Deliverable D2.1

UMF Specifications

Release 1

Grant Agreement	257513	
Date of Annex I	26-07-2010	
Dissemination Level	Public	
Nature	Report	
Work package	WP2 : Unified Management Framework	
Due delivery date	01 July 2011	
Actual delivery date	20 July 2011	
Lead beneficiary	TCF	Gerard Nguengang Gerard.Nguengang@fr.thalesgroup.com

SEVENTH FRAMEWORK PROGRAMME

FUTURE NETWORKS

Authors	TCF – Gerard Nguengang (editor), Mathieu Bouet, Marie-Noëlle Lepareux ALBLF – Leila Bennacer, Laurent Ciavaglia, Samir Ghamri-Doudane, Pierre Peloso, Benoit Ronot ALUD – Markus Gruber FT – Christian Destré, Imen Grida Ben Yahia, Zwi Altman TIS – Antonio Manzalini, Roberto Minerva TID – Alfonso Castro, Beatriz Fuentes Fraunhofer – Mikhail Smirnov VTT – Teemu Rautio, Jukka Mäkelä, Liinasuo Marja UCL – Alex Galis, Giovanni Toffeti, Stuart Clayman, Marinos Charalambides UniS – Majid Ghader, Stylianos Georgoulas NKUA – Alexandros Kaloxylos, Apostolis Kousaridas, George Katsikas, Panagiotis Spapis, Makis Stamatelatos UPRC – Panagiotis Demestichas, Kostas Tsagkaris, Vera Stavroulaki, George Athanasiou, Panagiotis Vlacheas, Yiouli Kritikou, Nikos Koutsouris, Aimilia Bantouna, Dimitris Karvounas, Evagelia Tzifa, Assimina Sarli, Evangelos Thomatos, Marios Logothetis, Andreas Georgakopoulos, Louiza Papadopoulou

Executive summary

UniverSelf core motivation is to arise as a driving force, with the duty of leading the autonomic networking research field into maturity by generating high industrial impact, keeping a business focused approach and federating the various valuable research results that have already been obtained. The design of a Unified Management Framework (UMF) that targets the embedding of autonomic paradigms in any type of network in a consistent manner is part of this great challenge.

The Deliverable 2.1 presents the first release of the UMF specifications. It provides an initial description of the UMF, its perimeter and its enablers. First, the "cleaned state" approach adopted by UniverSelf is motivated, follows a deep analysis of existing management and autonomic networks architectures. The positioning of UMF with respect to existing networks management systems is also detailed. Then, the analysis of the top down requirements related to UMF as well as the requirements defined in D4.1 based on specific use cases is presented. The current deliverable identifies and specifies the UMF core functional blocks and interfaces. Moreover, it addresses the technological challenges of knowledge and information management, network governance and intelligence embodiment. Candidate solutions are studied and a working approach is defined. The outcomes of these approaches will be described in the next release of UMF specification that is Deliverable 2.2.

Table of Contents

Foreword

Deliverable D2.1 represents a first concise description of the UMF design, as derived from the primary contributions of Task 2.1 (UMF Design) of work package 2 (Unified Management Framework) of the UniverSelf project.

According to the project lifecycle, the requirements expressed in work package 4 are transferred to work package 2 to guide the design of the Unified Management Framework (UMF). Work package 2 and specifically task 2.1 produces a specification, in terms of identification of required functional modules for the UMF, its interfaces and models, which also addresses the requirements deriving from the use cases handled by the project. The set of UMF-compliant specifications are first distributed inside all the tasks of work package 2. The embodying tasks of work package 2 will work together to design solutions to embed the algorithms provided by work package 3 (Network Empowerment). This design will respectively cover network monitoring and knowledge building aspects in task 2.2 – Information and Knowledge Management, enforcement of high level goals in task 2.3 – Network Governance and enabling mechanisms in task 2.4 – Intelligence Embodiment. Once the UMF functional specification of the embedded autonomic functions is completed, task 2.1 will feed the integrated solutions specifications to task 4.2 of work package 4 in order to instantiate and evaluate these autonomic functions through combination of simulations, emulation and prototype implementations . The interactions between work packages 2 and 4 and the relative synchronization points through specific documents are depicted in Figure 1 of deliverable D4.1.

The UMF design will be developed across three documents; each one corresponding to one UMF release, namely deliverable D2.1 (UMF release 1), deliverable D2.2 (UMF release 2) and deliverable D2.4 (UMF release 3). The scope of these deliverables, which is in line with the Description of Work and also reveals what each UMF release addresses, is as follows:

D2.1 – UMF Specifications – Release 1: The deliverable will feature a first concise description of the UMF design (primary contributions from task 2.1). This version of the UMF design will describe the basis (objectives and approach) for achieving the target of embodying autonomic paradigms in any type of network, in a consistent manner, spanning widely different technological contexts, and providing to operators a service-oriented abstraction of the network they are operating. Deliverable D2.1 will comprise the fundamental elements for achieving a network agnostic management of services, embedding advanced service and network management intelligence, and federating the management of multiple networks, hence, bridging wireless, wireline, access, core, services, etc. The fundamental elements comprise governance, information management, and feature embodiment (comprising the cognitive part) functions. This UMF core will be flexible enough to accommodate for different networking scenarios and use cases in a consistent manner. The specification will also address requirements deriving from the first burst of use cases that will be handled by the project. Emphasis will be placed to compatibility with existing and emerging industry standards, the incorporation of recent results from research, and to achieving a future-proof design. In particular, the UMF release 1 focuses on the identification of the common functional groups and their interfaces; the possible organization and cooperation modes between UMF elements and domains; an attempt to a system view of the UMF which includes the introduction of a number of specialized logical nodes and of a possible hierarchical structure, a discussion on orchestration issues, as well as a mapping of the identified functional blocks into these nodes and the elaboration on their functionalities and interfaces among them. The positioning and mapping of the UMF (and of its components and interfaces) onto deployed and standardised control and management architectures, which is an essential aspect for the industrial impact, is initiated in this document and will be further progressed in the next releases.

D2.2 – UMF Specifications – Release 2: The deliverable will be a first complete specification of the UMF. The specification will focus on the information and knowledge management capabilities, the governance mechanisms, and intelligence embodiment functionality. It will also address requirements deriving from the second burst of use cases that will be handled by the project. Enhanced and extensible information and knowledge management mechanisms will be presented, for assuring that UMF always performs informed decisions at the system and network levels. A continuum of governance tools (i.e., means for visualization, applying policies and associated languages, managing information-models/ontologies, etc.) will be specified, aiming at making a UMF-empowered self-managed network controllable by the human operator through high-level mechanisms. Embodiment mechanisms will be comprised, enabling the introduction, deployment and

orchestration of intelligence in the network in a plug and play fashion. The specification in deliverable D2.2 will be at a level suitable for standardisation and first certification, in the direction of building consensus and trust from operators and vendors on UMF. In addition to UMF release 1 (deliverable D2.1), UMF Release 2 will focus on the flexibility and change capabilities; and the convergence towards 'Everything as a managed Service'.

D2.4 – UMF Design – Release 3: The last version of the UMF will be applicable to the overall infrastructure, bridging wireless and wireline, as well as access, core and service segments. This version of the UMF will integrate requirements from all use cases handled by the project and will incorporate corresponding network empowerment solutions. Emphasis will be placed to the project-wide harmonization of the UMF components and to the assurance that the specification is ready for deployment. Deliverable D2.4 will provide the latest developments on the federation of management systems, model driven specifications, the information and knowledge management functionality and the context awareness patterns, the continuum of governance tools (cross-referencing, where appropriate, the deliverable D2.3), the intelligence embodiment mechanisms. The document will report on the contributions and those planned for standardisation, as well as with respect to the status and plans of certification activities. In addition to previous UMF releases, UMF Release 3 will focus on the complete description of the intelligence embodiment and network empowerment integration in the UMF and the network and service infrastructure; the definition of migration and deployment strategies.

1 Introduction

UniverSelf aims at overcoming the growing management complexity of emerging and future networking systems through the smooth and trustworthy embodiment and empowerment of autonomic principles and techniques in both services and networks. Significant research in the field of autonomic networking has been done during the past decade without any tangible impact on the way operators and service providers are managing their infrastructures. The reasons of such low adoption of autonomic features in the network management chain can be found in the complexity of the telecommunications ecosystem to be managed, the diversity of existing network management tools and the lack of trust of the operators about autonomic control loops that have not yet been tested in real life. UniverSelf has chosen to follow both top-down and bottom-up approaches in the design of a Unified Management Framework. The top-down approach consists in an analysis of the high-level requirements identified in the project and in the literature (prior art) to define evolvable placeholders for autonomic management functions from the federation of the previous research outcomes. The bottom approach is based on a use case methodology that tackles current and future service providers and network operators' challenges.

The main goal of this deliverable is to provide a first description or sketch of the UMF design. In particular, it represents a first document reporting on the UMF positioning with respect to current management practices and systems, the analysis of both top-level and use case-oriented requirements, the identification and organisation of UMF core functional blocks and a deep study of potential enablers. Deliverable D2.1 shall be considered as a first release of the Unified Management Framework and thus presents the project's initial design of the UMF functions. This release will be complemented by two others releases that will take into account the inputs and feedback coming from the work package 3 (Network Empowerment) and WP4 (Deployment and Impacts), respectively for the integration of the autonomic mechanisms and the implementation and evaluation of the UMF.

The document is structured as follow: Section 2 examines the current issues when managing heterogeneous systems and motivates the introduction of the UMF. Section 3 outlines the criteria used to prior-art autonomic management/networking architectures and presents the conclusions of this analysis. Section 4 gives a clear positioning of the UMF in the landscape of existing management systems and architectures. Section 5 describes the design of UMF following top-down and bottom-up requirements. The UMF functional blocks are identified and classified into four main functional groups. Section 6 highlights technological challenges that will be addressed by UMF enablers and provides some initial propositions on how to handle them. Section 7 concludes the deliverable by summarising the outcomes of this first release and by elaborating on the next steps. Finally, a number of Annexes provides additional information for several aspects of UMF as follows: Annex A provides the mapping between the requirements and the UMF Functional Groups ; Annex B provides initial message flow between functional blocks; Annex C provides a state of the art analysis in intelligent embodiments; Annex D provides a state of the art analysis in information modelling from the standard organization point of view; Annex E provides a review of the information models developed and used in research projects; Annex F provides a state of the art analysis in policy-based management; Annex G provides the human network operator questionnaire use in the UniverSelf project (link to network governance); Annex H provides a state of the art analysis in governance systems. References, Abbreviations and Definitions Sections are completing this document.

2 Motivations

In the evolving networking environment, the telecommunication operators are facing a number of issues. One major issue is that their existing network management systems have been designed and deployed in a stovepipe way and the automatic correlation of their data is very difficult to be performed. Thus, it is very difficult to deploy, manage and maintain their equipment and services in a simple, cost efficient and end-to-end way. The situation is expected to become worse since the new systems incorporate a diverse set of heterogeneous network infrastructures. Another issue is that up to now, the human factor is the dominant one in the network management process. The lack (or low-level) of automation leads to slower and less efficient reaction procedures. The difficulty to manage this complex environment seriously affects the process of introducing and managing new products and services for the network operators' customers.

Such increasing complexity and challenges cannot be handled by traditional networking and management schemes. The distribution of the decision making process and the provisioning of autonomic management and control capabilities under the operator's supervision seems to be a promising approach. The key idea is that the networks and services are able to monitor their behaviour, learn about their status, and then execute decisions that are in line with operators' business goals and policies. Therefore, autonomic network management and control consist of a decentralized network empowerment through self-x capabilities, which will assist operators to handle the increasing amount of devices, data, management and control operations, while at the same time keeping the overall control.

Although considerable effort has been placed in the past decade for introducing autonomic functionality in network and service management, the majority of autonomic functionalities were studied typically for a single domain of operation (e.g., access network, core network, services domain). As it is already the case today, in future networks it is expected that different management systems operating for different domains will coexist. It is imperative for an operator to unify their operation and enhance the overall performance of the system in a trusted way. By trust we mean that an operator must be confident that these distributed and autonomic systems will provide their promised efficiency without causing stability issues in the network. Also, the introduction of governance (i.e., setting high level business goals and translating them into policies), and the end-to-end evaluation of the network operation are expected to improve the management of infrastructures and services.

Moreover, the future networks, where all domains will be empowered with the desired autonomic management functionality, are expected to materialize gradually. During this evolution, legacy management systems will co-exist with autonomic management systems. Support for the existing legacy systems until the end of their lifecycle will be one of the main goals of the operators. Towards this end, the standardisation of a number of interfaces will be needed. Specifically, interfaces are necessary: a) for federation (i.e. the ones among peer autonomic systems or between the autonomic systems and the managed elements), b) between legacy and future autonomic management systems and c) for governance, covering both directions from the network operator side towards network elements and vice versa.

Another challenge concerns the application of autonomicity in an end-to-end manner that still leaves important aspects to be addressed for the empowerment of the network and evolution towards in-network management[1] functionality integration, as well as their cooperation and the cooperation between different topology or technology domains. This explicitly goes beyond the many studies on self-x functions that have unarguably appeared during the last few years. Advances can be also made in the incorporation of learning in the self-x functions and of course, in sharing knowledge between the network nodes. Also, there is a need to ensure that the orchestration of these autonomic functionalities in an end-to-end fashion will provide the desired efficiency and meet the goals set by the operators in a secure and trusted way. Although several proposals claim to have investigated the orchestration functionality, we firmly believe that there is a need for additional work in this area before reaching the desired level of maturity.

[1] In-network management is an approach where management and control functions are distributed and located in or close to the managed network and service elements. The potential benefits are the inherent support for self-management features, higher automation and autonomicity capabilities, easier use of management tools and empowering the network with inbuilt cognition and intelligence. Additional benefits include reduction and optimisation in the amount of external management interactions, which is key to the minimization of manual interaction and the sustaining of manageability of large networked systems and moving from a managed object paradigm to one of management by objective.

UniverSelf aims at identifying the missing parts in existing autonomic management efforts and propose the means to unify their most promising attributes towards a “cleaned state” realization of autonomic networks management, as opposed to more clean slate (i.e., revolutionary) approaches.

3 Prior art analysis

Applying human-like autonomy into the network management plane has resulted in the concept of autonomic management. The first autonomic computing concept was used by IBM [1] in 2001 while several researches have followed towards the same direction. More specifically, these researches have targeted the design of autonomic architectures and frameworks for the management of telecommunication networks and services through the use of the principles of self-management and the MAPE (Monitor-Analyse-Plan-Execute) loop.

A number of such architectures/ frameworks were selected and analysed according to ten criteria (7 basic and 3 supplementary) in terms of principles that often characterize autonomic management. The purpose of this analysis was neither to criticize the already developed architectures nor to classify them, e.g. according to how autonomic they are, but to identify their re-usable features and the potential gaps and areas that need to be improved or become priority in the world of research. The results of such an analysis proved to be helpful when considered during the specifications of the UMF.

European Research Projects

The set of the used architectures included mainly outcomes of European research projects, namely 4WARD [2][3], ANA [4][5], AutoI [6][7], CASCADAS [8][9], E3 [10][11], EFIPSANS (GANA) [12][13], Self-NET [14], BIONETS (SerWorks) [15][16] and SOCRATES [17][18], and individual research initiatives [19] - [43] and/ or proprietary solutions [44] as well. Accordingly, the seven basic autonomic management features considered as criteria are: a) Implementation level, b) Support for governance, c) Support for federation, d) Distribution level, e) Adaptability, f) Managed objects, g) Closed control loops while the three supplementary criteria included information related to a) Self-x functionality(-ies), b) Knowledge representation and c) Embodiment.

The above described analysis with respect to each criterion revealed the following conclusions. These conclusions also constitute today's limitations that are interesting to be addressed for the UMF.

a) Even if most of the initiatives have resulted in the production of research software prototypes, either for the whole architecture or for specific functions or features, none of them has managed to reach standardisation and deployment. More specifically, some of them had enough impact into standardisation area to reach pre-standardisation reports while others were confined in a single technology or domain or focused on specifying functional blocks in order to address a set of requirements with autonomic flavour. Finally, operators' interest in standardized interfaces, models and processes under a unified common framework should also be taken into account.

b) Analyzing governance criterion, i.e. the criterion related to the new way of management based on business goals and through policies, revealed on one hand that governance has been applied in different levels among the initiatives and on the other hand that more attention, necessary for the transition to this new type of management, should be paid in the communication of a human network operator (HNO) and the self-managed network.

c) When considering federation, i.e. support of an end-to-end service view with variant technologies in different domains (wireless, fixed and core at the same time) and enforcement of a flow-through management, the important role of ontology or (meta-)model, in terms of common vocabulary between disparate domains, was identified.

d) Moving to the requirement for distributed functionalities of the architecture raised once more the issue and accordingly the requirement for standardized interfaces (either by specifying new or by capitalizing on existing ones). Attention should also be paid in maintaining consistency of the distributed entities through a policy-based (governance) framework.

e) Adaptability positions the examined efforts with respect to the nature of network operations and self-x functions and their behaviour. Specifically, three main classes were identified, being a) static b) reactive, with perception of the environment and timely response to occurring changes and c) proactive that exhibit some sort of goal–directed behaviour in a continuous way and independently of any occurring change. The investigated architectures demonstrated different adaptability levels. The majority could be characterized as reactive. Four architectures (ANA, Self-NET, E3, and CSMTN) go one step further, integrating learning capabilities in order to change proactively their behaviour.

f) The analysis of the managed objects (legacy or future) and business oriented processes such as Fulfilment or Assurance revealed that their inclusion was merely or not at all captured by the ongoing autonomic activities and that the focus was instead mainly placed in processes intended to lower level decision making, policy derivation, translation etc. However, the way these processes would become really autonomic as part of an overall autonomic operator's solution still remains an interesting working area.

g) One or more self-adjusting closed control loops are by definition performed in an autonomic system during its autonomous operation, and the consideration of multiple control loops was quite common into the studies and the results of the d initiatives. However, their coordination (or "orchestration") for avoiding accidental interactions or network instabilities was not organized in all cases in the same way, although, both the importance of having well coordinated control loops and its bounds with a well designed governance plane, keeping humans to the loop to some extent e.g. for unforeseen problems, were identified.

h) Self-x functionality (-ies) are also interwoven with the autonomic systems. Although the examined initiatives identified a large list of self-x functionalities, with self-configuration, self-optimization, especially focusing on resource usage and self-healing for detecting and diagnosing problems/faults/anomalies being the most popular, more precise definitions are required for avoiding redundancy and overlapping issues. Furthermore, positioning of each self-x function with respect to networks is also needed and attention should be paid in stability, and thus coordination, issues when integrating different self-x functionalities. Finally, advancements can also be made in the incorporation of learning in the self-x functions and of course, in sharing knowledge between the network nodes.

i) All architectures highlight the central role of knowledge, both in terms of types/models of information/data and building blocks and protocols for exchanging them, and in particular of tasks associated with its building, representation, fusion and dissemination for the instantiation of the autonomic management solutions. Although, they all consider the existence of a knowledge base (e.g., profiles, policy rules) that feeds the various involved control loops, in the majority of them the interfaces have not been explicitly described. Additionally, new information models were specified in different levels, based on existing or standardised ones, while ontologies have been introduced for the knowledge sharing or fusion, as a result of the common identification of the need for semantically richer information models. In most of the cases information models and ontologies were partially incorporated in the developed prototypes. Finally, the dissemination of the defined information and knowledge was based on existing protocols and transport mechanisms.

j) The placement of control and management functionality into the network is known as embodiment in the autonomous systems world and can be done either by adding new features in runtime and in a plug-and-play fashion or by enabling the network equipment to explore and evaluate optimum states by itself. Although, the first approach has been followed by a large number of the initiatives in terms of either being driven by embedding capabilities or facilitating them, there is not much report with respect to the second approach.

Other Research Projects and Initiatives are listed in the following table.

Table 1. State-of-the-art on management frameworks – standardisation groups

Autonomic Computing	[10, 11] propose and explore a holistic vision for autonomic computing in which the system as a whole would attain a higher degree of automation than simply the sum of its self-managed parts. Various research initiatives have been based on this motivation. (IBM Study)
Network Self Management and Organization – NESTOR	[12] emphasises the role of a uniform object-relationship model of network resource, in order to allow any kind of manager (human or software) to configure and control the network behaviour. (Colombia University - Sponsored by DARPA – USA)
Complexity Oblivious Network Management – CONMan	[17] is based on the concept of "Network Managers (NM)", which are software agents, distributed in the network devices. It introduces the Module Abstraction that allows the NMs to generically manage all the network entities with the same simple primitives, and allows the managed entities to translate these primitives. (Cornell University - Sponsored by NSF – USA)
The 4D Architecture – 4D	[15] is based on four planes: a "brainless" data plane, a decision plane that controls and manages the network, a discovery plane and a dissemination plane that links the network elements to their managers. (Carnegie Mellon University - Sponsored by NSF – USA)
FAIN	[51] presents a management framework for programmable and active networks.
Future Networks	[18] presents a survey on Architectures for the Future Networks and the Next Generation Internet. (Washington University Survey)
Large scale experimentations	[47] describes testbeds, based on the concept of federation among different parties in a distributed geographical area allows achieving a scalable for testing new paradigms. [48] GENI (The Global Environment for Network Innovations) is a virtual laboratory for future networks which aims at supporting at-scale experimentation on shared, heterogeneous, highly instrumented infrastructure under a collaborative and exploratory environment.

Summarizing the conclusions above, the next high level messages were indentified and considered as a feedback for building UMF:

a) The main challenge would not definitely be to capitalize on a single architecture, but on the contrary, to design a framework that will unify the different approaches i.e. it will ensure that multiple diverse management systems implemented upon different autonomic architectures will be able to interoperate and federate.

b) The federation of heterogeneous administrative domains is a key feature for an end-to-end autonomic framework that will support existing and future services in an optimum way.

c) The application of autonomicity in an end-to-end manner leaves still important issues to be studied for the empowerment of the nodes as well as their cooperation and the cooperation between different topological (e.g., access, backhaul, core) or technology domains.

d) There is plenty of room for standardisation; the lack of which comprises one good reason for why none of the vendors and operators finally adopted any of the above mentioned architectures so far. Particularly, given the analysis above, standardisation is mostly important in the areas that crucial interfaces of the autonomic systems are provided namely, the governance interfaces, covering both interfaces towards elements but also towards human operator, on the one hand and the interfaces for federation i.e. the ones among peer autonomic systems or between the autonomic systems and the managed elements on the other hand.

e) A unifying solution should provide means for accelerating fixed and mobile convergence since network operators' services span technological boundaries between the wireless and wireline domains and that these services need also to be (self)-managed just like the systems supporting them.

f) Although the maturity level of self-x functionalities is good, further advancements can still be achieved by, for instance, unifying a large set of self-x functionalities, possibly belonging to different classes (self-configuration, self-optimization etc.). However, attention should be paid to their coordination and synchronization so as to avoid any misbehaviours and instabilities coming from their combination.

g) Finally, further advancements can also be made in the incorporation of learning in the self-x functions and of course, in building mechanisms that will embed these self-x functions into the network nodes towards empowering the network with embedded type of intelligence.

4 UMF positioning and definition

UMF shall enable a higher manageability of networks and services for the operators. The main requirements driving its design stem from the UniverSelf project objectives, which are namely: multi-faceted unification, network empowerment, industry impact and adoption fostered by means of trust.

To achieve these goals, UMF targets a management framework that defines how control and management intelligence can be embedded into existing management systems (e.g., EMS, NMS, OSS) as well as networking devices (e.g., routers, base stations) and service control and operation devices (e.g., IMS, VoIP servers). Thus, the UMF consists in a reference management framework specified to enable autonomicity composed of the framework itself and its internal and external interfaces, a unification structure for all autonomic networking functions and a set of enablers and toolbox in support of the operations life-cycle of the framework. The framework consists in a prescribed grouping of functions/enablers and their interfaces. It comprises a set of components (i.e. procedures, data structures, state machines, etc.), the common enabling functionality, the characterization of their interactions (i.e. messages, calls, events, etc.) and the non-functional qualities (i.e. performance, optimisation, integrity, scalability, robustness, flexibility, usability, programmability, etc.) The unification structure combines a set of common services for the integration of all control and management functions (i.e. integration and interworking of all self-x functions, coordination and interworking of all closed control loops in the managed/controlled systems, integration and interworking of heterogeneous managed objects) and a uniform set of interworking interfaces between management/control functions that allows introduction and migration of new functions without affecting the integrity and stability of the full system. UMF clearly defines the path to empower networks and services with self-x algorithms, procedures and tools. These are used to automatically create, deploy, activate, bootstrap, optimize, maintain, test, heal and deactivate network and service components. This automation is achieved via observation, awareness, cooperation, and embodiment of cognition and prediction mechanisms.

Then, UMF realizes an operator-governed ecosystem by unifying today's separate management systems in the different network domains (e.g., access, backhaul, core, services) and technologies, and hence breaking down current management silos. The objective is a transition towards a complete managed service-centric view of the network's and application's resources, realizing a substantial reduction in systems lifecycle OPEX.

UMF also enables trust in the self-x decision processes since all self-x actions are transparent, verifiable, and accountable. It offers a stability framework providing the handles to readjust or deactivate autonomic closed loops, orchestrate their behaviour, resolve conflicts and restore the system in a stable state (targeting both preventive and corrective actions).

A fundamental characteristic targeted by UMF, among the other frameworks on autonomic networking, is its impact on operators and manufacturers. Therefore, the positioning of UMF within the current and future management landscape is a key enabler for its large deployment. UMF shall not impose a radical shift on how network management will be performed but rather enable the introduction of new management functions that gradually will achieve a productive end-to-end network and service management. Therefore, UMF deployment scenarios are an important aspect that shall be thoroughly addressed in order to ensure that UMF is both legacy-compliant and future-proof.

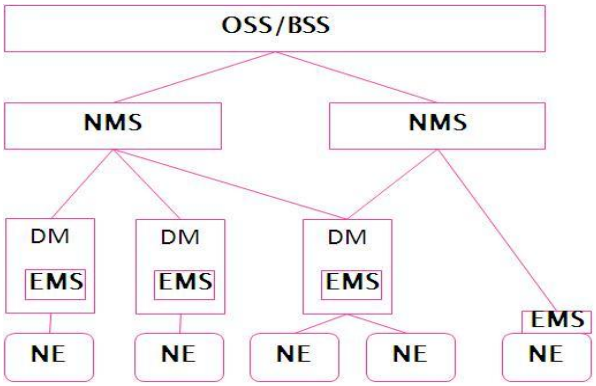


Figure 1. Traditional Operators organisation.

So far, operators and manufacturers follow a hierarchical model supported by several tools, interfaces, and standards. A generic reference model is composed of Networks Elements (NE), Element Management System (EMS), Domain Manager (DM), Network Management System (NMS), and Enterprise Management (OSS/BSS) [3GPP], as depicted in Figure 1. These are interacting according to different scenarios. On one hand, the EMS could be embedded within the NE, or could be specific to a NE with respect to the manager-agent paradigm. EMS could also be common to multiple NE coming from the same vendor and supporting multi technology. On the other hand, the NMS could be common to multiple EMS from different domains, or a specific NMS is needed for each EMS per domain manager.

On one hand, this model already faces serious problems (e.g., the different management entities are built in a stove pipe approach and their inter-working is not an easy task). These problems are expected to become more complex when the deployment of next generation networks such as LTE/SAE and future networks, supporting a number of different technologies, is going to materialize.

On the other hand, the operators have already deployed network management systems and they are supporting their networks in a well-established (even though complex and sub-optimum) way. Thus, it is expected that UMF will achieve its goals by supporting all realistic deployment scenarios. To illustrate these deployment scenarios, we provide the following definitions.

A UMF node can be either:

1. An existing network management system (e.g., EMS, NMS) enhanced with the required capabilities to communicate with other UMF nodes (e.g., through the use of wrappers). We call these UMF nodes as ***Enhanced Legacy Management System (ELMS)***
2. A network connectivity node (e.g., router, BS), which in the future will have embedded autonomic management functionality and resources. We call these UMF nodes ***Future Management Systems (FMS).***

A UMF domain is a collection of UMF nodes and network/service resources, which are on demand established and dynamically maintained and where management can be applied uniformly (e.g. administrative domains, type of networking segments: core, access, virtual, service domains, etc.).

When setting up the deployment scenarios for UMF, it is important to keep in mind that not all the network elements support empowerment. Some network elements (especially legacy ones) are not capable of self-management or empowerment: there is no probe, they have limited Operation System and reduced functionality in the north interfaces. Besides, in future networks, the potential transfer of management functions to the Network Elements (NEs) will be achieved progressively (using an incremental approach). Some pieces of the network (close to access points) have few management capabilities as they need to have as small a price as possible. Also, one of the main operators' objectives is about leveraging current and past investments as long as possible, which, by the way, are assuring revenues in the short and medium term. As such, operators will prefer a properly controlled and smooth evolution of current management chains NE - EMS - NMS - OSS - BSS. Indeed, technology is only one aspect of the process; capability of investments and regulatory are other important variables. Even more human factors should not be underestimated.

Therefore, through its deployment scenarios, UMF shall allow the operators to adopt gradually the specified functionalities and to envisage different deployment and intelligence embodiment strategies.

More specifically, the operators can initially deploy some functionality into some upgradeable network management systems (e.g., ELMSs). Using only ELMSs the operators should be able to use the same interfaces that are going to be simply enhanced with additional communication messages.

As a next step the operators will be able to introduce a number of ENEs. UMF will specify how these ENEs will be supported and interwork with ELMSs, how this larger number of management systems will be organized in the exchange of information and knowledge and how conflicts between these distributed and concurrent entities will be resolved.

5 UMF design

This section presents our first achievement on the design of the UMF. The release 1 of the UMF design is our view at the current time, yet over the period of the UniverSelf project, the design will be refined, updated and consolidated as our research highlights and clarifies the issues, and feedback from the integration of the network empowerment solutions (WP3) and feasibility/implementation of the UMF (WP4) is incorporated in the design work.

We aim at specifying UMF in terms of a new framework and new integration structure for unifying management functions. The release 1 of the UMF design emerged as a combined result of the prior art analysis (see section 3), the bottom-up analysis (see section 5.2) and the top-down analysis (see Section 5.1). Earlier results have been reported internally to the project in Milestone 24 – UMF Specifications. These constitute the three pillars for the development of the UMF Design release 1 as depicted in Figure 2.

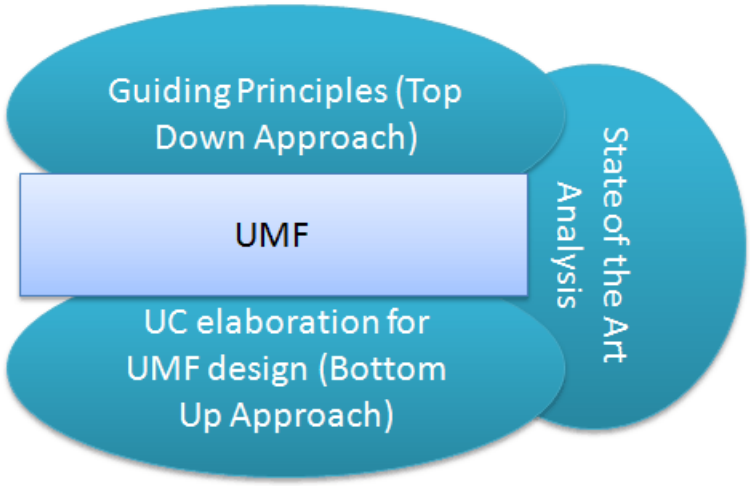


Figure 2: Three pillars used for the design of the UMF design release 1.

5.1 Top level requirements analysis

The UMF design follows a dual approach and will be realised by pursuing an integration the two axes: a "bottom-up" synonymous of synthesis (use case requirements) and a "top-down" synonymous of decomposition (of high-level requirements into functions and interfaces). The former approach aims at addressing the great set of requirements deriving from the first burst of use cases that were defined and developed so far within deliverable D4.1 – Synthesis of Use Case Requirements, Release 1 (WP4). Both "bottom-up" and "top-down" analysis have resulted in the definition of a set of functional blocks and interfaces that consider both services and networks and exhibit the flexibility to accommodate mixed networking scenarios and use cases spanning both wireline and wireless technologies. This approach points out the design of a system that aims at resolving operators' day-to-day problems identified in live networks and on existing service/network architectures. On the other hand, in the top-down axis, UMF will capitalize on previous autonomic architecture research in order to achieve a coherent set of autonomic network management functionalities that can interwork in a scalable manner. To this effect, this section provides a set of top level requirements and design goals that have to be covered by the design and based on those, it also outlines a set of "functional groups" where UMF should be involved.

The top level requirements and design goals cover the Description of Work (DoW), individual project partners' expertise, as well as the general vision and research directions for Future Networks, Service Oriented Computing and Networking, and Future Internet. The requirements together as a set, and not necessarily per individual requirement, describe what distinguishes UniverSelf from earlier network and service management technologies and what the UniverSelf project intends to design and deliver. They designate both "nice to have" and mandatory features/properties that the UMF shall exhibit and will be gradually addressed by the ongoing

work leading to the next UMF releases. It is shown how they are satisfied in the current release and particularly, how they would be used to refine the functional blocks and/or packages deriving from the bottom-up approach (additional or completely new functionalities). It must be also noted that the role of governance, information and knowledge management and feature (intelligence) embodiment as enablers/fundamental elements the technical options of which are elaborated in Section 6, is significant in the realization and addressing of the top level goals and requirements.

A more detailed description of the mapping between the UMF functional block and the UMF requirements is provided in Annex A.

Governance: The prominent role of governance in UniverSelf calls for explicit design of its management functionality and associated interfaces within UMF. First of all, the UMF design should designate and facilitate the development of a privileged, powerful and evolved human to network interface that will be used by the human operator for expressing their business goals and requests, thus shifting from network management to network governance. At the same time, UMF should provide a policy-based framework for translating those business level goals/requests (highest level policies) to low level policies and configuration commands. In general, UMF must facilitate high-level dialogues between self-managed networks and multiple human network operators. They will ensure that all well-formed queries to the network are answered in a pertinent way and also that every well-formed goal injected to a network is either enforced completely and instantly or its delay/modifications are negotiated per rules instantiated. In the opposite direction, UMF must take care so that every impossibility to continue self-managed operation or realistic danger of that will be reported to humans with pertinent details of the situation. Having a global coarse view of the network components and services, governance participates in the overall evaluation on the performance of services/network nodes/domains etc.

Unification and Federation: UMF must ensure that multiple diverse management systems implemented upon different autonomic architectures will be able to interoperate and federate. It will also guarantee that autonomic functions may be implemented (apart from optional interfacing) independently of the architecture chosen for the management system. Actually, UMF envisages a multi-faceted unification i.e. UMF is a unified and evolvable framework constituting a cross-technology (wireless and wireline) and common abstraction/substrate for supporting the management of both networks and services.

Service orientation: Much related to unification above is the service orientation of UMF. UMF will be service oriented and will offer a service view instead of the traditional resource view. This means that UMF should cover explicitly both network and services aspects in a unified manner and facilitate shifting and convergence towards “Everything as a managed Service”, which also includes “Network as a Service” (e.g. management of the integration of network and service aspects).

Automation/Autonomicity/Self-x: Autonomicity/automation and self-x networking are of topmost importance for UniverSelf and they should be facilitated by and demonstrated through UMF. A number of coordinated, autonomic, closed control loops per management function or group of management functions will need to be specified. In particular, UMF should provide a framework for understanding the behaviour of active self-x entities. It should be also able to assess their performance and when needed i.e. at ideal points in time, to re-optimize individual management processes. This last might also designate the need to satisfy extensibility (change of management functionality) requirements. That is, UMF must provide the enablers for activating new management functionality on demand in a plug-and-play / unplug-and-play fashion and programmatically, but also the capability to adapt the information flow and interactions between the functions of the UMF to face new system or operational requirements.

Orchestration/Coordination: In supporting autonomicity above, UMF should also provide a framework for the coordination and orchestration of the newly introduced self-x managing and managed entities. This can be based both on human control/directives (i.e. governance) and explicit functionality destined to this task. Additionally, this introduction of autonomic/self-x network capabilities into a network and services might cause instabilities, thus jeopardizing performances and integrity. Therefore, UMF must provide the means to monitor, detect/predict, resolve and manage external/internal disturbances/dynamics in networks and services.

Intelligence Embodiment/Network Empowerment: The seamless integration of the developed self-x functionalities within both existing and future networks also requires the outline of clear migration strategies/path. In particular, a migration path must complement the UMF in order to support the progressive introduction of self-x features in the existing NE/EMS/NMS/OSS/BSS management chain, thus justifying its cleaned-state claim. While a short-term UMF can proceed with deltas or incremental evolutions, in a longer-

term approach, UMF should propose a simplification, reduction and even at least a partial replacement of the NE/EMS/NMS/OSS/BSS chain. Moving towards more autonomic control instead of continuing to over-engineer and befuddle the management of the networks/systems is needed and this means that some transfer or refactoring of the functionality and operations of the management and control planes (as currently defined) shall occur. The UMF design takes care of this evolution by moving towards underpinning and elaborating on governance which should be a great simplification of the NMS/OSS/BSS part, and network empowerment by introduction of possibly distributed closed-control loops/self-x functions that will assist in greater control and, less management. In particular regarding the latter, UMF must enable the network empowerment mechanisms i.e. embedding intelligence to service and network domains

Future Networks: UMF will capitalize both on research done in autonomic networking and demonstrate its applicability to industry standards, whereas at the same time it will be forward looking, enabling future research and engineering to build on UniverSelf outcomes. The top level requirements regarding future networks that follow, were actually identified by ITU-T SG13 "Focus Group on Future Networks (FG-FN)" and are expected to play quite a role in the finalized UMF design and in demonstrating its future-proofing.

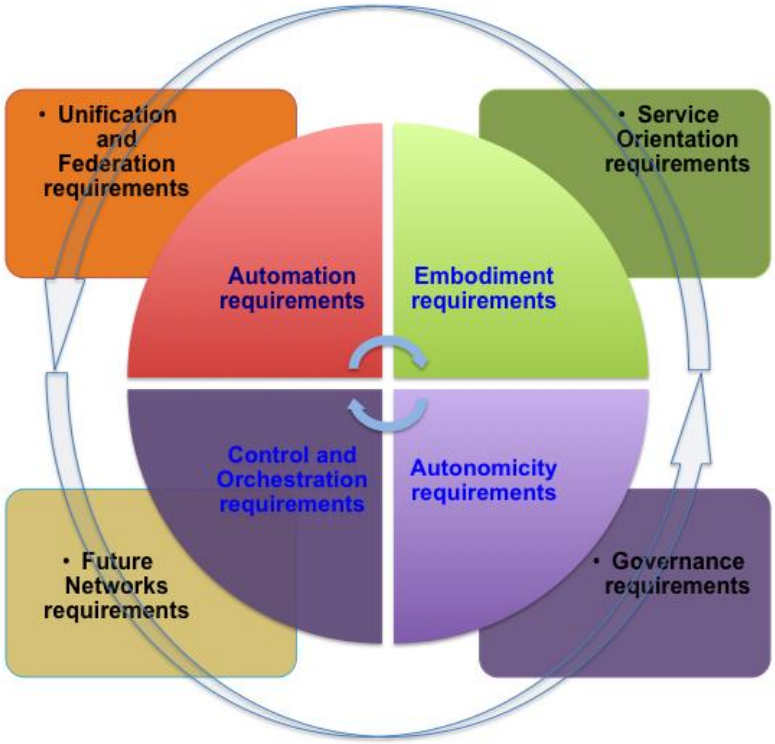


Figure 3. UMF top-down requirements synthesis.

The UMF functionalities to which the elicited requirements apply include:

- Aware and Self-aware functions: Monitoring groups of managed entities and operational context as well as internal operational network state in order to assess if the system current behaviour serve its purposes.
- Adaptive and Self-adaptive functions: It triggers changes in managed entities' operations (state, configurations, and works) in function of the changes in network context.
- Automatic self-functions: It enables self-control (i.e. FCAPS, self-FCAPS, and self-x) of its internal operations, functions and state. It also bootstraps itself and it operates without manual external intervention. The only manual/external input is the setting-up of the goal(s).
- Autonomic self-functionality including Self- configuration, -monitoring, -optimization, - organization, -healing, - diagnosis, -protection, - awareness, - governance, - testing, -management.
- Policy Management
- Life cycle management functions (i.e. design, deployment, activation/deactivation, operation, update, move, change) of all management functionality.

- Integration functions: the enablers and common functionality for interworking and orchestration of different management functions
- Extensibility functions: It adds new functions without disturbing the rest of the system (Plug_and_Play/Unplug_and_Play/ Dynamic programmability of management functions)
- Coordination, control and orchestration of the management functionality
- Empowerment functions: The management functionality is embedded in the managed entities (e.g. co-located with the managed entities functions)
- System Outlay functions: Minimise life-cycle network operations' costs, minimise energy footprint, minimise carbon footprint.

Each UMF function would apply and change any of the following **managed entities**:

- *Services:* large number of ICT and Telecom services offered by the network operator or different service providers needs to be managed (e.g., management of the mapping of service components into executable services on the network environments, deployment and activation of services, services run, the service profile/requirements, manage the e2e performance of the services, assurance management, charging/accounting management, etc.)
- *Networks:* Different technological (e.g., wired, wireless), topological (e.g., enterprise, access, core) and administrative domains need to be managed (i.e., enforce policies, configure components, monitor management data, etc.)
- *Resources:* the per node computational resources (e.g., buffers, memory, CPU), network resources (e.g., spectrum, radio channels, network interfaces, etc.) as well as virtual resources, which are dynamically created groups of physical resources need to be managed in an autonomous or cooperative way.
- *Domains:* a grouping of resources and managed objects with uniform set of policies (e.g. administrative domain, access-network domain, core network domain, virtual network domain, service domain, etc.).
- *Managed Things:* S/W objects, which are part of management applications/services, Virtual Machines representing service components and virtual routers, network attachments, domains, smart objects / Internet of things.

5.2 Use case elaboration for UMF design

The purpose of this chapter is to describe the methodology and results of the "bottom-up" approach as one of the two main axes (the other being the "top-down" one) used for deriving the first description of the UMF design. This approach is based on the elaboration of a great set of requirements as were elicited from the first burst of use cases (defined and elaborated so far within WP4 Task 4.1). It has resulted in the specification of a first functional view of the UMF aiming at resolving operators' day-to-day problems identified on existing service/network architectures considering both services and networks and spanning both fixed and mobile network domains.

The use cases were elaborated based on the so called "Black Box" methodology. This was actually an artefact used to identify and focus on the multiple and disparate problems that should be tackled within each of the use cases. The idea is to represent each use case problem as a black box, and use this so as to hide the details of this problem i.e. only describing it in terms of inputs and expected outputs and with no or limited knowledge of its internals. Accordingly, every use case was decoupled in a number of black-boxes (problems) with further decomposition of a black box leading to lowest level of abstraction e.g. a number of black boxes (sub-problems) can be the descendant of a parent black box.

The above decoupling/decomposition in black boxes were made for all use cases and at a level of detail that enables future evolution, alternate architectures and implementations. In the sequel and based on this elaboration, a rich set of use case requirements were extracted and apposed in Deliverable D4.1 "Synthesis of Use Case Requirements" as functional requirements. The visualization of the methodology for extracting those requirements is depicted in Figure 4 assuming a hypothetical use case x and following the nomenclature of D4.1.

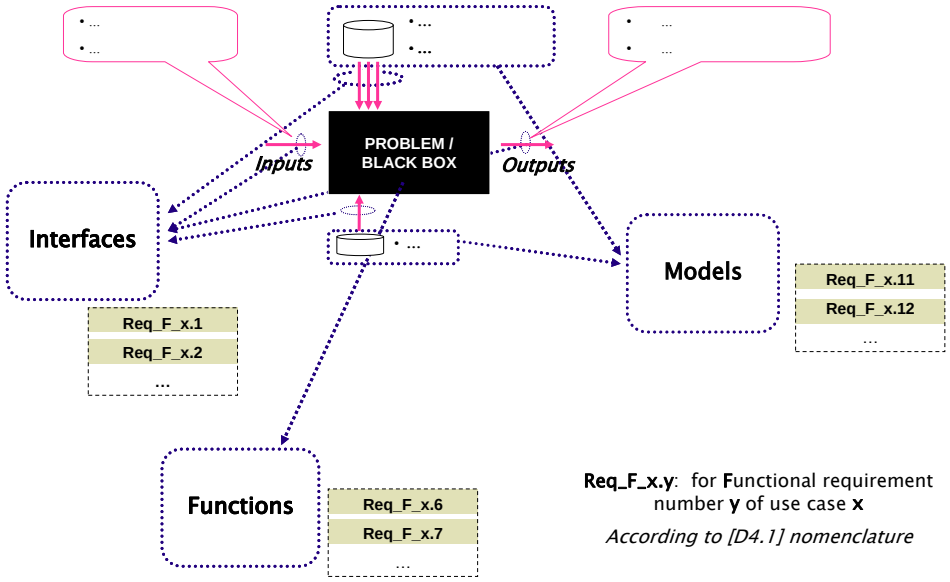


Figure 4. Visualization of the methodology for extracting requirements based on decoupling in black-boxes analysis.

In general, each problem/black box requires an input and provides an output. Therefore, in this first and lowest level of the used methodology, a great set of functional requirements has been derived per use case (see Deliverable D4.1 – Synthesis of Use Case Requirements, Release 1 for more details). The main goal of the bottom-up approach of UMF design is to address these functional requirements i.e. to carefully analyse and exploit them so as to identify a first functional view of the UMF.

More specifically, all these functional requirements derived from use cases and accumulated in D4.1 actually designated a set of Functions (Figure 4), which are required to solve the use case problems (transform input to output), and Models (Figure 4), consisting of information and knowledge bases, policy repositories, storage etc., required to fulfil the functions' operation

In the next step, the functions (from all use cases) that exhibit similar purpose/goal and/or similar inputs and outputs or operation are elaborated and they are grouped into a set of functional blocks. Accordingly, the functional blocks group functions with commonalities and irrespectively of the use case from which they eventually derive from. As such, they designate design blocks that exhibit great levels of reusability and cohesion and can be used to implement a core function of the UMF.

The list of the correspondingly identified functional blocks is depicted in Table 2. Each functional block in this table contains the relevant set of commonalities in a specific format: each use case is presented with bullets, accompanied with a description of the function that is indicative to the relation of the latter to the functional block it belongs to.

Table 2. List of identified functional blocks

Monitoring	
Network	UC1: topology information monitoring UC2: active self-monitoring, passive self-monitoring UC3: data monitoring, network monitoring for end-to-end (E2E) connection/session status/statistics, network data monitoring, topology information monitoring UC4: self-X monitoring, monitoring of the entire SON process, policies, entities, coordination and conflict resolution, KPIs UC5: network parameter monitoring UC6: RAN & Backhaul/Core network monitoring, network parameters monitoring UC7: network context monitoring, context discovery
Service	UC1: service data monitoring UC3: service data monitoring UC5: SLA compliance monitoring
Customer	UC1: reports from customer/customer care monitoring UC5: SLA compliance monitoring
Situation Analysis/Diagnosis	
UC1: root cause analysis from alarms, data aggregation, traffic anomaly detection UC3: data processing (aggregation & processing of monitored data) UC4: determination of the involved SON entities and their location based on the operator targets UC5: traffic aggregation, bandwidth estimation UC6: business level entries/request analysis to derive (translate them to) technology (network) specific requirements	
Candidate Solutions Computation	
UC1: potential faults prediction and reaction to failures/events identification UC3: provision of candidate networks or network domains to be reconfigured, capabilities discovery UC5: network configuration sharing, alternative configuration suggestion, configuration costs sharing UC6: candidate solutions discovery/determination, candidate solutions reasoning UC7: context condition determination (topology, devices, configuration parameters) associated with the network and technologies	
Solution Selection and Elaboration	
UC1: reparation/ mitigation plan selection UC2 : proactive self-stabilization actions, reactive self-stabilization actions, on-line self-prevention actions UC3: decision for specific reconfiguration actions, conflict management UC4: SON entities coordination in order to enforce the policies including conflict resolution UC5: triggering routing optimization, generation of connectivity configuration changes UC6: invocation of the selected network (RAN and backhaul/core) with a request, QoS optimization function, RAN management function, conflict resolution mechanisms for different self-optimization and/or self-healing actions UC7: (re-)configuration decision	

Configuration Enforcement
UC1: network (re-)configuration UC2: policies/rules application UC3: configuration of network resources UC5: network configuration changes application UC6: conversion of generic configuration into technology-specific configurations, self-configuration functions, application of actual configuration of RAN and Core network nodes UC7: global network configuration
Solution Evaluation/Assessment
UC1: network and services operation end-to-end evaluation UC2: validation of self-* features UC3: evaluation for end-to-end (E2E) connection/session status/statistics UC4: e2e evaluation UC6: autonomic functions for (self-)optimization actions UC7: QoS calculation, behaviour assessment
Policy Derivation and Management
UC1: mitigation policies UC2: translation of business goals to infrastructure level policies, rules and policies generation for on-line network self-stabilization/self-prevention or for network stabilization/prevention, for automatic verification and validation of decisions before their actual implementation UC3: policy language translation, translation of SLAs to policies, policies for conflict resolution UC4: policy repository, policy language, policy generation, policies for self-X operation, policy adjustment, translation of high level policies to SON entities specific policies, rules and policies to identify the involved self-X entities, to activate/ deactivate self-X functions, to allow the interaction between self-X entities, and between self-X and other network entities, to resolve conflicts between running self-X processes when coordination fails UC5: policy rules for routing table updates UC6: policies derivation, policy conflict resolution, business goals to policies translation, applied policy translation to traffic engineering compatible commands, policy repositories UC7: business goals translation, policy language, policy conflict resolution, policy-based trust management mechanisms
Governance
UC1: predicted event reporting to human, triggered mitigation reporting to human, OSS Interface to NMS for communicating operator's goals, EMS to OSS interface for reporting failure of re-configuration actions, H2N interface for reporting users' problems and evaluation of the system UC2: human de-activation of self-* features, human interface for on-line, human interface for off-line UC3: H2N interface to insert high level goals, to deliver control and management and to feedback system checks UC4: H2N interface for inserting operator targets and policies, interfaces for self-X governance, network planning tool UC5: human enhancement of bandwidth estimation, SLA and configuration commands manipulation by a H2N interface UC6: H2N interface for request and goals expression, feedback provision to the H2N governance GUI UC7: business goals language, H2N interface to insert high level business goals, network to human notifications, information retrieval from autonomic entities by governance tools, real time monitoring available to the operator on the fly

Profiles and Models
UC1: model for anomaly prediction pattern data exchange, model for normality prediction pattern data exchange, model for anomaly diagnosis exchange, model for normality diagnosis exchange, model for complex data, model for complex data exchange
UC2: network stability models, map of self-* features
UC3: traffic/load analysis/estimation mathematical models, network/service/user profile, traffic/mobility/energy/SLA models, RAN and backhaul/core models
UC5: network topology model
UC6: policy models, network model information provision, Application profile, user class profile, Behaviour/Mobility models, QoE/ QoS models, SLA models, energy models, resource models, network profiles of candidate networks, configuration models
UC7: information model of all the elements involved in the lifecycle of a service, service profiles, user profile, network profile and (RAN/backhaul/core) models, SLA models, charging profile, model(s) of traffic/mobility/energy requirements
Information and Knowledge Building
UC1: horizontal data correlation, vertical data correlation, time scale data correlation, knowledge about reparation/mitigation plan
UC2: network knowledge extraction, external knowledge (vulnerable state descriptions)
UC3: information and knowledge on resources and element capabilities, learning capabilities on conflict resolution
UC4: information for self-X operation (information and knowledge about SON entities and their location, already active policies, mobility, traffic, performance/QoS, service requirements, bandwidth allocation, KPI thresholds)
UC5: Knowledge data management (past observations, states and decisions, such as past traffic measurements, bandwidth estimations, network configurations)
UC6: information and knowledge on SLAs, Applications, User classes, RAN (network, resources, configuration), Backhaul /Core (network, resources, configuration), Traffic mobility requirements, and Traffic demand descriptions
UC7: knowledge on policy conflict resolution, active policies, resource availability and element capabilities, learning capabilities for evaluation
Cooperation
UC1: event prediction algorithms coordination
UC2: on-line self-prevention actions (e.g. coordination, conflict resolution of self-* features), orchestration of self-* features
UC3: conflict management
UC4: SON entities coordination in order to enforce the policies including conflict resolution
UC6: collaboration and negotiation between network segments (RAN & Backhaul/Core), domains, operators and service providers, and conflict resolution mechanisms for different self-optimization and/or self-healing actions

The above functional blocks are described in more detail in the sequel:

Monitoring: it is needed for collecting measurements in order to ensure that the desired performance is guaranteed. Three levels of monitoring are distinguished, namely network, service and customer monitoring.

