for cloud and IT

Inventory Update

File 2: Product, Service and resource inventory update for cloud and IT**6.4.3 3GPP****6.4.3.1 Basic Inventory Management Concepts****Basic inventory management concepts from 3GPP TS 32.690 version 10.0.0 Release 10**

The main task of the 3G network inventory management is to manage network inventory information about the various static resources of a 3G mobile telecommunication network. It provides support to network planning, to network operation and maintenance and to working craft management. Inventory management functions are distributed over different layers of a **Telecommunications Management Network (TMN)**. The main task of the inventory management function at Itf-N is to provide an efficient access for network management systems to the static inventory data of all related managed network elements.

The basic tasks of the inventory management IRP of this release are:



- to provide an efficient mechanism enabling IRPManagers to upload inventory data as follows:
 - to request an IRPAgent to prepare inventory data of a certain part of the current network for uploading;
 - to check the status of data preparation in the IRPAgent;
 - to request the IRPAgent to alert the IRPManager when the data preparation is completed; and to
 - to upload the prepared inventory data;
- to provide a standard data format so that all IRPManagers and IRPAgents involved have a common understanding of the uploaded inventory data.

The inventory data:

- is static data about the hardware equipment and firmware units (e.g. line cards, processing units, power supplies) constructing the network elements managed by the concerned element manager. Static data is the data which:
 - is usually provided by vendors and is basically vendor-specific;
 - is basically independent of the operation status of the related equipment/units;
 - is not changed frequently during the normal operation;
 - cannot be changed through Itf-N interface; and is
 - basically independent of configuration management;
- may either be integrated in the related equipment/units or be assigned to the related equipment/units during the installation or during operation;
- may include data showing static physical relations between equipment or units, e.g. card A is in slot B.

The following **requirements** shall apply for Inventory Management over Itf-N:

Inventory data is defined as information pertaining to Field Replaceable Unit (FRU) hardware, firmware and optionally software units of 3G Networks, and shall be manageable. The management of software unit information should be similar to the management of hardware and firmware information. Examples of inventory data and attributes are described in Annex A and Annex B and standardised inventory data for Itf-N is defined in 3GPP TS 32.692 [4]. Examples of inventory hardware units may be rack, shelf, slot, circuit pack and physical port, as long as they are FRUs.

The Inventory hardware information can be captured as a hierarchy or a flat model. In a hierarchical model, an inventory unit is contained by another inventory unit, thereby creating a containment relationship.

It shall be possible for the IRPManager to initiate the upload (IRPAgent to IRPManager) of inventory data over Itf-N. It shall be possible to scope the inventory data to be uploaded from the IRPAgent, e.g. inventory data for a NodeB, an RNC, or all the NEs managed by the IRPAgent.

It shall be possible to filter the inventory data to be uploaded from the IRPAgent, e.g. HW units of a certain type in the network.

It shall be possible to check the status of an Inventory Management operation.

Interface-N shall support a file-based mechanism for transferring inventory data.

The file format used for transferring of bulk inventory data shall include a standard part and shall also allow for vendor specific representation of inventory data. The meaning, syntax, units, etc. of the standard part of inventory information will be specified, e.g. standard fields for HW board identity (including version number of HW/SW/FW), board type and serial number.

A network resource model shall be defined for the standard part of inventory data.

As the files are transferred via a machine-machine interface, the file format shall be machine-readable using industry standard tools, e.g. XML or ASN.1 parsers.

The file format shall be specified by using a standardised language, e.g. the extensible mark-up language (XML).

The file format shall be independent of the data transfer protocol used to carry the file from one system to another.

The file transfer facility shall be implemented using a file transfer protocol as defined in 3GPP TS 32.101 [1].

The identification of IOC instances shall be consistent with alarm reporting and the network resource models used for configuration management.

All inventory units shall be uniquely identifiable.



6.4.3.2 1.4.3.2 Inventory Management Network Resources IRP Model

Network Resource Model from 3GPP TS 32.691/2/6 version 10.x.0 Release 10.

The IM NRM specifications provide solution in XML for the requirements:

- The NRM defined by this IRP shall provide inventory data over Itf-N of network entities in the 3G network.
- The NRM defined by this IRP shall provide inventory data over Itf-N of network entities in the EPS network including those installed by SON functionality.

The current solution provides a recursive structure of a single Object Class InventoryUnit.

Interface/Operations from 3GPP TS 32.691/2/6 version 10.x.0 Release 10:

Inventory management (IM), in general, provides the operator with the ability to assure correct and effective operation of the 3G network as it evolves. IM actions have the objective to monitor the actual configuration on the network elements (NEs) and network resources (NRs), and they may be initiated by the operator or by functions in the operations systems (OSs) or NEs. The final goal of IM is the establishment of an accurate and timely model of the actual inventory in the NEs or NRs.

IM actions may be requested to reflect changes initiated by configuration management (CM) actions or to make sure that the inventory model is in synch with the actual inventory. IM actions are initiated either as single actions on single NEs of the 3G network or as part of a complex procedure involving actions on many resources/objects in one or several NEs.

According to the above the transfer of the specified inventory data is by using bulk CM (3GPP TS 32.611/2/5 v10.x.y characterised by Bulk (file-oriented) data retrieval (configuration parameters) over Interface-N from single NEs, a collection of NEs or the whole network.

6.4.3.3 Alignment of 3GPP Generic Network Resource Model (NRM) Integration Reference Point (IRP) and the TMF Shared Information/Data (SID) /MTOSI Model

3GPP and TM F are working in joint work group for alignment 3GPP NRM and TMF SID/MTOSI models. Conclusion related to inventory management are presented in the TR 32 828 in a following way

The inventory model information in previous chapters shows clearly that both 3GPP and MTOSI inventory models are defined for hardware inventory. This view is visible in both object definitions and object attribute definitions. Both models also follow similar modelling approach where managed element contains hardware related inventory units with relatively simple predefined structure. 3GPP hardware inventory unit model is very generic (contains inventory unit that may contain another inventory units) and the interpretation is seen as an implementation issue of vendors i.e. how to implement inventory unit object hierarchy. MTOSI specification defines the relationship of equipment holders and equipment thus somewhat more specific than 3GPP model. Anyhow the 3GPP inventory model enables this equipment holder – equipment modelling. TMF SID defines objects for hardware and software modelling purposes. The hardware model is similar (but richer) to MTOSI model. SID defines equipment and equipment holder in more detail. SID defines an object for software also unlike 3GPP and MTOSI. And notably, the relationship between software and hardware objects (PhysicalContainer) is also defined.

Some NRM and Interface/Operations model changes are proposed in TR 32 828 Network Resource Model enhancements proposed in 3GPP TR 32.828 version 1.5.0

The technical report to be finalised in May 2011 proposes a new object class structure in alignment with TMF SID.

Network element consists of one or several logical functionalities. The logical functionalities are realized by one or several HW units, related SW and controlling licences combinations. Current 3GPP inventory NRM captures



already the NE level information in ManagedElement object. NE may contain one or several logical entities which may again contain either logical and/or physical entities. Current model enables only HW related split. To address this hierarchy of the logical and physical entities, the recommendation is to add following objects

- **inventoryUnit NE** representing and realizing logical and physical structure of the Network Element. inventoryUnit NE consists of HW units, SW units and licenses (LIC) controlling the functionalities. The SW and LIC may be the same for several entities of same type within one InventoryUnit NE. Same applies to HW in case of shared resources. To address properly the HW, SW and license items, the recommendation is to add following objects
- **inventoryUnit HW** (hardware)
- **inventoryUnit SW** (software)
- **inventoryUnit LIC** (license)

Interface/Operations enhancements proposed in 3GPP TR 32.828 version 1.5.0

Establish the method "FT IRP / Inventory NRM IRP" as a viable alternative for transferring inventory information.

The inventory data could be stored per network element in an inventory file. The files would be available for upload per request or per schedule using existing File Transfer IRP (3GPP TS 32.341/2/6) capabilities, which already allows for inventory data transfer.

File Transfer IRP makes reuse of generic transfer concepts defined by the file transfer protocol applied and has additional relevant operations/notifications:

- listAvailableFiles
- notifyFilePreparationError
- notifyFileReady

6.4.3.4 Subscription Management

3GPP specifications are addressing service management layer information handling especially in relation to Subscription Management.

The 3G environment requires more complex service delivery mechanisms than in 2G. The following drivers are leading to a need to standardize SuM interfaces:

- Use of different vendor's equipment for 2G/2.5G and 3G.
- The trend in 2/2.5G toward the support of virtual network operators and content providers requiring standardized interfaces amongst them.