Situation Analysis/Diagnosis: it is used to analyse events in order to trigger appropriate actions. Events may consist of performance measurements or business goals/policies/other governance triggers. The elements, which are analysed, are also used as basis of the diagnosis mechanism. The goal is not to provide solutions but to analyse the situation and trigger the corresponding actions.

Candidate Solutions Computation: it intends to identify potential solutions (reparation/mitigation plans, (re)configuration) to be enforced. Therefore, this problem concerns the determination/discovery (reasoning with) of the candidate solutions that can satisfy the derived performance requirements.

Solution Selection and Elaboration: it actually pertains to the decision making procedure. Decision may consist of either a reparation/mitigation plan or a configuration action. In some cases, the function also addresses the resolution of possible incompatibilities or conflicts among the different involved entities, in particular inside the same segment/domain. For that reason, some sort of negotiation and cooperation between these entities is needed (coherence) or specific mechanisms. When coordination/orchestration and conflict resolution are required among different segments/domains, Cooperation that will be analysed later is the functional block to undertake these tasks.

Configuration Enforcement: it is responsible to apply the configuration decision. First, it's necessary to identify concerned equipment and request each of them to perform the appropriate configuration actions. Then, each of the targeted equipment has to translate and enforce the decision. The term configuration implies self-configuration and includes both configuration and reconfiguration actions (re-optimizations). Reconfiguration actions can be triggered in order to adjust the configuration parameters following network, service and customer conditions.

Solution Evaluation/Assessment: it aims at evaluating and assessing the solution (e.g. configuration, reparation/mitigation plan). If the objectives of the solution are not met, further actions are triggered for fine-tuning/optimizations, in order to maintain the overall performance in the desired and planned levels.

Policy Derivation and Management: it is used to translate high level goals/objectives provided through Governance into low level policies and often into low level self-configuration enforcement policies. Therefore, policies are derived according to the higher level goals. Sometimes, the derived policies are assessed against existing goals/policies so as to identify and resolve conflicts (in fact, conflicts can arise if the defined goal/objective/policy is antagonist with respect to previous goals or the impact of these goals on already deployed services). After the policy derivation and management, the system is free to autonomously work out the situation and meet the objectives.

Governance: it allows the introduction of the business level goals/policies in high level terms through a human-to-network (H2N) interface. Then, after the policy derivation and management, the system is free to autonomously work out the situation and meet the objectives. In addition to high level business goals/policies, other input consists of reparation/mitigation plans, policies for conflict resolution that are needed in the Policy derivation and management, SLA insertion etc. The H2N interface allows also feedback, e.g. the result of diagnosis or a visualization of the monitoring to the system administrator/operator.

Cooperation: it provides the necessary functionality to address requirements such as Unification/Federation and Orchestration/Coordination. Therefore, this function is initially responsible for achieving federation/bridging among different segments, such as the wireless and wireline worlds e.g. for translating data from one Network Management System (NMS) to another etc. Furthermore, it represents the required functionality in order to coordinate/orchestrate self-x managing and managed entities including conflict resolution, when these entities belong to different segments or domains. When self-x entities are located in the same segment or domain, Solution Selection and Elaboration is the responsible functional block to resolve and manage any disturbances/dynamics in networks and services. Cooperation could also be used to enable the cooperation of UMF with legacy systems.

Information and Knowledge Building: it refers to any function (e.g. store/retrieve/update/modify/exploit) related to dynamic knowledge. It differs from Profiles and Models, which represent the static knowledge. An important aspect of this functional block is the learning functionality, which is essential for addressing complexity and scalability. Learning enables the system to gradually obtain knowledge on how to handle

complex situations. This can increase the speed of the decision making process, and also the degree of certainty on the quality of the decisions. Therefore, learning contributes to the offer of scalability. It has a strong relationship with almost all the previous functional blocks, since it interacts with them.

Profiles and Models: it represents static knowledge that is stored in databases. In this concept, they represent any existing information on the managed elements, the offered applications, the served users and equipment etc. Regarding the managed network elements and the served equipment, the focus is on reflecting their capabilities, i.e., the configurations with which they can operate. Regarding the applications, the focus is on the permissible QoS levels at which they can be offered. Regarding the served users, the focus is on their behaviour (when a certain application is used), preferences (in terms of QoS levels per application) and agreements (range of allowed QoS levels per application). They are different from the Information and Knowledge Building, which includes the dynamic knowledge.

The last part of requirements as per the use case elaboration and [D4.1] designate the need to have a set of Interfaces, covering interworking among black boxes, interfaces towards information and knowledge models/bases, towards policy repositories, storage etc. All these have eventually resulted in a set of interfaces among the different functional blocks so as to realize the use case flows (see also Figure 4). This is highlighted in Figure 5 for the general UMF overview and in Figure 6 for showing the exemplary instantiation to an arbitrarily selected use case e.g. use case 1.

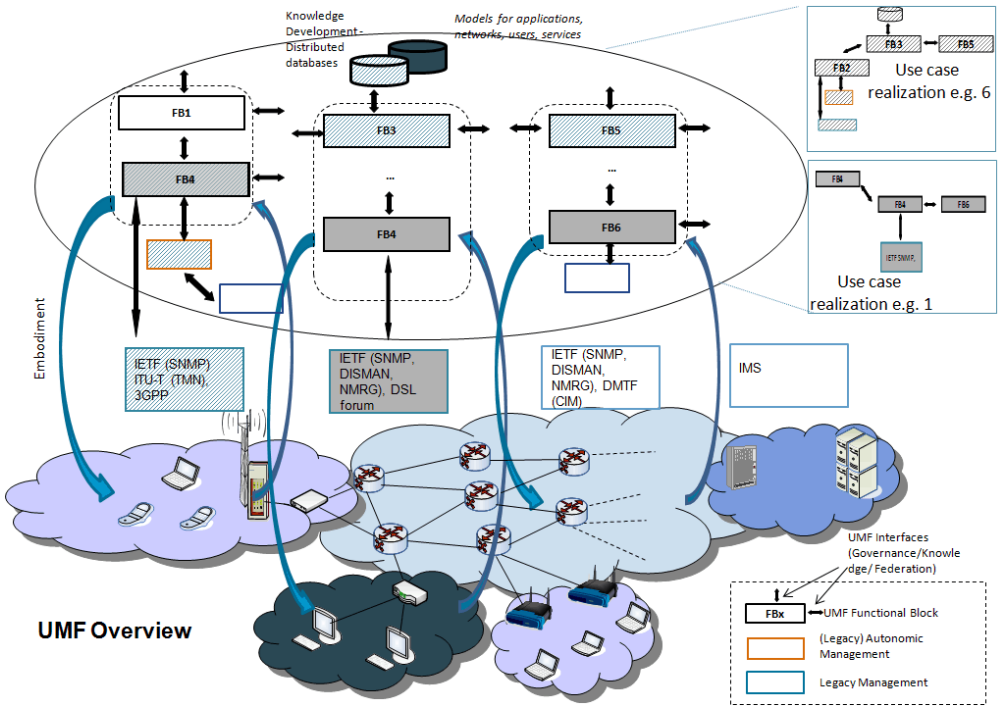


Figure 5. Interactions among functional blocks for realizing use cases.

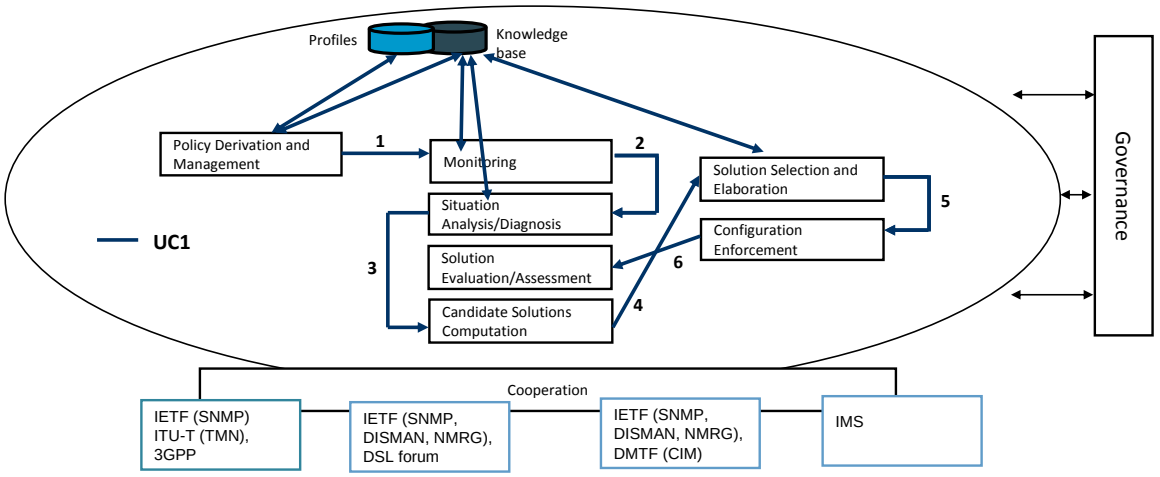


Figure 6. Instantiation of UC1 (Self-Diagnosis/Healing for IMS VoIP and VPN).

Last but not least, apart from the functional requirements the elaboration of which has resulted to the first functional view of UMF i.e. in terms of functional blocks, D4.1 reported a list of non-functional and business requirements as derived from the elaboration of the use cases, as well. Such business and non-functional requirements will be used to judge the UMF in terms of usability by network operators and extensibility for tackling new scenarios. In addition, they will also have certain impact to the system view of UMF (see section 5.4) e.g. by driving decisions with respect to level of distribution of specified functions among the management systems or network elements, which in essence may result in different performances.

In total, the non-functional and business requirements, together with the top level requirements identified as part of the top-down approach and reported in section 5.1 of this deliverable, need also to be taken into account in the UMF design. Towards this direction, Section 5.3 below provides a consolidation of the core functional blocks and their organization into the main Functional Groups constructing the UMF. These Functional Groups can be defined as 'an aggregation of functional blocks, which realizes a higher level management function'.

5.3 Core functional blocks consolidation and organisation

In this section, we introduce the main functional groups that are provided by UMF. Based on the identified top-level requirements in section 5.1 and the functional blocks identified in section 5.2, we are now in a position to define the main functional groups constructing the UMF.

The goal is to group the functional blocks into functional groups in consistency with the bottom-up approach but also in a way that satisfies the top-level requirements. This high-level functional grouping is the highest level of granularity. The template for the representation of each functional group is depicted in the Figure 7.

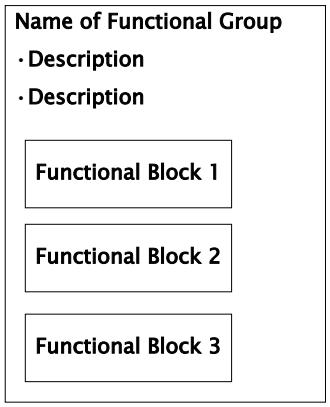


Figure 7. Template for visualizing the functional groups.

The first element in the template is the name of the group. After the name of the group, a brief description, using a vocabulary relative to the group, follows. The description is presented with bullets and consists of keywords. After the description, the list of the encompassed functional blocks is given. The derived functional groups are depicted in Figure 8, Figure 9, Figure 10 and Figure 11.

Four main functional groups are defined, namely:

- **Governance:** The operators specify their business requirements at a top level to the autonomic management system. These requirements are interpreted into policies, being derived and applied to the system autonomically. Policies specify rules that should govern the behaviour of the managed elements. Therefore, policies may specify constraints, optimization objectives and functionality that should be followed by the Intelligence functionality (see below), in the particular context. Essentially, policies can refine the information that reflects "what" is generally allowed in the current situation and constrain the options indicated by the context, which is acquired by the Knowledge Management (see below). This group consists of Governance and Policy Derivation and Management functional blocks.

  The human operator interacts with such a management system through a specific interface, called here H2N (human to network). The *Governance* is a group of functions that are running on a topmost level of network management interacting with the human operators, including self-governance, deployment of the policies into network components and support for a common structure from identification and specification of business problems to developing a deployable solution. Governance, as described also in section 5.2.1 is located at the highest level of abstraction.

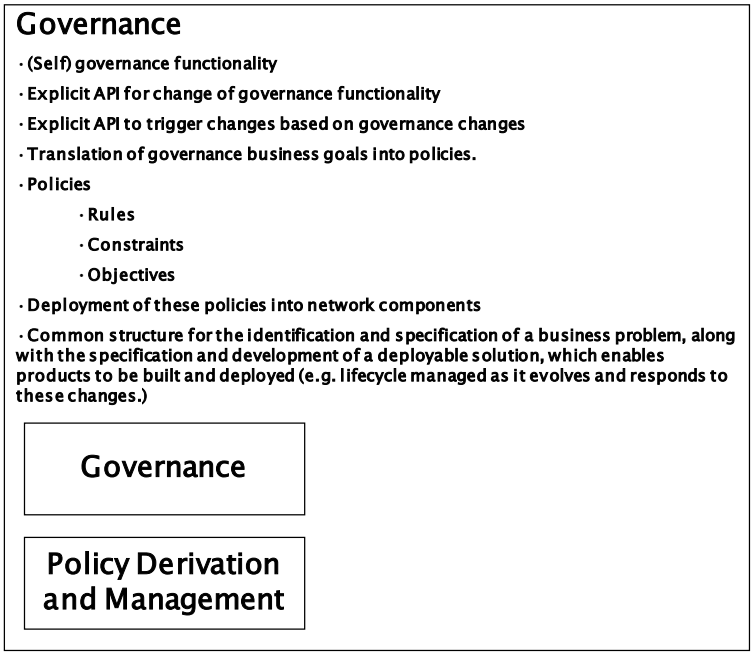


Figure 8. Governance Functional Group.

- **Knowledge Management:** Autonomic management heavily relies on the knowledge on different areas of the entire network. It covers a broad range of information distributed over the network including distributed data, Context (different forms) and Profiles (user/ network/ service). The knowledge management is required to coordinate all the monitoring, profiling and modelling, situation analysis and diagnosis, and information and knowledge building. The knowledge base elicits the following aspects:
  - Related to the context, the situations encountered by the system at various time epochs and locations.
  - Related to the profile, the user preferences and behaviour in the various contexts.
  - Related to the policy (above within Governance), the efficiency of the various policies in the various contexts.
  - Related to Intelligence functionality (see below), the best configurations for handling each contextual situation.

Knowledge management provides the means for perceiving and reasoning on the status of the managed reconfigurable elements and of their environment. In other words, it provides the context, in which the reconfigurable managed element operates, and includes monitoring and situation analysis/diagnosis in terms of functional blocks. From a wider perspective, Knowledge management reasons on "what" is generally possible in the current situation through the context, in collaboration with the acquired knowledge especially, and constrains the options specified in the profiles. Profiles are managed here, as described in the section 5.2.1 about profiles and models. From a wider perspective, profiles reflect "what" is generally possible.

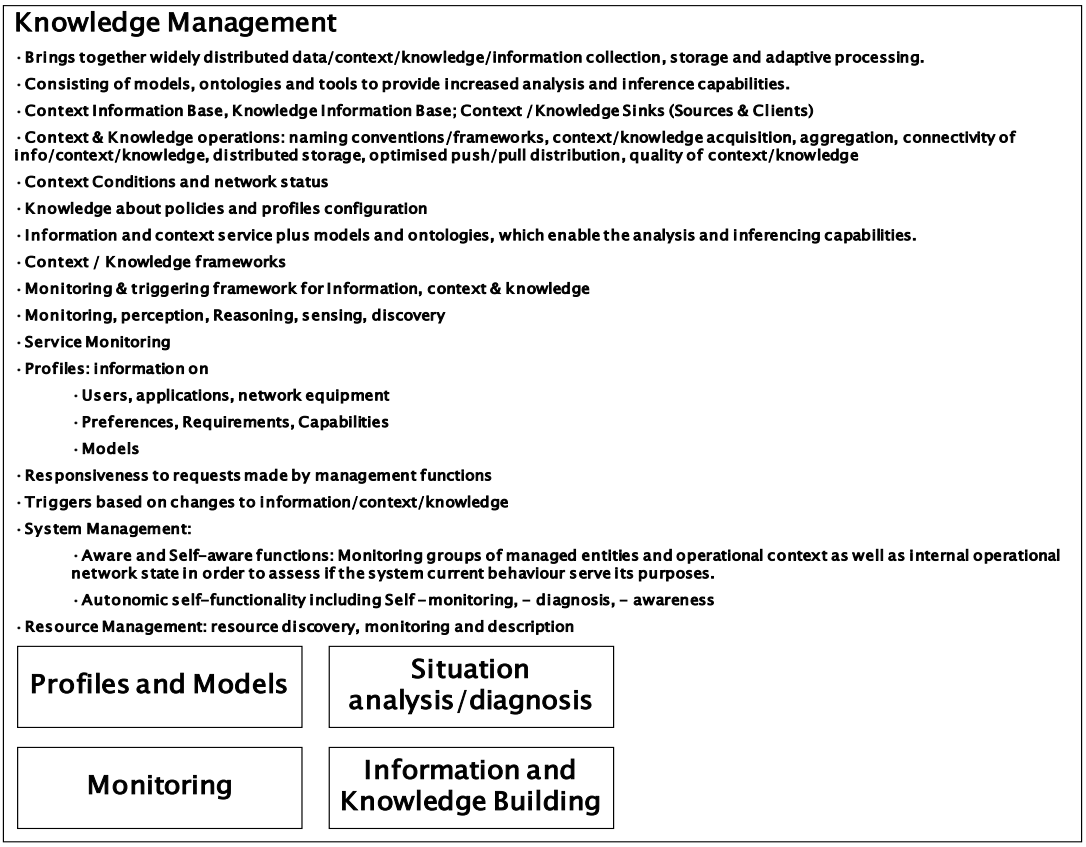


Figure 9. Knowledge Management Functional Group.

- **Enforcement:** Proper enforcement should be taken by applying the policies derived from Governance and after taking the proper action in the Intelligence functionality (see below) against the activated triggers, diagnosed problems and d situation by Knowledge Management. Enforcement is applied to a variety of the components of the system, including network resources (node and media) and services. Configuration enforcement, as described in section 5.2.1, is supported here.

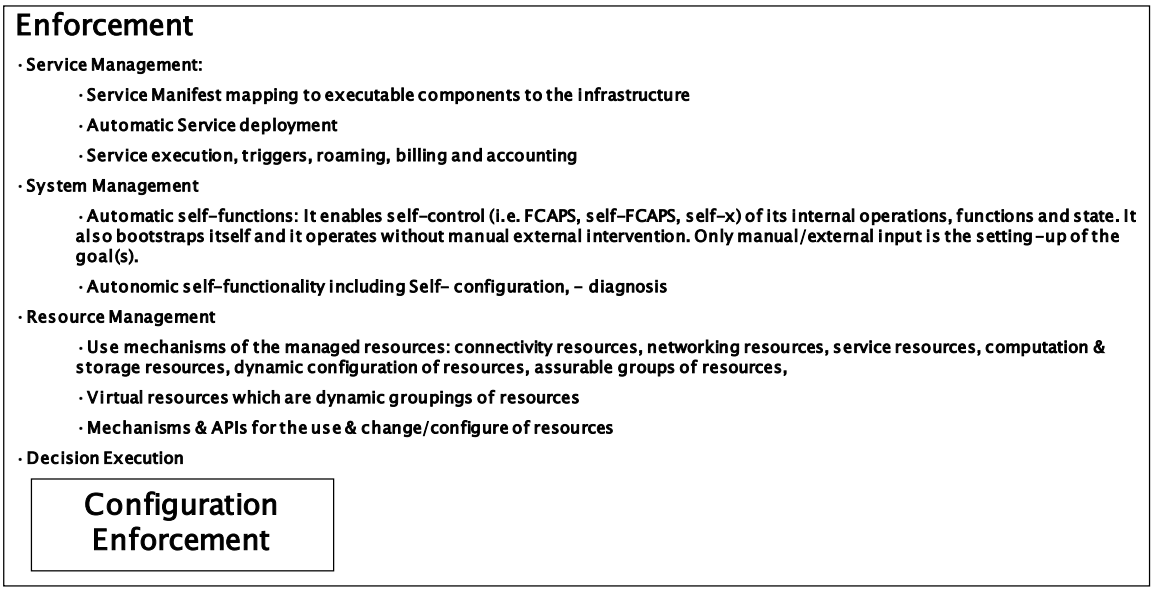


Figure 10. Enforcement Functional Group.

- **Intelligence:** All the functionalities identified above are in need of orchestration and the outcome of enforcement should be evaluated. The Intelligence group supports such an orchestration, assessment and evaluation (solution evaluation/assessment and candidate solutions computation in terms of functional blocks). Cooperation Functionality, as described in section 5.2.1, is supported here. Additionally, the Intelligence functionality is responsible for evaluations on the performance of service provisioning, node networking and resource utilisation.

  Moreover, Intelligence functionality provides the necessary optimization functionality (solution selection and elaboration in terms of functional blocks). Its input comprises the context, profile and policies information from the Knowledge Management. The output includes configuration decisions.

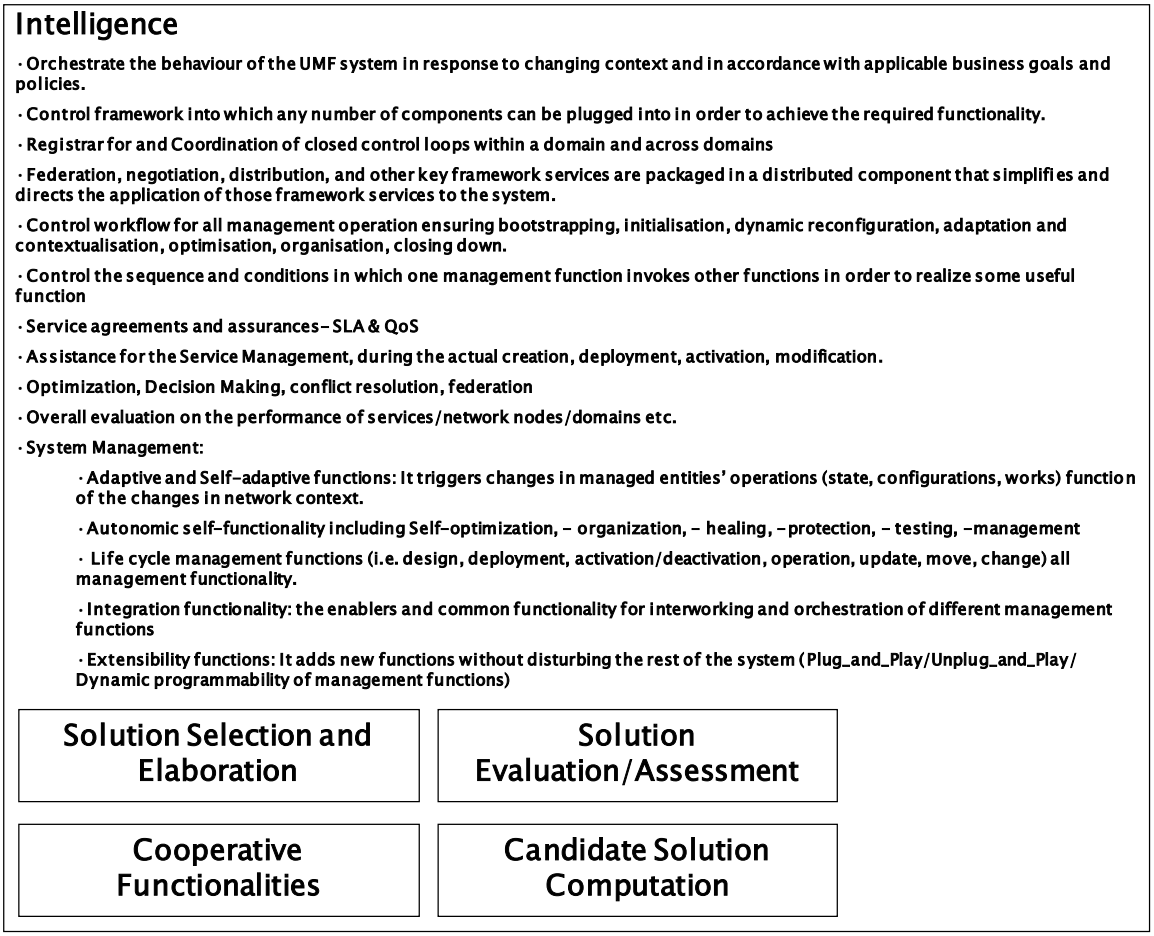


Figure 11. Intelligence Functional Group.

Finally, Figure 12 depicts a high level view of a first functional representation of the UMF based on the derived functional groups. In this figure, the possible interactions among the functional groups are shown.

An aspect of functional grouping in this figure is the concept of UMware. UMware is the set of modules running on different autonomic nodes within the network. The network nodes with different capabilities maintain and execute a number of selected functionalities. This will be more elaborated in section 5.4.

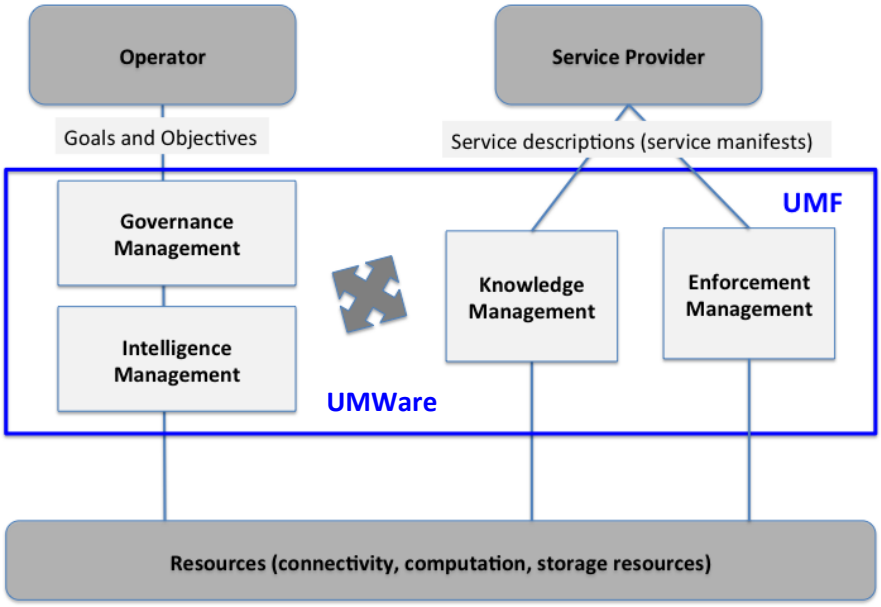


Figure 12. UMF high level functional analysis.

Another important issue is to show how the top-level requirements, identified in Section 5.1, are satisfied by the derived functional groups. *Governance* requirement is satisfied by Governance Functional Group. *Unification & Federation* and *Orchestration/Coordination* are mainly satisfied by Intelligence Functional Group, which offers orchestration and cooperation functionalities. Moreover, Knowledge Management may help in the direction to monitor, detect/predict external/internal disturbances/dynamics in networks and services. *Service orientation* is satisfied by both Knowledge Management and Enforcement, which are the main functional groups involved in service management. *Automation/Autonomicity/Self-x* is a requirement that encompasses all the UMF design, but is mainly supported by Intelligence Functionality, which enables through the orchestration and cooperation the coordination of the autonomic, closed control loops and the plug-and-play capability and through the optimization, evaluation/assessment the assessment of self-x entities performance and the re-optimization of individual management processes when needed, i.e. at ideal points in time. Finally, Intelligence *Embodiment/Network Empowerment*, although it has a strong relationship with all the UMware that will be elaborated in Section 5.4, is satisfied by Governance, which should be a great simplification of the NMS/OSS/BSS part, and Intelligence functionality, which enables the network empowerment by introduction of possibly distributed closed-control loops/self-x functions and embedding intelligence to services and network domains.

The fulfilment of the UMF requirements by the Functional Groups is presented in Annex A.

5.4 UMF System view

In this section, we discuss how the functional blocks and groups d in the previous sections can be organized inside the UMF domain. We provide some directions on how to manage the complex and heterogeneous environment by introducing the notion of a number of specialized logical nodes and introducing a possible hierarchical structure. We also discuss how the orchestration of distributed network management operations can be achieved. The next step is to map the previously defined functional blocks into these nodes and elaborate on their functionalities.

We also provide a detail description on how this organization can be adopted from existing network management systems and their hierarchy, as well as from future SON enabled network systems. This is done by applying the results of the bottom up and top down approaches (i.e., functional blocks and functional groups) into the topologies and the described nodes coming from the bottom up approach produced by the six use cases of WP4.

5.4.1 UMF components

UMF is aiming to provide a detailed description for the functionalities of future network management systems. To do this the first step is to provide some appropriate definitions to be used in the following sections about what a **UMF node and a UMF management domain** are.

As mentioned before, a **UMF node** can be either:

- An existing network management system (e.g., EMS, NMS) enhanced with the required capabilities to communicate with other UMF nodes (e.g., through the use of wrappers). We call these UMF nodes as ***Enhanced Legacy Management System (ELMS)***
- A network connectivity node (e.g., router, BS), which in the future will have embedded autonomic management functionality and resources. We call these UMF nodes ***Future Management Systems (FMS).***

A **UMF domain** is a collection of UMF nodes and network resources, which are on demand established and dynamically maintained and where management can be applied uniformly (e.g. administrative domains, type of networking segments: core, access, virtual, service domains, etc.)

The **external UMF interfaces** are: the governance interfaces for the operators' objectives description, the service description interfaces for the service deployment and the federation interfaces for the interoperability across multiple domains.

The **UMF enablers** – as resource-facing services – used for the integration of all management functions include the governance management, the information-context-knowledge management and the intelligence embodiment.

5.4.1.1 Organization of UMF functions

In previous sections it has been described that UMF will cater for a significant number of devices that will have embedded network management functionality. These will have to be organized somehow to achieve scalable and manageable solutions. Their organization is expected to greatly improve the efficiency of network management [159].

Moreover, it is imperative for an operator to orchestrate and federate the UMF functionality in order to establish an end-to-end autonomic management network. It is expected that each UMF node will be able to operate fully autonomously for a number of tasks (e.g., optimize its own resources, upgrade its software, monitor node related information etc.) and also communicate with other nodes for a different set of tasks where and when some cooperation is required (e.g., minimize interference among different base stations, or minimize the congestion between routers, etc.). It is clear that for scalability reasons, one needs to coordinate the actions of UMF nodes inside a pre-defined area instead of having them cooperating in a flat architecture. Such an area can clearly be defined based on the existing topological domains of an operator's network namely access, backhaul, core and service domain.

The reason for such a grouping is that inside each of these domains the management actions are executed to solve similar issues (e.g., interference minimization in a wireless access network, congestion avoidance inside a core IP network). Inside a domain we have the capability to filter and summarize information and knowledge that is to be shared with other domains.

Inside each one of these domains a UMF node is selected to act as a *'UMF intra-domain controller' (U_DC*). Such a node will be one of the many autonomic nodes located in the domain (although the selection should be among those with the most capabilities, e.g., memory, CPU). Such a node will be responsible to filter/translate and distribute information received from other domains. Moreover, it will be responsible for evaluating the performance of the domain and for addressing conflicts emerging from the simultaneous execution of tasks executed by more than one node inside a domain (Figure 13). A domain may have one or many U_DCs depending on the number of UMF nodes inside the domain as well as the number of manufacturer's types of products for these UMF nodes. The latter has its significance since although interfaces between nodes are going to be standardized this is not expected to be the case on how the decision process for an autonomic operation is going to be implemented (e.g., one manufacturer may implement the decision functionality based on a utility function while another manufacturer may implement the decision functionality based on a fuzzy logic mechanism).

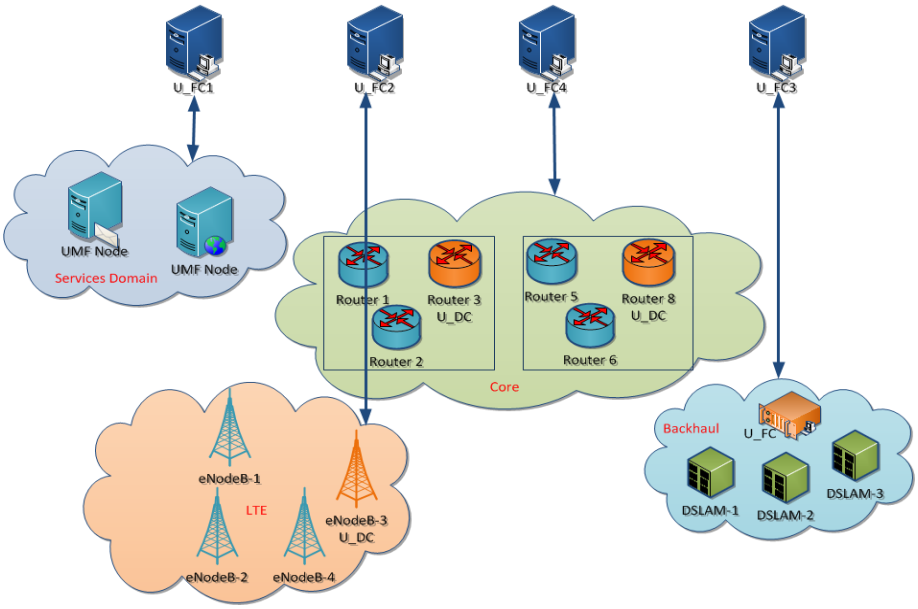


Figure 13. UMF nodes and domain controllers.

We also introduce another UMF node type called *'UMF Inter-domain controller/ federated controller' (U_FC)*. U_FC assists in orchestrating the operation of one or more U_DCs inside a domain. This means that the U_FC needs to translate context information from one type of U_DC to another (in the case of different manufacturers). It also communicates with other U_FCs to address inter domain management issues (e.g., conflict resolution, monitor and evaluation, information exchange). The introduction of U_FC can assist in summarizing the information needed to be exchanged between domains and reduce the number of logical communication connections (from N^2 to N) between U_DCs since their communication will take place through the U_FCs. Furthermore, the U_FC is a point where policies are distributed to the appropriate U_DCs and eventually to the UMF nodes, and also a point where policy conflicts among domains can be solved.

Based on the above description we see that we have a three level organization of functionalities. In every level the peer entities can communicate with each other. Note that this may not be the case for the services domain. As shown in Figure 13, in this domain the servers can support very different functions and their grouping under a distinct U_DC may not be reasonable. Thus, these nodes can communicate directly with the U_FC. In other words these nodes simply have the communication extensions that allow them to communicate with U_FC.

To summarize our proposal, UMF network management in the whole network can be introduced in two echelons interacting vertically through the relevant interfaces:

a) **UMF node (U_Node):** A ELMS or FMS that is able to monitor its environment and take autonomic decisions in a non-cooperative manner for a number of tasks (e.g., auto-inventory, software update, self-healing etc.)

b) **U_DC:** Inside a domain UMF nodes are organized in a group and elect a group controller. The group leader (i.e., UMF intra-domain controller) is a UMF node that assumes the role to coordinate the UMF nodes of its group and assist the management communication with different domains. Its main functions include: i) filtering of domain information (network context information) ii) translation of information to the nodes of the group (knowledge, policies, and network context information), iii) evaluation of the current status of the group, iv) conflict resolution inside the group.

c) **U_FC:** One or more U_DCs are controlled by a U_FC. The U_FC may be a UMF node or new management entity that has the role to coordinate the U_DCs of a domain node of its group and assist the inter-domain communication through other U_FCs. Its main functions include: i) filtering of inter-domain information (network context information) ii) translation of inter-domain information to the controlled U_DCs (knowledge, policies, and network context information), iii) evaluation of the current status of the whole network, iv) inter-domain conflict resolution. U_FC is also the place where the governance functional group is placed. Also, U_FCs provide the means to external service providers to set up their request for deploying new services.

5.4.1.2 UMF Interfaces

As far as the needed interfaces are concerned we need to define the following interactions:

a) The internal interfaces of a UMF node (i.e., between the protocols of the different functional groups).
b) Between UMF nodes and U_DCs
c) Between U_DC and U_FC
d) Between U_FCs of different domains
e) Between the administrator of the network and the U_FC,
f) Between the external service providers and the U_FC.

For the definition of these interfaces we need to review whether P2P interfaces are suitable or if brokers or message bus (content based, list based) or subscribe/publish mechanisms should be used as presented in Figure 14 [160]. Brokers and message bus mechanisms present advantages and disadvantages. The notion of semantic message bus could also be a candidate principle for a solution in UniverSelf [161].

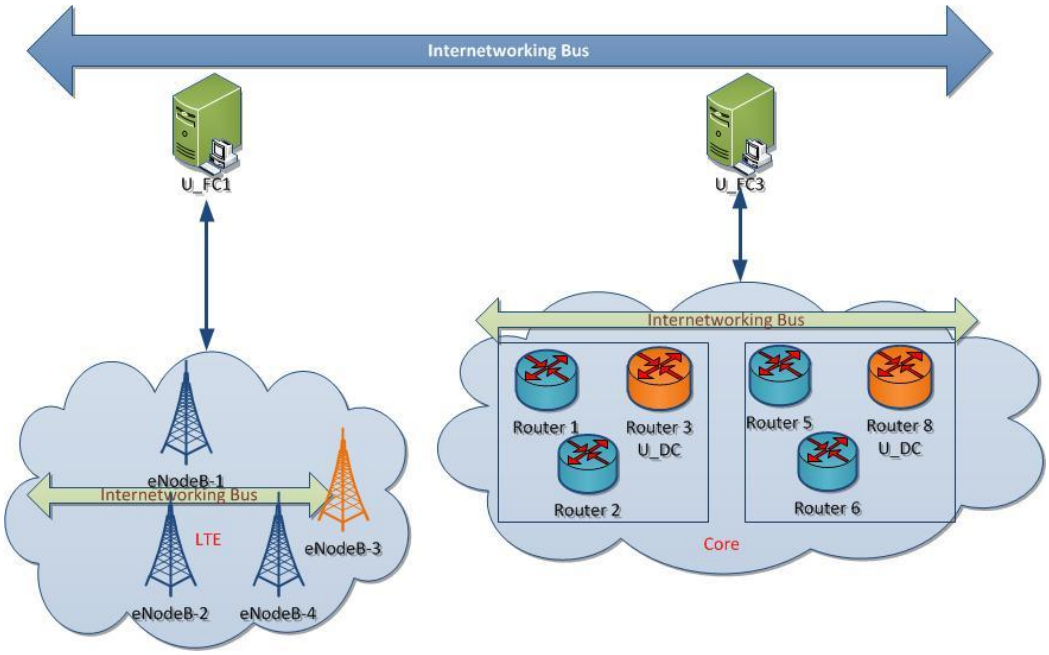


Figure 14. Interworking between UMF network management systems.

5.4.1.3 UMF Control Workflows

Based on the definition of the UMF nodes, we can define three typical levels of control loops. By control loop we mean the "Monitor – Decision – Execution – Learning" cycle that characterizes the operation of an autonomic management system. In UMF, the closed control loop of autonomic systems (Orchestrated MAPE) is expected to be observed at the following levels:

a) UMF node level (ELMS and FMS),
b) intra-domain level (e.g., core, access, backhaul, service) and
c) inter-domain level.

For the intra and inter domain control loops we need to orchestrate the operation of a number of autonomic nodes. This concept of multiple control loops appears in Figure 15.

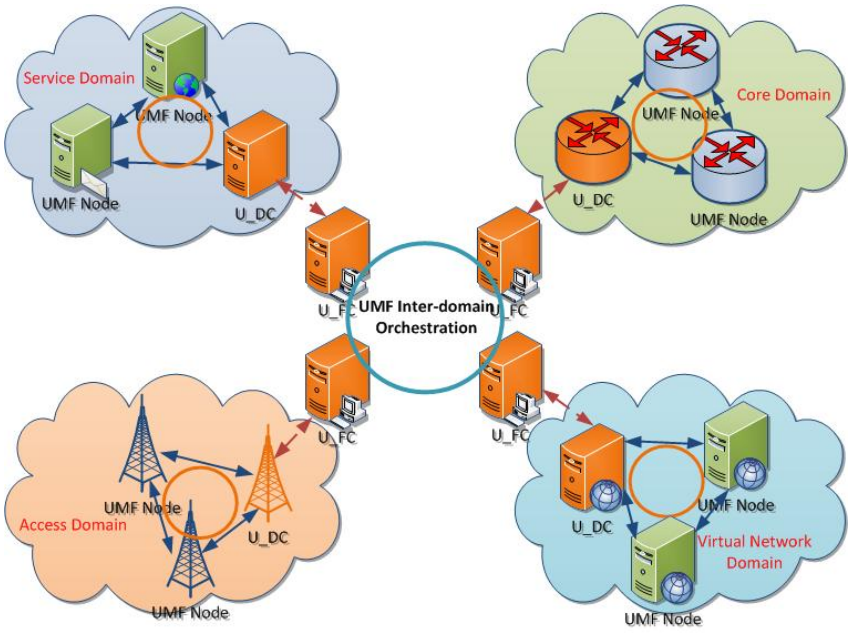


Figure 15. Intra and Inter domain orchestration.

An autonomic networking system is defined by its capacity to monitor the operating environment and its groups of resources, model its behaviour and take appropriate actions according to some knowledge elements. It consists of a number of *management control loops* capable of dynamically managing various aspects of the system functionality. In addition, an *orchestration control loop* regulates all control loops by harmonising individual decisions resulting to optimal or near optimal network configurations. Figure 16 presents the conceptual assembly of the autonomic networking system. Each management control loop is decomposed in four separate utilities (e.g. Monitor – Analyse – Plan – Execute) sharing a common knowledge of the operating environment and its group of resources as it interacts with the managed resources and with the orchestration control loop.

- *Monitor:* it probes the managed resources for operating parameters, management information and context information. It provides the information and context information for storage and distribution in the autonomic networking system. It also performs aggregation, filtering, verification and reporting operation over such data.
- *Analyse:* it provides the mechanisms to model the operating environment and its groups of resources. Such model is used in matching data with system dynamics and to forecast future trends.
- *Plan:* it provides the reasoning intelligence that identifies the actions to achieve specific goals.
- *Execute:* it implements the actions output of the plan function.

The *Manageability and Orchestration interfaces* present the autonomic networking system transparent interfaces to various management and managed technologies inclusive of full life cycle management phases (e.g. creation, provision, bootstrap, operation and secession).

In Figure 16 below, sensors are probes of the system dynamics, effectors are implementation points in the system, regulators are effectors, which modulate the group of managed resources for smooth operations and receptors are sensors, which detect changes in the operation integrity of the systems. Finally, the Knowledge Bases hold the derived data representing related experiences used in the control loops specific to the task at hand.

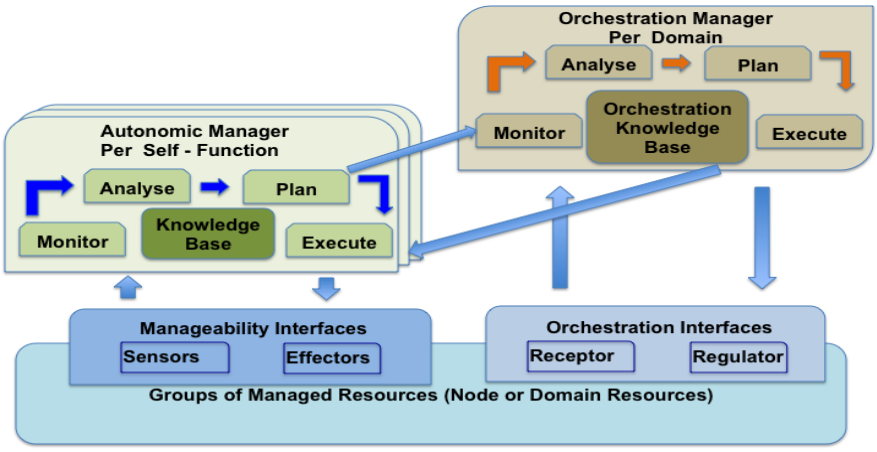


Figure 16. Orchestrated MAPE Process.

5.4.1.4 Mapping of functional blocks into UMF physical and logical nodes

In Figure 17 we have mapped the functional blocks described previously into the management nodes. In this figure we present only the types of nodes and not their different instances that will be in the different domains (access, backhaul, core, service) since in terms of functional groups the tasks to be performed seem the same. However, it is obvious that the further analysis of the autonomic tasks per different domains and nodes is needed (e.g., the autonomic tasks performed by an eNodeB are very different from the ones of a core network router, or an IPTV server) and it will give us a more detailed view of the overall system.

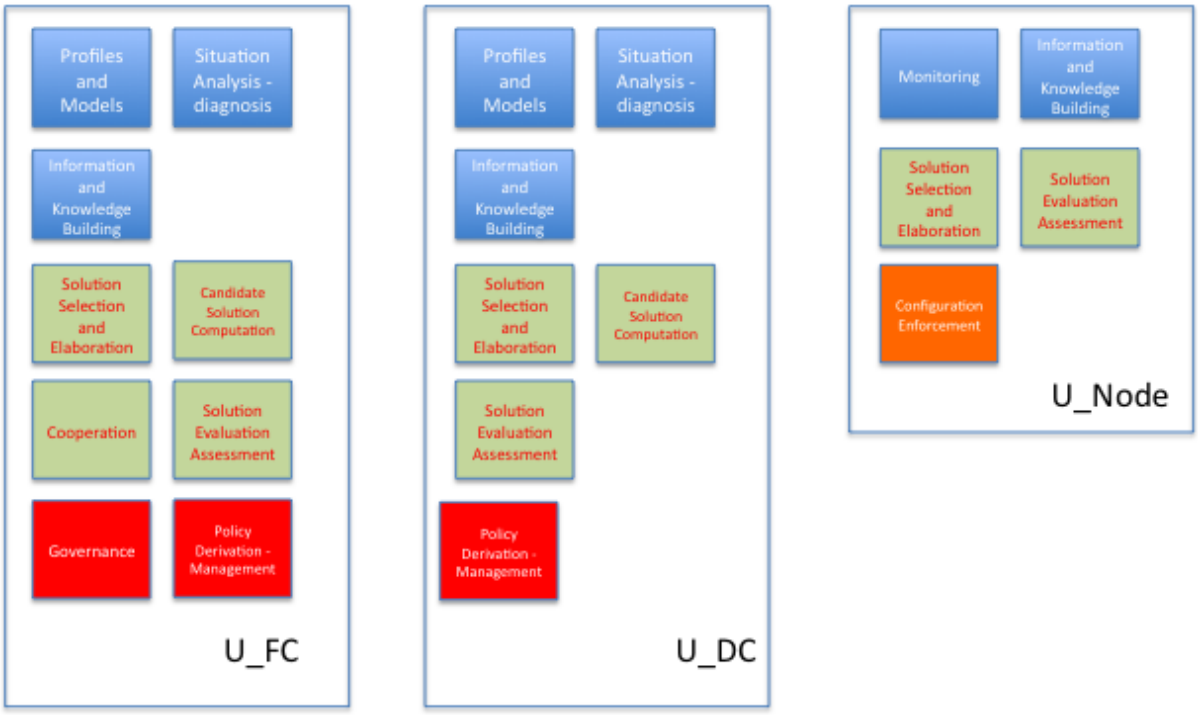


Figure 17. Functional blocks placement in U_Node, U_DC and U_FC.

In Figure 17, we can see that an U_Node is actually performing only monitoring actions and builds information and knowledge that are related solely to its own operation. The collected data are manipulated and locally stored by the information and knowledge building functional block. Afterwards they are processed autonomously by the two functional blocks (i.e., Solution selection and elaboration and Solution evaluation/assessment) of the "Intelligence's" functional group. These blocks constitute the autonomic decision making part of a UMF node and the evaluation of actions related only on the operation of an U_Node.

Finally, an U_Node has the configuration enforcement functional block. This block is present only in this level since it incorporates the actual execution of configuration actions.

The U_DC has embedded functionality related to knowledge management (all functional blocks except of monitoring that is already present in every U_Node). Note here that at this level these blocks are related only for intra domain information (e.g., summarized information received by all nodes of the domain, profiles and model in a detail level related to the equipment of the domain etc). U_DC has also policy derivation and management functionality (from the Governance functional group) that allows it to collect policies from U_FC and distribute them into all U_Nodes of the domain. As far as Intelligence Functional group is concerned, U_DC has all functional blocks (except from cooperation FB) that are related only to the orchestration of intra-domain actions.

Finally, U_FC contains all functional blocks from the Governance and Intelligence functional groups. However at this level the related functionality is only related to inter-domain communication, orchestration and evaluation of network management actions. At this level also the high level business goals are set and translated into policies. Finally, in relation to Knowledge management it has the same functional blocks as in the U_DC case. These however are involved only for operations related to inter U_FCs tasks.

5.4.2 UMF Global view and a migration path

In this section, a global view of UMF is presented as well as the migration path of existing network management systems is discussed. This migration path is very important for the operators since it is not realistic to expect that they will throw away the legacy network management systems. On the contrary, it sounds more logical to expect that the introduction of UMF functionalities is going to be gradual.

As mentioned in the previous section, different functional blocks are distributed among management systems, i.e. the required functionalities supported by FBs and grouped into FGs, can cover those needed to be supported by the existing network management structure. Looking again at the definition of U_DC and U_UFC we see that it is trivial to map U_DC functionality to EMS systems and U_FC functionality to NMS and OSS/BSS systems.

Thus, as a starting point, our inclusive management solution (UMF), involving all BSS/OSS, NMS and EMS, can be assumed as a stack of systems, considering business and operational support, network management and element management. The topmost level of this stack is where the human network operators interact with the system. From the other side, a management system consists of different components, which actually run and provide management functionalities. These functionalities are supported by FBs and in an abstract form by FGs. This concept is shown in Figure 18. UMware, residing on different systems of the UMF support various functionalities. Examples on how the UMware functional blocks are distributed among the management systems are provided in the following subsections, each addressing one of the UniverSelf use cases.

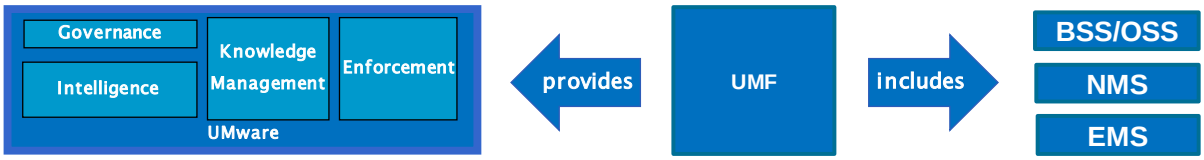


Figure 18. UMF functionalities and internal systems.

As defined earlier, any Legacy Management System, Enhanced with the capability of interacting with other management systems, called ELMS, and any Management System, supporting Future networks, called FMS can be operational within UMF. The nodes that host this ability are called U_Nodes. The U_Nodes can interact with each other within the same Management domain, by the coordination of the U_DCs. The federation between different management domains are coordinated by U_FCs. These UMF components consist of different management nodes (such as computers running management functional blocks) and can contain one or more management systems (e.g. B/OSS, NMS or EMS).

The interactions between different components of the UMF are supported by an Internetworking Bus, as shown in section 5.4.1.2.

An integrated global view of the UMF can be seen in Figure 19. In this figure, UMF components, each containing a number of Management or Support Systems, are shown as 3D shapes crossing BSS/OSS, NMS and

EMS planes. The management is applied to the Resource and Services (within NE plane) by the functionalities provided by management nodes.

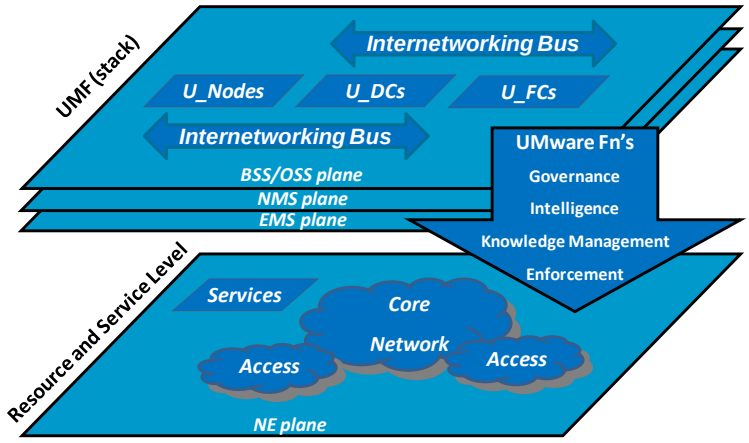


Figure 19. UMF Global View.

5.4.3 Overview of the identified functionalities and mapping to network layout per use case

In this section, each use case is investigated, and how different functional blocks are placed within this system. Details on the message exchange between different Functional Blocks can be found in Annex B.

UC1 - Self-Diagnosis and self-healing for IMS VoIP and VPN services

As it can be seen when combining sections 5.2 and 5.3, UC1 involves all the functional groups (FGs) with the functional blocks (FBs) that are presented in Figure 20.

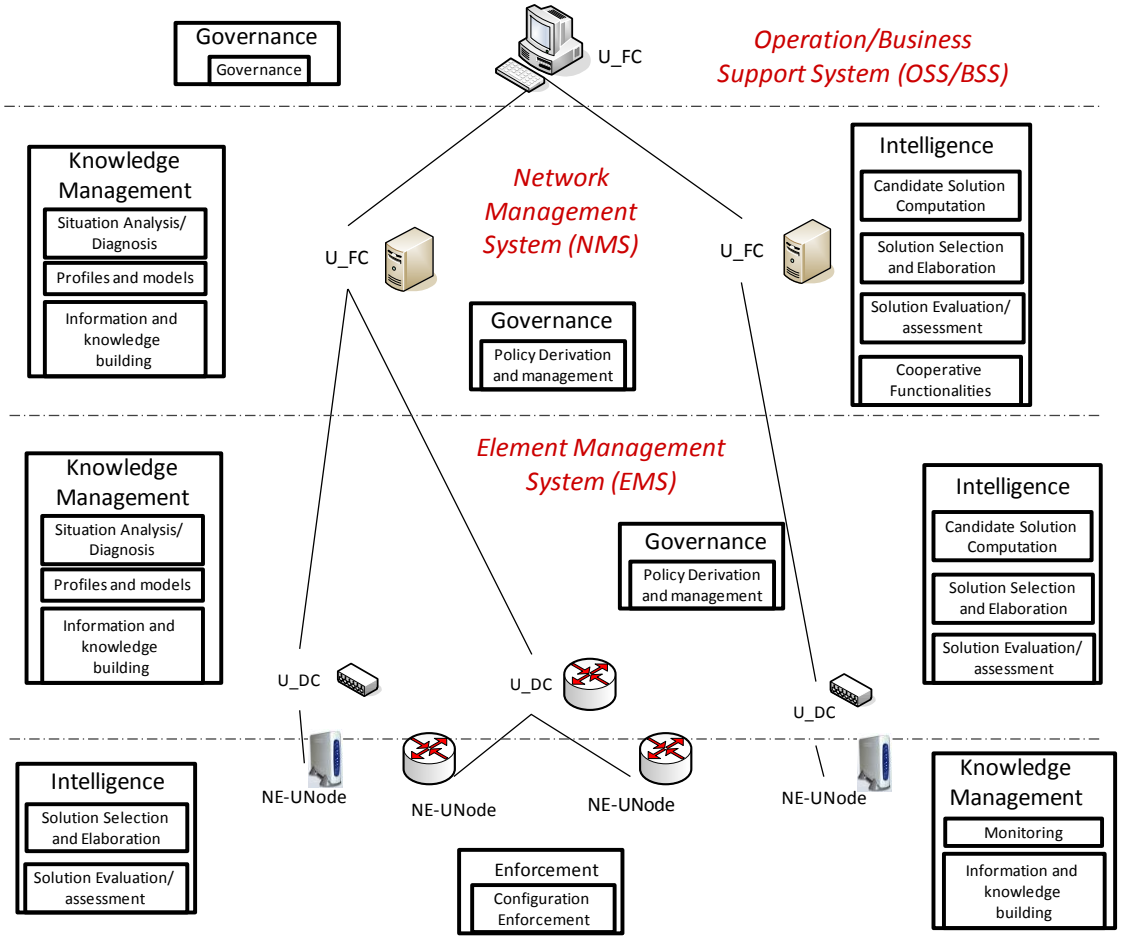


Figure 20. Mapping of UC1 FBs and FGs to the operator organization levels.

In particular:

- The responsibility for the insertion of the models, the goals and the reparation/ mitigation plans lies on Operator, thus Governance FB is mapped to OSS/ BSS.
- The elaboration of the models and the business goals for the creation of the policies is expected to take place in the NMS or in the EMS with respect to the level of the policy (high or low level policies), thus Policy Derivation and management FB is part both of NMS and the EMS.
- Situation Analysis/ Diagnosis FB can also be part of either the NMS or the EMS so as to create the necessary alarms depending on the origin of the alarm, e.g. if the alarm is related to a malfunction of an element or of an EMS then the alarm is generated by the respective EMS, alternatively, in case of an NMS malfunction, the NMS itself generated the respective alarm to be further elaborated.
- Regarding "monitoring" FB from the network view (network and performance data), this functionality should be encompassed in all elements, i.e. this functionality is part of the NE. Additionally, the overall monitoring of the network segments (in coarse mode) is also done by the NMS. If a problem is identified then the latter takes the responsibility of investigating it, collecting the related information and possibly coordinating the reparation actions. The same functionality is also present in the EMS but this is related only for the directly controlled NEs. Finally, regarding the customer view, this information will be possibly inserted to the system by the customers or the customer care service through a H2N interface to a database. This process, and thus the FB as well, are part of the BSS. The acquired data from this functionality will be then transferred to the mechanism of the NMS that will encompass the functionality of creating the alarms triggered by the users.
- As already stated, Information and Knowledge Building FB holds knowledge data with the latter taking part into many processes, thus it should be placed in a central management system, namely the NMS or the EMS depending to the type/ subject and how global or local is the respective knowledge.

- The actual (re-) configuration enforcement is performed (executed) by the respective network element. If the configuration action fails for certain reason then the operator will be notified to solve the issue manually.
- Candidate Solution Computation FB is triggered by the reported alarms (coming from the Situation Analysis/ Diagnosis FB of NMS or EMS) and its scope is to identify the possible reparation/ mitigation plans while Solution Selection and Elaboration FB is responsible for the selection of the most appropriate reparation/ mitigation plan given the context of the network. Due to their close cooperation to each other and with the Situation Analysis/ Diagnosis FB and given the capabilities of the existing management systems, these FBs are also mapped to the NMS or the EMS, accordingly. Eventually, NMS/EMS sends the respective configuration commands to the EMS of the involved elements.
- Since we are interested in an end to end evaluation, this should be made "above" the network and not in a specific element. Thus, Solution Evaluation/ Assessment FB is considered as an OA&M functionality hosted in the NMS. Furthermore, as the end to end evaluation is also based on customer/ user satisfaction and QoE, the functionality takes inputs from the (database of) BSS as well.

UC2 – Networks' Stability and Performance

As can be seen when combining sections 5.2 and 5.3, UC2 involves the functional blocks (FBs) that are presented in Figure 21 and thus all functional groups (FGs).

The mapping of these FBs is depicted in Figure 21 and is d bellow:

- The responsibility for the insertion of the models, the goals and the reparation/ mitigation plans lies on Operator, thus Governance FB is mapped to OSS/ BSS.
- The elaboration of the models and the business goals for the creation of the policies is expected to take place in the NMS, thus Policy Derivation and management FB is part of NMS.
- Regarding "monitoring" FB from the network view (network and service data), this functionality should be encompassed in all elements, i.e. this functionality is part of the NE. However, regarding the customer view, this information will be possibly inserted to the system by the customers or the customer care service through a H2N interface to a database. This process, and thus the FB as well, are part of the BSS.
- Information and Knowledge Building FB can be hosted in the NMS.
- The self-configuration of each element should be designated by its management system, thus EMS applies Configuration Enforcement functionality to the respective elements it manages.
- Eventually, since the network evaluation is selected to be end to end, the evaluation needs to have an "above" the network view and not an element specific one. Thus, Solution Evaluation/ Assessment FB is considered as an OA&M functionality hosted in the NMS.

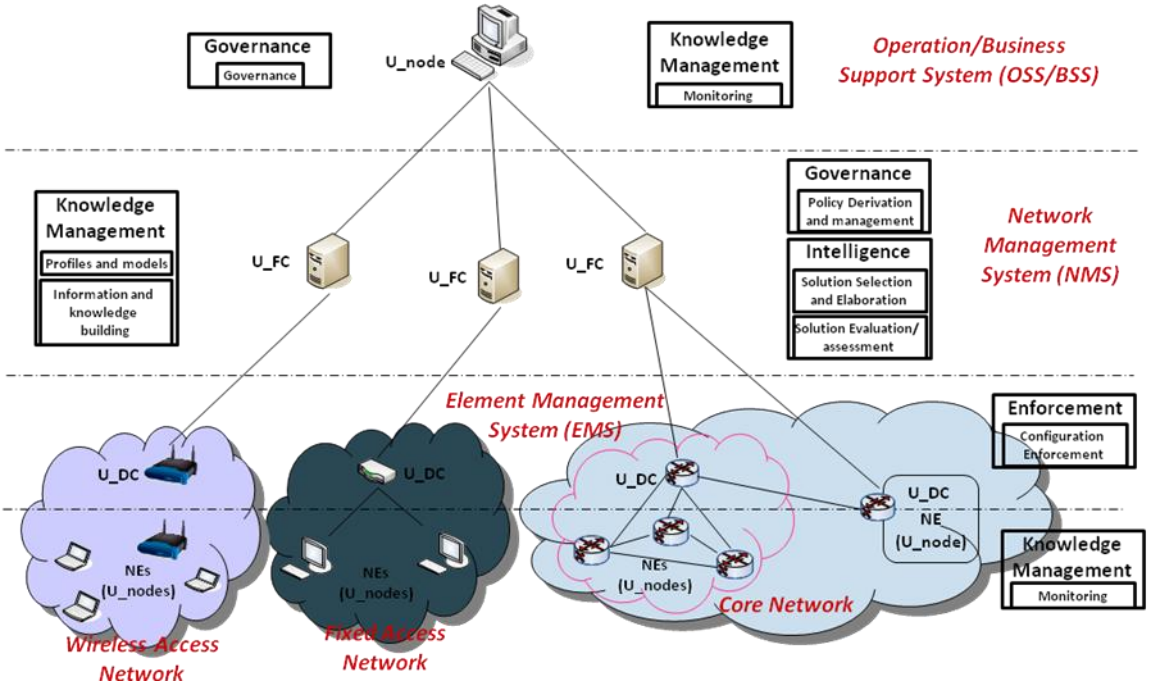


Figure 21. Mapping of UC2 FBs and FGs to the operator organization levels.

UC3 – Dynamic Virtualization and Migration Contents and Servers

Figure 22 depicts the mapping of the functional groups on the LTE network. More specifically:

- **Governance:** It is located both at OSS when H2N interface uses the Governance FB and at NMS when the Policy Derivation and Management FB is used.
- **Knowledge Management:** It is located at NMS and EMS (at eNodeBs and Core network). The Monitoring FB exists both at NMS and EMS and is used for the measurement of various network measurements. These measurements are processed by the Situation Analysis/ Diagnosis FB at NMS. Moreover, the Profiles and Models FB provides information about user preferences and behaviour for various contexts and is located at NMS. Finally, Information and Knowledge Building FB is located both at NMS and EMS.
- **Intelligence:** It is located at NMS. The Candidate Solution Computation FB provides a set of candidate networks or network domains to be reconfigured based on the parameters acquired from the Situation Analysis/ Diagnosis FB. Solution Selection and Elaboration FB computes specific reconfiguration actions that need to be accomplished. Solution Evaluation/ Assessment FB evaluates the decisions of the previous FB using a set of conflict resolution policies that are present in the network.
- **Enforcement:** It is located at EMS. The Configuration Enforcement FB applies the (re)configuration decisions of the Intelligence functionality to the network elements/ resources.

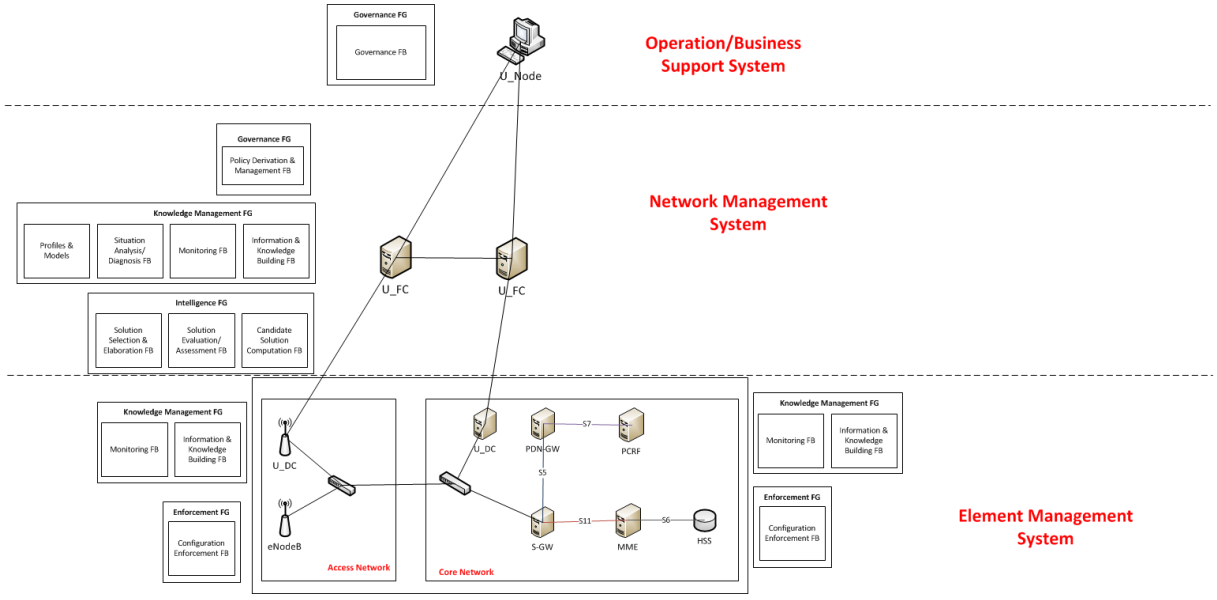


Figure 22. Mapping of functional blocks/groups on LTE network.

UC4 - SON and SON Collaboration According to Operator Policies

The network topology of this specific use case consists of a heterogeneous environment, including user terminals, eNodeBs, pico and femto cells, relays etc at a low architectural level and NMS, OSS at a higher level. However, only eNodeBs and NMS, OSS will be d here in detail since they are the main intelligent entities where the derived functional groups will be mapped. Communication between eNodeBs is enabled through X2 interface and between eNodeB and NMS through Itf-N interface.

An enhanced OSS is used at the highest level of hierarchy, namely an appropriate U_Node. Then, an U_FC is located inside NMS per domain in order to assist in orchestrating the operation of one or more U_DCs inside a domain. Finally, inside a domain, UMF nodes i.e. eNBs elect an U_DC to act as a group controller with the role to coordinate the UMF nodes of its group. Of course, extensions to the current interfaces (X2, Itf-N etc.) will be needed.

A possible mapping of the functional groups on the UMF components and the LTE network elements in the context of the specific use case is as follows:

Knowledge Management: it is located in both NMS and eNodeB, but different functional blocks may be instantiated in each case. Situation Analysis/Diagnosis FB is only used in the U_FC@NMS, in order to determine the involved SON entities based on the operator targets, since this needs to be done at a high layer. Monitoring FB is located both in eNodeBs, U_DC and in U_FC@NMS, since existing measurements should be processed in both of them. Finally, Information and Knowledge Building FB is located in both U_FC@NMS and U_DC. U_FC@NMS needs knowledge functionality about SON entities and their location (Situation Analysis/Diagnosis FB), already active policies (Policy Derivation and Management FB), bandwidth allocation and on how achieving efficient SON processes through coordination (Solution Selection and Elaboration FB). U_DC needs knowledge functionality about SON coordination (Solution Selection and Elaboration FB).

Governance: it is located in U_Node@OSS when Governance FB and the H2N tool are involved and in U_FC@NMS when the Policy Derivation and Management FB is used. Policy Derivation and Management FB intends to generate the SON entities specific policies based on the output of the Situation Analysis/Diagnosis FB that means the information about the involved SON entities and the operator targets. This functionality, related to governance and the translation of operator targets to SON specific policies, needs to be done at a high layer.

Intelligence: it is located in both U_FC@NMS and U_DC. When Solution Selection and Elaboration FB is involved, it resides both U_FC@NMS and U_DC, in order to coordinate the SON entities in order to enforce the policies derived by Policy Derivation and Management FB. Solution Selection and Elaboration FB is actually the decision making procedure and consists of various interacting, even conflicting, control loops. Moreover, network performance problems are tackled through the SON coordination. When Solution

Evaluation/Assessment FB is involved, it resides U_FC@NMS, in order to evaluate the SON process and coordination end-to-end and to trigger re-optimizations or for new operator goals when certain KPI thresholds are crossed and/or degradation exists. It is noted that evaluation is not explicitly introduced in eNodeBs, since it is considered that it is a typical, already existing prerequisite in SON. Cooperation FB resides U_FC@NMS and is used only in the case that different administrative domains exist, which are controlled by different U_FCs, and there is a need for this FB to assist the inter-domain communication among the container U_FC and other U_FCs.

Enforcement: it is not explicitly mapped somewhere, since the enforcement in SON takes place through the already existing, self configuration procedures.

Figure 23 depicts the aforementioned mapping of functional groups on the related to the use case LTE network topology and UMF components. Inside each functional group, the instantiated functional blocks exist, dependently on the location of the functional group.

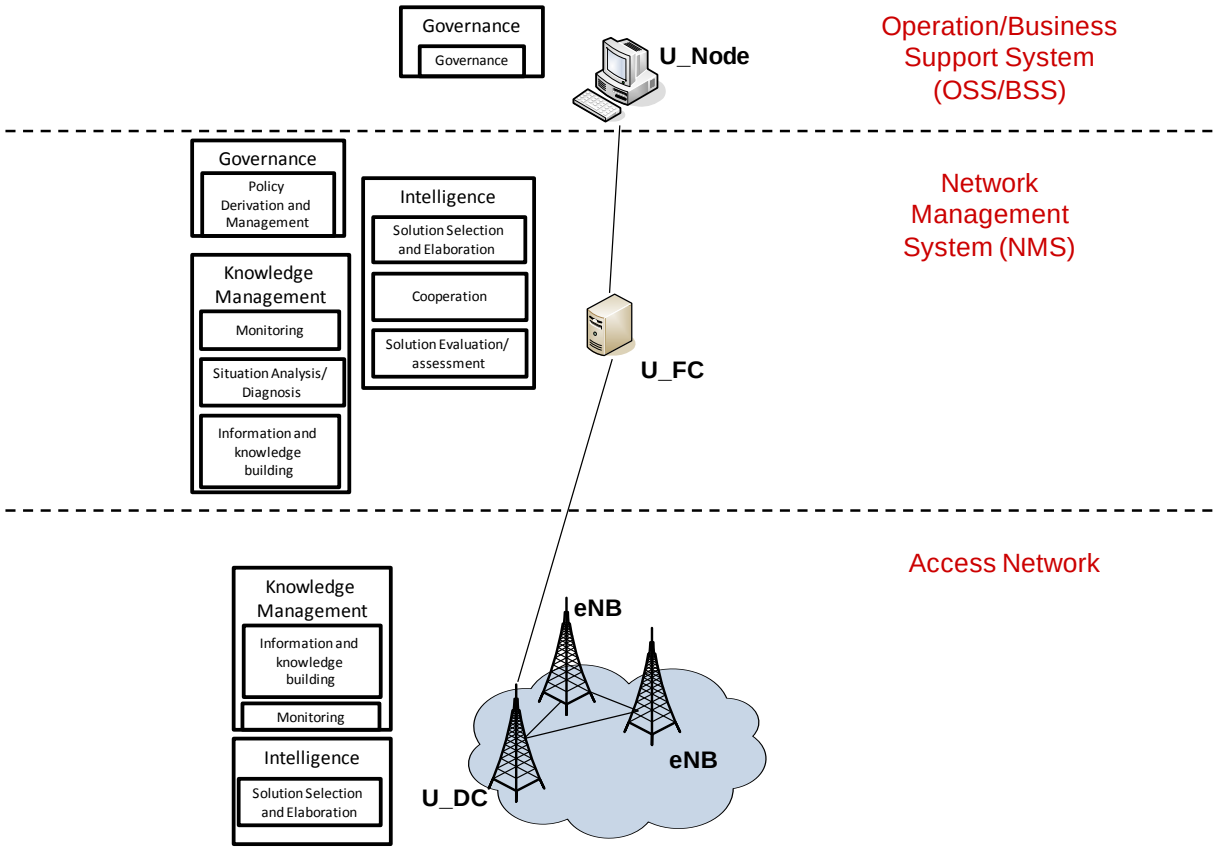


Figure 23. Mapping of functional groups to the UC4 network layout.

UC6 – Operator-Governed, End-to-End, Autonomic, Joint Network and Service Management[2]

The use case targets end-to-end topology, covering from the human console down to the wireless/mobile end-user device, considering all the main parts of a today's mobile network, namely access, backhaul and core segments, as well as the application domains (servers). In the wireless segment, the use case is extended so as to cover multi-hop networks based on relay nodes. In particular UC6 involves the following Functional Groups and Functional Blocks:

- Governance FG: Governance FB and Policy derivation and management FB
- Knowledge Management FG: Profiles and Models FB, Situation analysis/ diagnosis FB, Monitoring FB and Information and Knowledge Building FB
- Enforcement FG: Configuration enforcement FB

[2] Please note that UC5 – Network Morphing has been merged with UC6.

- Intelligence FB: Candidate Solution Computation FB, Solution selection and elaboration FB, Solution evaluation/ assessment FB and Cooperative Functionalities FB.

Finally, Figure 24 depicts a possible mapping of the UMF functional blocks (and accordingly of functional groups) into the different entities of the network layout envisaged in this use case.

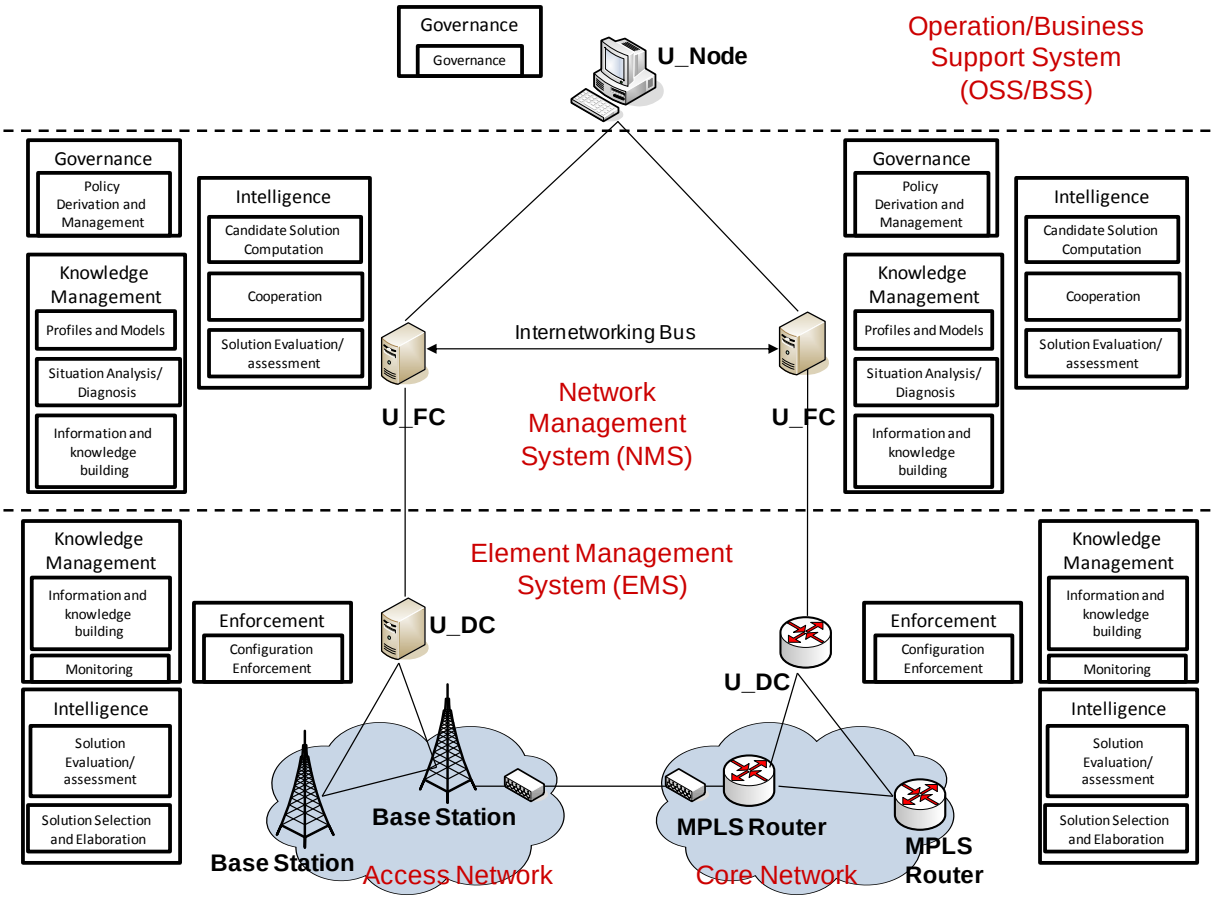


Figure 24. Mapping of UC6 in terms of Functional Blocks and Functional Groups to the network layout.

UC7 – Network and Services Governance

This Use Case defines the architecture, requirements and solution to manage processes associated with the whole lifecycle of the services related to IPTV and how it is possible to change the network conditions in order to provide the service with the correct quality.

This includes the deployment of an enhanced policy framework for managing the network elements according to business goals. Also, a human to network tool, policy modelling and methods for propagation through the infrastructure will be developed as part of this use case.

The main segments-actors in an IPTV network are (their roles are described):

- Customer premises: Where the IPTV stream is terminated and viewed.
- Access network: Distributes the IPTV streams to the DSLAMs
- Aggregation network. This is an Ethernet network aggregating the traffic from a number of bandwidth locations to be feed to an IP network.
- IP network. This is the backbone network composed by a few IP edge and transit routers.
- Both, aggregation and IP networks are using an transport SDH or WDM network owns by the Operator or rented from another Network provider
- TV head-end: Where most of the IPTV channels enter the network from national broadcasters

The main functionalities of the aforementioned elements are (required for the efficient execution):

- Express the business goals in a high level language, without the need of knowledge to represent the information in a more technical language.
- Translation of QoS requirements specified for a service need to network parameters, such as jitter, packet error rate, etc.
- Scheduling mechanisms for policy-based self-adjustment of resources to handle the network performance degradation.
- Management of network resources to different nodes (Access, Aggregation, Core networks)
    - RAN: Radio resource allocation, Admission/Congestion control and scheduling parameters, relay selection in case of multi-hop networks, link positioning, and compensation by means of SON mechanisms.
    - Core: LSP configuration in IP/MPLS case. At the core side, it also involves GW (e.g., SGW, PDN-GW) (re)selection/configuration, GW migration/dimensioning (in case of decentralized, virtualized GWs, cf. Use case 1.3) and content management.
    - Negotiation and cooperation between segments
- Mechanisms/algorithms for the network elements to self-discover their neighbours
- Network elements in FTTH environment must be able to monitor their operational context
- Network elements in FTTH environments must be able to make a probabilistic self-diagnosis based on their own state and their operational context.

Trust management schemes for detection of faulty/malicious behaviours of network elements based on operator policies.

- Coherence should be guaranteed when dealing with a multi-domain environment (e.g. resolution of incompatibilities between the offered QoS from RANs and backhaul/core segments, respectively).
- Network elements should collect measurements in order to ensure that the desired QoS level is guaranteed during the operational phase of the service.

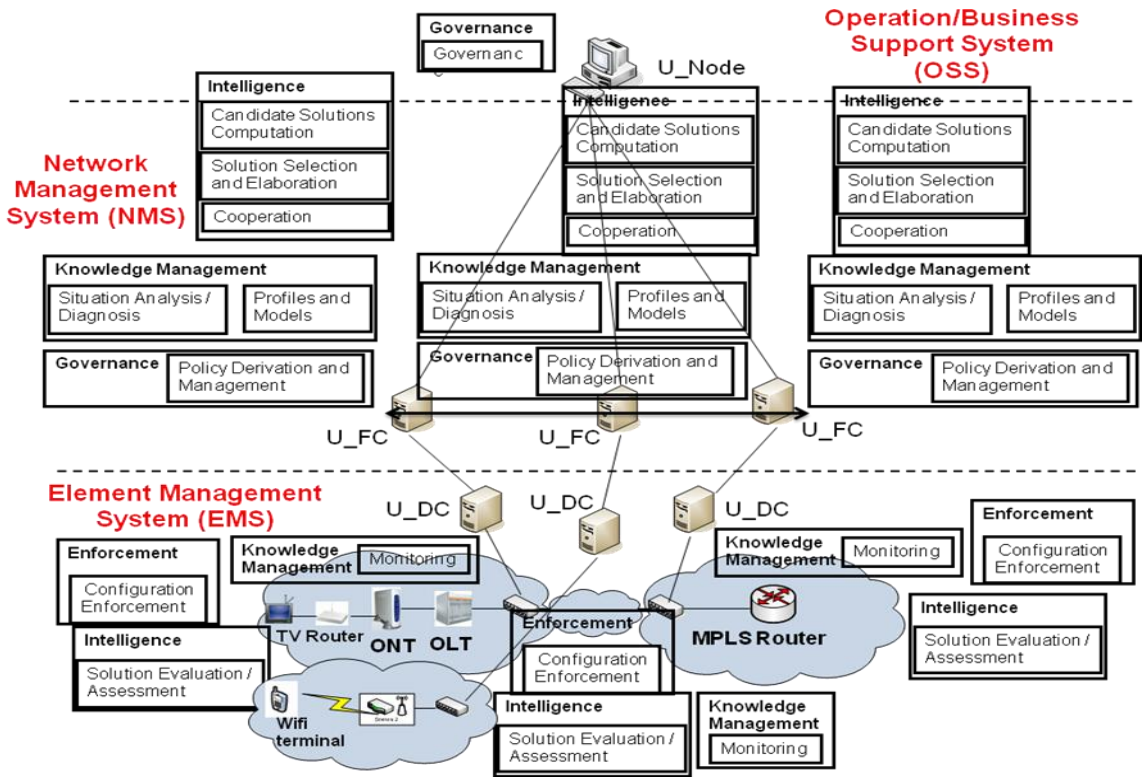


Figure 25. Mapping of UC7 functionalities in terms of functional groups/blocks to the network layout.

5.4.4 View per Network Segment: Consolidation of messages

As it can be observed, many messages among the UCs (see also Annex B) are common. The following table contains messages that can meet the needs of every functional block in every use case. The first column explains in brief the targeted segment of the message being an option among wireless access, wireline access, core and service segments, respectively. The second column provides the name of the FB that the message originates from. The third column is the destination FB of the message, while the fourth column contains a tentative name for the message.

Table 3. Table of consolidated among different UCs messages

Segment	Message purpose	Source	Destination	Message
Service	Notification on the number of new users to be served (per Application/User Class/ Location/Time etc)	Governance (@OSS)	Situation Analysis and Diagnosis (@NMS)	BusinessLevelEntryNotification
Service	Update the association of Applications to User Classes and Quality Levels	Governance (@OSS)	Policy Derivation and Management (@NMS)	AssociationNotification
Wireless, Wireline, Core	Update the policies that have to be followed when taking management decisions	Policy Derivation and Management (@NMS)	Solution Selection and Elaboration (@NMS)	GovPolicyNotification
			Configuration Enforcement (@EMS)	
Wireless	Provides SON-specific policies and triggers offline or/and online SON coordination	Policy Derivation and Management (@NMS)	Solution Selection and Elaboration (@NMS,@eNB)	SONPolicyNotification
Wireless, Wireline, Core	Notification of the number of new users to be served (per Application/User Class/Cell)	Situation Analysis/Diagnosis (@NMS)	Solution Selection and Elaboration (@NMS)	NewContextNotification
Wireless	Identifies the involved SON entities and their location	Situation Analysis/Diagnosis (@NMS)	Policy Derivation and Management (@NMS)	SONDetermination
Wireless, Wireline, Core	Apply new configuration according to decision	Solution Selection and Elaboration (@NMS)	Configuration Enforcement (@EMS)	ReconfigurationRequest
Wireless, Core	Requests checking of reconfiguration actions for conflicts	Solution Selection and Elaboration (@NMS)	Solution Evaluation and Assessment (@NMS)	ReconfigurationEvaluationRequest
Wireless, Core	Sends conflict-free reconfiguration actions	Solution Evaluation and Assessment (@NMS)	Solution Selection and Elaboration (@NMS)	ReconfigurationEvaluationResponse
Wireless, Wireline Core	Notify on new configuration application	Configuration Enforcement (@EMS)	Solution Evaluation and Assessment (@NMS)	ReconfigurationExecutionNotification

			Monitoring (@NE)	
Wireless, Wireline, Core	Request context information	Solution Selection and Elaboration (@NMS)	Monitoring (@EMS)	ContextRequest
Wireless, Wireline, Core	Send requested context information	Monitoring (@EMS)	Solution Selection and Elaboration (@NMS)	ContextReply
Wireless, Wireline, Core	Send unsolicited context information	Monitoring (@EMS)	Situation Analysis and Diagnosis (@NMS)	ContextNotification
Wireless	Provide the KPIs to be monitored	Solution Selection and Elaboration (@NMS,@eNB)	Monitoring (@eNB)	KPIDetermination
Wireless	Transfer KPIs measurements	Monitoring@NMS	Solution Evaluation/Assessment@ NMS	KPI_Information
		Monitoring@eNB	Monitoring@NMS	
Service	Notify about a violation and prompts for operator	Solution Evaluation and Assessment (@NMS)	Governance (@OSS)	ViolationNotification
Service	Information about user preferences	Monitoring (@EMS)	Information and Knowledge Building (@NMS)	UserInfoNotification
Wireless, Core	Information about area characteristics	Monitoring (@EMS)	Information and Knowledge Building (@NMS)	AreaInfoNotification
Service	Request user profile information	Solution Selection and Elaboration (@NMS)	Profiles and Models (@NMS)	UserProfileRequest
Service	Send user profile information	Profiles and Models (@NMS)	Solution Selection and Elaboration (@NMS)	UserProfileResponse
Wireless, Wireline, Core	Send application policies	Policy Derivation and Management (@NMS)	Solution Evaluation and Assessment (@NMS)	ApplicationPolicyNotification
Wireless, Wireline, Core	Feeds low level information/ data to the mechanism which is responsible for their analysis	Monitoring (@NE)	Situation Analysis and Diagnosis (@NMS)	LLDataProvision
			Information and Knowledge Building (@NMS)	

Wireless, Wireline, Core	Carries elaborated data (High Level) and correlations of low level data to the "candidate solution computation" mechanism	Situation Analysis and Diagnosis (@NMS)	Candidate Solutions Computation (@NMS)	HLDataProvision
Wireless, Wireline, Core	Informs the decision making mechanism for the diagnosis made	Candidate Solutions Computation (@NMS)	Solution Selection and Elaboration (@NMS)	DiagnosisProvision
Service	Provides the analysis mechanism with the reports coming from customers/ customer care service	Monitoring (@BSS)	Situation Analysis and Diagnosis (@NMS)	CustomerReport
Service	Gives feedback for the customers' QoE after the healing	Monitoring (@BSS)	Solution Evaluation and Assessment (@NMS)	QoEReport
Service, Wireless, Wireline, Core	Informs the Candidate Solution Computation FB about the available models	Profiles and Models (@NMS)	Candidate Solution Computation (@NMS)	ModelProvision
	Informs the database that holds the Profiles and the Models with the new models	Governance (@OSS)	Profiles and Models (@NMS)	
Service	Feeds Policy Derivation and Management FB with the business goals so as the latter to create policies	Governance (@OSS)	Policy Derivation and Management (@NMS)	Goals Provision
	Feeds knowledge database with business goals		Information and Knowledge Building (@NMS)	
	Feeds Policy Derivation and Management FB with the business goals so as the latter to create policies	Information and Knowledge Building (@NMS)	Policy Derivation and Management (@NMS)	
Wireless, wireline, Core	Transfers past knowledge (evaluation of past actions) to the candidate Solution computation FB	Information and Knowledge Building (@ NMS)	Candidate Solution Computation (@NMS)	EvaluationReport
	Feeds decision mechanism with knowledge related to (end to end) evaluation of solutions applied in the past		Solution Selection and Elaboration	
	Transfers the new configuration values and the respective QoEValue to knowledge base for future use	Solution Evaluation Assessment (@NMS)	Information and Knowledge Building (@NMS)	
Wireless, wireline, Core	Alerts the network administrator through the H2N interface if the reconfiguration was unsuccessful	Configuration Enforcement (@EMS)	Governance (@OSS)	ReconfFailure

Service	Feeds knowledge base, and the latter the decision mechanism, with the information of the available network resources	Governance (@OSS)	Information and Knowledge Building (@NMS)	NetResources
		Information and Knowledge Building (@NMS)	Solution Selection and Elaboration (@NMS)	
Service	Provides the knowledge building mechanism with user/customer data so as the latter to be collected, filtered and elaborated	Monitoring (@BSS)	Information and Knowledge Building (@NMS)	CustData
Wireless, wireline, core	Feeds current state of the networks (monitored values) to the solution evaluation/ assessment mechanism for the end to end evaluation	Information and Knowledge Building (@NMS)	Solution Evaluation/ assessment (@NMS)	NetwKnow

6 Areas covered by UMF enablers

This chapter will highlight technological challenges addressed by UMF enablers and provide initial thoughts on how to handle them. These main enablers corresponds to the three additional tasks of the work package 2 which are: Task 2.2 on Information and Knowledge Management, Task 2.3 on Network Governance, and Task 2.4 on Intelligence Embodiment. The material presented in the following sections is preliminary results based on the investigations and analysis done in the respective tasks. It is useful to mention that the work of these tasks has started at month 7 of the project (March 2011), while Deliverable D2.1 is primarily the result of the work achieved in Task 2.1 on UMF Design. Therefore, this material shall be considered as starting points/work in progress and directions of study for the respective tasks.

6.1 Intelligence Embodiment mechanisms

This section starts from a presentation of the concept of Intelligence Embodiment. Then, the challenges for Intelligence Embodiment in the UMF are addressed. An initial approach of design aspects of Intelligence Embodiment is presented as well as an evaluation of the Quality of embedding. An outline of relevant State of the Art areas is provided in Annex C.

6.1.1 Purpose of Intelligence Embodiment

As stated in the Description of Work (DoW) the main objective of WP2 is to "deliver a Unified Management Framework (UMF) that targets the embedding of autonomic paradigms in any type of network in a consistent manner". Furthermore, it is stated that the main goal of Task 2.4 on Intelligence Embodiment" is to enable the embedding of intelligence in the network equipment".

In the following, it is attempted to further detail the concept of intelligence embodiment in the scope of the UMF, starting from definitions of intelligence and embodiment in Artificial Intelligence. Despite the vast work in Artificial Intelligence, it is difficult to find one definition for intelligence in the scope of computer systems, autonomic computing, and autonomous systems. According to the IBM Autonomic Computing glossary Artificial Intelligence (AI) is "The capacity of a computer or system to perform tasks commonly associated with the higher intellectual processes characteristic of humans. AI can be seen as an attempt to model aspects of human thought on computers. Although certain aspects of AI will undoubtedly make contributions to autonomic computing, autonomic computing does not have as its primary objective the emulation of human thought". Embodiment in artificial intelligence is often viewed as the property of intelligent entities (e.g. agents) to "interact with the environment through a physical body within that environment" [46]. However, embodiment does not necessarily require a "material" body; the "dynamic relation with the environment" is the key requirement [47].

In this context, intelligence embodiment comprises all necessary mechanisms for enabling the *embedding* of intelligence into a global system. In this direction, intelligence in the scope of the UMF can be defined essentially as software corresponding to self-x (i.e. self-management, self-configuration, self-optimisation, self-healing) features/algorithms, autonomic capabilities/functionalities. Intelligence embodiment will facilitate the embedding of features/algorithms into network equipment and the UMF in a “plug-and-play” fashion. This includes the introduction (addition), deployment, invocation, integration and orchestration with existing solutions, potentially removal of self-x features/algorithms, autonomic capabilities such as autonomous decision-making algorithms, optimization algorithms, learning mechanisms. It should be noted that embodiment of intelligence into an entity does not have to be at the physical level, i.e. actually integrating intelligence into a hardware element. Embedding of intelligence may be "virtual", achieved by means of high-level interfaces and middleware components, thus allowing also for legacy network elements to become more intelligent.

6.1.2 Challenges

Innovative and sophisticated means are required that will facilitate the introduction of intelligence features and will allow for faster deployment of new services and applications. In the following, we call “intelligence component” any software unit that can add to the intelligence of a system, by acting autonomously, according to a set of goals or tasks for which they are designed, facilitating more efficient operation of the

system/network. An imperative requirement is the openness of the solution that will allow for the introduction of new intelligence components and functions as well as the re-use, exploitation, removal or replacement of legacy elements in a plug 'n play fashion. This in turn calls for the abstraction (generic description) of self-x features/algorithms and autonomic capabilities through high-level interfaces, so as to allow the decoupling of elements from intelligence components. This decoupling is important so as to allow for "virtual" embedding of intelligence, i.e. intelligence that is not strictly tied to specific hardware elements, thus enhancing legacy elements with autonomic capabilities. In this sense, an important requirement for achieving intelligence embodiment, are mechanisms for the high-level description and discovery of intelligence components. Description and discovery of intelligence components also provide the means for potential composition of intelligence components (through other components), thus enabling flexible reuse of various autonomic features, even for legacy elements. In other words, description and discovery of intelligence components can provide the means for identifying components/elements that can help manage an element/node does not have any autonomic capabilities. The requirement for description and discovery of intelligence components in turn calls for a common information model and appropriate semantics that will also allow for interoperability between different domains, heterogeneous devices and networks, intelligence components from diverse vendors. In more detail, in order to be able to discover within a system an intelligence component and use the functionalities/services it provides, mechanisms are required for its generic description. In addition, a common information model and semantics is necessary as a common language, enabling intelligence embodiment over different domains. Furthermore, there is a need for mechanisms for the setup, deployment, starting, of intelligence components once these are introduced into the system as well as information/knowledge models that will specify the scope for interactions with the others modules. In this sense, a set of guiding principles is required for intelligence embodiment, which will explain how to design a software component that can be easily integrated into a more global system. These should be encompassed in the UMF.

Orchestration functionalities are also very important in the scope of intelligence embodiment for the coordination of newly introduced intelligence components with already existing intelligence components/features. Interoperation of intelligence components potentially stemming from different vendors is required.

Various concepts related to intelligence embodiment have been the objective of several research efforts. Relevant findings can be exploited for the specification of intelligence embodiment within the UMF. More specifically, the use of ontologies may be exploited in the scope of defining a common information model comprising the type of components, their properties and relations. In this sense, ontologies may be exploited for the description of intelligence features thus also facilitating their discovery and exploitation by other components of the overall system. Notions from semantics may be applied for discovery. Programmable networks include the notion of mechanisms that allow for the automated configuration of the functionality and behaviour of network elements, as well as the separation of network infrastructure hardware (i.e., switching fabrics, routing engines) from control software. Similar concepts have also been developed in the scope of pervasive computing. The description, discovery and orchestration of various components are key features in service oriented computing. Concepts related to interoperability and virtualisation, i.e. abstraction of infrastructure elements through generic, high-level interfaces and their decoupling from intelligence that runs on top of them are key issues in areas such as Software-as-a-Service (SaaS) and Infrastructure-as-a-Service. All these relevant research efforts are further explained in the corresponding State of the Art overview Annex C.

6.1.3 Design approach

Following concepts from State of the Art (e.g. programmable networks, service oriented computing, etc) every intelligence feature or infrastructure element can be virtualised as a set of services. Infrastructure elements can offer various services to intelligence features. Intelligence features can issue actions (e.g. reconfiguration actions) towards the infrastructure through high-level interfaces. As already introduced, intelligence features may include autonomous decision-making algorithms, information acquisition and knowledge management, self-management functionalities, etc. Each device/network element in a system can provide a set of services. For example a Base Station may provide a combination of a Monitoring service/function (i.e. Base station status), a Profile service/function (i.e. Base station profile) and a Configuration enforcement service/function (i.e. Base station reconfiguration). In a similar manner intelligence components may utilise other services of other intelligence components. For example reasoning/decision making components can be seen as composite services that utilise Monitoring, Situation Analysis and Diagnosis, Profiles and Models, Information and Knowledge Building, Policy Derivation and Solution Selection. An indicative high-level view of this concept is

presented in Figure 26, which depicts a set of indicative functional blocks (as defined in section 5), each comprising a set of intelligence components. In accordance to the concepts described in the previous each intelligence component, device or network element should provide a description of its capabilities and the services/functions it provides so that it can be discovered and exploited by other components in the system.