Service delivery and support across multiple vendors' solutions and organizations is a feature of other industries, and the solutions are adopted are secure supply chain solutions based upon mainstream e-commerce principles, methods and technologies.

SuM is an area of service operation management that permits service providers and operators to provision services for a specific customer service subscription.

Specific 3G areas that SuM requirements must address are:

- Subscription information is distributed across in a number of locations including the home network, the Visited Network, the user equipment, application VASP equipment (e.g. servers accessed by the subscriber for content and information based services).
- SuM will allow service providers and operators to provision, control and monitor the subscription information.



- SuM is not simply an internal matter for a single operator but a capability that is achieved by linking together features across multiple operators' operations support systems (OSSs).
- SuM will need to manage subscription information in e.g. the OSSs, HSS, UE, OSA, MMS and IMS subsystems.

SuM is concerned with provisioning the subscription profile throughout all the systems and trading partners needed to realize the customer service, SuM provides specifications that define the interfaces and the procedures that interconnect the three points of the SuM triangle: customer care center, the user and the network (s) where the subscription profile resides (such as HSS, USIM, etc.).

SuM, in particular the configuration of resources, aligns with subset of the eTOM model in the area of fulfilment.

eTOM fulfilment processes	eTOM level 2 process	eTOM level 3 processes
CRM Fulfilment Order Handling	No	
Marketing Fulfilment Response	No	
Selling	No	
SM&O Service Configuration and Activation	Yes	Implement, Configure & Activate Service
RM&O Resource Provisioning S/P Relationship	Yes	Configure & Activate Resource
Management S/P Requisition Management	No	

Table 7: Relationship between SuM and the eTOM model

SuM manages subscriptions in the form of subscription profile components. The subscription profile components may be distributed across service management & operations (SM&O), resource management & operations (RM&O) and network domains in order to easily configure resources and support services at the network operations management level.

There may also be mappings of subscription profile components between the SM&O, RM&O and network domains. In particular, such mapping may exist between a model of services and service parameters in the SM&O layer and the model of service parameters in the SuM NRM. Similarly, such mapping may also exist for identifiers and the concepts of user and subscriber as found in the SuM NRM to/from other representations

6.4.4 ITIL

6.4.4.1 Closely Related ITIL Concepts

ITIL is widely used framework of best practice approaches intended to facilitate the delivery of high quality IT services. It outlines an extensive set of management procedures that are intended to support businesses in achieving value for money and quality in IT operations. Being still devoted to its foundations in support for IT services management the ITIL practices and related system solutions share an analogue problem with telecom inventory in how information about IT infrastructure components and services can be managed. A key concept for that within ITIL is **Configuration Item (CI)**. Similarly like telecom inventory information is used e.g. by other operations process the ITIL CI information is utilized by e.g. ITIL incident management, problem management and change management processes.

The basic definitions related ITIL configuration items and processes/systems are:

The **Service Asset and Configuration Management (SACM)** supports the business by providing accurate information and control across all assets and relationships that make up an organization's infrastructure. The



purpose of SACM is to identify, control and account for service assets and configuration items (CI), protecting and ensuring their integrity across the service lifecycle. The scope of SACM also extends to non-IT assets and to internal and external service providers, where shared assets need to be controlled. To manage large and complex IT services and infrastructures, SACM requires the use of a supporting system known as the configuration management system (CMS).

The **Configuration Management System (CMS)** is a coherent logical model of the IT organization's infrastructure, typically made up of several configuration management databases (CMDBs) as physical sub-systems. It is used to store information on all configuration Items (CIs) under the control of configuration management. CIs are mainly hardware or software items and are characterized by their attributes (recorded in the CI's configuration record) and their relationships to other CIs.

ITIL does not define any standard for data modelling for configuration items. The vendor's solution providing tools for ITIL process support however do include practical approaches how to model IT infrastructure components and services. Typical attributes what CI records include are: unique identifier, name, description, CI owner, classification (category e.g. service, HW, SW and type e.g. server, printer), manufacturer info, version, location, status history (lifecycle) and relationships to other CIs etc.

6.4.4.2 Converging TMF and ITIL concepts – TR143

TMF and itSMF have done a joint technical report for converging TMF and ITIL concepts – TR143 Building bridges ITIL and eTOM. As one of main conclusions of how to relate key modelling concepts of each framework it is expressed:

6.4.4.2.1 ITIL CMS/CMDB

The ITIL configuration management system, with its support for one or more configuration management data bases, provides support for a range of inventory and other information/data repositories that may be used within the enterprise.