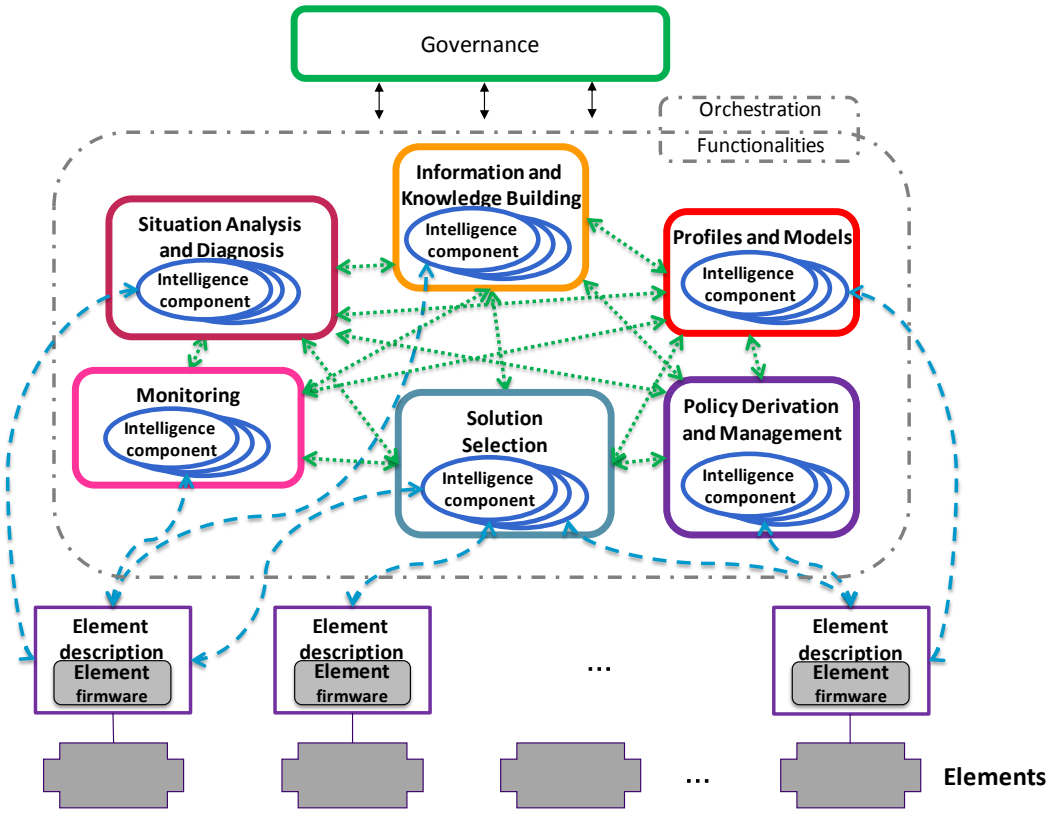


Figure 26. Indicative intelligence embodiment design approach.

6.1.4 Quality of Embedding

As mentioned in the previous intelligence embodiment comprises all necessary mechanisms for enabling the *embedding* of intelligence into a global system. Embedding is a non-functional property of an entity (thing, feature, property, function, mechanism, etc.) being surrounded by parts of another entity. Without loss of generality we are interested in such types of embedding, in which a surrounded entity itself is a part of a surrounding entity. Moreover we are interested in engineered entities – those that have a well defined structure. This way, the structure places the surrounding and surrounded parts of an entity into certain relation to each other.

Further understanding of the embedding and the explanation of what defines the Quality of Embedding (QoEm) require some deliberations about the notion of intelligence relevant to the project goals. Again, without loss of generality it is reasonable to define intelligence as a problem solving ability of an entity. Within the project this ability is domain-specific while domain definition can vary, meaning that e.g. radio-specific problem solving ability (e.g. interference sensing) is hardly useful for an entity within an optical segment of a network; at the same time intelligent load balancing algorithm might be useful in both wireless and wired domains.

The term "intelligent algorithm" used above holds the essence of intelligence embedding and explains what the entity is, into which intelligence is to be embedded. Indeed, an algorithm – the only subject of study in all computer science disciplines - might have a very broad variety of forms, however all of them, we conjecture can be compared based on the QoEm metric.

To work on the evaluation of the Quality of Embedding, would mean to be able to quantify the QoEm of a particular mechanism or method measured by its ability to sustain operational (network and service) changes experienced by the infrastructure. Such quantification will result in the assessment of particular self-x and cognitive solution in particular operator scenarios.

Assessment of self-x and cognitive methods is measured by their ability to solve particular problems, which ability can be characterised by the domain-specific know-how rather than by generic intelligence. The result of these studies also must be captured as project wide methodology that will contribute to the trend-setting certification process. Trust in Autonomics will be achieved via the [standardised] assessment within the certification [process] that verifies the process correctness of a system. This plan is captured by the concept map shown below (Figure 27).

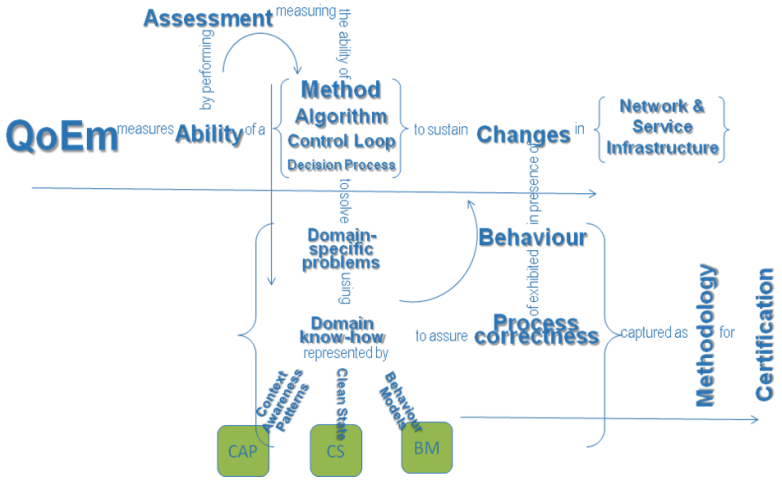


Figure 27. Concept map of the Quality of Embedding.

As the concept map demonstrates (by straight line arrows) there are several streams of activities in the studying of QoEm. The first stream is directly measuring the QoEm by relating environment changes to continuous correct operation of a method, algorithms, etc. The second stream is the assessment work per se, which includes definitions and formalisation of domain specific problems and the derivation and specification of a domain-specific know-how that appears useful for solving the above problems by the respective algorithms. The third stream of activities will capture the best current practices of the two previous streams and formulates methodology building blocks. There are two feedback streams (presented by curly arrows) between the previous streams. The first feedback stream will relate the direct QoEm measurements to the used assessment methods, which of course will need to be reshaped and tuned per and algorithm under assessment. The second feedback stream will relate the assessment to the definition of what per algorithm will be considered as process correctness under varying situations.

Following the DoW it is conjectured that the domain-specific know-how can be captured with enough completeness by the three types of information elements outlined below.

1. Context Awareness Patterns (CAP) are data structures reflecting per algorithm its sensitivity to temporal, spatial, operational contexts;
2. Clean State (CS) are algorithm internal data structures based on the trust predicates that are being defined in the T4.4;
3. Behaviour Models (BM) are algorithm's driving models that might exist in the two following formats:
    a. Abstract Behaviour Models that can be used in the algorithmical testing process and can be further offered for the certification process;
    b. Parameterised BM's that are actually guiding model-driven behaviour exhibited by an algorithm.

The parameterisation of BM might happen in a variety of ways including parameter optimisation, observation and action (learning) as well as through cooperation with other algorithms of the same or different type.

The above defined three types of domain-specific know-how are the key to define the three stages of embedding explained below.

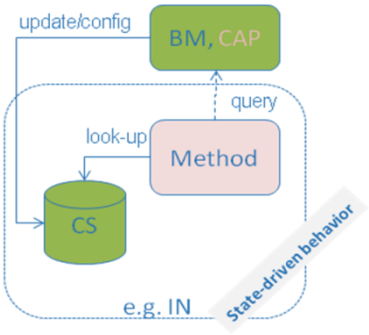


Figure 28. State embedding.

The QoEm=1 can be explained with the embedding of one type of the domain know-how. The figure demonstrates so called state embedding with the example of CS in the framework of Intelligent Networks (SS7). When a local database of an IN switch has no entry for a certain call in progress, the call is suspended and the Service Control Point is queried (and eventually, further the Service Creation Point); which returns to the switch a routable entry for the call. This process can be seen as an external BM (SCP) that updates and configures the switch internal CS. This way, the behaviour of IN switch can be termed as state-driven behaviour.

In this example a CAP, if any is recognised off-line and implemented manually into the SCP's logic (that is into BM). It seems very natural to have embedding of CS as the first stage of embedding, since it is questionable to have as the first step embedding of BM without CS or CAP without CS and BM.

The second stage of embedding (QoEm=2) can be termed state and model embedding and reflects model-driven behaviour.

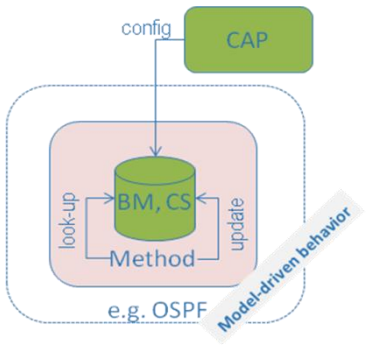


Figure 29. State and model embedding.

This case will be demonstrated by the example of OSPF. OSPF looks up the FIB for making forwarding decisions, synchronizes RIB's per adjacencies to converge on topology changes; the initial BM (topology database) is given from outside.

In this case CAP (e.g. virtual adjacencies) can be configured manually from outside. The model driven behaviour of OSPF is remarkable since e.g. a network of about 100 routers running OSPF can converge to any single link failure in just 0,1ms. The algorithm is organised at two levels the outer loop updates the RIB, while the inner loop that runs in the silicon makes look-ups the FIB and forwards the datagram.

After these two examples that do exist in real networks, it's time to speculate on the third stage of embedding (QoEm=3) since nothing like this is presently available in networks.

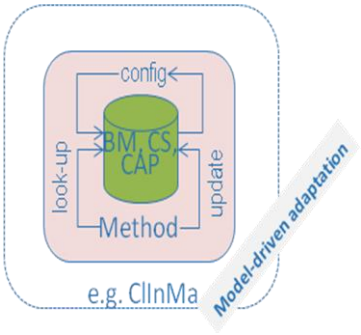


Figure 30. Model driven adaptation.

In this example that can be termed model driven adaptation all three types of domain know-how (CS, BM, CAP) are embedded in the algorithm. Since we describe non-existent mechanisms we shall refer to mechanisms addressed by Close Loop Interaction management (ClInMa) Task Force and ClinMa-enabled Control Loop will be able to exhibit radically new properties such as purpose-driven adaptation due to the fact that BM+CS+CAP has the power of adaptation model. In this case the CAP can take a number of forms. These can be problem-specific adjacencies that proved to be helpful in the past (HIP); this can be problem-specific HIP cooperation strategies; these can be problem-specific HIP optimisations, etc.

Obviously, the components and systems with the highest embedding QoEm=3 can no longer be managed by traditional means since all the three (CS, BM, CAP) are hidden from the stovepipe management; this was the purpose of embedding – empowerment of network components and systems. This does not mean however that a [human] manager can no longer be in a dialogue with such self-managed entities; the dialogue however has now a novel form. Human manager through extended policy interface (part of UMF) can provide governance inputs (goal policies) to self-managed entities and retrieve the entity's state and behaviour information at the desired level of aggregation. The dialogue is also bi-directional in a sense that when a self-managed entity finds itself unable to continue process-correct operation it issues through the same extended policy interface the Call for Governance.

For the three stages above the QoEm metric values are integers, however finer grained cases will populate the metric space with finer-grained numbers in-between these integers. Things in reality are very quickly getting much more complex. As the below figure demonstrates the BM+CS+CAP serves as the basis of a generic adaptation model, which has the number of properties such as self-description and cognition of different types.

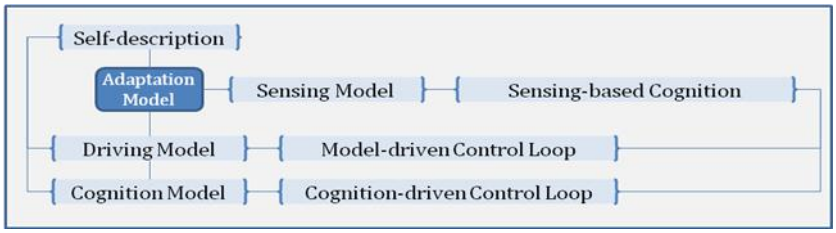


Figure 31. Generic adaptation model.

6.2 Information, knowledge management and sharing

The aim of the task is to design a unified information, context, and knowledge management system (ICKMS) for current and future Networks. This is a critical part of the UMF since it plays the role of information collection and distribution across all UMF functional areas. Moreover, such information management system is expected to enable UMF's features on network management interoperability.

In the following subsections we will first define *information*, *context*, and *knowledge* for networks and services, and then introduce the concepts of *Information Model* and *Information, Context, and Knowledge Management System* with their related state of the art. Subsequently we will provide an outline for the information modelling approach as an enabler for (i) managed system abstraction, (ii) semantic interoperability among different network management systems which could be "federated" through the UMF. Finally, we will address the motivation and positioning of the ICKMS and its design.

6.2.1 Introduction

The global knowledge necessary to manage a system resides in its intrinsic capacities (i.e., the functional capabilities of its constituents) and in their actualised status and/or data characteristics.

In order to clearly set the limits between the various concepts within an information and knowledge management system a differentiation is required to be provided among data-information-knowledge; this can be seen as part of the DIKW Hierarchy [125]. Within the DIKW Hierarchy information is defined in terms of data, knowledge in terms of information and wisdom in terms of knowledge. DIKW Hierarchy has been explored in the context of enterprise and organisational management ([126], [127]).

In [128] the following distinction is proposed:

- Data is observable and possibly measurable raw values that signal something of interest ; data have no meaning,
- Information is data, which a meaning has been attached to (various types of information can be distinguished, context information, policy information etc. as described in later sections),
- Knowledge is information, which has been attached a purpose to serve.

From the above, the importance of semantics is highlighted for network management framework. Information modelling is the first stage of defining semantics in a coherent way within a managed system.

In today's heterogeneous networking environment, information modelling to address integrated/federated management aspects can be really complex. This does not only concern the "core" modelling work (i.e. identification of concepts and hierarchies) but it also involves the semantic, which should be interpretable in a common or converged way by autonomic managers featuring different network management principles.

In a general manner, information concerning a set of Managed Objects, irrespective of their functionality, should be ideally exploitable by any type of monitoring, any type of triggering of behaviour, whatever the "autonomic manager" orchestrating these actions. A common usage of an *information model* simplifies the sharing of information among and across domains as well as the orchestration of self-features behaviours. Multiple platform- and language-specific data models should be federated in a single information model in order to establish a common understanding of shared data.

According to the authors in [95] [122], an information model is "an abstraction and representation of the entities in a managed environment. It includes definition of their properties, operations and relationships. It is independent of any specific type of repository, software usage, platform, or access protocol."

As stated in [94] an Information Model can be defined using a standard language; in general, the Unified Model Language (UML) class diagrams are commonly used since UML is widely accepted and standardised by the Object Management Group (OMG [83]). Several organisations have been using UML for Information Models specification including the DMTF, the ITU-T SG 4, 3GPP SA5, and the IEEE P1900.4 WG within the IEEE SCC41, the TeleManagement Forum, and the Autonomic Communication Forum. A summary of the state of the art of Information Models is presented in Annexes D and F.

Apart from a common agreement on an information ***model***, in order for the required information to flow across domains and functional areas, the UMF needs an information, context, and knowledge ***management system*** (ICKMS) to provide different means for data, context, information, and knowledge collection, aggregation, communication, storage, and query while the managed system is in operation. Several academic and commercial distributed monitoring systems have been designed to monitor the status of large networks. Some of these systems follow a centralised approach whereby all the monitoring data is collected and d at a central host. For instance, Ganglia [106] uses a hierarchical system where the attributes are replicated within clusters using multicast and then cluster aggregates are further aggregated along a single tree. Astrolabe [107] collects large-scale systems state, permitting rapid updates and providing on-the-fly attribute aggregation. This capability permits an application to locate a resource, and offers a scalable way to track system state as it evolves over time. Astrolabe is implemented with the main goal of being scalable using a peer-to-peer protocol, and uses a restricted form of mobile code based on the SQL query language for aggregation. It also natively addresses several security considerations using a built-in PKI. Other prominent information management solutions for very large distributed systems are Moara [108], and IBM Tivoli [109].

6.2.2 Set the Challenges for Information Modelling

As stated previously, an information model provides a conceptualisation of a managed system through defining key concepts, parameters and operations valid throughout that system. In integrated management approaches information modelling can be a complex procedure as it might be required not only to identify hierarchies of concepts and their relations but also to address semantic gaps, overlapping, mismatches or conflicts. Such issues result from the different modelling or management approaches which have been adopted by the systems that need to be federated at a conceptual and/or functional level.

In terms of UMF, system conceptualisation must be coupled to interoperability among management systems that need to interact under a common management framework. This means that a step-by-step approach needs to be defined in order to converge to a "single" Information Model for the UMF needs.

The UMF can be viewed as a container of generic management interface classes for all management domains (legacy, autonomic); such management interface classes are to enable integrated management realised by legacy and autonomic network nodes and devices deploying related autonomic and legacy capabilities. In order to enable the federation and orchestration of management actions in autonomic and non-autonomic domains, it should enable:

- Technology-agnostic end-to-end service management,
- Unification of existing management approaches and systems,
- Network governance, driven by high level policies as imposed by involved players and lower-level - more specific policies/rules as derived by related network management procedures,
- Management of future networks,
- Embedding of autonomic paradigm in any type of network.

The UMF "view" on information and knowledge modelling should reflect related concepts and models for enabling interoperability (between autonomic and legacy systems) within the UMF scope and for addressing related objectives as abstracted by related mechanisms.

The set of domain of interest for the UMF includes (Figure 32):

- Service Domain: services offered to users by value producers (e.g. telecom operators, service providers, etc.);
- Access Domain: part of telecommunication system, lying between the User Equipment and the Core Network;
- Core Domain: the central part of a telecommunication network that provides various services to customers who are connected by the access network;
- Backhaul: the "portion" of the network comprising the intermediate links between the core network, and the small sub-networks at the "edge" of the entire hierarchical network implying a high capacity line;
- Virtual Networks Domain: Network Virtualization is the process of combining hardware and software network resources and network functionality into a single, software-based administrative entity.

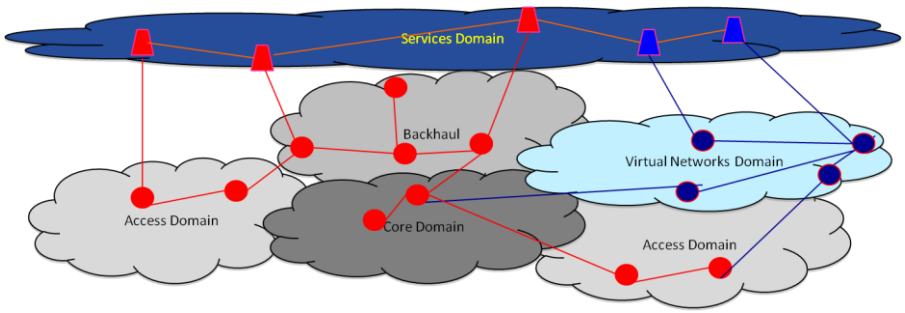


Figure 32. UMF domains.

Federation/Unification spans among different Autonomic management systems and different Legacy Management systems. At a functional level, federation and unification enable application of governance mechanisms to all the corresponding management domains as depicted in (**Figure 33**).

Information Modelling aims at enabling semantic "continua" for UniverSelf autonomic functionalities (Figure 34). In this context information/Knowledge and Policy continua are targeting:

- Unifying different consistencies – levels of management,
- Employing appropriate terminology and syntax,
- Preserving semantics within consistencies/levels of management

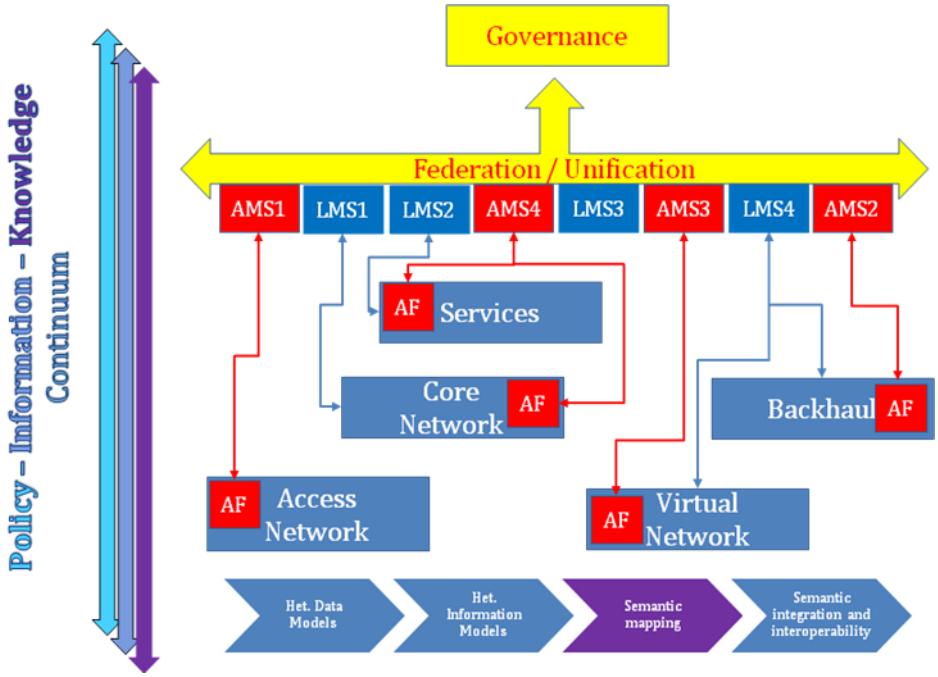


Figure 33. Management Domains and related continua.

6.2.3 The Procedure of Information Modelling

Two directions can be identified towards defining a single information model for the UMF and providing the required abstractions inside UniverSelf. The first direction (internal) is based on the UniverSelf technical scope and the networking objects which compose the UniverSelf technical ecosystem (as outlined through Use Cases and defined functionalities), the management domains under consideration (e.g. Service, Access Network, Core Network, etc.), the information flows, the required measurements, metrics and policies governing the elements' behaviour and interactions. A corresponding information model could be composed by corresponding entities, through breaking them down (conceptually) and assigning them parameters, capabilities, requirements and measurements.

However, in order to avoid starting from scratch as well as to overcome different and possibly conflicting approaches in defining the concepts and the hierarchies, existing initiatives have been considered which could serve as a basis not only for the concepts and the hierarchies' definition but also for assigning semantics (see Annexes E and F).

In this same direction, the following issues are under study and elaboration:

- Which are the domains forming each existing Information model scope,
- What kind of abstraction models and hierarchies have been identified,
- How such models can form a unified one covering the complete UniverSelf scope,
- What are the gaps to be filled

Once the information space has been outlined (at various levels of details) the information model requirement will further be considered under the scope of existing (standardised) approaches (CIM, SID, DEN-ng, etc).

This is depicted in the following figure:

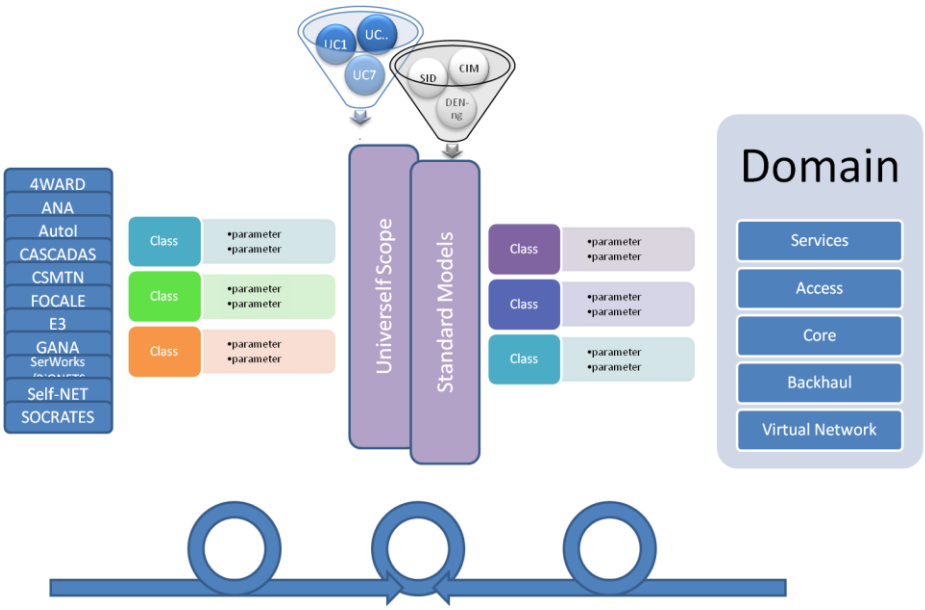


Figure 34. Unified Information Modelling - Unifying Information models.

In this context, the UMF managed system will have to be broken down into primitive concepts and relations (based on information flows and management tasks); in parallel such procedure will take into account initiatives and outcomes in previous projects and standardisation efforts in terms of semantics, hierarchies, application domains, business and technical scope, and usable results. Such elaboration will be based on the SOTA analysis and summarisation on existing information models and approaches as presented in Annexes E and F.

6.2.4 Context and Knowledge for Networks & Services

Context Definitions

Context is roughly any information used to characterize the situation of a managed entity/system except its intrinsic status. However this definition needs precisions and that we try to capture the meaning of "context" with respect to communication networks and services and to classify the relevant context information, before focusing on UMF contextual information management.

Contextual information is mainly environmental information to which temporal aspects are attached.

Moreover this information can be measurable or not. Temporal aspects concern persistence of the context, temporal situation, past present or future.

Another attribute of this information is the fact to be linked to the system or to services.

Environmental context includes the following points:

- Human User context characteristics = information representing the user's surroundings (user location, identity, user mobility, available devices, etc.) as well as his/her physical being (e.g. identity, preferences, history etc.)
- Device context characteristics include: IP address per machine, IP masks per sub-network or address per domain – parameters that vary according to our preferred level of abstraction. The complexity escalates when we look at the proliferation of mobile devices, e.g. mobile phones and PDAs that now have access to the Internet. In terms of location as context data, the mobile telephony research community has long developed a reliable system for hand-over between base stations and international roaming.
- Network context characteristics: network identity; network resources: bandwidth, available media ports; other parameters: available Quality of Service (QoS), security level, access-types, coverage. The network Context Information Base (CIB) – is a logical construct representing a distributed repository for network context data and operands, and it can be used by all networking functions and services.

The CIB's functionality includes: (i) methods & functions for keeping track of context sources, including context registration and naming, context data directory, indexing, context data monitoring and management, etc.; (ii) collection and distribution of context data to clients through context associations, including context data update and context processing such as aggregation, inference etc. to support higher-level context services.

- Flow context characteristics: flows are the physical and electronic embodiment of the interaction between the user and networks. Context information that characterizes these flows may be used to optimize or enhance this interaction including: the state of the links and nodes that transported the flow, such as congestion level, latency/jitter/loss/error rate, media characteristics, reliability, security; the capabilities of the end-devices; the activities, intentions, preferences or identities of the users; or the nature and state of the end-applications that produce or consume the flow. Because of the ephemeral nature of flows, flow context has to be handled differently than user or network context.
- Apart contextual information itself, the way it is accessed is also important. Interaction between Context Sources and Context Sinks can be characterized as follows:
- Context Push: The context sources periodically push updated context information to the context sinks. The context sinks maintain the information in a context store, from which they service client inquiries;
- Context Pull: The context sinks must explicitly request context information. They can either make these requests on a periodic basis (polling) or when an application demand arises. Each mechanism has advantages and disadvantages. A polling system collects data ahead of need and thus may offer better performance. However, it may consume substantial resources transferring and storing information that is never required, though this may be worthwhile if information freshness is important. In some circumstances, it may be possible to use pre-fetch and/or caching mechanisms to alleviate these problems, but this may increase resource utilization.

The categorisation above describes how context information is gathered and how it evolves over time. However this discussion gives rise to some questions about how context information should be managed, stored, aggregated, disseminated and used, considering its changing nature. For example we might consider it helpful to associate a timestamp, a period of validity and a Quality of Context with each piece of context information.

UMF Context Management Infrastructure design

In order to use context information in UMF the following context management framework is identified.

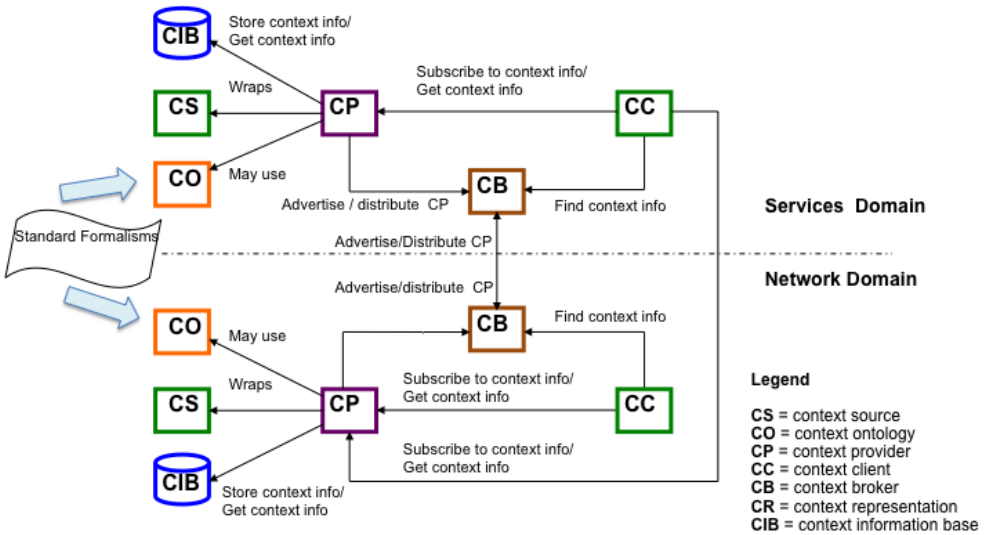


Figure 35. UMF - Context Management Infrastructure (CMI) – Overview.

Context Sources (CS) delivers raw context information, such as GPS coordinates or calendar data available on a mobile device. **Context Providers (CP)** are software entities that produce context information from internal or external information. The internal working of a CP might range from simply wrapping a body sensor for a single user to a full-fledged inference reasoner that combines information gathered from other CPs. Many different

Context Providers can co-exist. A software entity that uses CPs as input is called a **Context Client (CC).** A CC can itself provide input to (multiple) other CCs, and thus be a CP for them. What binds the CPs together is that they all implement a minimal set of common interfaces (i.e. described in XML-based definitions), which make use of a **Context Ontology** describing the logical relations (e.g. OWL-DL) between the different context concepts. We call this minimal set of common interfaces the **Context Representation Framework (CRF).** The CCs discover the correct CPs using a **Context Broker (CB),** which takes care of registration and lookup of CPs and provides a single point of entry for users of context. The Context Broker (CB) is a powerful query and repository service component within the **Context Management Infrastructure (CMI).** The Context Broker component aims to address two functional concerns for proper operation and interaction between Context Provider (CP) and Context Consumer (CC) software entities within the CMF. Firstly, the CB allows Context Providers to publish context information by registering, updating and deregistering Context Provider advertisements that uniquely describe the functionalities of the Provider. Secondly, it accepts searches as queries from CCs that can be matched against Context Provider advertisements.

Knowledge Definitions and UMF Knowledge Infrastructure

A commonly used definition of knowledge as provided in the context of Artificial Intelligence is included in [123] where knowledge is defined as "justified true belief". In this sense, a knowledge-based system integrates collections of information structures – representation of beliefs for which formal interpretations need to be defined granting them as true beliefs; such status can be further checked for providing them a formal justification [124]. Moreover, the process of Knowledge sharing is the defined as the process of conveying knowledge embedded into one Knowledge-based System to another [124].

In UniverSelf we will adopt a Knowledge Management Infrastructure such as the one presented in Figure 36. The clouds in the figure represent ad-hoc knowledge realms/domains that are distributed. The solid lines that connect two domains show the flow of knowledge between difference Knowledge Sources and applications, whereas the dotted line shows the registration of Knowledge sources.

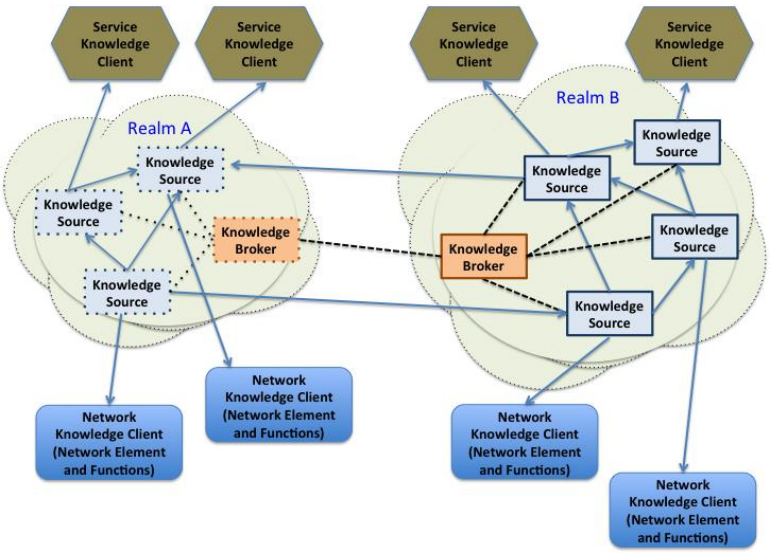


Figure 36. UMF - Knowledge Management Infrastructure (KMI) overview.

The main objective of the KMI is to enable the generic discovery and exchange of processed information, which enables services and networks to act more "intelligently". Within the knowledge layer of the KMI, information originating from context sources and / or providers is added semantics and used to derive entailed and predicated information that we refer to as knowledge. This knowledge encompasses rules, facts, recommendations, preferences, prediction, or context data at various semantic levels.

A **knowledge source** is any functional element that is able to provide knowledge. The knowledge that is provided by a knowledge source is semantically described using annotations. We can define the following knowledge source components:

Context Wrappers: knowledge sources that 'wrap' context information, coming from outside the KMI

Knowledge Reasoner: knowledge sources that enrich, aggregate, or infer entailed knowledge from other knowledge contained in a knowledge base

Knowledge Storage (Knowledge Base): knowledge sources that store knowledge, for example usage patterns for learning

The knowledge originating from any knowledge source component can be provided to applications, services, network elements, network functions and other knowledge sources via a uniform interface that abstracts a number of internal details of the underlying context source or reasoning mechanism.

Knowledge brokers provide the interfaces to support federated knowledge sharing across domains. The main functions of knowledge sources in the KMI include:

- Obtain knowledge from external sources and/or other sensors.
- Process the obtained knowledge.
- Output the result of processing as knowledge for future processing.

Knowledge state of the art

DARPA started a project, called Knowledge Plane [100], which aimed to add intelligence and self-learning to the network management. It aims to eliminate the unnecessary multi-level configuration. If one specifies the high-level design goals and constraints, the network should make the low level decisions on its own. The system should reconfigure itself according to the changes in the high-level requirements. A distributed cognitive system, which permeates the network, is proposed that is called: knowledge plane (KP). Each networking element (end-node, router) has a KP. The KPs at a number of nodes interact with each other in order to keep themselves informed about global (network-wide) states and events. This interaction is also used to reconcile contradictory service levels and requirements. The KP must function in the presence of partial, inconsistent and possibly misleading or malicious information. It must operate appropriately even if different stakeholders of the Internet define conflicting higher-level goals. In order to meet these challenges, the authors suggest that cognitive techniques will be needed, because analytical methods generally require precise and complete information. Nowadays, the network is usually divided into two architectural planes: a data plane and a control (or management) plane. The authors [100] believe that a new construct is needed instead of fitting knowledge into an existing plane. The KP would not move data directly so it is not the data plane. However, unlike the control plane it tries to provide a unified view of the network rather than partition the world into managed segments. The KP integrates behavioural models and reasoning processes into a distributed networked environment. It supports the creation, storage, propagation and discovery of information: observations (current conditions), assertions (high-level goals, constraints) and explanations (conclusions). Based on this information, the KP manages the actuators that change the behaviour of the network components.

A light version of the Knowledge plane was identified, designed and used in [101]. Components of the Knowledge Plane realization were developed in 4WARD [103]. Distributed Knowledge Plane (KP) defined in the ANA architecture [104] was used to interconnect different network elements. In CASCADAS [102] data and information gathered from Autonomic Communication Elements (ACE) are eventually transformed into knowledge, which is playing a central role in the behaviour of autonomic control loops of ACEs. In Self-Net [105] the knowledge plane is formed by specification of an Information Model.

6.2.5 Context and knowledge interaction and interoperability mechanisms

Inside a domain UMF nodes are organized in a group and elect a group controller. The group leader or **U_DC** (UMF intra-domain controller) can be a natural candidate for information, context, and knowledge aggregation, storage, and reasoning within a domain.

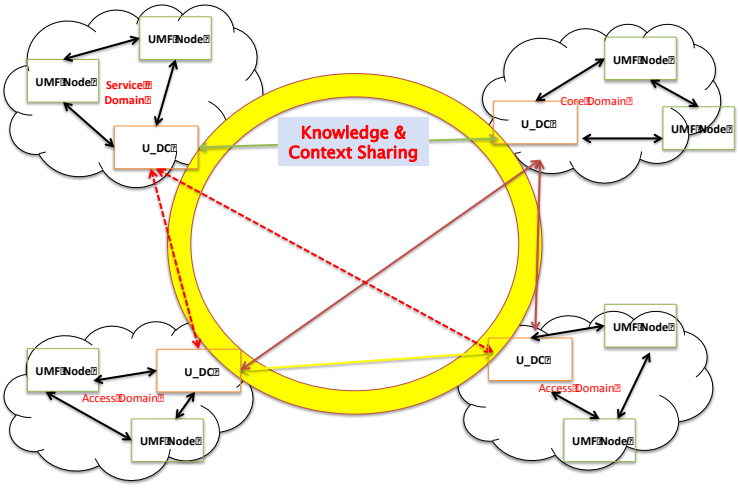


Figure 37. Inter-domain information sharing.

In a multiple-domain scenario, information can be shared among intra-domain controllers and be used for inter-domain negotiation / collaboration / orchestration through knowledge brokers.

6.2.6 Motivation, positioning and description of the ICKM in UniverSelf

Knowledge is the most important part of an autonomic-system, the way knowledge is represented, the way it is updated locally or remotely, the way information are translated into knowledge, the way reasoning and adaptation is achieved on this knowledge, the way to reason on limited part of the global knowledge are all challenges. For these reasons, the ICKM functionality in UniverSelf will have to support and interface with all other functionalities and information, context, and knowledge management (ICKM) should include:

- Mechanisms for representing and manipulating information, context and knowledge in Networks & Services
- Mechanisms for managing (i.e. aggregation, filtering, distribution, storage, optimisation, sources/sinks, usage, quality of information, information agreements) the information flows between UMF components in domain and inter-domain
- Interfaces for the collection, dissemination and use of information from/to the network and service entities
- Interaction with and between the self-management functions (i.e. self-configuration, self-adaptation, self-optimisation, self-awareness, self-healing, self-modelling, self-protection, and other self-x functionalities).

Concerning design goals, the ICKM should:

- Bring together widely distributed data/context/knowledge/information collection, storage and adaptive processing;
- Provide increased analysis and inference capabilities through models, ontologies, and relevant tools;
- Provide a unified Context Information Base and a Knowledge Information Base;
- Support Context & Knowledge operations with naming conventions/frameworks, context/knowledge acquisition, aggregation, connectivity of info/context/knowledge, distributed storage, optimised push/pull distribution, quality of context/knowledge, conflict resolution, federation
- Support a Monitoring & triggering framework based on changes to information/context/knowledge
- Respond promptly to requests made by management functions

6.3 Network governance

Autonomic infrastructure implies a quantum step on network operations automation and intelligence that requires human technicians to go further from the command and control paradigms. Researching and developing new methods for operators to efficiently manage this intelligent infrastructure is the goal of Task 2.3 Network Governance.

This chapter is an introduction to the work being carried out in Task 2.3. It starts with a definition of the Network Governance as part of the UMF, followed by the most important challenges this task must face in

order to successfully accomplish its mission. Section 6.3.3 summarizes the method for gathering requirements for governance from the interviews with human operators of the UniverSelf industrial partners. These requirements, provided directly by the people that are currently managing the network in different telco companies, must be the lighthouse that guides the work in this task. Therefore, the output of the Network Governance task at the end of the project should fulfil these specifications. Section 6.3.4 presents the approach that will be followed in order to achieve the above mentioned goals. Annex H presents a summary of the current state of the art of the governance topic and of the different models and mechanisms that could be of utility when building a governance framework

6.3.1 Definition of Network Governance

Although one of the goals of Autonomic Infrastructure is that of self-management, a framework aiming to manage an autonomic network must include tools to facilitate the control and supervision of the network. It is Operators who need to lead the business transformation and it is a must to ensure human to network communication, if they are to control the infrastructure, focusing on the business level rather than technical aspects of the network—which ought to self-manage thanks to autonomics. Policies seem to take new relevance on this scenario. The perception brought to the operator by this paradigm shift is that of keeping focused on network governance while network management goes autonomic.

Governance is a high level mechanism which involves all functionalities necessary to address the gap between high-level specification of human operators' objectives and existing resource management infrastructures towards the achievement of global goals. Governance also encompasses Human-to-Network (H2N) communication and the introduction of policies and business goals to the network. It should be underlined that orchestration features are required so as to coordinate various network management entities towards achieving global goals (Control of control loops). In short, Network Governance provides the necessary support for an enhanced business oriented policy framework deployment.

6.3.2 Challenges

An increasing number of heterogeneous devices used from different places to access a myriad of very different services and/or applications require a new reliable, dynamic, and secured communication infrastructure with highly distributed capabilities [129]. The autonomic network envisions meeting these features. The complexity of managing such infrastructure exceeds the capabilities of current Operational Support Systems (OSS) and is one of the main challenges that the telecommunications industry is currently facing. Operators need to change their vision on current management paradigms; otherwise they will collapse under the operational weight of managing complexity [129].

The new network infrastructure is highly adaptive and autonomous, and the resources that compose it operate with dynamic relationships. Some functions that were traditionally performed by management systems are no longer held by them, but autonomously carried out by the network itself.
Operators will be mostly settled about decision-oriented operational tasks for the different network elements. What these decision-oriented tasks are and how they impact the decision elements are main issues. After introducing autonomicity, there is a re-assignment of tasks carried out by human network managers, which will focus on the network exploitation enforcement and planning for the future, rather than continuously monitoring the behaviour of particular components.

The building of a network governance framework also faces technological challenges in five main topics: business language, translation, reasoning, policies and configuration enforcement:

Network governance is meant to provide a mechanism for the operator to adjust the features of the demanded service/infrastructure using a high level language. In order to achieve this objective a business language may be required that will help the operator to express what is needed from the network. Such a business language may be modelled by the use of ontology to add semantics and enable machine reasoning on the goals. These indications are afterwards translated to a set of policies that will clearly define the valid operating region for the autonomic functionality.

These high level directives must be translated into low level policy rules that can be enforceable to control behaviour of self managed resources whatever their type, either a single device or a set of devices that can group their self-features (set of devices managed by one single autonomic manager).

Reasoning is also an important challenge in the scope of network governance, as it can be exploited for the mediation and negotiation between separate federated domains. In other words, to allow interoperability between semantically equivalent, but differently instantiated models, it is required to cover multiple standards instead of relying on a single information model. This leads to the use of ontologies for allowing semantic fusion and reasoning with knowledge extracted from data/information.

Moreover, policies are inherent to network governance. Policies specify rules that should govern the behaviour of the managed elements. In particular, network governance is almost always interwoven with policies lying at the highest level of the so called policy continuum. In network governance policies are required for the selection of the optimal configuration of a service and for the translation from business level entries and high-level policies to low level policies.

Furthermore, configuration enforcement mechanisms are necessary in order to apply the configuration decision. First, it is required to identify concerned equipments and request each of them to perform the appropriate configuration actions. Then, each of the targeted equipments has to translate and enforce the decision. The term configuration implies self-configuration and also includes reconfiguration actions (re-optimizations). Reconfiguration actions can be triggered in order to adjust the configuration parameters following network, service and customer conditions.

In short, the real challenge of this task is to design a Network Governance framework based on four technological pillars (high-level language, reasoning, policies and distribution) that results in a system that is able to:

- Work with proper business rules and policies, while connecting high level goals and network resources in order to provide the administrator with an appropriate governance interface.
- Guide infrastructure behaviours while offering a service view
- Offer mechanisms that assist the operator to express goals, objectives, constraints and rules to ensure the desired operation of his autonomic network.
- Provide a friendly human interface, aiming to be easy of use that is not to be used only by highly specialized technicians.
- Work in a reliable way, and be able to demonstrate its reliability
- Help to convince the operators of the bondage of adopting autonomic approaches

  The results of the research activities carried out in this task will be of special importance for building trust in autonomic networks.

6.3.3 Operator's requirements

When designing tools for human operators, it is of outmost importance to produce solutions that are usable by human users and meaningful in the context of their work. This aim can be supported by learning what the characteristics of the operator work are, from the perspective of these professionals, and what the specific demands are that the work sets on human performance. The conceptions regarding autonomic tools can also be elaborated.

Thirty-four human network operators have been interviewed, 33 operators from Spain (TID) in face-to-face interviews and one French operator (ALU) over phone. Since the duration of the interviews varied from 10 minutes to 1.5 hours, the level of detail was quite low in some interviews. Furthermore, operators' expressed opinions of autonomic functionalities may be biased because the implementation of self-x functionalities may diminish the need for human network operators. This all must be taken into account when analysing the interviews.

Human network operators were asked about the following items:

- interviewee's work on a general level;
- general characteristics of the network;
- interviewee's work with the network;
- work experience and
- Interviewee's conceptions of self-x functionalities.

A set of 40 questions were asked (see Annex G). The analysis of the interviews will be performed during summer 2011 and presented in Milestone 25, "Human factors in network management".

6.3.4 Adopted approach

This section describes the main building blocks of a Network Governance framework, aiming to give a human operator a mechanism for controlling the network from a high level business point of view, that is, without the need of having a deep technical knowledge of the network. The three main components of the governance framework are described in the following subsections and depicted in Figure 38, a Human to Network (H2N) interface providing a friendly way of creating and editing policies using a high level business language; a Policy Manager, providing the main functionalities for the management of business and network policies; and an information flow process, a mechanism to efficiently distribute network profiles to the nodes in the infrastructure.

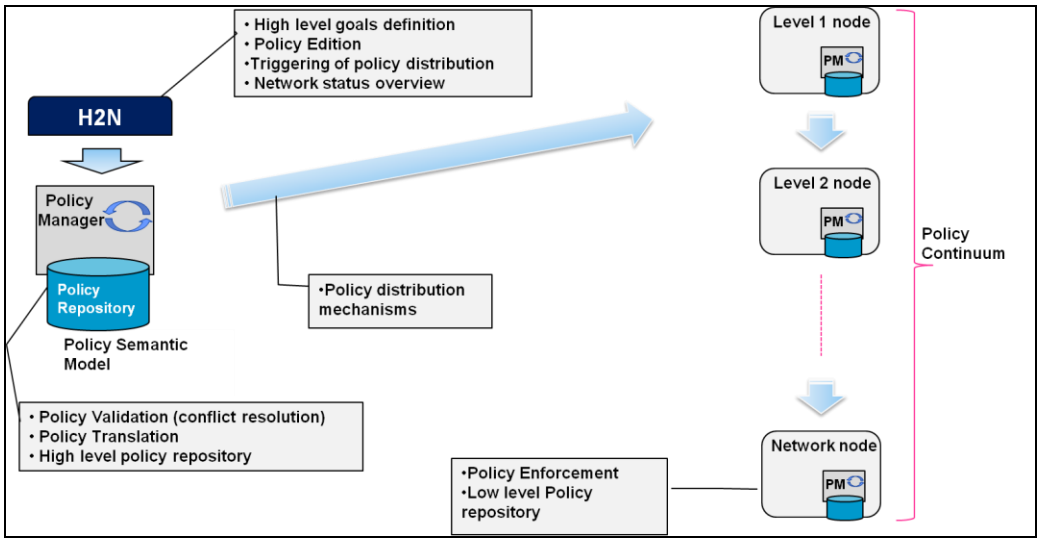


Figure 38. Network Governance Framework.

6.3.4.1 Human-To-Network (H2N) interface

This section describes a Human-to-Network (H2N) interface that falls under the Governance functional group, within the Governance functional block.

The main functionality of this H2N interface is to provide a tool for the human operator to insert high-level business goals, which will be later on translated autonomously into technology-specific terms autonomously so that the human operator does not need to deal with any technical details. A high-level view of the role of the H2N interface is depicted in Figure 39. Business goals may be related to the introduction of a new application, sets of user classes for the application, sets of Quality of Service (QoS) levels for each user class of the application, etc. This introduction can be related to a specific location, time period, volume of users, etc. Furthermore, the H2N interface allows the associations of applications/services with User Classes, QoS levels, network technologies, other applications, QoS levels with QoS parameters. This influences the Policy Derivation and Management functional block, within the Governance functional block.

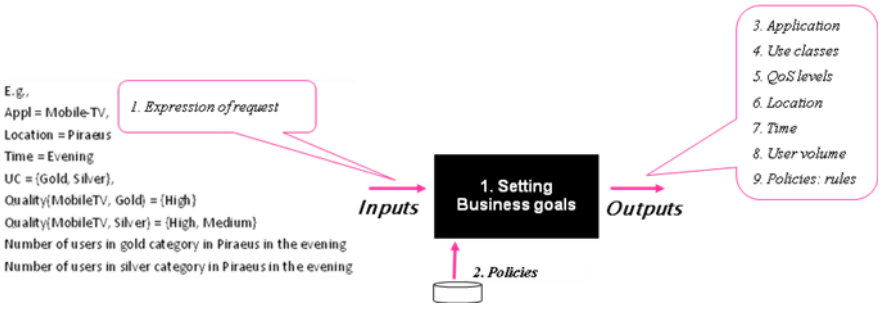


Figure 39. High-level view of H2N interface role.

Moreover, the H2N interface allows for the configuration of the number of users anticipated for an application, the corresponding user class and Quality Level in a certain location and at specific time zone. These high-level objectives/policies need to be further propagated to the network going through an arbitrary set of levels (related to different aspects of the management of a communications network) and be transformed into lower level policies so that they reach the element(s) in which to be enforced in terms of low level, technology-specific commands. Consequently, the already set business goals are forwarded to the Policy Derivation and Management functional block in order be translated from service requirements into network configuration (technology-specific terms) and leave the system to autonomously work out the situation and meet the objectives. The H2N interface also allows feedback, e.g. the result of diagnosis or a visualization of the monitoring to the system administrator/operator.

6.3.4.1.1 Examples of high-level goals

In the scope of the work done so far on the H2N interface, two types of high-level policies have been identified: Business level entries and Associations. A potential approach for these is provided in sub-sections 6.3.4.1.1.1 and 6.3.4.1.1.2 respectively. However it should be noted that high-level as well as low level policies (enabling self-configuration of user devices and network elements) need to be further elaborated on. Their specification will be realised in the context of the overall information model for the UMF.

6.3.4.1.1.1 Business level entries

This sub-section presents business level entries as an example of high-level goals (policies). Business level entries are information provided at the business level related to the number of users anticipated for an application, user class, in a certain location and time zone. In more detail, as can be observed in Figure 40, business level entries comprise information on the Number of users, the Traffic percentage, i.e. the number of concurrently active users anticipated, the Location (e.g. Piraeus, Athens-centre, …), the Time Zone (e.g. 08:00-11:00, 21:00-22:00, …), the Application (e.g. IPTV, …), User Class (e.g. Gold, Silver, Bronze, …), Quality Level (e.g. High, Medium, Low), Quality level parameters (e.g. Bit rate, delay, jitter, packet losses …) and the Mobility pattern (e.g. High, low, train, car, …).

6.3.4.1.1.2 Associations

This sub-section presents associations entries as an additional example of high-level goals (policies). Associations are high-level policies that specify rules related to the relationship of applications with user classes and quality levels, the relation of a certain application with other applications and the relations between user classes. As can be observed in Figure 40 (b), an association comprises information on a set of applications. Each application may be associated with one or more User Classes. Each User Class may be associated with one or more QoS levels. Each QoS level is associated with one or more QoS parameters.

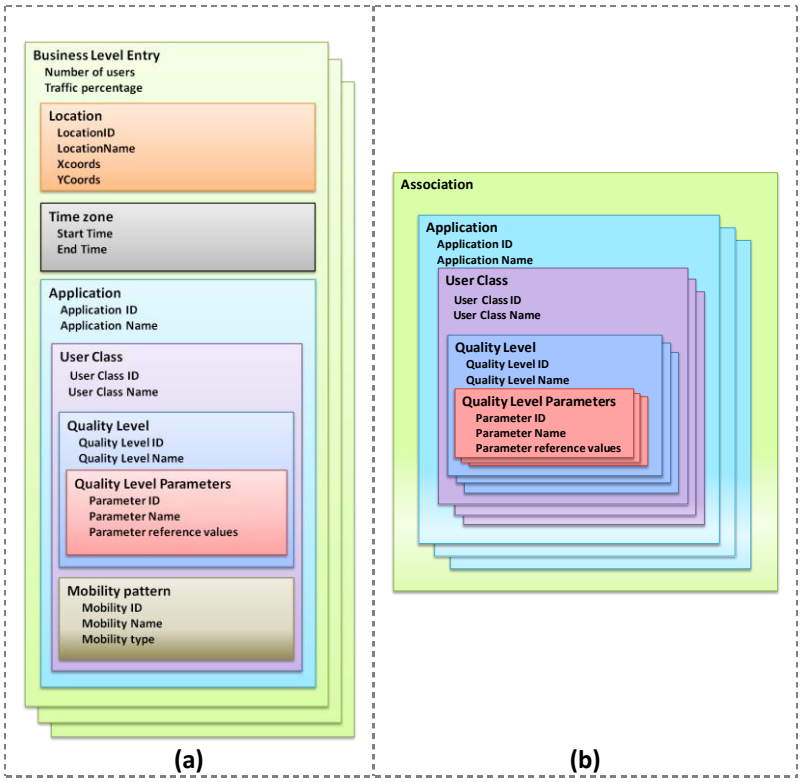


Figure 40. Examples of high-level goals: (a) Business level entries; (b) Associations.

6.3.4.2 Policy Manager

This component falls under the Governance functional group, within the Policy Derivation and Management functional block. It is in charge of (i) providing storage for the policies, (ii) providing mechanisms for the management of the Policy Repository (insertion, modification, retrieval and removal of policies) (iii) translating business language to more specific policy language statements, (iv) checking whether the different indications given by the operator have conflicts and (v) in case conflicts appear, resolve them according to the well defined conflict resolution mechanisms.

In short, the Policy Manager encapsulates all the functionality closely related to policies. It contains a Policy Repository for the effective storage of policies. Policy Manager provides interfaces to the H2N tool, which offers functionalities for policy edition, and therefore needs to access the Policy Repository for inserting, modifying, accessing and deleting policies.

It is worth mentioning that the decoupling of the Policy Manager from the H2N tool allows the reuse of the Policy Manager through the different levels of the Policy Continuum, as shown in Figure 41. Policies must be go through different layers before arriving to the network nodes, that is, mechanisms for distribution, translation, validation of policies must be available at the different levels. Therefore, the encapsulation of all the policies management functionalities in one independent entity allows the deployment of the Manager at different layers.

Section 6.3.4.1.1 already provided examples of high-level business goals. Concerning network policies, there already exist different policy languages to cover different network and vendor technologies (see Annex H for a summary on the state of the art), based on different information models such as CIM. Therefore, the specification of the Policy Manager should allow interoperability among the different languages used by the different devices of different providers. Semantic technologies are a promising approach for achieving this goal. The use of ontologies will enable reasoning capabilities to map the service requirements to the network level without direct intervention of the operator.

6.3.4.3 Information flow

As already introduced, governance allows the introduction of the business level goals/policies in high level terms through the human-to-network (H2N) interface. Typically, business level policies define high level

expression of business objectives. In the sequel and according to the policy continuum concept these policies are propagated to the network going through an arbitrary set of levels (related to different aspects of the management of a communications network) where they are being transformed into lower level policies, until they finally reach the element(s) in which to be enforced in terms of low level, technology-specific commands. Figure 41 depicts the set of levels and the corresponding policies of the proposed architecture in accordance to the policy continuum paradigm. It is obvious that the direction of the enforcement flow is top-down, while the evaluation/monitoring flow has inverse direction. Policy derivation and management translates high level goals/objectives provided through Governance into low level policies and often into low level self-configuration enforcement policies.

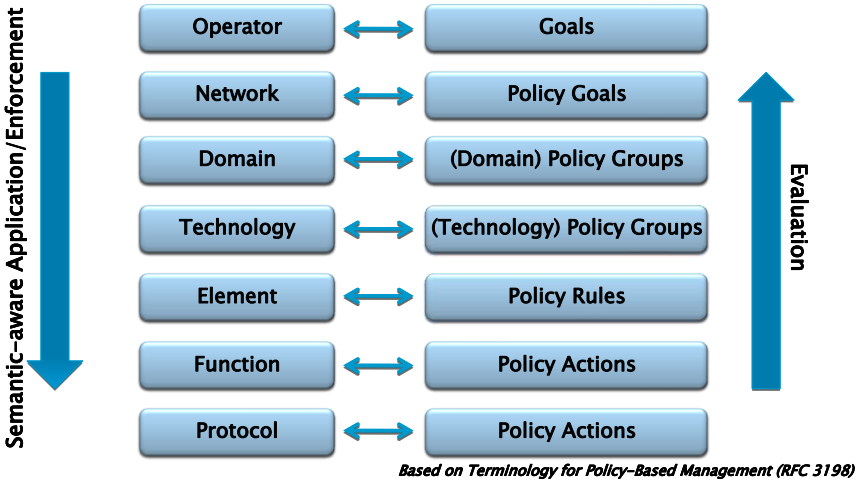


Figure 41. Network Decomposition and corresponding Policy Framework.

7 Conclusion

Deliverable D2.1 “UMF Specifications – Release 1” provides a first concise description of the UMF design. Part of this document is devoted to setting the scene around the current issues, problems and lacks in contemporary network and service management that point out the need for UMF. Specifically, the document starts with the motivations that leverage the need for designing and delivering a UMF. Then, the prior art analysis covers autonomic networking research findings, but also standard groups and management frameworks. The results of the prior art analysis are provided as input to the design of UMF. A clear UMF position within the landscape of existing management systems and architectures, in particular as it is perceived from the industrial point of view is provided

In the core section of this deliverable, “UMF design”, UMF is presented as the outcome of the dual approach namely, “bottom-up” and “top-down”, covering the complementarities and consistencies among them. The contribution from the prior art analysis to the UMF principles is also acknowledged. In the bottom-up analysis, the requirements elicited from the first burst of use cases (reported in D4.1) have been used to provide a first functional view of the UMF in terms of core, reusable and cohesive “Functional Blocks” and associated interfaces. Accordingly, these functional blocks provide the means to resolve day-to-day problems identified on existing service/network architectures considering both services and networks and spanning both fixed and mobile network domains, as reported by operators. On the other hand, the top-down approach consists in an analysis of the high-level requirements identified in the project and in standard and research activities on Future Networks and Future Internet in order to define additional/complementary features and properties so as to enhance and consolidate the UMF design. These highlight what distinguishes UniverSelf from earlier network and service management technologies and also capitalize on previous autonomic architecture research. Core functional blocks deriving from the bottom-up analysis are consolidated and organised into the so called “Functional Groups”, in a way in a way in which the top-level requirements would also be satisfied. Last but not least, a system view of the UMF is also attempted. This includes the introduction of a number of specialized logical nodes and of a possible hierarchical structure, a discussion on orchestration issues, as well as a mapping of the identified functional blocks into these nodes and the elaboration on their functionalities and interfaces among them.

The role of the identified UMF enablers, as fundamental and common elements of the UMF functionality, which include Intelligence embodiment, Network governance, as well as Information and knowledge management as fundamental elements of the UMF provided functionality is also discussed. Technical challenges, issues, variant options associated with these enablers are highlighted and initial thoughts on their handling are also proposed.

Capitalizing on these first specifications, next steps include: a) fine-tuning and stabilizing the functional view of UMF so as to accommodate the algorithms and methods studied and developed within WP3 for solving use-case specific problems, b) consolidation of the system design for UMF, c) the accommodation of further, future use cases as a means to prove a great level of reusability of functional blocks and/or interfaces and most importantly d) putting emphasis on the work pertinent to enablers namely, information and knowledge management capabilities (definition/development/management of ontologies, selection of information model, knowledge structures etc.), network governance mechanisms (H2N interface, policy based framework etc.) and intelligent embodiment mechanisms.

8 References

[1] (Online) IBM Corporation, Autonomic computing – a manifesto, www.research.ibm.com/autonomic, 2001

[2] The FP7 4WARD Project, home page of the project, http://www.4ward-project.eu/

[3] Pentikousis, K.; Meirosu, C.; Miron, A.; Brunner, M.; "Self-Management for a Network of Information," Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on , vol., no., pp.1-5, 14-18 June 2009

[4] ANA Project Homepage, http://www.ana-project.org/ [Last accessed: 25 Sep 2010]

[5] Schuetz, S. Zimmermann, K. Nunzi, G. Schmid, S. Brunner, M. , "Autonomic and Decentralized Management of Wireless Access Networks", Vol. 4, no. 2, Page(s): 96 – 106, 2007

[6] Autonomic Internet (AutoI) project http:/ist-autoi.eu/

[7] A. Galis, et. al., "Management Architecture and Systems for Future Internet Networks" in "Towards the Future Internet – A European Research Perspective" ISBN 978-1-60750-007-0, Apr. 2009, IOS Press.

[8] CASCADAS Project Web Site, http://acetoolkit.sourceforge.net/cascadas/

[9] A. Manzalini, F. Zambonelli, "Towards Autonomic and Situation-Aware Communication Services: the CASCADAS Vision". Proceedings of the IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS'06), pp. 383-388, 2006.

[10] E3 Website. https://ict-e3.eu/

[11] K.Nolte, A.Kaloxylos, K. Tsagkaris, et al., "The E3 architecture: Enabling future cellular networks with cognitive and self-x capabilities", To appear in the International Journal of Network Management, 2010

[12] EC funded- FP7-EFIPSANS Project: http://efipsans.org/

[13] R. Chaparadza, S. Papavassiliou, T. Kastrinogiannis, M. Vigoureux, et al. Creating a viable Evolution Path towards Self-Managing Future Internet via a Standardizable Reference Model for Autonomic Network Engineering. FIA Prague 2009 Conference, published in the FI Book produced by FIA, 2009.

[14] A. Kousaridas, G. Nguengang, J. Boite, V. Conan, V. Gazis, T.Raptis, N. Alonistioti, "An experimental path towards Self-Management for Future Internet Environments", Book Chapter In: "Towards the Future Internet - Emerging Trends from European Research" [ISBN 978-1- 60750-539-6], Edited by Georgios Tselentis, Alex Galis, Anastasius Gavras, Srdjan Krco, Volkmar Lotz, Elena Simperl, Burkhard Stiller pp. 95 - 104, 2010.

[15] BIONETS Project Web Site, http://www.bionets.eu/

[16] D. Miorandi et al, "D1.1.1 Application scenario analysis, network architecture requirements and high-level specification", 2007 (http://www.bionets.eu/docs/BIONETS_D1_1_2.pdf)

[17] EC funded- FP7-Socrates Project: http://www.fp7-socrates.org/

[18] Neil Scully, Kristina Zetterberg, Szymon Stefanski et al., "D5.10: Measurements, architecture and interfaces for self-organising networks", 2010 (http://www.fp7-socrates.org/files/Deliverables/SOCRATES_D5.10%20Measurements,%20architecture%20and%20interfaces%20for%20self-organising%20networks.pdf).

[19] P. Demestichas, "Introducing cognitive systems in the wireless B3G world: motivations and basic engineering challenges", Telematics and Informatics journal, No.27, pp. 256-268, February 2010, doi:10.1016/j.tele.2009.08.002, Journal Papers

[20] P. Demestichas, D. Boscovic, V. Stavroulaki, A. Lee, J. Strassner, "m@ANGEL: autonomic management platform for seamless wireless cognitive connectivity", IEEE Commun. Mag., Vol. 44, No.6, pp. 118-127, June 2006, Journal Papers

[21] P. Demestichas, G. Dimitrakopoulos, J. Strassner, D. Bourse, "Introducing reconfigurability and cognitive networks concepts in the wireless world: research achievements and challenges", IEEE Vehicular Technology Mag., Vol. 1, No. 2, pp. 33-39, June 2006

[22] V. Stavroulaki, N. Koutsouris, K. Tsagkaris, P. Demestichas, "A platform for the integration and management of cognitive systems in future networks", In Proc. IEEE Global Communications Conference (GLOBECOM 2010), Miami, USA, December 2010

[23] Java Agent DEvelopment Platform (JADE), Web site: http://jade.tilab.com, accessed June 2010.

[24] JADEX Projects, http://vsis-www.informatik.uni-hamburg.de/projects/jadex/, accessed June 2010

[25] V. Stavroulaki, Y. Kritikou, P. Demestichas, "Acquiring and learning user information in the context of cognitive device management", In Proc. IEEE International Conference on Communications 2009 (ICC 2009), Dresden Germany, June 2009

[26] Y. Kritikou, V. Stavroulaki, P. Demestichas, "Learning user preferences for the realization of intuitive cognitive devices", submitted for publication to the Wireless Personal Communications

[27] K. Tsagkaris, A. Katidiotis, P. Demestichas, "Performance evaluation of artificial neural networks based learning schemes for cognitive radio systems", Computers & Electrical Engineering, Volume 36, Issue 3, May 2010, Pages 518-535

[28] P. Demestichas, A. Katidiotis, K. Tsagkaris, E. Adamopoulou, K. Demestichas, "Enhancing channel estimation in cognitive radio systems by means of Bayesian networks", Wireless Personal Communications, Vol. 49, No. 1, pp. 87-105, April 2009

[29] K. Tsagkaris, A. Katidiotis, P. Demestichas, "Neural network based learning schemes for cognitive radio systems", Computer Communications, Vol. 31, No. 14, pp. 3394-3404, Sept. 2008

[30] A. Bantouna, K. Tsagkaris, P. Demestichas, "Self-Organizing Maps for improving the channel estimation and predictive modeling phase of cognitive radio systems", In Proc. 20th International Conference on Artificial Neural Networks (ICANN 2010), Thessaloniki, Greece, September 2010

[31] K. Tsagkaris, A. Bantouna, P. Demestichas, "Self-organizing maps for advanced learning in cognitive radio systems", submitted for publication to the IEEE Transactions on Vehicular Technology

[32] K. Tsagkaris, G. Dimitrakopoulos, P. Demestichas, "Policies for the management of services in CDMA-based segments of the B3G world", IEEE Vehicular Technology Mag., Vol. 2, No. 3, pp. 21-28, Sept. 2007

[33] K. Tsagkaris, G. Dimitrakopoulos, P. Demestichas, "Policies for the reconfiguration of cognitive wireless infrastructures to 3G radio access technologies", Wireless Networks journal, Vol. 15, No. 3, pp. 391-405, April 2009

[34] P. Demestichas, A. Katidiotis, V. Stavroulaki, D. Petromanolakis, "Management system for terminals in the wireless B3G world", Wireless Personal Communications journal, Vol. 53, No. 1, pp. 81-109, March 2010

[35] A. Saatsakis, K. Tsagkaris, P. Demestichas, "Exploiting context, profiles and policies in dynamic sub-carrier assignment algorithms for efficient radio resource management in OFDMA networks", accepted for publication to the Annals of Telecommunications

[36] A. Saatsakis, P. Demestichas, "Context matching for realizing cognitive wireless networks segments", accepted for publication to the Wireless Personal Communications journal

[37] K. Tsagkaris, I. Chadjifotis, P. Demestichas, "Ant colony optimization for subcarrier allocation in OFDMA-based wireless systems", In Proc. of ICT Mobile Summit 2009, Santander, Spain, June 2009

[38] D. Karvounas, K. Tsagkaris, P. Demestichas, "Position optimization of moving access points", In Proc. Future Networks and Mobile Summit 2010, Florence, Italy, June 2010

[39] P.Demestichas, V.Stavroulaki, "Issues in introducing resource brokerage functionality in B3G, composite radio, environments", IEEE Wireless Commun. Mag., Vol. 11, No. 10, pp. 32-40, Oct. 2004

[40] A. Galani, K. Tsagkaris, P. Demestichas, "Information flow for optimized management of spectrum and radio resources in cognitive B3G wireless networks", accepted for publication in the Journal of Network and Systems Management

[41] K. Tsagkaris, M. Akezidou, A. Galani, P. Demestichas, "Evaluation of signaling loads in cognitive network management architecture", submitted for publication to the IEEE Transactions on Network and Service Management

[42] K. Tsagkaris, N. Koutsouris, A. Galani, P. Demestichas, "Performance assessment of a spectrum and radio resource management architecture for heterogeneous wireless networks", In Proc. Future Networks and Mobile Summit 2010, Florence, Italy, June 2010

[43] A. Galani, K. Tsagkaris, N. Koutsouris, P. Demestichas, "Design and assessment of functional architecture for optimized spectrum and radio resource management in heterogeneous wireless networks", accepted for publication in the International Journal of Network Management

[44] Strassner, John and Agoulmine, N. and Lehtihet, E. (2006) *FOCALE: A Novel Autonomic Networking Architecture.* In: Latin American Autonomic Computing Symposium (LAACS), 2006, Campo Grande, MS, Brazil.

[45] IBM Autonomic computing glossary, http://www.research.ibm.com/autonomic/glossary.html, accessed June 2011

[46] Embodiment. (2010, February 28). In Wikipedia, The Free Encyclopedia. Retrieved 13:58, June 15, 2011, from http://en.wikipedia.org/w/index.php?title=Embodiment&oldid=346909542

[47] Razvan V. Florian, "Autonomous artificial intelligent agents", Technical Report, Center for Cognitive and Neural Studies (Coneural), 2003

[48] I. Grida Ben Yahia, E.Bertin, N.Crespi, "Ontology-based Management Systems for the Next Generation Services: State-of-the-Art," icns, pp.40, International Conference on Networking and Services (ICNS '07), 2007

[49] T. R. Gruber, "A translation approach to portable ontology specifications" Knowl. Acquis. Vol. 5, No 2, pp. 199-220, June 1993

[50] J. Hendler, "Agents and the Semantic Web", IEEE Intelligent Systems Vol. 16, No 2, pp. 30-37, March 2001.

[51] Y. Kalfoglou, "Exploring Ontologies", In: Handbook of Software Engineering and Knowledge Engineering: vol. 1: Fundamentals, pp. 863-887, World Scientific Publishing, 2001

[52] Ontology Definition Metamodel http://www.omg.org/technology/documents/spec_catalog.htm

[53] K. Baclawski, M. K. Kokar, P. Kogut, L. Hart, J.E. Smith, J. Letkowski, and P. Emery, "Extending the Unified Modeling Language for ontology development", International Journal Software and Systems Modeling (SoSyM) 1(2) , pp. 142-156, 2002

[54] IST-2000-29243 OntoWeb (Ontology-based information exchange for knowledge management and electronic commerce) Deliverable 1.3: A survey on ontology tools, 2002, available at icc.mpei.ru/documents/00000826.pdf

[55] A. Galis, S. Denazis, C. Brou, C. Klein, "Programmable Networks for IP Service Deployment", Artech House Books,

2004
[56] I. Grant, "Programmable Networks: the goal as industry giants hook up", online article on Computer.Weekly.com, available at http://www.computerweekly.com/Articles/2011/03/24/246038/Programmable-networks-the-goal-as-industry-giants-hook.htm, March 2011
[57] A.T. Campbell, M. E. Kounavis, J. Vicente, M.K. Villela, and H. De Meer, "A Survey of Programmable Networks", ACM SIGCOMM Computer Communication Review, Vol. 29, No. 2, pp. 7-24, April 1999
[58] Open Networking Foundation, http://www.opennetworkingfoundation.org/
[59] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, E. Jansen, "The Gator Tech Smart House: A Programmable Pervasive Space," Computer, pp. 50-60, March, 2005
[60] Ubiquitous Intelligence and Computing, 7th International Conference Proceedings, UIC 2010, Xi'an, China, October 26-29, 2010
[61] T. Erl, "SOA Principles of Service Design", Service-Oriented Computing Series Prentice Hall/Pearson PTR, 2007
[62] Open Virtual Machine Format Whitepaper for OVF Specification. Available at http://www.vmware.com/resources/techresources/1003
[63] Open Virtualization Format Specification. Document number: DSP0243. Available at : http://www.dmtf.org/sites/default/files/standards/documents/DSP0243_1.1.0.pdf
[64] Open Cloud Community Interface, http://occi-wg.org/
[65] Open Grid Forum, http://www.ogf.org/
[66] GDF-P-R.183. Open Cloud Computing Interface – Core. Available at http://ogf.org/documents/GFD.183.pdf
[67] GDF-P-R.184. Open Cloud Computing Interface – Infrastructure. Available at http://ogf.org/documents/GFD.184.pdf
[68] Xen hypervisor, http://www.xen.org/
[69] Kernel Based Virtual Machine, http://kvm.qumranet.com/kvmwiki
[70] libvirt: The virtualization API, http://libvirt.org/
[71] FP7/ICT Reservoir (Resources and Services Virtualization without Barriers) project, http://www.reservoir-fp7.eu/, February 2008 - March 2011
[72] Virtual Infrastructure Management in Private and Hybrid Clouds, B. Sotomayor, R. S. Montero, I. M. Llorente, I. Foster. IEEE Internet Computing, vol. 13, no. 5, pp. 14-22, Sep./Oct. 2009.
[73] Amazon Elastic Compute Cloud (Amazon EC2), see http://aws.amazon.com/ec2/
[74] ElasticHosts, see http://www.elastichosts.com/
[75] Resource Leasing and the Art of Suspending Virtual Machines, B.Sotomayor, R.Santiago Montero, I.Martín Llorente, I.Foster. The 11th IEEE International Conference on High Performance Computing and Communications (HPCC-09), June 25-27, 2009, Seoul, Korea.
[76] Xcalibre Flexiscale, see http://flexiscale.com/index.html
[77] GoGrid, see http://www.gogrid.com/
[78] GoGrid API, see https://www.gogrid.com/how-it-works/gogrid-API.php
[79] Cloud Foundry, http://www.cloudfoundry.com/
[80] The Internet Engineering Task Force, IETF, RFC 3198, Terminology for Policy-Based Management
[81] The Internet Engineering Task Force, IETF, RFC 3444, On the Difference between Information Models and Data Models
[82] Unified Model Language, UML, http://www.uml.org/
[83] The Object Management Group, OMG, http://www.omg.org
[84] A., K., Dey, (2001). "Understanding and Using Context". Personal Ubiquitous Computing, Springer London, Volume 5, Number 1, February, 2001
[85] The Distributed Management Task Force, Inc., http://www.dmtf.org/
[86] Common Information Model (CIM) Specification, http://www.dmtf.org/standards/cim/cim_spec_v22
[87] The Open Mobile Alliance, http://www.openmobilealliance.org
[88] IEEE 1900.4 Standard for Architectural Building Blocks Enabling Network-Device Distributed Decision Making for Optimised Radio Resource Usage in Heterogeneous Wireless Access Networks, IEEE Std 1900.4-2009, Feb. 27, 2009.
[89] 3GPP TS 22.278: "Service requirements for the Evolved Packet System (EPS)".
[90] 3GPP TS 23.402 "Architecture enhancements for non-3GPP accesses (Release 9)"
[91] 3GPP TS 24.312 "Access Network Discovery and Selection Function (ANDSF) Managment Object (MO);(Release 10.1.0)
[92] N. Agoulmine, Autonomic Network Management Principles - From Concepts to Applications, Academic Press (Elsevier), 2011.
[93] K. Nolte et al, The E3 architecture: enabling future cellular networks with cognitive and self-x capabilities, International Journal of Network Management, Wiley, to appear
[94] The Internet Engineering Task Force, IETF, RFC 3198, Terminology for Policy-Based Management
[95] Strassner, J., "DEN-ng: achieving business-driven network management," Network Operations and Management Symposium, 2002. NOMS 2002. 2002 IEEE/IFIP, vol., no., pp. 753-766, 2002
[96] Strassner, J.; Hong, J.W.-K.; Kyo Kang; "A framework for modelling and reasoning about network management

resources and services to support information reuse," Information Reuse & Integration, 2009. IRI '09. IEEE International Conference on, vol., no., pp.85-90, 10-12 Aug. 2009
[97] Dey, A.K., Abowd, G.D., "Towards a better understanding of context and context awareness" in Workshop on the What, Who, Where, When and How of Context-Awareness, affiliated with the 2000 ACM Conference on Human Factors in Computer Systems (CHI 2000), April 2000
[98] Dey, A.K., Salber, D., Abowd, G.D., Futakawa, M., "An architecture to support context-aware applications" GVU Technical Report: GIT-GVU-99-23, 1999
[99] Raz, D., Juhola, A., Serrat, J., Galis, A. –"Fast and Efficient Context-Aware Services" ISBN 0-470-01668-X; pp250, April 2006; John Wiley & Sons, Ltd.; http://eu.wiley.com/WileyCDA/WileyTitle/productCd-047001668X.html;
[100] Clark, D. D., Partridge, C., Ramming, J. C., Wroclawski, J. T. - "A Knowledge Plane for the Internet" SIGCOMM2003, Karlsruhe, Germany, 2003
[101] AutoI FP7 project http://ist-autoi.eu/autoi/index.php
[102] CASCADAS Project http://acetoolkit.sourceforge.net/cascadas/
[103] 4WARD Fp7 Project http://www.4ward-project.eu/
[104] ANA Project, http://www.ana-project.org/
[105] Self-Net Project Home Page, https://www.ict-selfnet.eu
[106] The Ganglia monitoring system: http://ganglia.sourceforge.net/
[107] Robbert Van Renesse, Kenneth P. Birman, and Werner Vogels. 2003. Astrolabe: A robust and scalable technology for distributed system monitoring, management, and data mining. ACM Trans. Comput. Syst. 21, 2 (May 2003), 164-206. DOI=10.1145/762483.762485 http://doi.acm.org/10.1145/762483.762485
[108] Moara: Flexible and Scalable Group-based Aggregation System http://kepler.cs.uiuc.edu/~sko/moara/
[109] IBM Tivoli: Integrated Service Management Software, //www-01.ibm.com/software/tivoli/
[110] IETF Policy Framework Working group, Charter available at: http://www.ietf.org/html.charters/OLD/policy-charter.html.
[111] D.C. Verma, "Simplifying Network Administration Using Policy-Based Management," IEEE Network, Vol. 16, No. 2, March 2002.
[112] B. Moore, E. Ellesson, J. Strassner, A. Westerinen, "Policy Core Information Model," RFC 3060, IETF, February 2001.
[113] D. Duurham, J. Boyle, R. Cohen, S. Herzog, R. Rajan, A. Sastry, "The COPS (Common Open Policy Service) Protocol," RFC 2748, IETF, January 2000.
[114] S. Boros, "Policy-based Network Management with SNMP," CTIT Technical Report Series, No. 00-16, ISSN 1381-3625, 8 pages, 2000.
[115] W. Yeong, T. Howes, S. Kille, "Lightweight Directory Access Protocol," RFC 1777, March 1995.
[116] M.S. Sloman, J. Moffett, "Domain Management for Distributed Systems," Integrated Network Management, Vol. 1, North Holland, Amsterdam, 1989.
[117] M.S. Sloman, "Policy Driven Management for Distributed Systems," Journal of Network and Systems Management, Vol. 2, No. 4, pp. 333-360, Plenoum Press, December 1994.
[118] M.S. Sloman, E.C. Lupu, "Policy Specification for Programmable Networks," proceedings of International Conference on Active Networks, Springer-Verlag, Berlin, Germany, June 1999.
[119] N. Dulay, E. Lupu, M.S. Sloman, N. Damianou, "A Policy Deployment Model for the Ponder Language," proceedings of IEEE/IFIP International Symposium on Integrated Network Management, Seattle, USA, May 2001.
[120] N. Damianou, N. Dulay, E.C. Lupu, M.S. Sloman, "The Ponder Policy Specification Language," proceedings of IEEE Workshop on Policies for Networks and Distributed Systems, Bristol, UK, January 2001.
[121] D. Agrawal, K.W. Lee, J. Lobo, "Policy-based Management of Networked Computing Systems," IEEE Communications Magazine, Vol. 43, No. 10, pp. 69-75, October 2005.
[122] The Internet Engineering Task Force, IETF, RFC 3198, Terminology for Policy-Based Management
[123] J. P. Delgrande, J. Mylopoulos, Knowledge representation: features of knowledge, W. Bbel, P. Jorrand (Eds.), Fundamental of Artificial Intelligence: An Advanced Course, LNCS 232 Springer, Berlin, 1986.
[124] F. S. Correa da Silva, et al, On the insufficiency of ontologies: problems in knowledge sharing and alternative solutions, Elsevier, Knowledge-Based Systems 15 (2002) 147-167
[125] DIKW from Wikipedia, http://en.wikipedia.org/wiki/DIKW
[126] Bierly, P., E., Kessler, E., H., Christensen, E., W., Organizational learning, knowledge and wisdom, available at: http://www.emeraldinsight.com/
[127] C. Polychronopoulos, M. Stamatelatos, N. Alonistioti, Tracing the Parallel Paths of Knowledge Management in the Organizational Domain and the Autonomic Paradigm , JCKBSE 08, 8th Joint Conference on Knowledge - Based Software Engineering 2008
[128] N. Agulmine, Autonomic Network Management Principles, From Concepts to Applications, ISBN 978-0-12-382190-4
[129] J.A. Lozano, J.M. González, J. Morilla. A Telco approach to autonomic infrastructure management.
[130] Multi-operator telecommunication distribution of service content. http://www.wipo.int/pctdb/en/wo.jsp?IA=SE2005000483&DISPLAY=DESC
[131] UniverSelf Milestone MS24. UMF specifications.
[132] Autonomic Internet (AutoI) project http:/ist-autoi.eu/

[133] Strassner, John and Agoulmine, N. and Lehtihet, E. (2006) FOCALE: A Novel Autonomic Networking Architecture. In: Latin American Autonomic Computing Symposium (LAACS), 2006, Campo Grande, MS, Brazil.
[134] EC funded- FP7-EFIPSANS Project: http://efipsans.org/
[135] ANA Project Homepage, http://www.ana-project.org/ [Last accessed: 25 Sep 2010]
[136] Information Management Datasheet. Available at http://www.tmforum.org/BestPracticesStandards/DatasheetsInformation/4274/Home.html.
[137] A. Andrieux et al. Web Services Agreement Specification (WS-Agreement), Version March 2007. Grid Forum Document, GFD.107, Proposed Recommendation, Open Grid Forum, 2007. Available at http://www.ogf.org/documents/GFD.107.pdf.
[138] Keven T. Kearney, Francesco Torelli, Constantinos Kotsokalis. SLA*: An Abstract Syntax for Service Level Agreements. Service Level Agreements in Grids Workshop 2010.
[139] Carlos Bueno Royo, JuanLambea Rueda, Óscar L. Dueñas Rugnon, Beatriz Fuentes, Alfonso Castro. Business Terms. Model for a Telecom Operator Business View of SLA. ICE-B 2010
[140] Ranganai Chaparadza, Martin Vigeraux, José-Antonio Lozano-López, and Juan-Manuel González-Muñoz, "Engineering Future Network Governance", European Journal for the informatics professional 2010.
[141] Costantinos Kotsokalis, Ulrich Winkler. Translation of Service Level Agreements: A Generic Problem Definition. Workshop on Non-Functional Properties and SLA Management in Service-Oriented Computing .NFPSLAM'09
[142] Gruber, T. R., Toward Principles for the Design of Ontologies Used for Knowledge Sharing. International Journal Human-Computer Studies, 43(5-6):907-928, 1995.
[143] OWL-S: Ontology Web Language Overview. W3C. http://www.w3.org/TR/owl-features/
[144] SWRL proposal: http://www.w3.org/Submission/SWRL/
[145] Rule Markup Web Site: http://www.ruleml.org/
[146] Yavatkar et al., "A Framework for Policy-based Admission Control", RFC 2753, January 2000. John Strassner1, José Neuman de Souza, Sven van der Meer1, Steven Davy1, Keara Barrett1, Dave
[147] John Strassner, José Neuman de Souza, Sven van der Meer, Steven Davy, Keara Barret, Dave Raymer, Srini Samudrala. The Design of a New Policy Model to Support Ontology-Driven Reasoning for Autonomic Networking.
[148] Steven Davy, Brendan Jennings, John Strassner. The Policy Continuum. A formal model.
[149] A. Uszok, J. Bradshaw, R. Jeffers, N. Suri, P. Hayes, M. Breedy, L. Bunch, M. Johnson, S. Kulkarni, J. Lott. KAoS Policiy and Domain Services: Toward a Description-Logic Approach to Policy Representation, Deconfliction, and Enforcement.
[150] Steven Davy. Harnessing Information Models and Ontologies for Policy Conflict Analysis. PhD. Thesis.
[151] Steven Davy, Brendan Jennings, and John Strassner. On harnessing information models and ontologies for policy conflict anaylsis. In Proc. 11th IFIP/IEEE International Symposium on Integrated Network Management (IM 2009), pages 821–826. IEEE, 2009.
[152] Charalambides, M.; Flegkas, P.; Pavlou, G.; Rubio-Loyola, J.; Bandara, A. K.; Lupu, E. C.; Russo, A.; Dulay, N. and Sloman, M. (2009). Policy conflict analysis for diffserv quality of service management. Network and Service Management, IEEE Transactions on, 6(1), pp. 15–30.
[153] Steven Davy, Keara Barrett, Sasitharan Balasubramaniam, Sven van der Meer, Brendan Jennings, John Strassner. Policy-Based Architecture to Enable Autonomic Communications – A Position Paper
[154] Arosha K Bandara Emil C Lupu Jonathan Moffett Alessandra Russo Arosha: A Goal-based Approach to Policy Refinement. 5th IEEE International Workshop on Policies for Distributed Systems and Networks. June 2004
[155] Arosha K Bandara, Emil C Lupu, Alessandra Russo: Using event Calculus to formalise specification and analysis. IEEE 4th International Workshop on Policies for Distributed Systems and Networks. Jun 2003
[156] John Strassner and José Neuman De Souza and Sven Van Der Meer and Steven Davy and Keara Barrett and David Raymer and Srini Samudrala, "The Design of a New Policy Model to Support Ontology-Driven Reasoning for Autonomic Networking, Journal of Network and Systems Management - JNSM , vol. 17, no. 1-2, pp. 5-32, 2009.
[157] "CIM Core Model White Paper", DMTF, August 2000
[158] "CIM Policy White Paper", DMTF, June 2003
[159] Famaey, J.; Latrea, S.; Strassner, J.; De Turck, F.; , "A hierarchical approach to autonomic network management," Network Operations and Management Symposium Workshops (NOMS Wksps), 2010 IEEE/IFIP , vol., no., pp.225-232, 19-23 April 2010
[160] http://technet.microsoft.com/en-us/library/ff647958.aspx
[161] Jeroen Famaey, Steven Latré, John Strassner and Filip De Turck, An Ontology-Driven Semantic Bus for Autonomic Communication Elements, MODELLING AUTONOMIC COMMUNICATION ENVIRONMENTS Lecture Notes in Computer Science, 2010, Volume 6473/2010, 37-50, DOI: 10.1007/978-3-642-16836-9_4

9 Abbreviations

API	Application Programming Interface
ANA	Autonomic Network Architecture
AutoI	Autonomic Internet
BM	Behaviour Model
BSS	Business Support System
CAP	Context Awareness pattern
CASCADAS	Component-ware for Autonomic Situation-aware Communications, and Dynamically Adaptable Services
CIM	Common Information Model
ClInMa	Control Loop Interaction Management
CPU	Central Processing Unit
CS	Clean State
DM	Domain Manager
DMTF	Distributed Management Task Force
DoW	Description of Work
E3	End to End Efficiency
EFIPSANIS	Exposing the Features in IPv6 protocols that can be exploited/ extended for the purposes of designing/building autonomic Networks and Services
ELMS	Enhanced Legacy Management System
EMS	Element Management System
ENE	Empowered Network Elements
FB	Functional Block
FCAPS	Fault, Configuration, Accounting, Performance, Security
FIB	Forwarding Information Base
FG	Functional Group
FG – FN	Focus Group on Future Networks
FMS	Future Management Systems
H2N	Human to Network
HNO	Human Network Operator
HTTP	HyperText Transfer Protocol
IaaS	Infrastructure as a Service
IMS	IP Multimedia Subsystem
IN	Intelligent Network
ITU – T SG13	International Telecommunication Union Study Group 13
LTE/ SAE	Long Term Evolution/ Service Architecture Evolution
MAPE	Monitor – Analyse – Plan – Execute
NE	Network Element
NMS	Network Management System
OCCI	Open Cloud Computing Interface
OCL	Object Constraint Language
ODM	Ontology Definition Metamodel
OMG	Object Management Group

OPEX	Operational Expenditure
OSPF	Open Shortest Path First
OSS	Operating Support System
OVF	Open Virtual machine Format
OWL	Ontology Web Language
P2P	Peer to Peer
RAT	Radio Access Technology
RIB	Routing Information Base
SaaS	Software as a Service
SCP	Service Control Point
Self - NET	Self-Management of Cognitive Future InterNET Elements
SOCRATES	Self-Optimisation and self-ConfiguRATion in wirelEss networkS
SOAP	Simple Object Access Protocol
S/W	Software
U_DC	UMF Intra-domain controller
U_FC	UMF Inter-domain controller/ federated controller
UMF	Unified Management Framework
UML	Unified Modelling Language
VoIP	Voice over Internet Protocol
VDS	Virtual Dedicated Server
XML	eXtensible Markup Language

10 Definitions

Business requirement – *it is a description in business terms of what must be delivered or accomplished to provide value.*

Compliance – *the conformance to a rule, such as a specification, policy, standard or regulatory.*

Extensibility – *the ability to extend a system and the level of effort and complexity required to realize an extension. Extensions can be through as the addition of new functionality, new characteristics or through modification of existing functionality/characteristics, while minimizing impact to existing system functions.*

Functional Block – *is a group of functions which have derived from all use cases and exhibit similar purpose/goal and/or similar inputs and outputs or operation. Accordingly, the functional blocks group functions with commonalities and irrespectively of the use case from which they eventually derive from. As such, they designate design blocks that exhibit great levels of reusability and cohesion and can be used to implement a core function of the UMF.*

Functional Group – *is an aggregation of functional blocks, which realizes a higher level management function, i.e. this high-level functional grouping is the highest level of granularity.*

Functional requirement – *it is a description of a function, or a feature of a system, or its components, capable of solving a certain problem or replying to a certain need/request. The set of functional requirements present a complete description of how a specific system will function, capturing every aspect of how it should work before it is built, including information handling, computation handling, storage handling and connectivity handling.*

Interoperability – *the ability of diverse systems and subsystems to work together (inter-operate).*

Network Governance – *a framework which enables operators to describe their goals and objectives, through high-level means and govern their network. Include the derivation of network policies from the business goals through the use of semantic techniques.*

Non-functional requirement – *it is a description of how well a system performs its functions, it represents an attribute that a specific system must have. The non-functional requirements are controlled by other aspects of the system.*

Operability – *the ability to keep a system in a safe and reliable functioning condition, according to pre-defined operational requirements.*

Performance – *it describes the degree of performances a system (according to certain predefined metrics, e.g. convergence time).*

Privacy – *the ability of system or actor to seclude itself or information about itself and thereby reveal itself selectively.*

Scalability – *the ability of a system to handle growing amounts of work or usage in a graceful manner and its ability to be enlarged to accommodate that growth.*

Security – *the ability to prevent and/or forbid access to a system by unauthorized parties.*

Stakeholder – *a person, group or organization with an interest in something.*

Usability – *the ease with which a system performing certain functions or features can be adopted and used.*

Use Case – *it is a descriptor of a set of precise problems to be solved. It describes steps and actions between stakeholders and/or actors and a system, which leads the user towards a value added or a useful goal. A UC describes what the system*

shall do for the actor and/or stakeholder to achieve a particular goal. Use-cases are a system modelling technique that helps developers determine which features to implement and how to gracefully resolve errors.

Annex A: Fulfilment of the UMF requirements by the Functional Groups

Apart from the consolidation of the outcomes of the top down and the bottom up approach; functional groups need to also address UMF top level requirements. Accordingly, Table 4 names these requirements and their respective correlation with the functional groups in terms of which functional group(s) is (are) responsible for each UMF top level requirement.

The expected properties of the top-level requirements are:

1. atomic: each requirement addresses one functional group and only one thing.
2. complete: each requirement is fully define with no missing information.
3. dependable: each requirement does not contradict any requirements and it is fully consistent with all relevant references.
4. current: the requirements have not been made obsolete by the existing networking functions.
5. feasible: the requirements can be implemented as supported by the enabling technology
6. prioritized (must have Vs. nice to have): each requirement represents a characteristic the absence of which will result in a deficiency that cannot be ameliorated. if must-have requirements are not met, efficiency at major scale fails.
7. verifiable: the implementation of a design goal/objective can be determined through one of five possible methods: inspection, demonstration, test, trial or analysis.

Table 4. UMF top level requirements realization through the functional groups

Req. ID	Name	Definition / Description	Source or link	Responsible Functional Group
		UMF Top Level Requirements - Universal Project Internal		
R1	Automation of networks and services	UMF must enable the automation of networks and services on-going management works in an adaptable, flexible and scalable way.	Project Internal - (CD)	All UMF FGs
R2	Coordination and Orchestration	UMF must provide the coordination and orchestration of the managing and managed elements based on human control/directives.	Project Internal - (CD)	Intelligence FG
R3	Migration	UMF must provide a migration path to support the progressive introduction of self-x management features in the existing NE/EMS/NMS/OSS/BSS management chain.	Project Internal – (CD)	All UMF FGs
R4	High-level Interchange	UMF must facilitate high-level dialogues between self-managed networks and multiple human network operators satisfying the following properties: ▪ every well-formed query is answered by a network pertinently; ▪ every well-formed goal injected to a network is either enforced completely and instantly or its delay/modifications are negotiated per rules instantiated; ▪ every impossibility to continue self-managed operation or realistic danger of that must be reported to humans with pertinent details of the situation.	Project Internal – (MS)	UMF Governance FG
R5	Competing Goals, Optimisation,	UMF must be able to ▪ adequately weight competing goals	Project Internal –	Intelligence FG

	Triggers	▪ re-optimize individual management processes at ideal points in time ▪ provide a set of specific events that enforce policies	(MG)	
R6	System Stability	As the introduction of autonomic/self-organise network capabilities into a network and services might cause instabilities jeopardizing performances and integrity UMF must provide the means of monitoring, detecting, resolving and managing stability problems in networks and services.	Project Internal - (AM)	Intelligence FG
R7	Performance of Network Empowerment	Performance requirements: For instance, problem X (or function X) must be solved (must perform) in less than 1 millisecond -or- solution must guarantee to scale up to 100K concurrent states per [interface/node] -or- function should work with partial information available. There may be also system constraints for feasibility such as model size must be inferior to 1MB, if distributed process is required.	Project Internal – (LC)	All UMF FGs
R8	Self-Functionality	UMF must enable integrated self-functions, interworking and orchestration of different self-functions and the embodiment of self-functionality in networks and services. Self-functionality includes: Self- configuration, -monitoring, -optimization, -organization, - healing, - diagnosis, -protection, -awareness, - governance, - testing, -management, -learning	Project Internal – (AG)	All UMF FGs
R9	Extensibility/ Change of management functionality	UMF must provide the enablers for activating new management functionality on demand in a plug-and-play / unplug-and-play fashion and programmatically	Project Internal – (AG)	Intelligence FG
R10	Life cycle management functions	UMF must provide design, deployment, activation/deactivation, operation, update, move, and change for all management functionality.	Project Internal – (AG)	Intelligence FG
R11	Integration functionality	UMF must provide the enablers and common functionality for interworking, communication and orchestration of different management functions	Project Internal – (AG)	Intelligence FG
		UMF Top Level Requirements - DoW		
R12	Interoperability and Federation of Multiple management systems	UMF will first ensure that multiple diverse management systems implemented upon different autonomic architectures will be able to interoperate and federate. Secondly, it will also guarantee that autonomic functions may be implemented (apart from optional interfacing) independently of the architecture chosen for the management system →UMF would be separated in several logically independent management functional groups dealing with different management tasks in order to allow interoperability and federation of different management systems at each area level	DoW	Intelligence FG
R13	Network Empowerment	Demonstrate that UMF enable Network Empowerment (embed intelligence into the systems and network equipments that constitute the infrastructure and support service delivery) →UMF adds intelligence to services and network domains	DoW	Intelligence FG
R14	Multi-faceted Unification	UMF is an unified and evolvable framework constituting a cross-technology (wireless and wireline)	DoW	Intelligence FG

		and common substrate for both systems and services →Explicit abstraction / substrate in support of the management of both networks and services		
R15	Autonomicity	UMF will demonstrate 'autonomic networking (self-x networking) →Explicit control & orchestration of a variety of autonomic closed control loops for each separate management function / group of management functions	DoW	UMF Intelligence FG
R16	Information management	Implement the enhanced and extensible information management to assure that UMF always makes informed decisions at both system and network levels → Explicit Information / Context / Knowledge Management	DoW	UMF Knowledge Management FG
R17	Autonomic Governance	UMF is represented by behavioural requirements of autonomic blocks and standardized interfaces. It will ensure that these blocks contain capabilities and mechanisms to govern the integrated behaviour and operations of all networking blocks →UMF would be separated in several logically independent management functional groups dealing with different management tasks in order to allow governance	DoW	UMF Governance FG
R18	Service orientation	Convergence towards 'Everything as a managed Service', which includes Network as a Service. →UMF is service oriented; it offers a service view instead of the traditional hardware view. →UMF covers explicitly both network and services aspects in an unified manner →UMF covers explicitly 'Network as a Service' (e.g. management of the integration of network and service aspects)	DoW	All UMF FGs
R19	Abstraction of Resources	UMF will be able to provide operators with an abstraction of the network they are operating, and this abstraction will be unified through the various network types →UMF covers explicitly the creation and use of an abstraction of physical resources and a mapping to each type of network resources	DoW	UMF Enforcement and Knowledge FGs
R20	Federation Management	UMF is a network agnostic management of services, able to federate the management of multiple networks →enable federation of domains and federation interfaces and behaviour requirements for dynamic composition / decomposition of different domains of resources	DoW	Intelligence FG
R21	Human-to-Network Interfaces	Develop a privileged, powerful and evolved human to network interface shifting from network management to network governance →Explicit design of management functionality & interfaces of the governance	DoW	UMF Governance FG
		UMF Top Level Requirements - Future Networks		
R22	Management of Future Networks	UMF would manage all new networking and servicing functionality of Future Networks as depicted in R22.1-R22.15.	ITU-T Y.3001 (Q2/2011) 'Future	All UMF FGs

		Phased deployment of Future Networks falls roughly between 2015 and 2020.	Networks: requirements and design goals'	
R22.1	Management of FN – Service awareness	UMF would provide full life cycle management of services whose functions are designed to be appropriate to the needs of applications and users. The number and range of services is expected to explode in the future. UMF is aimed to accommodate manageability of these services without drastic increases in, for instance, deployment and operational costs.	ITU-T Y.3001 (Q2/2011) 'Future Networks: requirements and design goals'	UMF Knowledge Management FG
R22.2	Management of FN – Data awareness	UMF would provide the optimised enablers for handling enormous amounts of data in a distributed environment, and the users' enablers to access desired data safely, easily, quickly, and accurately, regardless of their location. In the context of this requirement "data" is not limited to specific data types like audio or video content, but describes all information accessible on a network.	ITU-T Y.3001 (Q2/2011) 'Future Networks: requirements and design goals'	UMF Knowledge Management FG
R22.3	Management of FN – Environment awareness	UMF would provide a number of environmentally friendly enablers. The design, resulting implementation and operation of UMF are aimed to minimize their environmental impact; such as the consumption of materials and energy and reducing greenhouse gas emissions. UMF are recommended to be also designed and implemented so it can be used to reduce the environmental impact of other sectors (i.e. transport, intelligent cities, utilities, etc.)	ITU-T Y.3001 (Q2/2011) 'Future Networks: requirements and design goals'	All UMF FGs
R22.4	Management of FN – Social and Economic awareness	UMF would provide enablers for reflecting social and economic issues to reduce barriers to entry for the various participants of the networks. UMF's lifecycle costs would be reduced and managed in order for UMF to be deployable and sustainable. These factors will help to universalize the services and allow appropriate competition and an appropriate return for all participants.	ITU-T Y.3001 (Q2/2011) 'Future Networks: requirements and design goals'	All UMF FGs
R22.5	Management of FN – Service Diversity	UMF would provide enablers for life-cycle management of diversified ICT services accommodating a wide variety of traffic characteristics. UMF would also manage a huge number and wide variety of communication objects such as sensors and terminal devices.	ITU-T Y.3001 (Q2/2011) 'Future Networks: requirements and design goals'	All UMF FGs
R22.6	Management of FN – Functional Flexibility	UMF would provide enablers for to manage and sustain new end-user facing services and new network resource facing services. UMF would be design and implemented to support agile deployment of new services keeping pace with their rapid growth and change. UMF would provide facility to accommodate experimental services for testing and evaluation purposes, and it should also enable graceful migration from experimental services to established services, from experimental protocols to established protocols to lower the obstacles for new service deployment.	ITU-T Y.3001 (Q2/2011) 'Future Networks: requirements and design goals'	All UMF FGs
R22.7	Management of FN – Virtualisation of Resources	UMF would provide enablers for managing and support virtualisation of resources associated with networks, computation and storage in order to support partitioning of resources so that a single	ITU-T Y.3001 (Q2/2011) 'Future Networks:	UMF Enforcement FG

		resource can be used concurrently by multiple virtual resources. UMF would also support isolation of any virtual resource from all others. Virtual resource need not directly correspond to its physical characteristics. Virtualization of resources can realize networks without interfering with the operation of other virtual networks while sharing the network resources among virtual networks. Since multiple virtual networks can simultaneously coexist, different virtual networks can use different network technologies without interfering with other virtual networks and it is possible to improve utilization of physical resources. Also, the abstraction property enables to provide standard interfaces for accessing and managing the virtual network and resources and helps to support updating of the capability of virtual networks.	requirements and design goals'	

Annex B: Tentative message flow between Functional Blocks within different use cases

UC1 - Self-Diagnosis and self-healing for IMS VoIP and VPN services

Taking into account the mapping presented in Section 5.4, the necessary data for each FB (to be used as input) and the outcome of the FBs involved in UC1 in case of self-diagnosing and self-healing a congestion of a link between two elements that belong in the same domain, we concluded in a tentative Message Sequence Chart (MSC) of UC1, like the one depicted inFigure 42 Table 5 and Table 6 resume the respective tentative messages and their possible parameters.