The eTOM business process model and the SID information model also support a structured view of this, with process areas within eTOM focused on inventory management: customer, product, service, resource, etc, and within SID, domains and information elements ("ABEs") focused on the same areas.

There is therefore a basis for applying the CMS approach within eTOM, and through SID within NGOSS overall, with a consistent and compatible structuring approach

Guideline G6: The ITIL configuration management data base / configuration management system (CMDB/CMS) can be realized through the SID information model (and supported by processes within the eTOM business process framework) using compatible and consistent information/data structuring in both the ITIL and the SID/eTOM/NGOSS views.



7 ALL REFERENCES

3GPP. (o. J.). TR 32 828.

3GPP. (o. J.). TR 32.828 version 1.5.0.

3GPP. (o. J.). TS 32 140.

3GPP. (o. J.). TS 32.101.

3GPP. (o. J.). TS 32.300 Telecommunication management; Configuration Management; Name convention for Managed Objects.

3GPP. (o. J.). TS 32.341/2/6. Abgerufen Juli 18, 2011f, von about:blank

3GPP. (o. J.). TS 32.611/2/5 v10.x.y. Abgerufen Juli 18, 2011g, von about:blank

3GPP. (o. J.). TS 32.622.

3GPP. (o. J.). TS 32.622 Generic network resources IRP: NRM.

3GPP. (o. J.). TS 32.690 version 10.0.0 Release 10.

3GPP. (o. J.). TS 32.691/2/6 version 10.x.0 Release 10.

3GPP. (o. J.). TS 32.692.

3GPP. (o. J.). TS 32.xyz series on NRM.

ATM Forum, Technical Committee, Network Management,. (o. J.). M4 Network View CMIP MIB Specification, CMIP Specification for the M4 Interface, Sep, 1995.

CCITT Rec. X.733 specification. (o. J.). . Abgerufen von T-REC-X[1].733-199202-!!!PDF-E.pdf

ITU-T. (o. J.). ITU-T X.733 Correction.

JOSIF Guidebook. (o. J.). . Abgerufen von

http://sourceforge.net/apps/mediawiki/openoss/index.php?title=JOSIF_Guidebook

NGMN Alliance. (o. J.). NGMN Top OPE Requirements Version 1.0, Abstract.

NGWW. (o. J.). 3GPP SA5-TM Forum model alignment JWG meeting in Budapest, April 4-6, 2011.

NGWW. (o. J.). Fixed Mobile Convergence (FMC) Network Management – Federated Network model (FNM) Umbrella, Version 1.2 JWG meeting in Budapest, April 4-6, 2011.

S5-102610 S5vTMFa033 E NSN Proposed enhancement of Generic NRM IOCs v3. (o. J.). .

TMF. (2008a, Mai). DDP BA, TMF518_MRI, Version 1.1.



TMF. (2008b, Mai). DDP BA, TMF518_MSI, Version 1.0.

TMF. (2011, Januar). TMForum Interface Program, Inventory Interface, Progress Status at TAW Paris.

TMF. (o. J.). GB922, Information Framework (SID) Suite, Release 9.0. Abgerufen von

<http://www.tmforum.org/browse.aspx?catID=9285&artf=artf2048>

TMF. (o. J.). MTOSI 2.0. Abgerufen von

<http://www.tmforum.org/MTOSIRelease20/MTOSISolutionSuite/35252/article.html>

TMF. (o. J.). MTOSI 2.0: Network Resource Fulfillment DDP IA, TMF612_NRF, Version 1.0.

TMF. (o. J.). TM Forum Information Framework (SID) Suite; Release 9.5. Abgerufen von

<http://www.tmforum.org/Guidebooks/GB922InformationFramework/45046/article.html>

TMF. (o. J.). TM Forum MTOSI Rel. 2.1 supporting document SD2-5_Communication_Styles.

TMF. (o. J.). TM Forum SID Rel. 9.5.

TMF, & itSMF. (o. J.). TR 143 Building bridges ITIL and eTOM.

TR143. (o. J.). . Abgerufen Juli 18, 2011, von about:blank

TR166 - Federated Information Framework - Concepts and Principles - v0.1.docx. (o. J.). .

UML Superstructure Specification, v2.1.1. (o. J.). .