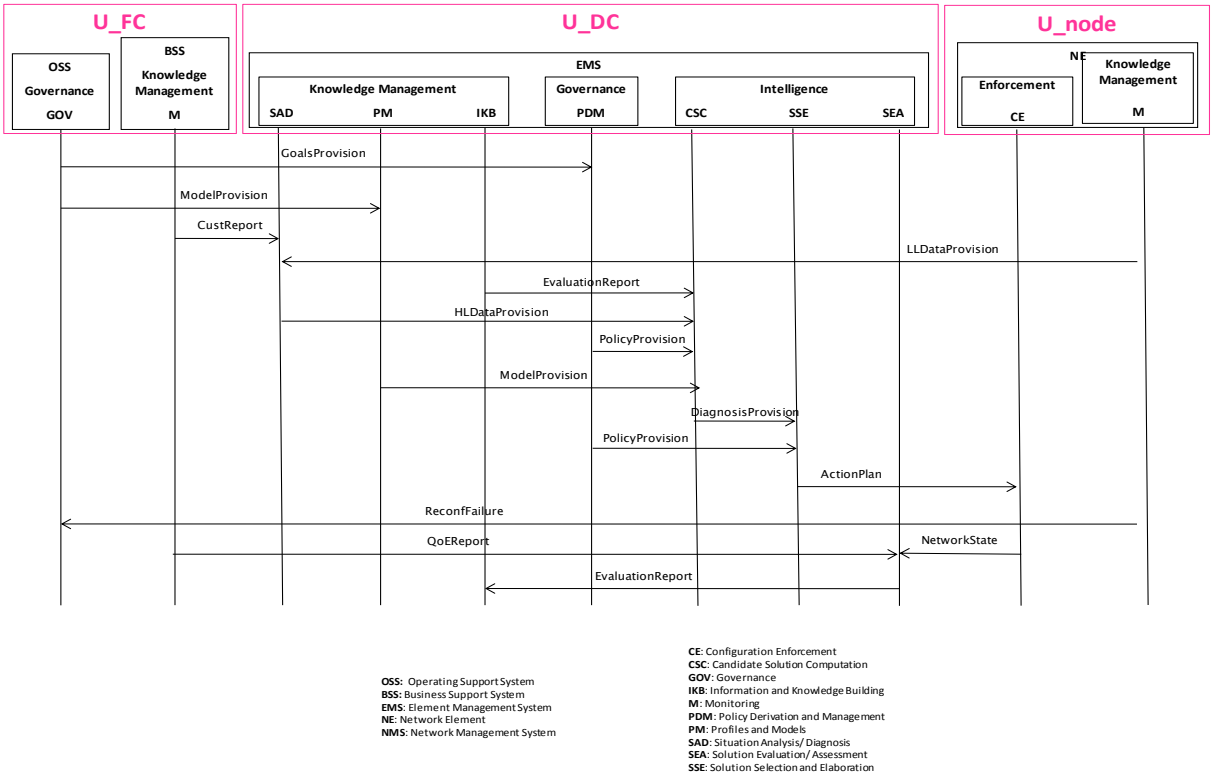


Figure 42. Content Exchanges between the management systems of UC1.

Table 5. Tentative Messages of UC1

Source	Destination	Primitive	Parameters	Scope
Governance @ OSS Service Segment	Policy Derivation and Management @ NMS NMS segment	GoalsProvision	List of BusinessGoals	Feeds Policy Derivation and Management FB with the business goals so as the latter to create policies
Governance @OSS Service segment	Profiles and Models @NMS NMS segment	ModelProvision	Models of • reparation/ mitigation plans • service • network • predefined faults	Informs the database that holds the Profiles and the Models with the new model

			• events • anomaly • normality • KPIs	
Monitoring @ BSS (customer reports) Service segment	Situation Analysis/ Diagnosis @ NMS NMS Segment	CustReport	Service_id Malfunction_id	Provides the analysis mechanism with the reports coming from customers/ customer care service
Monitoring @ EMS Access (wireless or Wireline)/ Core Segment	Situation analysis / Diagnosis @ NMS NMS segment	LLDataProvision	NDataType DataType_id DataValue	Feeds low level information/ data to the mechanism which is responsible for their analysis
Information and Knowledge Building @NMS NMS Segment	Candidate Solution Computation @NMS NMS segment	EvaluationReport	NConfPara ConfParam_id ParamValue QoEValue	Transfers past knowledge (evaluation of past actions) to the Candidate Solution Computation
Situation Analysis/ Diagnosis @ NMS NMS segment	Candidate solution computation @ NMS NMS segment	HLDataProvision	NDataType DataType_id DataValue	Carries elaborated data (High Level) and correlations of low level data to the "candidate solution computation" mechanism
Policy Derivation and Management @ NMS NMS segment	Solution Selection Elaboration @NMS NMS segment	PolicyProvision	List of Policies	Feeds the decision making mechanism with the policies that it has to obey
Profiles and Models @NMS NMS segment	Candidate Solution Computation @NMS NMS segment	ModelProvision	Models of • reparation/ mitigation plans • service • network • predefined faults • events • anomaly • normality • KPIs	Informs the Candidate Solution Computation FB about the available models
Candidate solution computation @ NMS NMS segment	Solution selection and elaboration @ NMS NMS segment	DiagnosisProvision	DiagnosisType_id DiagnosisValue	Informs the decision making mechanism for the diagnosis made

Policy Derivation and Management @ NMS NMS segment	Solution Selection Elaboration @NMS NMS segment	PolicyProvision	List of Policies	Feeds the decision making mechanism with the policies that it has to obey
Solution selection and elaboration @ NMS NMS segment	Configuration enforcement @ EMS Access (Wireless/ wireline)/ Core Segment	ActionPlan	ConfParam_id NewParamValue	Transfers the new configuration values to the elements that need to be reconfigured
Configuration enforcement @ EMS Access (Wireless/ wireline)/ Core Segment	Governance @ OSS Service Segment	ReconfFailure	ReconfFailureValue	Alerts the network administrator through the H2N interface if the reconfiguration was unsuccessful
Configuration enforcement @ EMS Access (Wireless/ wireline)/ Core Segment	Solution Evaluation/ Assessment @ OSS Service Segment	NetworkState	NConfPara ConfParam_id ParamValue	Provides evaluation mechanism with the current (new) state of the network
Monitoring @ BSS (customer reports) Service segment	Solution Evaluation/ Assessment @ NMS NMS Segment	QoEReport	Customer_id QoEValue	Gives feedback for the customers' QoE after the healing
Solution Evaluation/ Assessment @ NMS NMS segment	Information and Knowledge Building @NMS NMS Segment	EvaluationReport	NConfPara ConfParam_id ParamValue QoEValue	Transfers the new configuration values to the elements that need to be reconfigured

Table 6. Message Parameters of UC1

Parameter	Type	Description
List of BusinessGoals	List of composite types	Describes the Business Goals according to which the system decides whether to replace the whole faulty equipment by a working one or to repair the malfunctioning units. This decision depends on the severity of the event and the equipment functionalities.
List of Policies	List of composite types	Describes the Policies created according to the business goals.
Models of • reparation/ mitigation plans • service • network • predefined faults • events • anomaly	List of composite types	Describes the • reparation/ mitigation plans • service • network • predefined faults • events • anomaly • normality

• normality • KPIs		• KPIs
NDataType	Integer	Integer representing the number of data types carried in the message
DataType_id	Integer	Integer representing the type of the respective data
DataValue	Integer	Value on the data
DiagnosisType_id	Integer	1 for proactive, 2 for reactive diagnosis
DiagnosisValue	String or integer	Name of the fault or an integer representing the respective fault
ConfParam_id	String/ integer	Name of the parameter to be configured or an integer representing the respective parameter
NewParamValue	integer	New value of the parameter
ReconfFailureValue	boolean	Failure of the reconfiguration of the element or not
NConfPara	integer	Number of the configuration parameters
ParamValue	integer	value of the parameter
Service_id	String/ integer	Name of the service related to the report of the customer or an integer representing the respective service
Malfunction_id	String/ integer	Name of the malfunction of the service or an integer representing the respective malfunction
Customer_id	Integer	Integer representing the id of the customer
QoEValue	Integer	Integer expressing customers' satisfaction by the service and the time needed for the healing

UC2 - Networks' Stability and Performance

A tentative MSC for this UC, taking into account the above presented mapping (section 0), the necessary data for each FB (to be used as input), the outcome of the FBs involved in UC2 and attempting to specify the parameters exchanged between the functionalities under each message name we concluded in Figure 43. Table 7 and

Table 8 resume these tentative messages and their possible parameters.

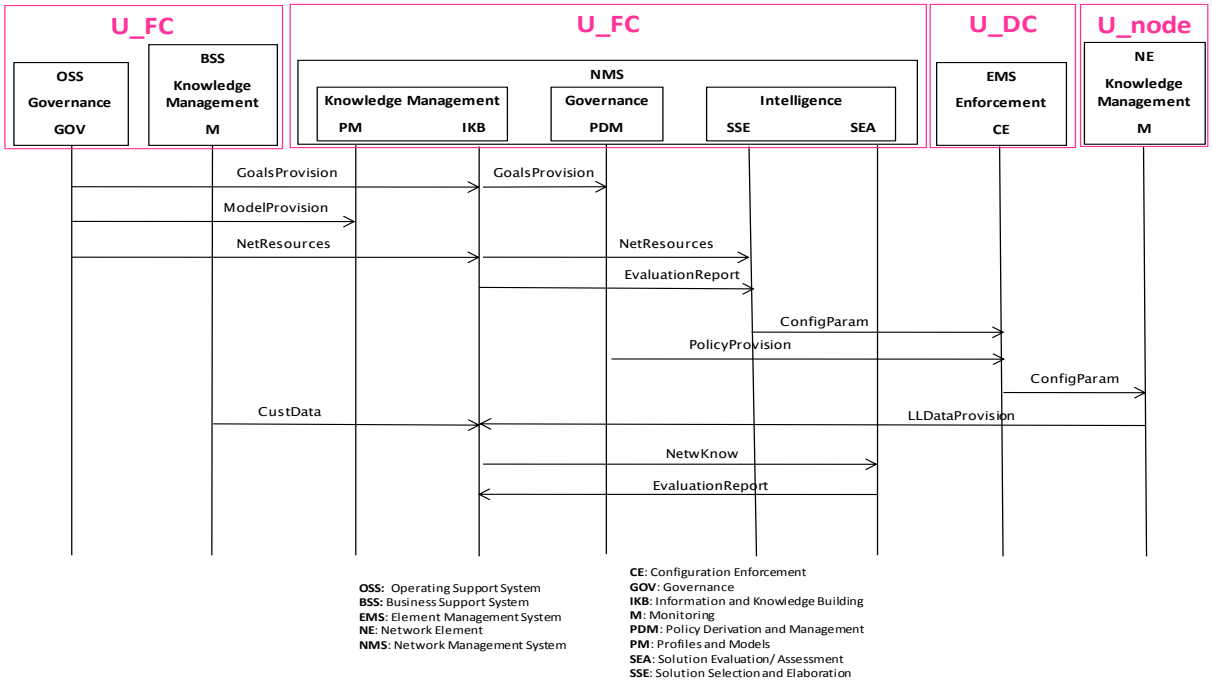


Figure 43. Content Exchanges between the management systems of UC2.

Table 7. Tentative Messages of UC2

Source	Destination	Primitive	Parameters	Scope
Governance @ OSS Service Segment	Information and Knowledge Building NMS segment	GoalsProvision	List of BusinessGoals	Feeds Knowledge database with business goals
Information and Knowledge Building NMS segment	Policy Derivation Management NMS segment	GoalsProvision	List of BusinessGoals	Feeds Policy Derivation and Management FB with the business goals so as the latter to create policies
Governance @OSS Service segment	Profiles and Models @NMS NMS segment	ModelProvision	Models of • network stability, • technologies, • topologies, • mobility, • traffic, • network, • service and • energy.	Informs the database that holds the Profiles and the Models with the new models
Governance @OSS Service segment	Information and Knowledge Buuilding @NMS NMS segment	NetResources		Feeds knowledge base with the information of the available network resources
Information and Knowledge Buuilding @NMS NMS segment	Solution Selection and Elaboration @NMS NMS segment	NetResources		Feeds decision mechanism with the information of the available network resources
Information and Knowledge Buuilding @NMS NMS segment	Solution Selection and Elaboration @NMS NMS segment	EvaluationReport	NConfParam ConfParam_id ParamValue QoSValue	Feeds decision mechanism with knowledge related to (end to end) evaluation of solutions applied in the past
Solution selection and elaboration @ NMS NMS segment	Configuration enforcement @ EMS Access (Wireless/ wireline)/ Core Segment	ConfigParam	ConfParam_id NewParamValue	Transfers the new configuration values to the elements that need to be reconfigured
Policy Derivation and Management @ NMS NMS segment	Configuration Enforcement @EMS Access (Wireless/ wireline)/ Core Segment	PolicyProvision	List of Policies	Feeds the configuration enforcement FB with the policies that it has to obey
Configuration enforcement @ EMS	Monitoring @NE Access (Wireless/ wireline)/ Core Segment	ConfigParam	ConfParam_id NewParamValue	Transfers the configured in the elements

Access (Wireless/ wireline)/ Core Segment				parameters with their new values to Monitoring FB so as the latter to monitor them
Monitoring @ NE Access (wireless or Wireline)/ Core Segment	Information and Knowledge Building @ NMS NMS segment	LLDataProvision	NDataType DataType_id DataValue	Feeds low level information/ data to the knowledge building mechanism so as to be collected, filtered and elaborated
Monitoring @ BSS (customer reports) Service segment	Information and Knowledge Building @ NMS NMS Segment	CustData	customer_id custdata_id custdata_value	Provides the knowledge building mechanism so as to be collected, filtered and elaborated
Information and Knowledge Building @ EMS NMS Segment	Solution Evaluation/ Assessment @ NMS NMS segment	NetwKnow	NDataType DataType_id DataValue	Feeds current state of the networks (monitored values) to the solution evaluation/ assessment mechanism for the end to end evaluation
Solution Evaluation/ Assessment @NMS NMS Segment	Information Knowledge Building @NMS NMS segment	EvaluationReport	NConfParam ConfParam_id ParamValue QoSValue	Transfers knowledge coming from the evaluation of past actions to the knowledge database so as to be saved and exploited in future decisions

Table 8. Message Parameters of UC2

Parameter	Type	Description
List of BusinessGoals	List of composite types	Describes the Business Goals according to which the system decides whether to replace the whole faulty equipment by a working one or to repair the malfunctioning units. This decision depends on the severity of the event and the equipment functionalities.
List of Policies	List of composite types	Describes the Policies created according to the business goals.
Models of • network stability, • technologies, • topologies, • mobility,	List of composite types	Describes the models of • network stability, • technologies, • topologies, • mobility,

• traffic, • network, • service and • energy.		• traffic, • network, • service and • energy.
NDataType	Integer	Integer representing the number of data types carried in the message
DataType_id	Integer	Integer representing the type of the respective data
DataValue	Integer	Value on the data
ConfParam_id	String/ integer	Name of the parameter to be configured or an integer representing the respective parameter
NewParamValue	integer	New value of the parameter
NConfParam	integer	Number of the configuration parameters
ParamValue	integer	value of the parameter
Customer_id	Integer	Integer representing the id of the customer
custdata_id	String/ integer	Name of the under question customer data or an integer representing the respective customer data
custdata_value	integer	Value of the under question customer data
QoSValue	Integer	Integer expressing the quality of the offered service

UC3 - Dynamic Virtualization and Migration Contents and Servers

Interfaces and potential content exchanges between the elements. Having mapped the functionalities to the network topology and using the way that the black boxes interact (which black box sends information to which and where do the inputs of the black box originate from, inputs, outputs) are sufficient for the identification of the first interfaces and content exchanges between the elements. Table 9 depicts the messages that are exchanged between the functional blocks. In Table 10 the main parameters of UC3 are indicated.

Table 9. Main messages of UC3

Purpose	Source	Destination	Name
Information about user preferences	Monitoring(@EMS)	Information&KnowledgeBuilding(@NMS)	UserData
Information about areas characteristics	Monitoring(@EMS)	Information&KnowledgeBuilding(@EMS)	AreaMeasures
Sends current context	Monitoring(@NMS)	Situation Analysis and Diagnosis (@NMS)	ContextNotification
Sends users profile information	Profiles&Models(@NMS)	SolutionSelection&Elaboration(@NMS)	UserProfileData
Sends information about capabilities of network elements	Monitoring(@NMS)	SolutionSelection&Elaboration(@NMS)	ElementCapData
Sends network measurements	Monitoring(@EMS)	SolutionSelection&Elaboration(@NMS)	NetworkMeasureData
Sends reconfiguration actions	SolutionSelection&Elaboration(@NMS)	SolutionEvaluationAssessment(@NMS)	ReconfigurationData

Sends service policies	PolicyDerivation&Management (@NMS)	SolutionEvaluationAssessment(@NMS)	ServicePolicyData
Sends conflict-free reconfiguration actions	SolutionSelection&Elaboration(@NMS)	ConfigurationEnforcement(@EMS)	ReconfigurationData

Table 10. Main parameters of UC3

Message/Parameter name	Type	Description
ServiceMeasuresData		
servicesInfo	ServiceMeasures array of	class describing service measurements
ServiceMeasures		
serviceMeasuresID	integer	unique ID of measurement
serviceID	integer	ID of the service that is measured
e2eServiceLatency	double	e2e service latency
accessFrequency	integer	access frequency to this service
concurrentUsers	integer	number of concurrent users to be served
NetworkMeasures		
networkRssi	double	Received Signal Strength Indicator is a measurement of the power present in a received radio signal
networkRtt	long	Round-Trip Time is the length of time it takes for a signal to be sent plus the length of time it takes for an acknowledgment of that signal to be received
networkBer	double	Bit Error Rate is the number of received bits of a data stream over a communication channel that have been altered due to noise, interference, distortion or bit synchronization errors
NetworkMeasureData		
networkMeasureInfo	NetworkMeasures	class describing network measurements
UserData		
userInfo	User array of	class describing a user
User		
userID	integer	unique ID of user
userProfileID	integer	ID of the user profile
userCurrentQos	double	user's current QoS
userLocationX	integer	user's X coordinate
userLocationY	integer	user's Y coordinate
userSpeed	double	user's speed
userPlatform	string	user's platform
userAssignedGW	ElementConfig	assigned GW
userSubscriptions	ServiceProfile array of	user's subscriptions

areaPreference	AreaMeasures	class describing the areas that the user has been to
ServicePolicy		
serviceID	integer	unique ID of service
serviceName	string	service name
serviceDelay	double	minimum delay required by service
serviceJitter	double	minimum jitter required by service
servicePER	double	minimum packet error rate required by service
ServicePolicyData		
serviceInfo	ServicePolicy array of	class describing service profiles
UserProfile		
userProfileID	integer	unique ID of user profile
userProfileName	string	profile name
userQos	double	maximum QoS for this user profile
UserProfileData		
userProfileInfo	UserProfile array of	class describing user profiles
ElementCap		
elementID	integer	unique ID for a network element
elementMaxStorage	long	storage capacity of element
elementEnergyConsumption	integer	energy consumption of element
elementMaxTransmission	integer	maximum transmission of element
ElementCapData		
elementCapInfo	ElementCap array of	class describing capabilities of a network element
ElementConfig		
elementID	integer	unique ID of the network element
elementLocationX	integer	X coordinate of element
elementLocationY	integer	Y coordinate of element
elementStorageRate	double	the percentage of the storage capacity that is used
elementTransmissionRate	double	the percentage of the element's maximum transmission that is used
ElementData		
elementInfo	ElementConfig array of	class describing the elements' configuration
ContextNotification		
userDataInfo	UserData	class describing users
serviceDataInfo	ServiceMeasuresData	class describing service measurements
elementDataInfo	ElementData	class describing network elements' configuration
ReconfigurationData		
reconfigInfo	ElementConfig array of	class describing network elements configuration
AreaData		

areaID	integer	unique ID of the area
areaFrequency	integer	how many times the area was visited
areaTimeVisited	time array of	the time that the area was visited
AreaConfig		
areaID	integer	unique ID of an area
areaName	string	area name
areaLocationX	integer	X coordinate of the area
areaLocationY	integer	Y coordinate of the area
areaRadius	integer	radius of the area
AreaMeasures		
areaInfo	AreaData array of	class describing area measurements

UC4 - SON and SON Collaboration According to Operator Policies

Finally, in Figure 44, an instantiation of UC4 is depicted. The triggering event consists of operator goals, which are inserted via a H2N tool (Governance FG) at an enhanced OSS (U_Node), perhaps after a Violation_Notification from Intelligence FG (Solution Evaluation/Assessment FB) at U_FC@NMS. The GoalsProvision primitive carries these operator goals to the Knowledge Management FG (Situation Analysis/Diagnosis FB) at U_FC@NMS and the SON_Determination primitive informs Governance FG (Policy Derivation and Management FB) at U_FC@NMS about the involved SON entities. Then, Governance FG triggers the Intelligence FG (Solution Selection and Elaboration FB) either at U_FC@NMS as an offline process or at U_DC@eNB as an online process with SON-specific policies through PolicyProvision primitive. At this point of time, the SON coordination takes place via control loops and conflicts are resolved based on the provided policies. The parameters are also configured in a self and automatic way. Then, Intelligence FG notifies Knowledge Management FG (Monitoring FB) at U_DC@eNB for the metrics to be monitored via KPI_Determination. Knowledge Management FG at U_DC@eNB reports to Knowledge Management FG (Monitoring FB) at U_FC@NMS the monitoring results (KPIs) periodically through KPI_Information and Knowledge Management FG at U_FC@NMS sends KPI information to Intelligence FG (Solution Evaluation/Assessment FB) at U_FC@NMS either periodically (KPI_Information) or when there is KPI violation (KPI_Violation). Finally, the operator is informed through H2N tool (Governance FG) at U_Node@OSS via a Violation_Notification message about a KPI violation.

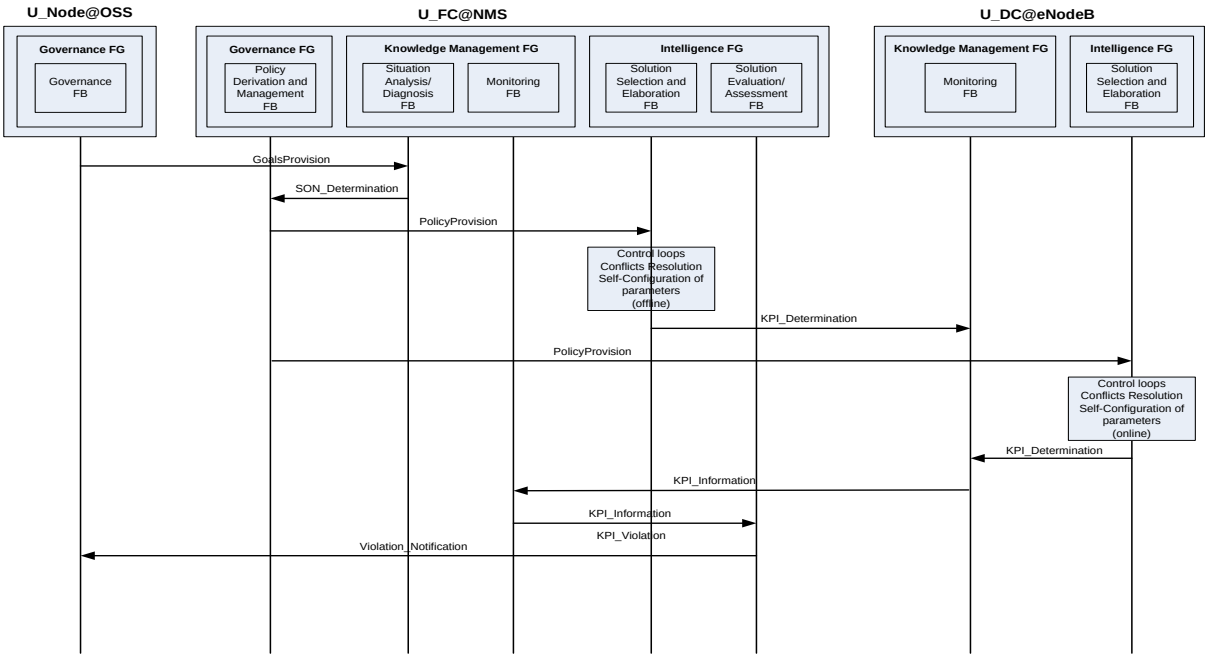


Figure 44. Example of MSC for UC4.

Having mapped the functional groups to the network topology and UMF components, the primitives exchanged and the relative parameters are depicted in the two following tables. In the table above, in Source and Destination columns, the relative FBs inside the FG are used, in order to show a further level of detail in the interactions that take place, e.g. inside the FG.

Table 11. Table of Primitives for UC4

Source	Destination	Primitive	Parameters	Scope
Governance @U_Node @OSS	Situation Analysis/Diagnosis@ U_FC@NMS	GoalsProvision	List of Business_Goals	Provides the operator goals
Situation Analysis/Diagnosis@ U_FC@NMS	Policy Derivation and Management@ U_FC@NMS	SON_Determination	List of SON_alg_ID List of Phy_ID List of NM_ID List of Business_Goals	Identifies the involved SON entities and their location
Policy Derivation and Management@ U_FC@NMS	Solution Selection and Elaboration @U_FC@NMS (offline mode) @U_DC@eNB (online mode)	PolicyProvision	List of SON_alg_ID List of Phy_ID List of NM_ID List of { Metric_id RelOper_id Threshold}	Provides SON-specific policies and triggers offline or/and online SON coordination
Solution Selection and Elaboration @U_FC@NMS (offline mode) @U_DC@eNB (online	Monitoring@ U_DC@eNB	KPI_Determination	List of { Metric_id RelOper_id Threshold}	Provides the KPIs to be monitored

mode)				
Monitoring@ U_FC@NMS	Solution Evaluation/Assessment@ U_FC@NMS	KPI_Violation	(metrics which violate thresholds) List of { Metric_ID Value Threshold}	Notifies about violated KPIs
Monitoring@ U_DC@eNB	Monitoring@ U_FC@NMS	KPI_Information	List of { Metric_ID Value Threshold}	Transfers KPIs measurements
Monitoring@ U_FC@NMS	Solution Evaluation/Assessment@ U_FC@NMS	KPI_Information	List of { Metric_ID Value Threshold}	Informs about KPIs values periodically
Solution Evaluation/Assessment @U_FC@NMS	Governance@U_Node@OSS	Violation_Notification	(metrics which require action) List of Metric_ID (to be completed)	Notifies about a violation and prompts for operator action

Table 12. Table of Parameters for UC4

Parameter	Type	Length	Value	Description
SON_alg_ID	Integer	1 byte	1-9 1: Coverage and Capacity Optimization 2: Energy Savings 3: Interference Reduction 4: Automated Configuration of Phy_ID 5: Mobility Robustness Optimization 6: Mobility Load Balancing Optimization 7: RACH Optimization 8: Automated Neighbour Relation 9: Inter-Cell Interference Coordination	This parameter identifies the SON algorithm
Phy_ID	Integer	4 bytes	$1\text{-}2^{32}$	This parameter identifies the eNB by identifying the corresponding cell's physical ID (L1 cell identifier)
NM_ID	Integer	4 bytes	$1\text{-}2^{32}$	This parameter identifies the corresponding Network Manager (U_FC)

List of Business_Goals	List of composite types	Business Level Entry (num_of_users, traffic_percentage, location, time_zone, application, user_class, quality_level, quality_level_parameters, mobility_pattern) and Association Notification (application, user class, quality_level, quality_level_parameters) as depicted in H2N ontology (to be elaborated)		
Metric_id	Integer	1 byte	1-2^8 1: Spectral efficiency 2: Call Blocking Probability 3: HO Blocking Probability 4: Ping-pong Probability 5: HO Failure rate 6: Num of successful outgoing HOs 7: User Throughput 8: Cell Throughput 9: Cell Load 10: Call Dropping Probability 11: Inter-cell interference (To be completed)	This parameter identifies the metric
RelOper_id	Integer	1 byte	1-5 1: equals 2: greater 3: less 4: lessequal 5: greaterequal	This parameter identifies the relational operator of the policy expression
Threshold	Integer	4 bytes	1-2^{32}	This parameter identifies the right part of the policy expression
Value	Integer	4 bytes	1-2^{32}	This parameter identifies the value of a metric

UC6 - Operator-Governed, End-to-End, Autonomic, Joint Network and Service Management

An instantiation of UC6 is depicted in the message sequence chart of Figure 45 below.

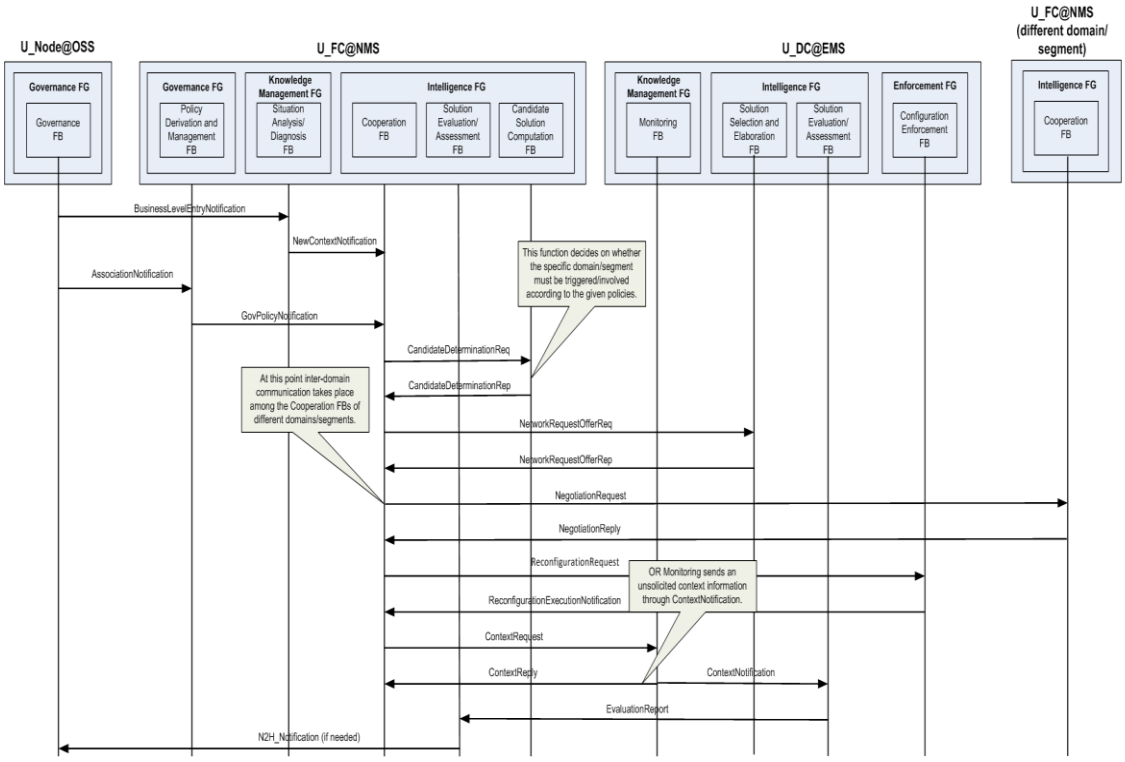


Figure 45. Example of MSC for UC6.

In addition and in order to capture the requirements concerning the communication (interfaces) between the various functional blocks, Table 13 and Table 14 below a) provide information on which functional blocks need to communicate between each other, b) give a summary of the main messages that need to be conveyed among these functional blocks for realizing the use case, as well as c) identify what information (i.e. parameters) needs to be exchanged.

Table 13. Main messages of UC6

Message purpose	*Source (FB@SystemEntity)*	*Destination (FB@SystemEntity)*	*Message Name*
Notification on the number of new users to be served (per Application/User Class/ Location/Time etc)	Governance (U_Node@OSS)	Situation Analysis/ Diagnosis (U_FC@NMS)	BusinessLevelEntryNotification
Update the association of Applications to User Classes and Quality Levels	Governance (U_Node @OSS)	Policy Derivation and Management (U_FC @NMS)	AssociationNotification
Update the policies that have to be followed when taking management decisions	Policy Derivation and Management (U_FC @NMS)	Cooperation (U_FC @NMS)	GovPolicyNotification
Notification of the number of new users to be served (per Application/User Class/Cell)	Situation Analysis/ Diagnosis (U_FC @NMS)	Cooperation (U_FC @NMS)	NewContextNotification
Request to decide on whether one domain/ segment must be triggered/involved according to the given	Cooperation (U_FC @NMS)	Candidate Solution Computation (U_FC @NMS)	CandidateDeterminationReq

policies			
Reply on whether one domain/ segment must be triggered/involved according to the given policies	Candidate Solution Computation (U_FC @NMS)	Cooperation (U_FC @NMS)	CandidateDeterminationRep
Requirements (traffic and mobility) addressed to candidate network	Cooperation (U_FC @NMS)	Solution Selection and Elaboration (U_DC @EMS)	NetworkRequestOfferReq
Reply to the requirements (traffic and mobility) addressed to candidate network	Solution Selection and Elaboration (U_DC @EMS)	Cooperation (U_FC @NMS)	NetworkRequestOfferRep
Cooperation request among different domains/segments	Cooperation (U_FC @NMS)	Cooperation (U_FC @NMS) (different domain/segment)	NegotiatonRequest
Cooperation reply among different domains/segments	Cooperation (U_FC @NMS)	Cooperation (U_FC @NMS) (different domain/segment)	NegotiatonReply
Apply new configuration according to decision	Cooperation (U_FC @NMS)	Configuration Enforcement (U_DC @EMS)	ReconfigurationRequest
Notify on new configuration application	Configuration Enforcement (U_DC @EMS)	Cooperation (U_FC @NMS)	ReconfigurationExecutionNotification
Request context information	Cooperation (U_FC @NMS)	Monitoring (U_DC @EMS)	ContextRequest
Send requested context information	Monitoring (U_DC @EMS)	Cooperation (U_FC @NMS)	ContextReply
Send unsolicited context information	Monitoring (U_DC @EMS)	Cooperation (U_FC @NMS)	ContextNotification
	Monitoring (U_DC @EMS)	Solution Evaluation/Assessment (U_DC @EMS)	
Report about evaluation results	Solution Evaluation/Assessment (U_DC @EMS)	Solution Evaluation/Assessment (U_FC @NMS)	EvaluationReport
Notification to the operator/administrator	Solution Evaluation/Assessment (U_FC @NMS)	Governance (U_Node@OSS)	N2H_Notification

Table 14. Main Parameters of UC6

Message / Parameter Name	Type	Description
AssociationNotification		
GovApplicationInfo	GovApplication array of	class describing the application
BusinessLevelEntryNotification		
BusinessLevelEntryInfo	BusinessLevelEntry	class describing new traffic demand to be served
BusinessLevelEntry		
NumberOfUsers	integer	total number of users
TrafficPercentage	integer	percentage of concurrent active users

GovLocationInfo	GovLocation	class describing the location of the users
GovTimezoneInfo	GovTimezone	class describing the time zone when the users will appear
GovApplicationInfo	GovApplication array of	class describing the application
GovLocation		
GovLocationID	integer	unique ID of the location area
GovLocationName	string	human friendly name of this location area
GovLocationXcoords	integer	X coordinate of the location area's center
GovLocationYcoords	integer	Y coordinate of the location area's center
Range	integer	range of the location area
GovTimezone		
TimezoneName	string	human friendly name of this timezone
StartTime	string	formatted representation of start time
EndTime	string	formatted representation of end time
GovApplication		
GovApplicationID	integer	unique ID of the application
GovApplicationName	string	human friendly name of the application
GovUserClassInfo	GovUserClass array of	class describing the user class of the application
GovApplicationPriority	integer	priority of this application comparing to the others
GovApplicationRATInfo	RAT array of	list of preferred RATs for the provision of this application
GovUserClass		
GovUserClassID	integer	unique ID of the user class
GovUserClassName	string	human friendly name of the user class
GovMobilityInfo	GovMobility array of	class describing the mobility pattern of this user class
GovQualityLevelInfo	GovQualityLevel array of	class describing the allowed quality levels for this user class, used only in an AssociationNotification message class
GovMobility		
GovMobilityID	Integer	unique ID of the mobility pattern
GovMobilityName	String	human friendly name of this mobility pattern
GovMobilityType	Integer	ID declaring one of the predefined mobility pattern types
GovQualityLevel		
GovQualityLevelID	Integer	unique ID of the quality level
GovQualityLevelName	String	human friendly name of this quality level
GovQualityLevelParamInfo	GovQualityLevelParam array of	class describing a generic parameter
GovQualityLevelParam		

ParamID	Integer	unique ID of the parameter
ParamName	String	human friendly name of this parameter
ParamReferenceValues	integer array of	reference values of this parameter
ParamUnit	String	measurement unit of this parameter
GovPolicyNotification		
GovApplicationInfo	GovApplication array of	class describing the application
NewContextNotification		
BaseStationContextInfo	BaseStationContext array of	class describing the new traffic demand to be served by the BS
ContextNotification		
BaseStationContextInfo	BaseStationContext	class describing the current context of the BS
BaseStationContext		
BaseStationID	Integer	unique ID of the base station
TRXContextInfo	TRXContext array of	class describing the current context of a specific transceiver of this base station
TRXContext		
TRXID	Integer	unique ID of this transceiver
AggregateLoad	Integer	the aggregate load of this TRX in kbps
ApplicationLoadInfo	ApplicationLoad array of	class describing the current load caused by a application
TransmissionPower	Double	transmission power of this transceiver
ComputingLoadPercent	Integer	percentage of the used computational power
EnergyConsumption	Double	indicator of the consumed energy with the current load and configuration
ApplicationLoad		
ApplicationInfo	Application	describing the application
UserClassLoadInfo	UserClassLoad	load caused by a user class of this application
Application		
ApplicationID	Integer	unique ID of the application
ApplicationName	String	human friendly name of this application
QualityLevelInfo	QualityLevel array of	acceptable quality levels for this application
QualityLevel		
QualityLevelID	Integer	unique ID of the quality of application level
QLavailability	Double	probability that this Quality Level will be available
QLreliability	Double	probability that this QL will be maintained as long as necessary
QLperformance	Double	indicator of how well this QL will serve the

		corresponding application
UserClassLoad		
UserClassInfo	UserClass	class describing the user class
QualityLevelLoadInfo	QualityLevelLoad array of	class describing the current load caused due to the provision of the application to the users of this user class at the specific quality level
UCmobility	Double	indicator of the behaviour of this user class when using this application
UserClass		
UserClassID	Integer	unique ID of the user class
UserClassName	String	human friendly name of this user class
QualityLevelLoad		
QualityLevelInfo	QualityLevel	class describing the quality level
UserInfo	User array of	class describing the user belonging to the user class and receiving the application at a specific quality level
User		
UserID	Integer	unique ID of the user
ContextResponse		
BaseStationContextInfo	BaseStationContext	describing the current context of the BS
ContextRequest		
BaseStationID	Integer	unique ID of the base station whose context is requested
ReconfigurationRequest		
BaseStationConfigInfo	BaseStationConfig	class describing the current base station configuration
BaseStationContextInfo	BaseStationContext	class describing the current context of the BS
BaseStationConfig		
BaseStationID	Integer	unique ID of the base station
TRXConfigInfo	TRXConfig array of	class describing the current configuration of a specific transceiver of this base station
TRXConfig		
TRXID	Integer	unique ID of this transceiver
Range	Integer	the range of the area covered by this transceiver
AllocatedDownlinkBandwidth	Integer	total allocated bandwidth in the downlink direction
AllocatedUplinkBandwidth	Integer	the total allocated bandwidth in the uplink direction

RATInfo	RAT	the Radio Access Technology in which this transceiver operates
FrequencyInfo	Frequency	the frequency range which is used by this transceiver
ApplicationInfo	Application array of	the supported applications on this transceiver
RAT		
RATID	Integer	unique ID of the RAT
RATname	String	human friendly name of this RAT
FrequencyInfo	Frequency array of	supported frequencies when operating in this RAT
ParamConfigInfo	ParamConfig array of	class describing a generic parameter
Frequency		
FreqBand	String	frequency band
ParamConfig		
ParamName	String	name of the parameter
ParamValue	String	value of the parameter

UC7 - Network and Services Governance

Based on the mapping of the functional blocks to the network layout, Figure 46 describes the interactions between the entities (segments) of the network (where black box functionality and functional block instances are active). The figure tries to depicture the two different network configurations that use case 7 will deal with: fixed FTTH-based and ADSL-based wireless access networks.

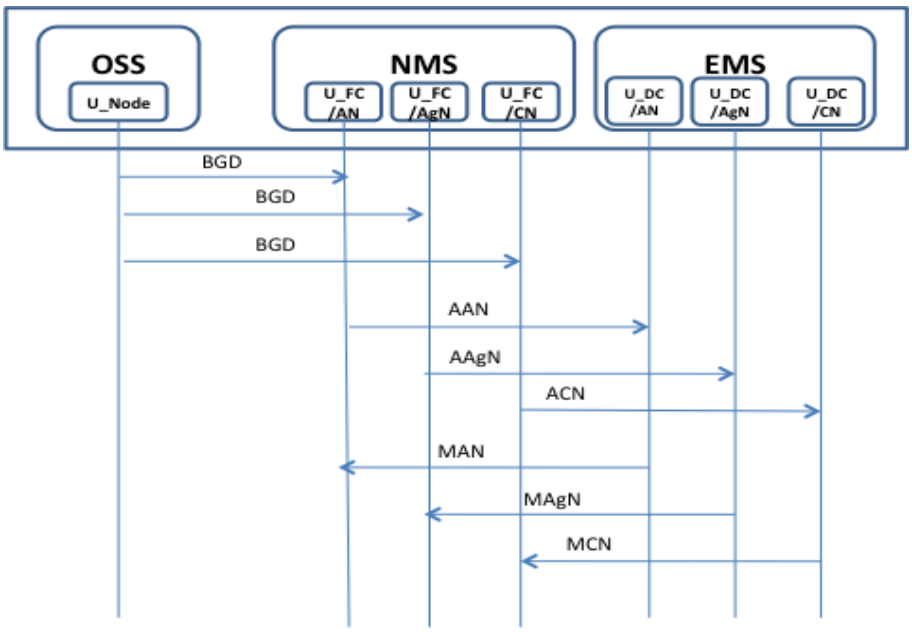


Figure 46. Message passing between the elements/ functional blocks (NMS – U_FC/AN: controller for Access Network, U_FC/AgN: controller for Aggregation Network, U_FC/CN: controller for Core Network, EMS – U_DC/AN: Access Network, U_DC/AgN: Aggregation Network, U_DC/CN: Core Network).

We describe in detail the visualized messages communicated between the elements of the OSS, NMS and EMS and their main parameters:

- BGD – Business Goals Definition:
- AAN - Adapt Access Network segments:
- AAgN - Adapt Aggregation Network segments:
- ACAN - Adapt Core Network segments:
- MAN - Monitoring data for Access Networks:
- MAgN - Monitoring data for Aggregation Network:
- MCN - Monitoring data for Core Network

Table 15. Detailed description of the message parameters

Message / Parameter Name	Type	Description
AssociationNotification		
GovApplicationInfo	GovApplication array of	class describing the application
BusinessLevelEntryNotification		
BusinessLevelEntryInfo	BusinessLevelEntry	class describing new traffic demand to be served
BusinessLevelEntry		
NumberOfUsers	integer	total number of users
TrafficPercentage	integer	percentage of concurrent active users
GovLocationInfo	GovLocation	class describing the location of the users
GovTimezoneInfo	GovTimezone	class describing the time zone when the users will appear
GovApplicationInfo	GovApplication array of	class describing the application
GovLocation		
GovLocationID	integer	unique ID of the location area
GovLocationName	string	human friendly name of this location area
GovLocationXcoords	integer	X coordinate of the location area's center
GovLocationYcoords	integer	Y coordinate of the location area's center
Range	integer	range of the location area
GovTimezone		
TimezoneName	string	human friendly name of this timezone
StartTime	string	formatted representation of start time
EndTime	string	formatted representation of end time
GovApplication		
GovApplicationID	integer	unique ID of the application
GovApplicationName	string	human friendly name of the application
GovUserClassInfo	GovUserClass array of	class describing the user class of the application
GovApplicationPriority	integer	priority of this application comparing to the

		others
GovApplicationRATInfo	RAT array of	list of preferred RATs for the provision of this application
GovUserClass		
GovUserClassID	integer	unique ID of the user class
GovUserClassName	string	human friendly name of the user class
GovMobilityInfo	GovMobility array of	class describing the mobility pattern of this user class
GovQualityLevelInfo	GovQualityLevel array of	class describing the allowed quality levels for this user class, used only in an AssociationNotification message class
GovMobility		
GovMobilityID	integer	unique ID of the mobility pattern
GovMobilityName	string	human friendly name of this mobility pattern
GovMobilityType	integer	ID declaring one of the predefined mobility pattern types
GovQualityLevel		
GovQualityLevelID	integer	unique ID of the quality level
GovQualityLevelName	string	human friendly name of this quality level
GovQualityLevelParamInfo	GovQualityLevelParam array of	class describing a generic parameter
GovQualityLevelParam		
ParamID	integer	unique ID of the parameter
ParamName	string	human friendly name of this parameter
ParamReferenceValues	integer array of	reference values of this parameter
ParamUnit	string	measurement unit of this parameter
GovPolicyNotification		
GovApplicationInfo	GovApplication array of	class describing the application
NewContextNotification		
BaseStationContextInfo	BaseStationContext array of	class describing the new traffic demand to be served by the BS
ContextNotification		
BaseStationContextInfo	BaseStationContext	class describing the current context of the BS
BaseStationContext		
BaseStationID	integer	unique ID of the base station
TRXContextInfo	TRXContext array of	class describing the current context of a specific transceiver of this base station
TRXContext		
TRXID	integer	unique ID of this transceiver

AggregateLoad	integer	the aggregate load of this TRX in kbps
ApplicationLoadInfo	ApplicationLoad array of	class describing the current load caused by a application
TransmissionPower	double	transmission power of this transceiver
ComputingLoadPercent	integer	percentage of the used computational power
EnergyConsumption	double	indicator of the consumed energy with the current load and configuration
ApplicationLoad		
ApplicationInfo	Application	describing the application
UserClassLoadInfo	UserClassLoad	load caused by a user class of this application
Application		
ApplicationID	integer	unique ID of the application
ApplicationName	string	human friendly name of this application
QualityLevelInfo	QualityLevel array of	acceptable quality levels for this application
QualityLevel		
QualityLevelID	integer	unique ID of the quality of application level
QLavailability	double	probability that this Quality Level will be available
QLreliability	double	probability that this QL will be maintained as long as necessary
QLperformance	double	indicator of how well this QL will serve the corresponding application
UserClassLoad		
UserClassInfo	UserClass	class describing the user class
QualityLevelLoadInfo	QualityLevelLoad array of	class describing the current load caused due to the provision of the application to the users of this user class at the specific quality level
UCmobility	double	indicator of the behaviour of this user class when using this application
UserClass		
UserClassID	integer	unique ID of the user class
UserClassName	string	human friendly name of this user class
QualityLevelLoad		
QualityLevelInfo	QualityLevel	class describing the quality level
UserInfo	User array of	class describing the user belonging to the user class and receiving the application at a specific quality level
User		
UserID	integer	unique ID of the user

ContextResponse		
BaseStationContextInfo	BaseStationContext	describing the current context of the BS
ContextRequest		
BaseStationID	integer	unique ID of the base station whose context is requested
ReconfigurationRequest		
BaseStationConfigInfo	BaseStationConfig	class describing the new base station configuration
BaseStationContextInfo	BaseStationContext	class describing the new context of the BS
BaseStationConfig		
BaseStationID	integer	unique ID of the base station
TRXConfigInfo	TRXConfig array of	class describing the current configuration of a specific transceiver of this base station
TRXConfig		
TRXID	integer	unique ID of this transceiver
Range	integer	the range of the area covered by this transceiver
AllocatedDownlinkBandwidth	integer	total allocated bandwidth in the downlink direction
AllocatedUplinkBandwidth	integer	the total allocated bandwidth in the uplink direction
RATInfo	RAT	the Radio Access Technology in which this transceiver operates
FrequencyInfo	Frequency	the frequency range which is used by this transceiver
ApplicationInfo	Application array of	the supported applications on this transceiver
RAT		
RATID	integer	unique ID of the RAT
RATname	string	human friendly name of this RAT
FrequencyInfo	Frequency array of	supported frequencies when operating in this RAT
ParamConfigInfo	ParamConfig array of	class describing a generic parameter
Frequency		
FreqBand	string	frequency band
ParamConfig		
ParamName	string	name of the parameter
ParamValue	string	value of the parameter
ElementCapData		
elementCapInfo	ElementCap array of	class describing capabilities of a network element

Parameter types

The following table provides a more detailed description of the messages and the parameters that are conveyed in these messages.

Table 16. Tables of Parameters of Consolidated Messages

Message / Parameter Name	Type	Description
AssociationNotification		
GovApplicationInfo	GovApplication array of	class describing the application
BusinessLevelEntryNotification		
BusinessLevelEntryInfo	BusinessLevelEntry	class describing new traffic demand to be served
BusinessLevelEntry		
NumberOfUsers	integer	total number of users
TrafficPercentage	integer	percentage of concurrent active users
GovLocationInfo	GovLocation	class describing the location of the users
GovTimezoneInfo	GovTimezone	class describing the time zone when the users will appear
GovApplicationInfo	GovApplication array of	class describing the application
GovLocation		
GovLocationID	integer	unique ID of the location area
GovLocationName	string	human friendly name of this location area
GovLocationXcoords	integer	X coordinate of the location area's center
GovLocationYcoords	integer	Y coordinate of the location area's center
Range	integer	range of the location area
GovTimezone		
TimezoneName	string	human friendly name of this timezone
StartTime	string	formatted representation of start time
EndTime	string	formatted representation of end time
GovApplication		
GovApplicationID	integer	unique ID of the application
GovApplicationName	string	human friendly name of the application
GovUserClassInfo	GovUserClass array of	class describing the user class of the application
GovApplicationPriority	integer	priority of this application comparing to the others
GovApplicationRATInfo	RAT array of	list of preferred RATs for the provision of this application
GovUserClass		
GovUserClassID	integer	unique ID of the user class

GovUserClassName	string	human friendly name of the user class
GovMobilityInfo	GovMobility array of	class describing the mobility pattern of this user class
GovQualityLevelInfo	GovQualityLevel array of	class describing the allowed quality levels for this user class, used only in an AssociationNotification message class
GovMobility		
GovMobilityID	integer	unique ID of the mobility pattern
GovMobilityName	string	human friendly name of this mobility pattern
GovMobilityType	integer	ID declaring one of the predefined mobility pattern types
GovQualityLevel		
GovQualityLevelID	integer	unique ID of the quality level
GovQualityLevelName	string	human friendly name of this quality level
GovQualityLevelParamInfo	GovQualityLevelParam array of	class describing a generic parameter
GovQualityLevelParam		
ParamID	integer	unique ID of the parameter
ParamName	string	human friendly name of this parameter
ParamReferenceValues	integer array of	reference values of this parameter
ParamUnit	string	measurement unit of this parameter
GovPolicyNotification		
GovApplicationInfo	GovApplication array of	class describing the application
SONPolicyNotification		
SON_alg_ID	integer array of	this parameter identifies the SON algorithm
Phy_ID	integer array of	this parameter identifies the eNB by identifying the corresponding cell's physical ID (L1 cell identifier)
NM_ID	integer array of	this parameter identifies the corresponding Network Manager (NM)
ParamConfigInfo	ParamConfig array of	class describing a generic parameter
ApplicationPolicyNotification		
ApplicationPolicyInfo	ApplicationPolicy array of	class describing application policies
ApplicationPolicy		
ApplicationID	integer	unique ID of the application
ApplicationName	string	human friendly name of the application
ParamConfigInfo	ParamConfig array of	class describing a generic parameter
NewContextNotification		
BaseStationContextInfo	BaseStationContext array of	class describing the new traffic demand to be served by the BS

SONDetermination		
SON_alg_ID	integer array of	this parameter identifies the SON algorithm
Phy_ID	integer array of	this parameter identifies the eNB by identifying the corresponding cell's physical ID (L1 cell identifier)
NM_ID	integer array of	this parameter identifies the corresponding Network Manager (NM)
BusinessLevelEntryInfo	BusinessLevelEntry array of	class describing new traffic demand to be served
ContextNotification		
BaseStationContextInfo	BaseStationContext	class describing the current context of the BS
BaseStationContext		
BaseStationID	integer	unique ID of the base station
TRXContextInfo	TRXContext array of	class describing the current context of a specific transceiver of this base station
TRXContext		
TRXID	integer	unique ID of this transceiver
AggregateLoad	integer	the aggregate load of this TRX in kbps
ApplicationLoadInfo	ApplicationLoad array of	class describing the current load caused by a application
TransmissionPower	double	transmission power of this transceiver
ComputingLoadPercent	integer	percentage of the used computational power
EnergyConsumption	double	indicator of the consumed energy with the current load and configuration
NetworkRSSI	double	Received Signal Strength Indicator is a measurement of the power present in a received radio signal
NetworkRTT	long	Round-Trip Time is the length of time it takes for a signal to be sent plus the length of time it takes for an acknowledgment of that signal to be received
NetworkBER	double	Bit Error Rate is the number of received bits of a data stream over a communication channel that have been altered due to noise, interference, distortion or bit synchronization errors
ApplicationLoad		
ApplicationInfo	Application	describing the application
UserClassLoadInfo	UserClassLoad	load caused by a user class of this application
e2eApplicationLatency	double	e2e application latency
concurrentUsers	integer	number of concurrent users to be served
Application		
ApplicationID	integer	unique ID of the application

ApplicationName	string	human friendly name of this application
QualityLevelInfo	QualityLevel array of	acceptable quality levels for this application
QualityLevel		
QualityLevelID	integer	unique ID of the quality of application level
QLavailability	double	probability that this Quality Level will be available
QLreliability	double	probability that this QL will be maintained as long as necessary
QLperformance	double	indicator of how well this QL will serve the corresponding application
UserClassLoad		
UserClassInfo	UserClass	class describing the user class
QualityLevelLoadInfo	QualityLevelLoad array of	class describing the current load caused due to the provision of the application to the users of this user class at the specific quality level
UCmobility	double	indicator of the behaviour of this user class when using this application
UserClass		
UserClassID	integer	unique ID of the user class
UserClassName	string	human friendly name of this user class
QualityLevelLoad		
QualityLevelInfo	QualityLevel	class describing the quality level
UserInfo	User array of	class describing the user belonging to the user class and receiving the application at a specific quality level
User		
UserID	integer	unique ID of the user
xGeographicalCoordinate	string	parameter indicating user's location
yGeographicalCoordinate	string	parameter indicating user's location
UserSpeed	double	user's speed
UserPlatform	string	user's platform
UserAssignedElementID	integer	assigned element (GW)
UserProfileInfo	UserProfile array of	class describing this user's profile
UserProfile		
UserID	integer	unique ID of the user
NetworkOperatorID	integer	unique ID of the network operator
UserProfileID	integer	unique ID of this profile
UserProfileName	string	human friendly name of this profile
UserClassInfo	UserClass	class describing the user class in which this user belongs
UserApplicationInfo	Application array of	class describing the applications that are available when using this profile

UserAreaInfo	AreaData array of	class containing data about the areas that the user has been to
AreaData		
AreaID	integer	unique ID of the area
AreaPresenceCounter	integer	how many times the area was visited
AreaPresenceTime	string array of	the time periods that the area was visited
UserInfoNotification		
UserInfo	User array of	class containing user information
AreaInfoNotification		
AreaInfo	AreaData array of	class containing area information
UserProfileRequest		
UserID	integer	unique ID of the user
UserProfileResponse		
UserProfileInfo	UserProfile array of	class describing this user's profile
ContextResponse		
BaseStationContextInfo	BaseStationContext	describing the current context of the BS
ContextRequest		
BaseStationID	integer	unique ID of the base station whose context is requested
ReconfigurationRequest		
BaseStationConfigInfo	BaseStationConfig	class describing the new base station configuration
BaseStationContextInfo	BaseStationContext	class describing the new context of the BS
ReconfigurationExecutionNotification		
Result	integer	code indicating the reconfiguration result
BaseStationConfigInfo	BaseStationConfig	class describing the new base station configuration
BaseStationContextInfo	BaseStationContext	class describing the new context of the BS
BaseStationConfig		
BaseStationID	integer	unique ID of the base station
TRXConfigInfo	TRXConfig array of	class describing the current configuration of a specific transceiver of this base station
TRXConfig		
TRXID	integer	unique ID of this transceiver
Range	integer	the range of the area covered by this transceiver
AllocatedDownlinkBandwidth	integer	total allocated bandwidth in the downlink

		direction
AllocatedUplinkBandwidth	integer	the total allocated bandwidth in the uplink direction
RATInfo	RAT	the Radio Access Technology in which this transceiver operates
FrequencyInfo	Frequency	the frequency range which is used by this transceiver
ApplicationInfo	Application array of	the supported applications on this transceiver
RAT		
RATID	integer	unique ID of the RAT
RATname	string	human friendly name of this RAT
FrequencyInfo	Frequency array of	supported frequencies when operating in this RAT
ParamConfigInfo	ParamConfig array of	class describing a generic parameter
Frequency		
FreqBand	string	frequency band
ParamConfig		
ParamName	string	name of the parameter
ParamID	integer	ID of the parameter
ParamValue	string	value of the parameter
ParamRefValue	string	reference value(s) of the parameter
KPIDetermination		
ParamConfigInfo	ParamConfig array of	class describing a generic parameter
ViolationNotification		
ParamConfigInfo	ParamConfig array of	class describing a generic parameter
BaseStationProfile		
BaseStationID	integer	unique ID of the base station
TRXProfileInfo	TRXProfile array of	class describing the profile of a specific transceiver of this base station
xGeographicalCoordinate	string	parameter indicating BS location
yGeographicalCoordinate	string	parameter indicating BS location
TRXProfile		
TRXID	integer	unique ID of this transceiver
PossibleOperatingRATs	RAT array of	class describing the RATs in which this TRX can operate
PossibleOperatingFreqs	Frequency array of	class describing the Frequencies which this TRX is able to use
ElementProfile		
ElementID	integer	unique ID of this element
xGeographicalCoordinate	string	parameter indicating element's location

yGeographicalCoordinate	string	parameter indicating element's location
ElementMaxStorage	long	storage capacity of element
ElementMaxStorage	long	storage capacity of element
ElementMaxStorage	long	storage capacity of element
ElementEnergyConsumption	integer	energy consumption of element
ElementMaxTransmission	integer	maximum transmission of element
ElementConfig		
ElementID	integer	unique ID of this element
ElementStorageRate	double	the percentage of the storage capacity that is used
ElementTransmissionRate	double	the percentage of the element's maximum transmission that is used
Area		
AreaID	integer	unique ID of this area
AreaName	string	human friendly name of this area
xGeographicalCoordinate	string	parameter indicating area's location
yGeographicalCoordinate	string	parameter indicating area's location
AreaRadius	integer	radius of this area
LLDataProvision		
ParamConfigInfo	ParamConfig array of	class describing a generic parameter
HLDataProvision		
ParamConfigInfo	ParamConfig array of	class describing a generic parameter
DiagnosisProvision		
ParamConfigInfo	ParamConfig array of	class describing a generic parameter
CustomerReport		
ApplicationID	integer	unique ID of the application
MalfunctionID	integer	unique ID of the malfunction
QoEReport		
UserID	integer	unique ID of the user
QoEValue	integer	integer expressing customers' satisfaction by the service and the time needed for the healing
ModelProvision		
Model	Model array of	Models of • reparation/ mitigation plans • service • network • predefined faults

		• events • anomaly • normality KPIs
ElementCap		
elementID	integer	unique ID for a network element
elementMaxStorage	long	storage capacity of element
elementEnergyConsumption	integer	energy consumption of element
elementMaxTransmission	integer	maximum transmission of element
ElementCapData		
elementCapInfo	ElementCap array of	class describing capabilities of a network element

Annex C: Intelligence embodiment - State of the art

This section provides an overview of various research areas that address concepts and issues related to intelligence embodiment, the results of which can thus be utilised for the design specification of the corresponding intelligence embodiment mechanisms.

Ontologies and Semantics for description and discovery of intelligence components

Several definitions have been given for ontologies as stated in [45]; the most widely cited one defines ontology as a specification of a conceptualization [49]. Hendler in [50] enhances the above definition as following: "Ontology is a set of knowledge terms, including the vocabulary, the semantic interconnections, and some simple rules of inference and logic for some participants".

The important enhancements are the semantics interconnections inference and logic. Semantics interconnections mean that ontology specifies the meaning of relations between the used concepts. It can be also understood that ontologies can be interconnected. The inference and logic part means that ontologies enable some forms of reasoning.

Kalfoglou stresses another important issue related to ontologies: "ontology is an explicit representation of a shared understanding of the important concepts in some domain of interest" [51]. Shared means that ontology represents knowledge accepted by a group of community. Ontology cannot be a subjective knowledge of some individuals. The goals of the sharing aspects are the reusability of that knowledge which enables semantic interoperability between intelligent agents and applications.

To this effect, the use of ontologies enable intelligence embodiment by providing a common communication language that comprises the types of components that exist, their properties and relations.

The advantage of using ontology compared to Object oriented data models

In the telecommunications domain, the most widely used data models are Object Oriented. In this regard, this section aims to compare the structure of Object Oriented data models and ontologies. Ontologies are purely declarative. Declarative paradigm means the specification of what to be modelled or computed. In this sense, ontologies describe concepts and their relationships. They also have very similar approaches as the Object Oriented paradigm, for the declaration of static structures, such as classes (concepts), class hierarchies (using inheritance), attributes, relationships and instances. However, the significant differences are within the representations of semantics.

Ontologies use constraints (metadata on slots), axioms and rules whereas Object-Oriented models rely on methods (sequences of imperative commands). Both approaches are equally expressive. A declarative approach is then more suited for a domain or system modelling.

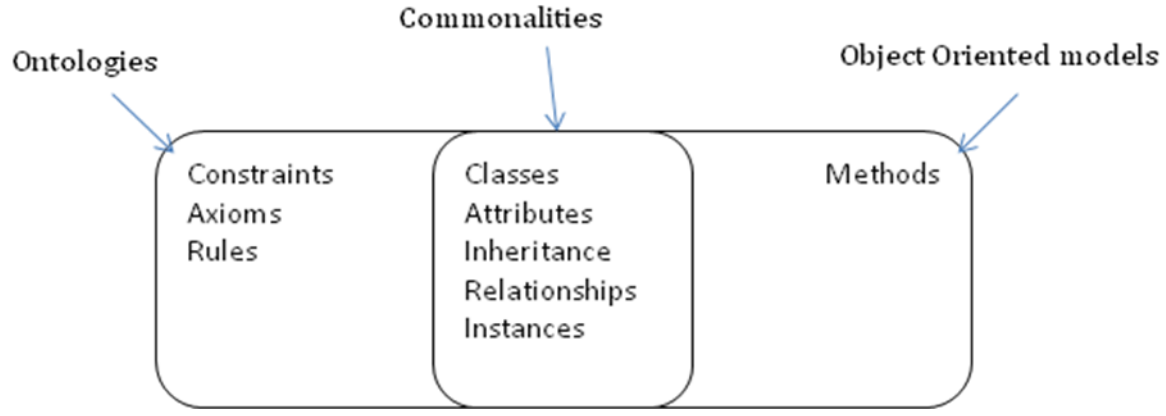


Figure 47. Comparative figure between Ontology and Object oriented models.

The Object Management Group in the Ontology Definition Metamodel (ODM) specification [52], explain the fact that: “The lack of reliable set semantics and model theory for Unified Modelling Languages prevents the

use of automated reasoner on UML models. Such a capability is important to applying Model Driven Architecture to systems integration. UML lacks a formal model theoretic semantics, OCL (Object Constraint Language) also has neither a formal model theory nor a formal proof theory, and thus cannot be used for automated reasoning (today)".

The following table, based on [53], compares Ontology Languages to Object Oriented Modelling (UML).

Table 17. Comparison of Ontology Languages to Object Oriented Modelling

Differences	Descriptions
Monotonic	Ontology languages are monotonic, whereas UML and Object Oriented languages are non monotonic. A system is monotonic if adding new facts never cause previous facts to be falsified.
Metalevels	Ontology does not have a rigid separation between metalevels. For example, in OWL full, an instance of class can be another class.
Modularity	Ontology languages do not have profiles, packages, or any other modularity mechanism supported by UML and Object oriented languages.
Cardinality Constraints	In ontology languages one can specify cardinality constraints for every domain of a property all at once whereas in UML this must specified separately for each association.
Sub properties	Ontology languages allow one property to be a sub-property of another. UML does not support ontology features, such as: sub properties

Tools towards building semantic or ontology based systems

In the last years, a high number of tools for ontology construction and ontology use have appeared. Tools are important both for the ontology development cycle (building, annotation, merge) and for the ontology usage within applications (Knowledge Management, Semantic Web).

Tools for ontologies could be browser, reasoner, language, store…A grouping of ontology tools has been proposed in [54].

- Ontology development tools
- Ontology merge and integration tools
- Ontology evaluation tools
- Ontology-based annotation tools
- Ontology storage and querying tools
- Ontology learning tools

Programmable networks/spaces

Programmable networks are networks that allow the functionality of some of their network elements to be programmable dynamically. These networks aim to provide easy introduction of new network services by adding dynamic programmability to network devices such as routers, switches, and applications servers.

Dynamic programming refers to executable code that is injected into the network element in order to create the new functionality at run time. The basic idea is to enable third parties (operators, and service providers) to inject application-specific services (in the form of code) into the network. Applications may utilize this network support in terms of optimized network resources and, as such, they are becoming network aware.

Programmable networks allow dynamic injection of code as a promising way of realizing application-specific service logic, or performing dynamic service provision on demand. As such, network programming provides unprecedented flexibility in telecommunications. However, viable architectures for programmable networks must be carefully engineered to achieve suitable trade-offs between flexibility, performance, security, and manageability. A comprehensive review of the programmable networks technology is provided by [55].

The programmable networks paradigm comprises mechanisms that allow for the automated configuration of the functionality and behaviour of network elements [55]. The aim is to facilitate and accelerate the creation and deployment of new network services, as well as the coexistence of various network architectures and provide operators better control of their networks [56]. The programmability of network elements is based on the separation of network infrastructure hardware (i.e., switching fabrics, routing engines) from control software [57]. In this sense, research efforts in the area of programmable networks can be exploited for the specification and design of intelligence embodiment features in the scope of the UMF. One of the most recent approaches in this direction is the notion of Software Defined Networking, promoted by a non profit organization, the Open Networking Foundation, comprising several big companies of the telecommunications market (including operators, manufacturers and software vendors).

Pervasive computing

Initially, efforts in the area of pervasive computing mainly focused on the integration of various devices in the environment of the user such as sensors, actuators and other computing devices. With the evolution of devices and networking technologies, the integration of new elements in a pervasive system in a more dynamic manner became an important issue, thus programmability is also relevant in pervasive spaces, so as to enable greater flexibility and dynamic introduction of infrastructure elements as well as intelligence [59], [60]. A lot of research and development effort in the area of pervasive computing and smart spaces aims to develop off-the-shelf technologies which users can install and use without requiring any assistance. Various technologies are employed for realising this aim, including service-oriented technologies such as OSGi. Approaches and findings of programming pervasive spaces can also be exploited for the specification of intelligence embodiment within the UMF.

Service-oriented computing

Service oriented computing is one of the most recent paradigms of distributed computing. In a service oriented world, as the name implies, everything revolves around services. Services are entities that can be described, discovered, used separately or combined with others to form composite services [61]. Therefore, service oriented computing inherently meets the requirements outlined in the previous sections, specifically for the abstraction (generic description) of self-x features/algorithms and autonomic capabilities, and presents an excellent approach for intelligence embodiment.

Software-as-a-Service

Software-as-a-Service (SaaS), also known as “software on demand”, refers to software being deployed over the internet instead of being installed on local end-user machines. It is therefore a multi-tenant platform in the sense that it uses common resources and a single instance of both the object code of an application as well as the underlying database to support multiple end-users simultaneously. Compared to the traditional model of locally installed software, from an end-user point of view, SaaS offers the benefit of requiring no local installation, saving therefore some cost on installation servers, and being accessible from anywhere as long as internet connectivity is available. From a provider’s point of view it offers cost savings, since the providers only need to maintain one instance of the program.

On the other hand, the deployment model of SaaS raises some issues with respect to privacy, performance and customization. Since the application is installed remotely and with the same instance of it serving multiple end-users, data security and privacy is a concern. Furthermore, performance issues are raised when low application

response times are necessary (e.g. in case of interactive applications). A single application instance serving multiple end-users also means that the application cannot be customized on a per-user basis.

Infrastructure-as-a-Service

Infrastructure as a Service (IaaS) is a provision model in which an organization outsources the equipment used to support operations, including storage, hardware, servers and networking components. In other words, IaaS refers to hardware components, such as storage capacity, CPU cycles, memory capacity, and network bandwidth and so on, being delivered as services over the internet. The service provider owns the equipment and is responsible for housing, running and maintaining it. IaaS providers offer hardware level services in the form of raw resource provisions, meaning that it is up to their customers to combine these raw resources in terms of their own needs. End-users/consumers have control over the operating system, storage, deployed applications, and possibly networking components (firewalls, load balancers) but not the underlying cloud infrastructure beneath them. Aside from higher flexibility, a key benefit of IaaS is the ability of employing a usage-based payment scheme. This allows customers of the offered resources to request resources, and pay, as they grow and as they need them minimizing the need for upfront investment. Another important advantage is that by using the latest technology, as offered by the IaaS providers, customers can achieve a much faster delivery time and time to market.

IaaS is one of three main categories of cloud computing service. The other two are Software as a Service (SaaS) and Platform as a Service (PaaS). While Network as a Service (NaaS) is also sometimes treated as a separate category, it can be regarded as a subcategory of IaaS focusing on the hardware components needed for network connectivity and provisioning of network resources.

Platform as a Service aims at providing a development environment as a service; that is the operating system and environment needed to develop applications. Compared to conventional application development this approach can reduce the development time by offering various readily available tools needed for application development.

With respect to IaaS, similar though to SaaS, security and performance issues are raised. For example when virtualization is used as the technology to isolate resources, attacks are possible that aim to exploit vulnerabilities in hypervisors and compromise virtual machine separation. Interoperability issues between different virtualization platforms also are a possibility as well as performance degradation issues due to the additional virtualization layer.

Three main topics should be considered with reference to IaaS: models, virtualization and platforms.

Models

Two standards are worth mentioning in this subsection: Open Virtual Machine Format (OVF) and Open Cloud Computing Interface (OCCI). The Open Virtual Machine Format (OVF) describes an open, secure, portable, efficient and extensible format for the packaging and distribution of (collections of) virtual machines. Open Virtual Machine Format Specification have been submitted by Dell, HP, IBM, Microsoft, VMware, and XenSource to the Distributed Management Task Force (DMTF) for further development into an industry standard. As described within the OVF whitepaper [62], OVF format has specific features designed for complex, multi-tier services, supporting the configuration and composition of virtual machines to deliver composite services. It also permits the specification of both VM and application-level configuration On the other hand, OVF does not describe the non-functional properties of the virtual machines.

OVF has direct relationships with CIM as it uses CIM schemas and vocabularies to help describing its schema. Version 1.1.0 of the specification was published in January 2010 [63].

The Open Cloud Computing Interface (OCCI) [64] comprises a set of open community-lead specifications delivered through the Open Grid Forum [65]. OCCI is a Protocol and API for remote management of IaaS model based Services, allowing for the development of interoperable tools for common tasks including deployment, autonomic scaling and monitoring [66][67]. Current OCCI specifications define a core model, a REST API to communicate with the core model, and extensions for the IaaS domain. Extensions for monitoring, billing or negotiation are still in progress. OCCI has been proposed by a community of cloud based projects such as RESERVOIR, SLA@SOI or Eucalyptus, or the open source community MORFEO.

Virtualization

A key concept of the IaaS approach is the provision of a unified interface to virtualization, independent to particular virtualization implementations. Currently within the open source community Libvirt is one of the most prominent and promising projects. Sponsored by Redhat, Libvirt [70] is an open source API that provides a generic way to interact with different types of open source virtualization technologies (such as Xen [68] and KVM [69] among many others) for the management of the lifecycle of virtual machines.

From an open source strategy perspective Libvirt allows Redhat to abstract away from the particular implementations or vendors and use the most suitable virtualization approach available.

From a management point of view, it is worth mentioning the work of the EU FP7 funded project Reservoir [71] in the development of OpenNebula [72], an open source distributed Virtual Machine Manager, able to dynamically provision VMs on a pool of physical resources. This allows the decoupling of the server not only from the physical infrastructure but also from the physical location.

OpenNebula also presents potential service consumers with a unified interface to several providers supporting virtualization, from small players using standard open source solutions to virtualization based commercial infrastructure providers such as Amazon's EC2 [73] or Elastic Hosts [74]. Since it is based on an extensible plug-in based architecture, it is possible to add or enhance different OpenNebula components. For example Haizea [75] extends the basic scheduler capabilities to include virtual image management (transfer, lifecycle) and resource allocation.

Platforms

Finally, it is important to mention the existing commercial approaches, such as:

- Amazon Elastic Compute Cloud (EC2) [73] is a web service that provides resizable compute capacity in the cloud, providing customers complete control of their computing resources hosted on Amazon's computing environment. EC2 has matured into perhaps the most significant and substantial implementation of a hosted service oriented infrastructure in the marketplace today. EC2 adopts a "Paying for What You Use" model.
- XCalibre Flexiscale [76] provides computing infrastructure resources in a "Pay As You Go" model. It claims to provide all the power or storage resources needed in less than a minute. Self-provisioning of Virtual Dedicated Servers (VDS) is provided via the Control Panel or API, which allows customers to start/stop/delete VDS and to change memory/CPU/storage/IPs. The API provides a SOAP/XML web services interface for integration.
- ElasticHosts [74] offers a flexible and scalable infrastructure at competitive prices that allows customers to instantly add capacity for growth or peaks on demand. Customers have complete control to choose the operating system, applications and configuration of their virtual servers, and can even upload and boot their own custom disk images. ElasticHosts provide a web interface to manage the virtual servers, complemented by a HTTP API and a command line tool. The HTTP API allows users to create drives, upload and download drive images and create and control virtual servers on ElasticHosts infrastructure. The API works in a REST style, and ElasticHosts also provides a simple command line tool and a drive upload tool for Unix or Windows Cygwin users to control the infrastructure from users' own scripts without writing any code.
- ServePath GoGrid [77] is a multi-tier, cloud computing platform that allows customers to manage their cloud hosting infrastructure. GoGrid is ServePath's cloud hosting service with custom-built dedicated servers to create a hosted server network that can scale to meet seasonal or sudden spikes of internet traffic while providing the high I/O and CPU performance demanding database servers require. GoGrid provides a REST-like API [78] query interface to programmatically control the cloud hosting infrastructure. The GoGrid API specification is available under a Creative Commons Sharealike license.
- Cloud Foundry [79]: "an open source "Platform as a Service" (PaaS) from VMware allowing easy deployment of applications written using Spring, Rails and other modern frameworks. It can support multiple frameworks, multiple cloud providers, and multiple application services all on a cloud scale platform."

Annex D: Information Models in standardisation bodies and fora

The Common Information Model (DMTF CIM)

Distributed Management Task Force (DMTF) is an industry consortium that develops, supports, and maintains standards for systems management of PC systems and products for reducing total cost of ownership [84]. The DMTF is also participating in an industry effort to create a standard for management over the Internet. Among DMTF's main activities and achievements is the object-oriented Common Information Model (CIM) [85][86].

The CIM utilizes object-oriented techniques for conceptualisation and structuring; this provides a uniform modelling formalism for the development of an object-oriented schema across multiple organisations.

The CIM is a uniform modelling formalism for the development of an object-oriented schema across multiple organisations; an open standard defining how managed elements in an IT environment are represented as a common set of objects and relationships.

A common conceptual framework is provided through the management schema that includes a basic set of classes for establishing a common framework for a description of the managed environment. The management schema is divided into these conceptual layers.

The Core Model is an information model that captures notions applicable to the whole area of management. The core model is a set of classes, associations and properties that provide a basic vocabulary for describing managed systems. From the classes in the Core Model, the model expands in many directions, addressing many problem domains and relationships between managed entities. The Core Model incorporates entities abstracting (among others):

- ManagedElement: acts as a reference for associations that apply to all entities in the hierarchy,
- Capabilities: describes the various capabilities of specific ManagedElements (physical element capabilities, power management capabilities, localisation capabilities)
- Configuration: aggregates Settings and Dependencies, representing a certain behaviour or desired functional state for Managed System Elements
- ManagedSystemElement: represents Systems, components of Systems, any kinds of services (functionality), software and networks. The definition of "System" in the CIM context is quite broad, ranging from computer systems and dedicated devices, to application systems and network domains.
- Product: represents contracts between vendors and consumers, and capture information about how the Product was acquired, how it is supported, and where it is installed,
- Setting, SettingData: defines specific, pre-configured parameter data to be "applied" to one or more Managed System Elements
- StatisticalData, StatisticalInformation: any kind of statistical data for a Managed Element

The **Common Model** is an information model that captures notions that are common to particular management areas, but independent of a particular technology or implementation. The Core and Common models together are expressed as the CIM schema.

Specific models within the Common Model include:

- Application model captures information commonly required to deploy and manage software products and applications and is based on the need to manage the lifecycle and execution of applications, software product, software element, and software features.
- Database model defines management components for a database environment.
- Device Models abstracts the functionality provided by hardware, configuration and state data addressing low-level concepts such as sensors, controllers, batteries and fans, and high-level abstractions such as Storage Volumes.
- Event Model is modelling changes in the state of the environment or of the behaviour of some component of the environment (e.g. changes in the state of a service, a device or the overall system).

- The Network Model describes and manages communications connectivity and the network "cloud", as well as the individual services and protocols in the network. The managed entities in the model may be grouped into broad categories describing:
    - Network services (e.g. route calculation service, forwarding service, SNMP service),
    - Logical interconnection and access (e.g., protocol endpoints, routes and network pipes)
    - Network protocols (e.g. SNMP, OSPF and BGP),
    - Networking technologies (e.g., Switching/Bridging and VLANs),
    - Quality of Service (QoS) technologies (such as meters, markers and queues).
- The Physical Common Models describes the enclosures, cards and physical components, and cabling information.
- The DMTF Policy Model provides a common framework for specifying system behaviours that are both sufficiently abstract to be independent of implementation-specific details and scalable to configuring large complexes of computer systems. The DMTF Policy Model is a specific model for expressing such policies in a general and scalable way. The DMTF Policy Model enables constructing policy rules of the form: *if <condition(s)> then <action(s)>.* The subclasses are used to create rules and groups of rules that work together to form a coherent set of policies within an administrative domain or set of domains.
    - Policy Set: represents a set of policies that form a coherent set. The set of contained policies has a common decision strategy and a common set of policy roles,
    - Policy Group: An aggregation of PolicyRules that have the same decision strategy and inherit policy roles,
    - Policy Rule: represents the 'If Condition then Action' semantics of a policy rule,
    - Policy Role: identifies the resource(s) to be managed using the PolicySet.
- The System Common Models define computer-system related abstractions related to computer system. Besides the concept of the computer system itself, the System Model also addresses compute components and functionality, associated with most computer systems (e.g. FileSystem, OperatingSystem, Processor and Memory, Power, etc.
- The User/security Common Model defines classes to manage General contact and white pages information for organizations, organizational units and people as well as "Users" of services, and the related security information to authenticate and authorize those "users".

Alike, Figure 48 presents a specific model from the User CIM, the Person model. As depicted in the figure the model conceptualizes personal information, contact information, possible roles etc.

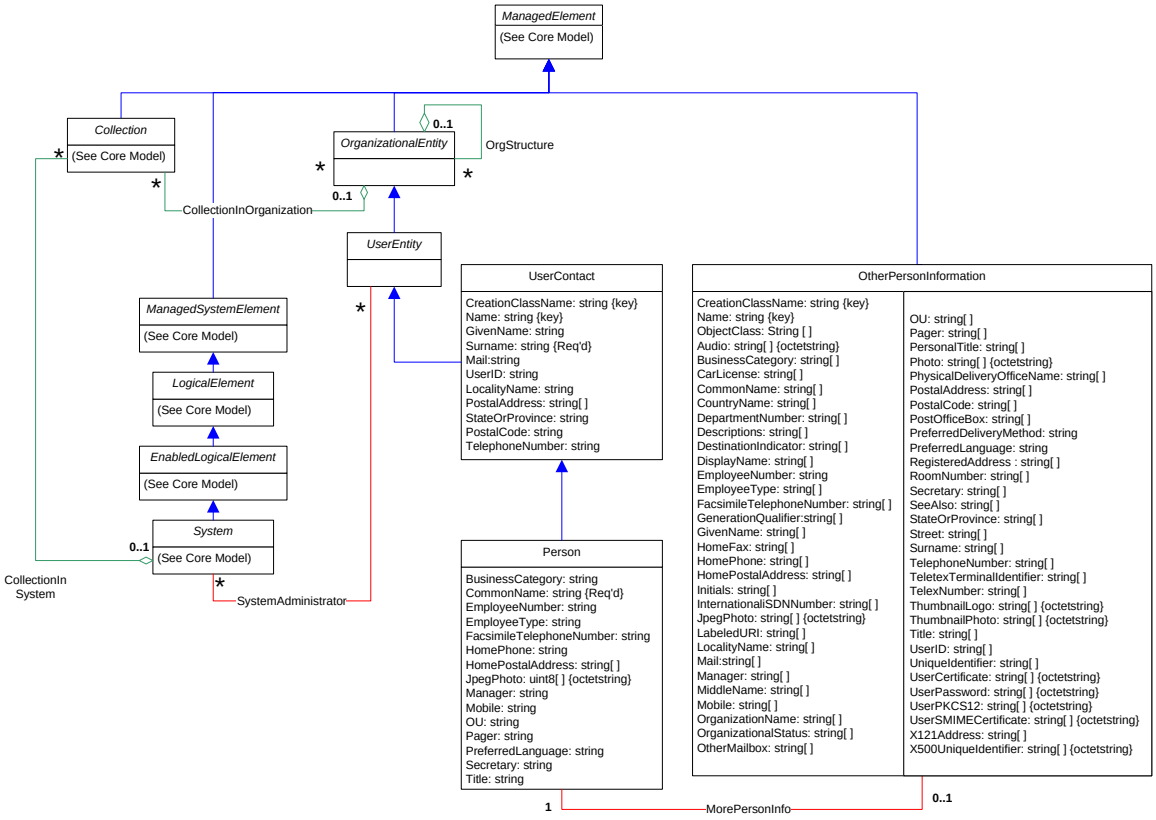


Figure 48. DMTF CIM: Person Model from the User CIM.

Trying to summarise/comment on the CIM approach, the following could be noted:

- CIM is a generic model and incorporates numerous of models, classes and parameters applicable to IT environments,
- It is close to be a data model as it uses database concepts,
- CIM mainly abstracts computer networks environments and not communication networks,
- The CIM Infrastructure Specification defines a method for mapping CIM to other models, such as SNMP
- There is no notion of policy-based behavioural modelling (policy application, etc.) which is centric to autonomic architectures,
- However, CIM could provide a good basis for conceptualising the "computing" part of the overall managed system, which is under consideration in UniverSelf.

The Shared Information and Data Model (TMF SID)

The Shared Information and Data Model (SID) scope covers the information needed to implement use cases base on the eTOM (enhance Telecom Operations Map) processes thus covering a large part of the information required by a service provider. The SID applies primarily to service provider's businesses and engaged stakeholders, system integrators, independent software vendors, and network equipment providers.

The SID is therefore a common information model will streamline the processes associated with information exchange within an enterprise and between the enterprise and its external entities:

- Enables simplification of information management
  - Provides common terminology,
  - Removes unnecessary variation.
- Unification of information within an enterprise and between enterprises
- Bridge between business and information technology groups,
  - Definitions understandable by the business,

- Usable for software development.

The SID business view framework (Figure 49) incorporates the following domains as well as Aggregate Business Entities (ABEs) within each domain. An ABE is a well defined set of information and operations that characterise a highly cohesive, loosely coupled ser of business entities.

The rest of the section will present the service domain specific entities.

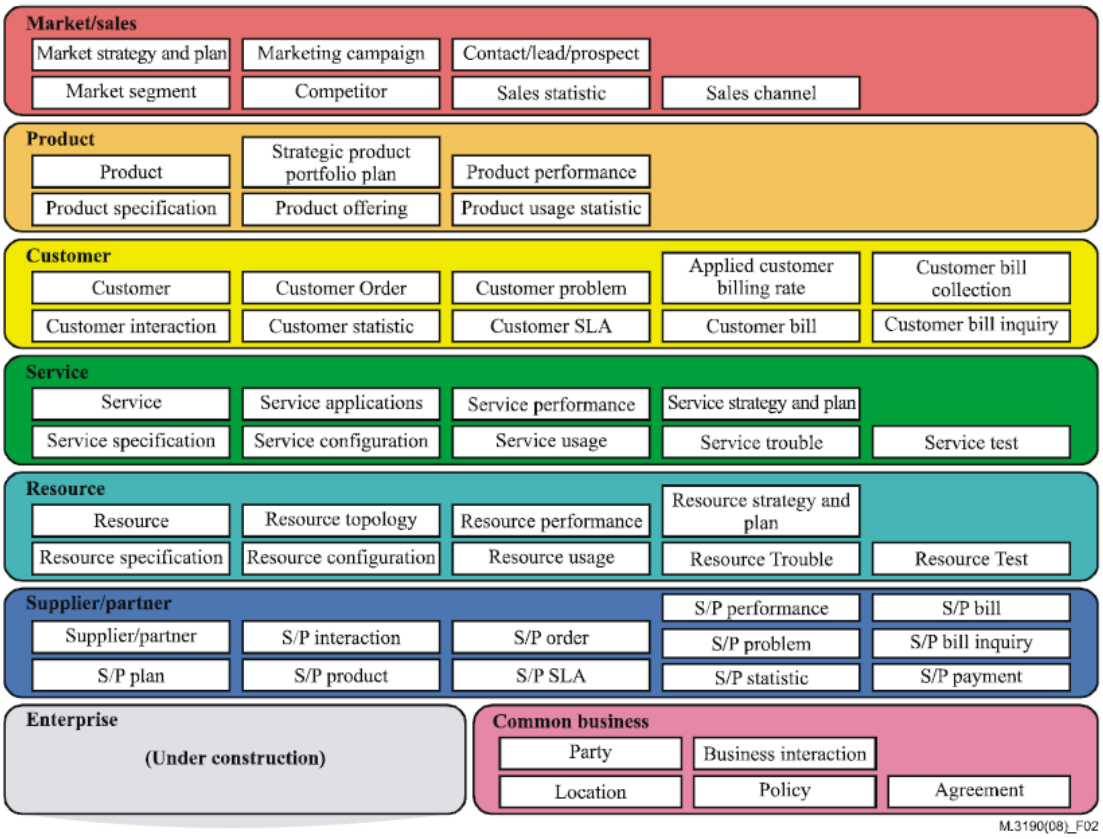


Figure 49. SID Business view framework (from ITU-T M.3190).

ABEs within Service Domain are used to manage the definition, development and operational aspects of services provided by a management system including

- 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.

As presented in previous figure the Service domain includes:

- Service: represent both customer-facing and resource-facing types of services,
- Service Specification: entities defining the invariant characteristics and behaviour of service entities focusing on
  - Distinguishing features of a service
  - Dependencies (logical, physical, to other services),
  - Quality and, cost.

Entities in this ABE enable services to be bound to products and run using resources.

- Service applications: define different types of services implemented as applications:
  - QoS fine tuning,
  - VPN transport,
  - Distance learning, VoIP.

- Service configuration: represent and manage configurations of service entities; provide details on how such configurations can be changed, depending on entities in the resource domain – infrastructure for implementing a service.
- Service performance: collects, correlates, consolidates and validates various performance statistics as well as network performance assessment, trend analysis (cause analysis, error rate, and service degradation), traffic management, traffic trend analysis.
- Service Test: testing services during service installation, or after trouble repair.
- Service trouble: manages faults, alarms, and outages from a service point of view, and direct the recovery from those problems, as well as differentiate between customer-reported problems and network-induced problems.
- Service usage: service consumption data, generates service usage records.
- Service strategy and plan: address the needs for enhanced or new services or retirement of existing services by the enterprise.

Trying to summarise/comment on the SID approach, the following could be noted:

- Concepts in Service Domain are used to manage the definition, development and operational aspects of services provided by a management system including
    - 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
- Initially SID covered the business (BSS) and the device management field well but not aspects such as logical networks and capacity resulting in poor utilisation of the model in certain telecom fields,
- There is no Notion of behavioural modelling (e.g. policy application etc.).
- SID can provide a basis for conceptualising service related aspects within UniverSelf.

IEEE P1900.4

The IEEE P1900.4 WG on Draft Standard for Architectural building blocks enabling network-device distributed decision making for optimized radio resource usage in heterogeneous wireless access networks The standard defines the building blocks comprising (i) network resource managers, (ii) device resource managers, and (iii) the information to be exchanged between the building blocks, for enabling coordinated network-device distributed decision making which will aid in the optimization of radio resource usage, including spectrum access control, in heterogeneous wireless access networks.

The Information Model as developed in P1900.4 WG has been based on an object-oriented approach abstracting the Composite Wireless Network and the Terminal as two sets of managed objects.

The Terminal MO is composed of the following concepts:

- User,
- Application,
- Device
- Link,
- Observed Channel,
- Radio Resource Management Policy.

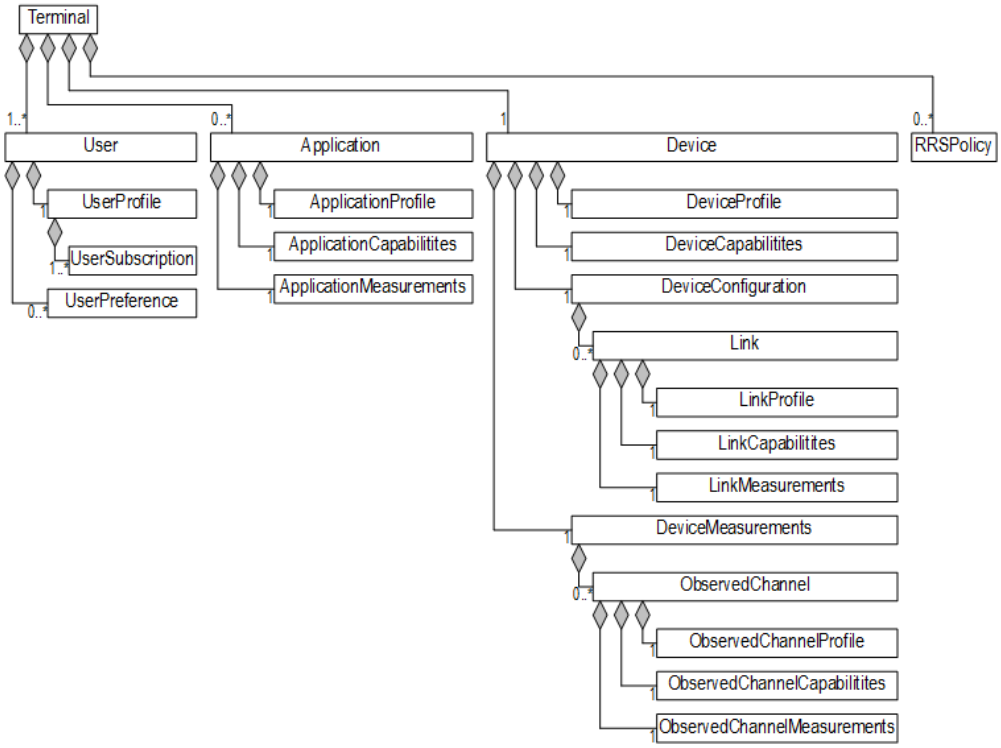


Figure 50. IEEE P1900.4 WG Information Model: the Terminal Manage Object.

More specifically, concepts in P1900.4 information model have been modelled to incorporate each concepts profile, capabilities and measurements. In this way, the Information Model provides a means for further specialisation to capture more specific capabilities and measurements, which are applicable to more specific contexts.

For example, the Device concept includes:

- Device Profile: the device incorporates a number identifier and a string,
- Device Capabilities:
  - List of supported device measurements: the location of the device, the battery power, etc.,
  - List of supported device options: the maximum Tx power, the maximum number of radio interfaces, etc.,
  - List of supported radio interfaces: UMTS, HSDPA, WiMax, LTE, WiFi, GSM, etc.,
- Device Measurements
  - List of active device measurements: the location of the device, the battery power, etc.,

Each of the included information classes incorporates a number of attributes and access related information. Composite Wireless Network (CWN) classes are shown in Figure 51.

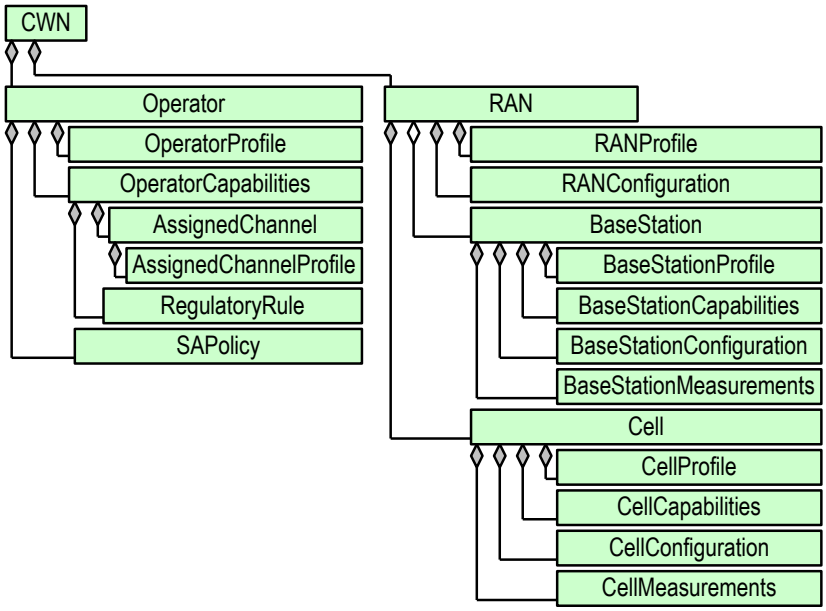


Figure 51. IEEE P1900.4 WG Information Model: the Composite Wireless Network Manage Object.

CWN classes are used to abstract:

- Operator
  - Assigned Channel,
  - Regulatory Rule,
  - Spectrum Access Policy.
- RAN,
- Base Station,
- Cell.

For each of the classes described above, IEEE 1900.4 standard describes its members and their type, where data types are specified using ASN.1 notations.

Trying to summarise/comment on the P1900.4 approach, the following could be noted:

- The IEEE P1900.4 Information Model provides managed objects hierarchies for the abstraction of the user equipment and the network in a composite, heterogeneous wireless network context, in reconfigurable and cognitive radio systems,
- The Managed Objects hierarchies have been developed in a uniformly structured way and use generic classes and attributes, which can be extended with (sets of) specific parameters.
- P1900.4 hierarchies and concepts can be used in the context of the access domain of the UniverSelf scope.

ANDSF Management Object in 3GPP

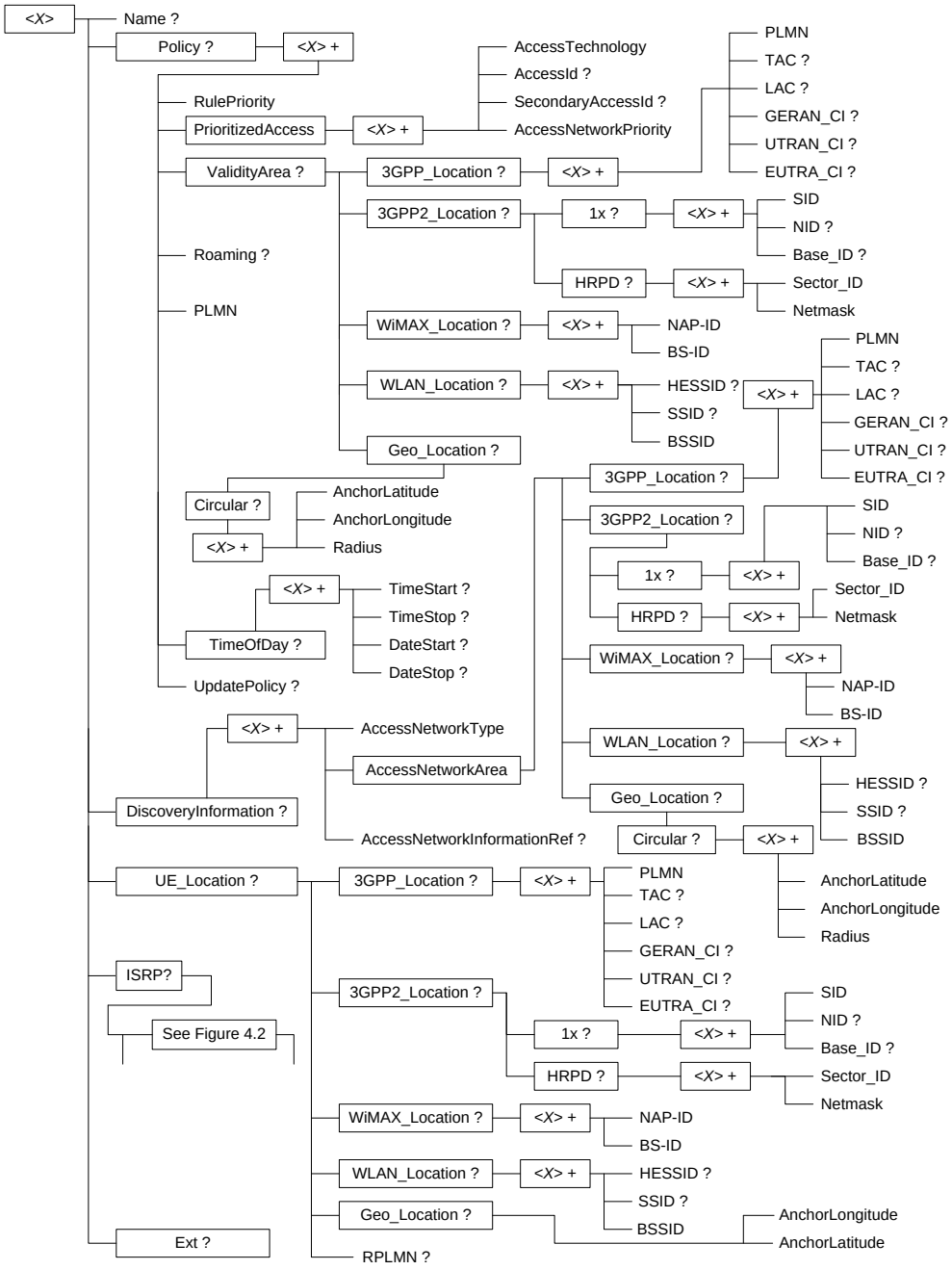
3GPP TS 24.312 [91] defines information models (Management Objects) that can be used by the Access Network Discovery and Selection Function (ANDSF) and the User Equipment. The Management Object (MO) is compatible with the OMA Device Management (DM) protocol specifications, version 1.2 and upwards. The MO consists of relevant parameters for intersystem mobility policy- and access network discovery information that can be managed by the ANDSF.

The service requirements and the functional requirements for the access network discovery and selection are described in 3GPP TS 22.278 [89] and in 3GPP TS 23.402 [90] respectively.

The ANDSF MO is used to manage intersystem mobility policy- as well as access network discovery information stored in a UE supporting provisioning of such information from an ANDSF. Specifically, ANDSF assists the UE in selecting non-3GPP accesses. The objective is to reduce UE battery consumption by keeping its air interface(s) down and retrieving RAN/RAT information via a centralised server. ANDSF can be queried by both single-radio and multi-radio UEs under a maximum allowed number of selection attempts. According to operator requirements, the ANDSF provides a) operator-defined policies for RAT selection (both inter-system mobility and preference over a specific access from a list of common RATs), and b) discovery information on access

networks. The latter may include the RAT type, the identifier of the access point (e.g., WLAN SSID), the carrier frequency, and validity information (e.g., location of the advertised RAT).

The relation between Policy and Discovery Information is that Policies prioritize the access network, while Discovery Information provide further information for the UE to access the access network defined in the policy. The possible nodes and leaf objects are possible under the ANDSF node as described in Figure 52 and Figure 53.


<X>
Name ?
Policy ?
<X> +
RulePriority
PrioritizedAccess
<X> +
AccessTechnology
AccessId ?
SecondaryAccessId ?
AccessNetworkPriority
ValidityArea ?
3GPP_Location ?
<X> +
PLMN
TAC ?
LAC ?
GERAN_CI ?
UTRAN_CI ?
EUTRA_CI ?
3GPP2_Location ?
1x ?
<X> +
SID
NID ?
Base_ID ?
HRPD ?
<X> +
Sector_ID
Netmask
Roaming ?
PLMN
WiMAX_Location ?
<X> +
NAP-ID
BS-ID
WLAN_Location ?
<X> +
HESSID ?
SSID ?
BSSID
<X> +
PLMN
TAC ?
LAC ?
GERAN_CI ?
UTRAN_CI ?
EUTRA_CI ?
Geo_Location ?
Circular ?
AnchorLatitude
AnchorLongitude
Radius
<X> +
3GPP_Location ?
3GPP2_Location ?
SID
NID ?
Base_ID ?
1x ?
<X> +
Sector_ID
Netmask
HRPD ?
<X> +
WiMAX_Location ?
<X> +
NAP-ID
BS-ID
WLAN_Location ?
<X> +
HESSID ?
SSID ?
BSSID
Geo_Location ?
Circular ?
<X> +
AnchorLatitude
AnchorLongitude
Radius
<X> +
TimeStart ?
TimeStop ?
DateStart ?
DateStop ?
TimeOfDay ?
UpdatePolicy ?
<X> +
AccessNetworkType
AccessNetworkArea
DiscoveryInformation ?
AccessNetworkInformationRef ?
UE_Location ?
3GPP_Location ?
<X> +
PLMN
TAC ?
LAC ?
GERAN_CI ?
UTRAN_CI ?
EUTRA_CI ?
3GPP2_Location ?
1x ?
<X> +
SID
NID ?
Base_ID ?
HRPD ?
<X> +
Sector_ID
Netmask
WiMAX_Location ?
<X> +
NAP-ID
BS-ID
WLAN_Location ?
<X> +
HESSID ?
SSID ?
BSSID
Geo_Location ?
AnchorLongitude
AnchorLatitude
RPLMN ?
ISRP?
See Figure 4.2
Ext ?

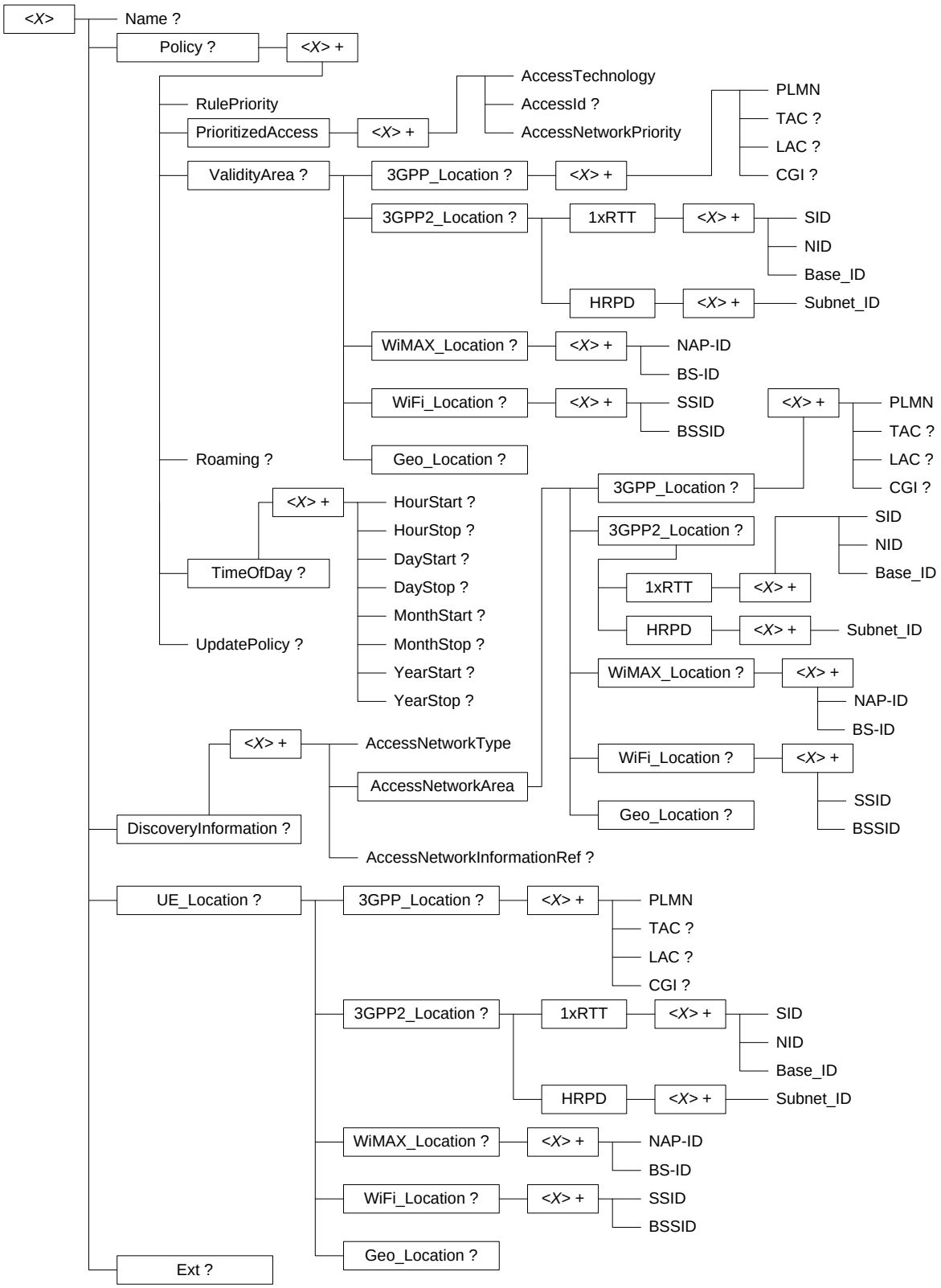


Figure 52. The ANDSF MO (Source: 3GPP TS 24.312 V10.1.0) [91].

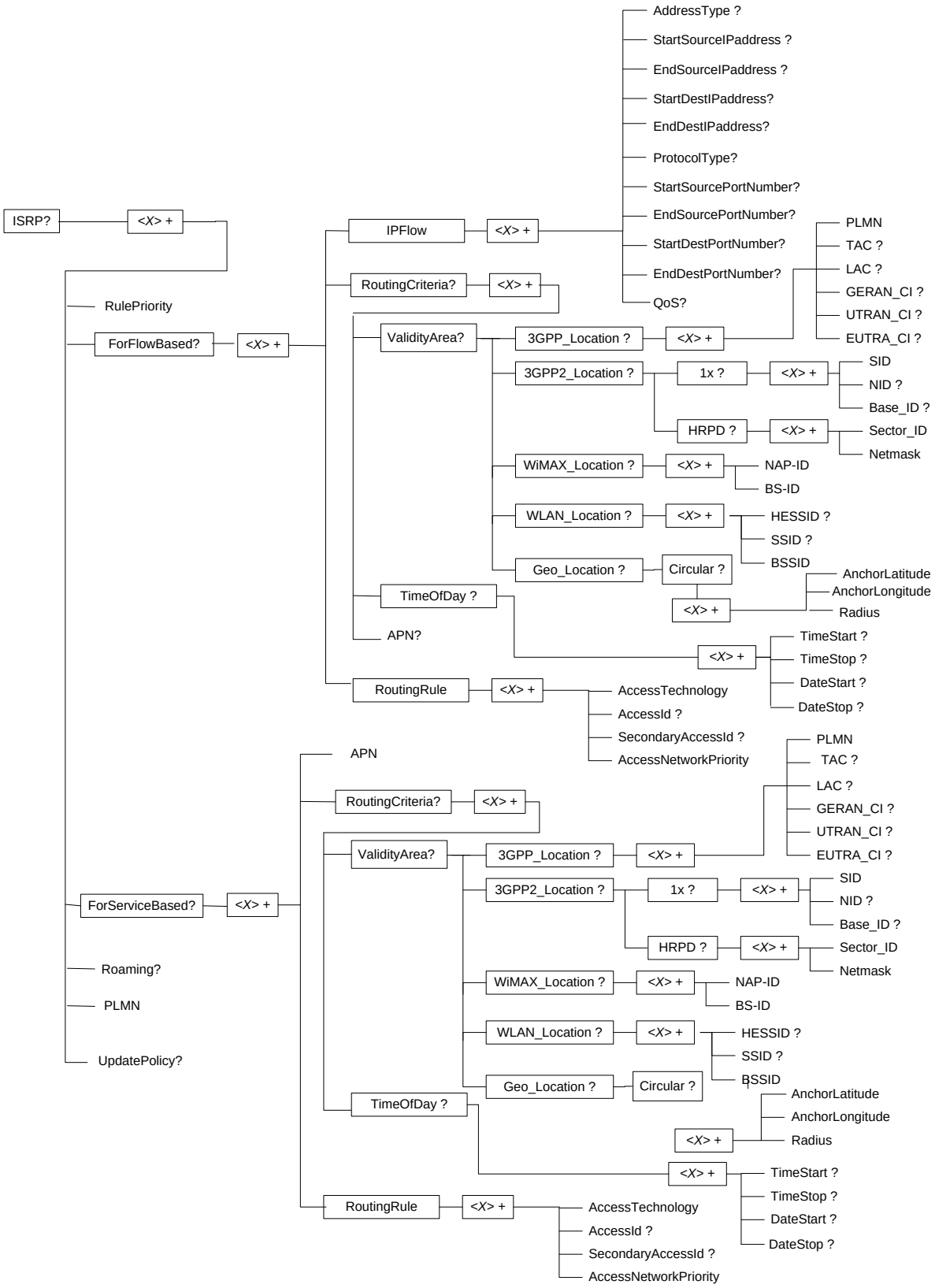


Figure 53. ANDSF MO continuation (Source: 3GPP TS 24312 v.10.1.0) [91].

Trying to summarise/comment on the ANDSF approach, the following could be noted:

- the ANDSF managed object abstracts information related to

  - access technologies discovery,
  - access technology prioritisation, and,
  - (policy-based) connection of UE.
- ANDSF could be considered as complementary to IEEE P1900.4 as they share the same technical scope (i.e. operation in heterogeneous wireless networks).

DEN-ng

DEN-ng is the information model framework which has been developed to enable autonomic network management. DEN-ng has been developed within the TMF and has been adopted by the FOCALE autonomic networking architecture. FOCALE is based on the following key principles:

- FOCALE uses information and data models in order to establish a common "lingua franca" for enabling interoperability between technology specific network management functionalities,
- FOCALE uses ontologies in order to provide semantic interoperability throughout the considered models as well as to address polysemy, synonymy, meronymy, antonymy, etc,
- FOCALE uses context-aware policy rules to govern the functionality of the management system and to direct the operation and execution of its control elements.

The DEN-ng model is based on the DMTF CIM and TMF SID models, however, the DEN-ng is focusing more in communications network management, and the SID on the business oriented aspects of management. Moreover, CIM is not perceived as an information model but a model similar to a relational database. The fact that the SID and CIM do not have any notion of policy application and negotiation, state machine, context, metadata, and ontology compatibility means that they do not provide enough technical features to satisfy the needs of autonomic architectures. A simplified comparison between DEN-ng, CIM and SID is presented in the next figure.

Feature	DEN-ng	SID	CIM
True information model?	YES	YES	NO – uses database concepts
Classification theory	Strictly used	Not consistently used; not used by mTOP	Not used at all
Patterns	Many more used that the SID	4	Not used at all
Views	Life-chain, not life-cycle	Business only; mTOP defining informal system & impl views	One view
Policy model	DEN-ng v7	DEN-ng v3.5	Simple IETF model
Policy application model	DEN-ng v7	NONE	NONE
Policy negotiation model	DEN-ng v6.7	NONE	NONE
State machine model	DEN-ng v6.8	NONE	NONE
Ontology compatibility	STRONG	NONE	NONE
Capability model	DEN-ng v6.6.4	NONE	Very simplified
Context model	DEN-ng v6.6.4	NONE	NONE
Metadata model	DEN-ng v7	NONE	NONE

Figure 54. A simplified comparison of the DEN-ng, CIM and SIF models ([92]).

DEN-ng was designed to represent the management information of large scale communications networks, from a business through to implementation perspective. DEN-ng represents the relationships among networking components and network services represented as well as business concepts and relationships.

The primary enhancement provided by DEN-ng over previous information models is that it did not just represent the structure of management information, but also defines a management methodology that instructed how the management of a large scale communications network should be carried out. Key differentiating factor of DEN-ng is a **Policy-based network management driven by an information model**. The DEN-ng information model describes concepts, relationships and attributes related to products, services, customers down to routing protocols and QoS services.

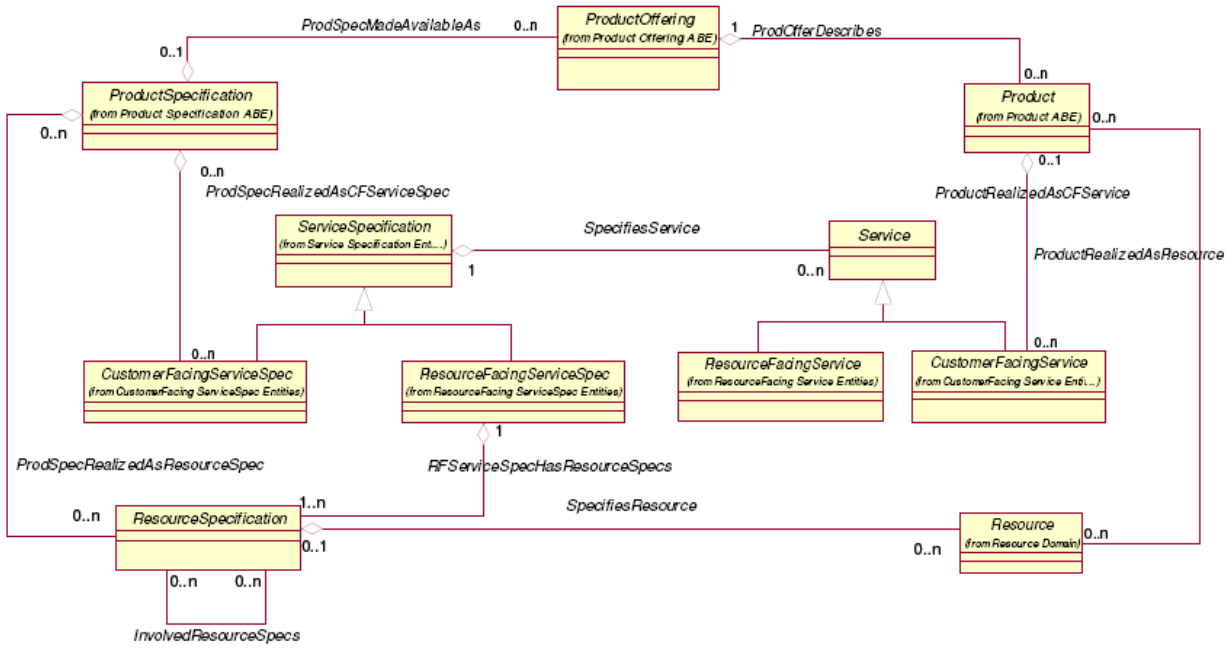


Figure 55. DEN-ng product-service (TMForum, 2008).

A typical information model diagram is depicted in Figure 55 and illustrates the relationship between a product that can be purchased by a customer from an ISP, to a set of services that must realise that product in the communications network. The DEN-ng information model was designed with a policy information model built in; policy could be related to business managed entities down to network managed entities.

DEN-ng enables:

- Context-aware policy rules to govern the functionality in a managed system: use context to select only those policy rules applicable to management task being performed.
- Policy continuum: a continuum of policies in which each policy captures the requirements of a particular consistency
    - Business people define an SLA in business terms,
    - This is transformed into architectural requirements,
    - …
    - Finally it is translated into network configuration commands.

  This is performed in a way that preserves the semantics of different policy rules which could otherwise be lost if simple syntactic translations were used instead.
- The operation of each managed governed by the autonomic system entity is specified through state machines
    - Information and data models populate the state machines,
    - Management information monitored by the autonomic system consists of sensor data,
    - Such information is d to derive the current state of the managed resource,
    - State comparison follows against the desired state,
    - In case, the transitions are defined and triggered towards the desired state.

A single information model embedding governance and behavioural model can obtain the characteristics and behaviour of managed entities and highly facilitate information sharing and reusability: a mapping to that single model is needed instead of multiple mappings between all involved vendor-specific models.

DEN-ng is used to

- Federate knowledge extracted from different sources,
- Provide final knowledge representation to be used by a specific application
- Representation is based on clearly differentiating between data-information-knowledge.

  - Data is characterized as observable and possibly measurable raw values that signal something of interest. In and of themselves, data have no meaning they are simply raw values,
  - Data is transformed into information when meaning can be attached to data. This provides the data with a simple, starting context,
  - The process of transforming information into knowledge attaches purpose, forms a more complete context, and provides the potential to generate action.
- Modelling observed/measured facts as well as inferred facts,
- Model the aspects of a managed entity separating the entity from its roles thus providing reusable different views of the same entity.

Trying to summarise/comment on the DEN-ng approach, the following could be noted:

- DEN-ng modelling approach can provide a reference framework for UniverSelf Information and Knowledge Model,
- The "contents" of the model - specific concepts can come from CIM, SID, etc.,
- This will enable mapping to a common reference model instead of mapping among all "involved" existing vendor-specific models.
- Actual models are restricted within TMF members. However, the approach has been well disseminated; this means that we could base our work in Den-ng principles and build a methodology,
- Building a methodology will help us avoiding building information meta-model for all existing information models. It is also future proof for any further systems.

Annex E: Information Models in Research Projects

This section presents two information models from previous research projects, in the area of cognitive radio systems (E3) and self-managed Future Internet Network Elements (Self-NET) highlighting also how those models (E3 mainly) have been based and extended existing information models.

Information Model in Cognitive Radio Systems (E3)

This section presents the E3 Information Model [93] in terms of a number of high level information concepts that are further abstracted using specific low level attributes. For each one of the key concepts a UML model has been developed as well as a comparison with two reference models (i.e. IEEE P1900.4 and 3GPP ANDSF).
The managed environment of E^3 concerns B3G/4G telecommunications, which is mainly characterised by:

- a plethora of user devices, with different capabilities and configurations, being operated in a dynamic and reconfigurable mode,
- a vast number of mobile applications that are developed and provided from different vendors and with different requirements,
- high availability of different types of networks (e.g. WLAN, 3G, UMTS, WiMAX and so on).

A set of the key concepts have been also identified that form the basis for high-level information abstraction. These include:

- The Network Operator that owns and operates different types of networks,
- The User that utilises a number of terminals for service provision/consumption purposes,
- The Terminal that moves within the heterogeneous ecosystem and gets attached to different RATs in a preference and policy-based approach,
- The RAN and the deployed RAT that form the heterogeneous environment,
- The Base Station, the Access Point and the Home Base Station that serve different RAT types within a RAN,
- The Cell that form the coverage area of the various RANs,
- The Ad-hoc Networks and the Terminals and/or Base Stations Clusters that can be formed in a dynamic way within the overall coverage area,
- The Services, both network and application that are provided within the ecosystem,
- The Hotspots that can be emerged within an area.

The primary information items above are also depicted in Figure 56. In this ontology, a set of defined interrelations are also presented:

- A User uses a Terminal and consumes a Service,
- An Operator provides Service and operates a RAN that deploys a RAT,
- A Terminal is attached to a RAN, may form an Ad-Hoc Network and is served by a HotSpot or Base Station,
- Conversely, a Base Station serves a Terminal, may form a Cluster and covers a (number of) Cells.

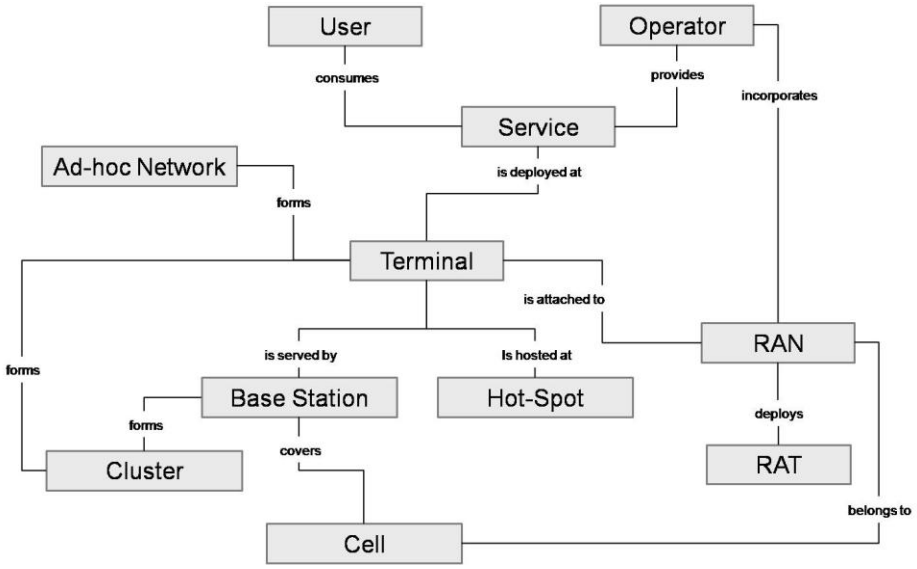


Figure 56. E3 Key Concepts.

The IEEE 1900.4 [88] and 3GPP ANDSF MO [87] have been identified to serve as reference models. In the following subsections indicative concepts as part of the E3 Information Model will be presented together with a comparison to the IEEE P1900.4 and 3GPP ANDSF models.

User Concept

The User concept includes the User profile that consists of the home and the work profile of a user. Moreover, the model incorporates information about the user preferences, the user requirements and the user experience as outlined by the respective classes. The user subscription provides information about the user's subscriptions to network and application service and is associated with the user requirements and the running service that abstracts any deployed service in user equipment.

Table 18. E3 User Concept

User		
E3	**P1900.4**	**ANDSF**
User Profile Home Profile Work Profile	☑ ☒ ☒	☒ ☒ ☒
User Preferences	☑	☒
User Requirements Minimum Data Rate	☒ ☒	☒ ☒
User Experience Degree of QoS	☒ ☒	☒ ☒
Running Service Subscribed Service Service Type	☒ ☑ ☒	☒ ☒ ☒

RAN and RAT Concepts

Within the E^3 context the RAN concept is related to the RAT; in P1900.4 such concepts have been incorporated through the Base Station and Cell related classes. For example, the ***listOfSupportedRadioInterfaces*** attribute is included in the ***BaseStationCapabilities*** class. It is noted that the term “Base Station” in the scope of P1900.4 is

used to refer to any radio node on the network side and is represented by the Base Station Model. A set of identified RAN related measurements that are listed in the table below can be considered as already incorporated in the P1900.4. For example, the Radio Load can be considered as an extension to the Cell Measurements class.

The concept of the RAT Protocol is included in the E^3 context, although such concept is not incorporated in the P1900.4 information model. A ***RATProtocol*** class can be added by extending the ***BaseStationCapabilities*** in order for the RAT Protocol to be abstracted in the model. Such a class aggregates the concepts of the Protocol Component, Protocol Metadata and Protocol Configuration as member classes whereas the corresponding attributes shall be defined. This is quite important for E^3 as the concept of a reconfigurable protocol has been subject of the project work and has been included in several use cases. In this way, the protocol related information fills any gaps in the standardised information models regarding reconfigurable protocols.

Table 19. E3 RAN Concept

Radio Access Network (RAN)		
E3	**P1900.4**	**ANDSF**
RAT Type RAN Policies	☑ ☒	☑ ☑
RAN Status Radio Load Processing Load Transport Load	☒ ☒ ☒ ☒	☒ ☒ ☒ ☒

Table 20. E3 RAN Concept

Radio Access Technology (RAT)		
E3	**P1900.4**	**ANDSF**
RAT Protocol Protocol Component Protocol Metadata Protocol Configuration	☒ ☒ ☒ ☒	☒ ☒ ☒ ☒
Assigned Spectrum	☑	☒

Policy Modelling

A number of different types of policies are incorporated within the scope of E^3. Such policies are derived from the involved actors strategies and objectives (i.e. the Network Operator, the User, etc) and target the system entities behaviour in a specific technical area (i.e. Flexible Spectrum Management, RAT Selection, etc.). The different types of policies that have been identified in the context of E^3 are presented in the following list:

- Dynamic Spectrum Access (DSA) Policy
- Radio Resource Assignment (RRA) Policy
- Mobile Terminal Assignment (MTA) Policy
- RAT Selection Policy
- Energy Saving Policy
- Handover Policy
- SON Policy

[93] presents a comparison between the E3 Policy Types and the corresponding work in the P1900.4 and the ANDSF.

Table 21. Policy Model Comparison

Policies		
E3	**P1900.4**	**ANDSF**
Dynamic Spectrum Access Policy	☑	☑
Radio Resource Assignment Policy	☑	☑
Mobile Terminal Assignment Policy	☒	☑
RAT Selection Policy	☑	☑
Energy Saving Policy	☒	☑
Handover Policy	☒	☑
SON Policy	☒	☑

Information Modelling in Self-Managed Future Internet Systems (Self-NET)

A set of key concepts has been derived from the use cases forming the first level of abstraction in the information model. Such concepts include:

- The **Network**, in terms of the different types that operate within the Self-NET managed environment (i.e. wired, wireless),
- The **Cell** that forms the minimum area of service provision in a wireless network,
- The **User** that operates a mobile device,
- The **Operator** that owns and operates different types of networks,
- The **Network Element** (autonomic) that form the main concept in the Self-NET framework; this can be Router, Switch, Gateway, Access Point, Base Station, Mobile Device, etc.,
- The **Link** between Network Elements.

The whole diversity of the draft information model is presented in the following text. In this, the various system entities are conceptualised through a number of parameters that are presented in a various level of abstractions.

Network Concept

The network is mainly abstracted through its type which can be wireless, wired or wireless mesh and the respective network resources. Some of them are aggregated measurements. Network concept incorporates information about the corresponding network operator. Operator concept includes attributes reflecting operator requirements on coverage and capacity; these are mostly related to operator strategies. Additionally, the area of service, the operator policies (for example dynamic spectrum access policies) are included together with information on the licensed and the shared spectrum bands; the later captures cases of dynamic spectrum management as included in the Self-NET use cases.

The Cell concept is considered part of the Network concept in case that the network is a cellular wireless one. The Cell Profile includes the identification of a cell, information about the traffic and the coverage area, the maximum number of subscribers who can be services inside the cell, and information about the cell throughput (maximum or targeted). Alike, the cell status is abstracted through measurements about the load, the traffic, the blocked and dropped call rates as well as the handover success ratio. The concept of the neighbouring call has been also included to capture inter-cell compensation cases as also described in the Self-NET use cases.

Finally, the channel concept includes information on the assigned frequency, the interference and the load as well as the respective sub-channels.

Network Element Concept

The Network Element is the main concept in the Self-NET information modelling. In that abstraction model, an initial set had been presented based on categories inside a Network Monitoring System. Such abstractions included information about the Router/Switch, the AP/Bs as well as the authentication and authorisation procedures. In the model that is presented in this subsection, the initial parameters have served as an initial basis, and the abstraction model has been elaborated to form a more complete abstraction of the different “instances” of the Network Element as it is considered in the Self-NET scope.

Specifically, the model incorporates classes and attributes that abstract the generic type of the Network Element, such as the identification, the brand, the CPU, information about the power supply and supported wireless and wired interfaces. Moreover, the status of a generic form of the NE is abstracted through respective context measurements, such as the observed congestion, traffic load, the used memory and the power consumption.

In turn the more specific types of NEs are abstracted as they have been identified inside the project. These include the Router, the Base Station, the Gateway and the Mobile Device and the mentioned specific abstraction is performed through more specific information items (classes and attributes) that are tailored to the different NE type and extend the mentioned generic ones.

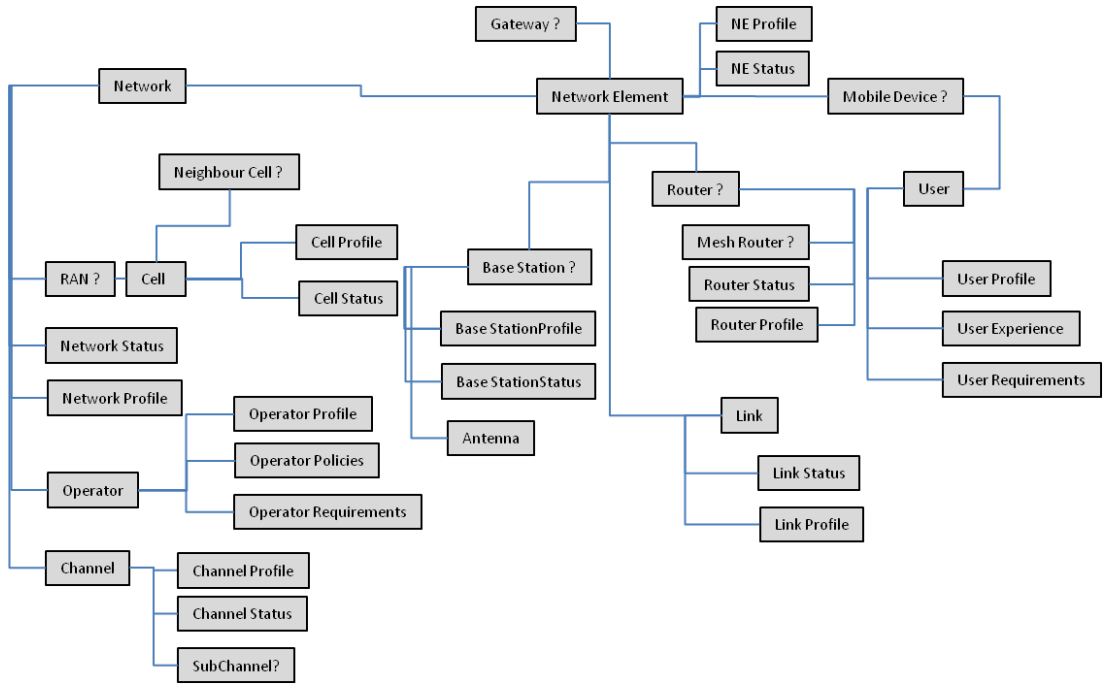


Figure 57. Self-NET Information concepts.

The AutoI Information Model

The AutoI Information Model (AIM) uses a set of abstractions and software patterns that enable services to express their needs to the management overlay, which translates those needs into a form that the network can understand.

A subset of the DEN-ng information model is used as the core of the AIM model. Then, new extensions are added to support new concepts such as virtualisation. In addition, some appropriate refinements to the model are done in conjunctions with the AutoI requirements and constraints. Thus, the core of the AIM model is based on a DEN-ng subset.

The AIM model plus a set of domain-specific ontologies can then be used as a common language to advance interoperability and understanding across the disparate components of the AutoI architecture. In the same way, the common language enables network resources to be defined in such a way that the services can use them and work with them.

In the final release of the AutoI information model, mechanisms for model migration where assessed and the Information Model was allowed to evolve by fully decoupling it from DEN-ng.

Domain Specific Languages (DSL) are used to address the specific system tasks in an interoperable manner by defining translation rules and facilitating extensibility of the AIM. This allows DSLs to address domain-specific tasks while preserving interoperability.

The AIM Model is used to define facts, and ontologies are used to augment facts with additional semantics. The ontologies extend the model knowledge with semantics to enable reasoning that can be used to provide added value by taking into consideration the contextual relevance of the data.

The Model-based Translator is a flexible component of the AutoI solution that aids in the abstraction of heterogeneous data models. It makes use of state of the art model driven software development techniques to adapt data at runtime for use in configuration of networking resources. It is designed to be deployed independently from the types of data models it may encounter, and management systems that make use of the MBT are also abstracted from having to know the multitude of data models they may encounter. The dynamic loading of translation templates allows the MBT to adapt to new networking device types and new configuration interfaces.

Policy Domain

The Policy information model for AutoI (AIM) is a subset of the model defined in DEN-ng. There are two main types of concepts defined in the AIM with respect to policy based management. On one side, concepts that are used to describe the components of individual policy rules and how they relate to the managed entities defined in the rest of the AIM. On the other side, concepts that describe the use of policies in managing sets of managed entities.

The AIM is primarily concerned with describing the structure and relationships of *ManagedEntities* from the perspective of AutoI. The Policy related concepts of the AIM are concerned with describing a management methodology that is flexible enough to be used for the management of all defined *ManagedEntities*. Interestingly, Policies are themselves *ManagedEntities* and so can be managed by other policies. This decision yields a highly flexible management methodology for use in the management of autonomic communications networks.

Annex F: SOTA on Policy-Based Management - Frameworks and Languages

The increasing complexities and heterogeneity of modern networking technology, and the vast number of resources to be managed, pose significant challenges to network management models. Policy-Based Management (PBM) is a promising solution for these demands, providing the means by which the administration process can be simplified and automated to a large extent. A policy, the basic building block of the policy-based paradigm, is a set of rules that govern the behaviour of a managed system. As these rules constitute interpreted logic, the approach facilitates flexibility and adaptability in that policies can be dynamically changed without modifying the underlying implementation. This section presents the three most important PBM frameworks and associated specification languages in the literature.

IETF Policy Management Framework

The joint effort of the IETF [94] and DMTF resulted in a generic policy architecture, which consists of four major functional elements: the *Policy Management Tool* (*PMT*), *Policy Repository*, *Policy Decision Point* (*PDP*), and *Policy Enforcement Point* (*PEP*).

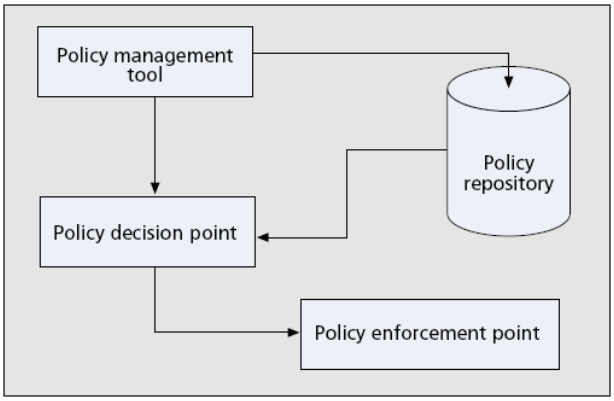


Figure 58. The IETF/DMTF Policy Framework.

The PMT is used by an administrator to define or update the policies to be enforced in the managed network. Resulting policies are stored in a repository in a form that must correspond to the information model in [112] so as to ensure interoperability across products from different vendors. When new policies have been added in the repository, or existing ones have been changed, the PMT issues the relevant PDP with notifications, which in turn interprets the policies and communicates them to the PEP. The latter is a component that runs on a policy-aware node and can execute (enforce) the different policies. The components of the architecture can communicate with each other using a variety of protocols. The preferred choice for communicating policy decisions between a PDP and network devices (PEPs) is the Common Open Policy Service (COPS) [113], or SNMP [114], and LDAP [115] for the PMT/PDP–repository communication.

The simplest approach for policy specification is through a sequence of rules, in which each rule is the form of a simple condition-action pair. The IETF policy framework adopts this approach and considers policies as rules that specify actions to be performed in response to defined conditions:

The conditional part of the rule can be a simple or compound expression specified in either conjunctive or disjunctive normal form. The action part of the rule can be a set of actions that must be executed when the conditions are true. The IETF does not define a specific language to express network policies but rather a generic object-oriented information model for representing policy information (PCIM) [112]. This model is a generic one, specifying the structure of abstract policy classes by means of association, thus allowing vendors to implement their own set of conditions and actions to be used by the policy rules.

Ponder Policy Framework

Initial work in [116] describes the concept of policies in distributed systems management. Here, policies are viewed as objects which define the relationships between subjects (managers) and targets (managed objects), and are separated from the managers' functionality. This facilitates the dynamic change of the behaviour and adaptation to new requirements without re-implementing the management applications. In [117] the authors identify that specifying policies for individual managed entities in large-scale systems is not a practical approach. They propose the use of domains as the means of grouping objects representing managed entities to which policies apply, thus partitioning the management responsibility.

The concept of domains is a key aspect of the Ponder policy framework which is depicted in Figure 59 [118]. Here, an administrator can create and modify policies using a policy editor. Authorisation policies are disseminated to target agents as specified by the target domains and obligation policies to manager agents (PMAs) as specified by the subject domains. Policies can be subsequently enabled, disabled or removed from the agents. Obligation policies are interpreted by manager agents, which register with the monitoring service to receive events relevant to their activation. Upon receiving an event, the agent queries the domain service to determine the target objects and performs the policy action(s).

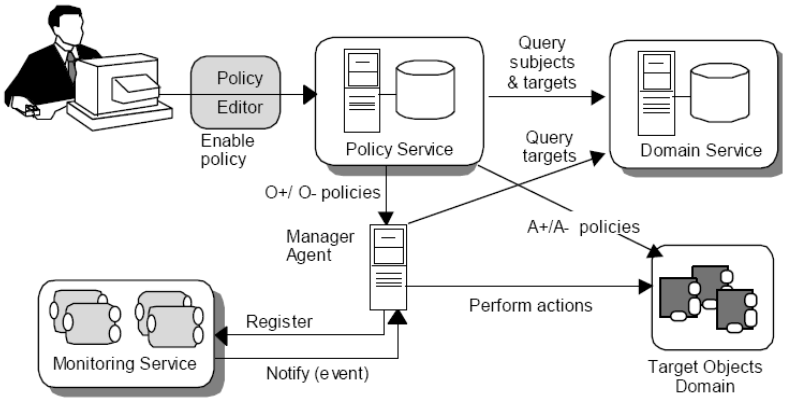


Figure 59. The Ponder Policy Management architecture.

Subsequent work on Ponder [119] involved the design of a deployment and enforcement model and the development of a toolkit integrating the various components of the framework to support the whole policy life-cycle relating to the specification and management of deployed policies. The toolkit provides a comprehensive policy-based management platform based on an object-oriented Java implementation and has been widely used in the research community.

Ponder is a declarative, object-oriented language [120] that can be used to specify both security and management policies. It supports two main policy types as described below: authorization and obligation policies.

Authorization policies define what actions a manager (subject) can perform on target objects. These policies are enforced by access controllers running in the target objects' environment aiming to protect resources from unauthorized access. A positive authorization policy is used to define the actions that subjects are permitted to perform on target objects, whereas negative authorizations define the actions that subjects are prohibited from performing.

Obligation policies are *event-condition-action* (ECA) rules that define the operations that must be performed by managers of the subject domain on objects of the target domain when certain events occur, given some supplementary conditions being true. While authorizations are executed by access controllers, obligation polices are enforced by PMAs which facilitate adaptation of the managed system according to emerging conditions.

Policy Management for Autonomic Computing

The Policy Management for Autonomic Computing (PMAC) platform [121] is part of IBM's initiative on autonomic computing, which defines a framework for self-managing IT systems. PMAC is a generic middleware platform that can be used to manage aspects of large-scale distributed systems including QoS, security and

auditing. The architecture of the platform is depicted in Figure 60 which provides components for policy creation, policy evaluation, and enforcement at managed resources.

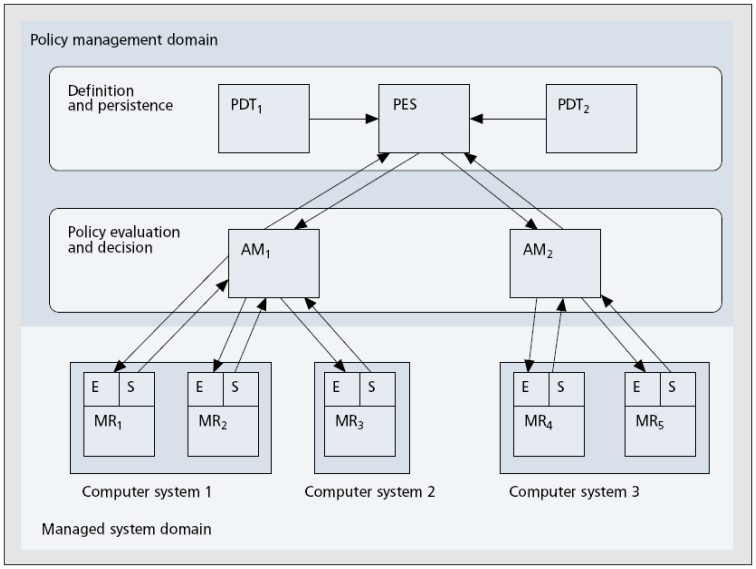


Figure 60. The PMAC Architecture.

At the highest level, multiple Policy Definition Tools (PDT) are supported for concurrent policy authoring. Policies are stored in a centralised Policy Editor Storage (PES) which can also hold metadata such as templates for policy re-use. The main component of PMAC is the Autonomic Manager (AM), the role of which is similar to that of the PDP in the IETF framework, but supports additional features such as state monitoring, event correlation and notification. AMs obtain policies from the PES and register Managed Resources (MR) that are interested in receiving policy directives from them. MRs provide two interfaces, *Sensors* (S) and *Effectors* (E), which represent the attributes that can be read from the resource and the management operations that can be performed on the resource respectively. AMs evaluate policies based on the sensed state of resources, which can invoke actions on MRs via the effector interface and consequently changing their behaviour.

Policies in the PMAC framework are specified using the Autonomic Computing Policy Language (ACPL), the structure of which is defined using an XML schema. They are ECA rules, where the conditional part is specified with a generic constraint language, which is also XML-based. The advantage of using such an approach is that the resulting policies can be parsed and type checked by XML parsers, thus making it attractive to applications that can consume XML format. Furthermore, the language can be extended relatively easy with new operations by modifying the schema and adding the extension operators. The problem with an XML representation is that policies can become quite verbose and not easily interpreted by human administrators.

Annex G: UniverSelf human network operator interview questions

This Section contains the questionnaire that will be sent to operators.

General characteristics of the network

First, some general questions related to your work will be asked.

Questions related to interviewee's work in general

1. What is your occupation?
2. What is the main responsibility of a [occupation]?
3. What are the main tasks of a [occupation]?
4. For how long have you been working in this domain *(i.e., work relevant to your skills in network operations)*
5. In what situations does your work require cooperation *(sharing information, assistance in your work; is cooperation formal and/or informal etc.)*
    a. with your team?
    b. with other stakeholders (inside and outside your company)?

Questions related to general characteristics of the network (as perceived in your work)

6. Network uncertainties
    a. Does the network, as an object of work, include uncertainties *(e.g., unpredictable events)*?
    b. What are the uncertainties?
7. Network as a complex system
    a. Is the network, as an object of work, a complex system *(e.g., it has many interdependencies)*?
    b. What are the factors that make the network complex?
8. Network as a dynamic system
    a. Is the network, as an object of work, a dynamic system *(e.g., situations change rapidly)*?
    b. What are the factors that make the network dynamic?
9. Changes in network
    a. How does the network change? (*large-scale changes, controlled and uncontrolled)*
10. With what criteria would you describe good functioning of a network?
11. Critical sources of error/factors resulting in network breakdown:
    a. How long do you think network would function without human operations?
    b. What would be the most likely reasons affecting the network deterioration and finally breakdown?

Questions related to your work with the network

12. What are the items that draw your attention in network functioning when managing the network?
13. What is the share of proactive and reactive actions in your work?
14. Is it possible to anticipate network problems? How?
15. Questions related to serious network problems
    a. What are serious network problems in your work?
    b. How do you evaluate the seriousness of a network problem?
    c. Are you satisfied with your possibilities of detecting a network problem?
16. Questions related to changes in the network
    a. How do network changes affect your work?
    b. How does network growth affect your work?
    c. In what situations do you have to make changes to the network (other reason than error correction)?
    d. How would you characterise situations which are especially hard to interpret (about whether a change is needed or not)?

17. Does your work include any other important tasks related to network operations, which have not been covered yet?
    a. In what situations [these tasks] have to be made?
    b. What are the situations like that are especially hard to interpret (about whether [these tasks] are needed or not)?
    c. Are you satisfied with your possibilities of performing [these tasks]?
18. Questions related to tools in your work with the network (*no need to name the tools!*)
    a. What are the good qualities of the tools?
    b. Are there poor qualities in the tools? If there are, what are they?

Questions related to work experience

19. What is a good network operator like? *What are the qualities needed in your occupation?*
20. What is the most straining aspect in your work?
21. What is the most rewarding aspect in your work?

Autonomic Functionalities

In the following, the pros and cons in possible future autonomic functionalities of the network, related to your work, will be asked from several perspectives.

22. What do you understand by autonomic functionalities?
23. Are there autonomic functionalities in network operation at the moment? If any, what are they?
24. What could be the greatest benefit(s) of autonomic functionalities?
25. What could be the greatest danger(s) of autonomic functionalities?
26. What autonomic functionalities would be most beneficial in your work? Why?
    a. *Do you encounter problems in your work that could be mitigated with autonomic functionalities? What would such functionalities be?*
    b. *Do you have tasks in your work that could be assisted with autonomic functionalities? What would such functionalities be?*
27. What are the present tasks that should not or cannot be left done automatically, why?

28. To what extent should the human operator know what the autonomic functionalities are doing? *Options, one of the following should be chosen if the interviewee doesn't give a free reply:*
    a. *Should everything be known – give examples of "everything" (such as each time the autonomic functionalities are affecting the network, then it should be known how it is affected or the like)?*
    b. *Is it enough to know of only the most important aspects (such as when the functionalities are related to a network problem or the like?)*
    c. *Is it enough to know only when autonomic functionalities fail and human intervention is needed?*

29. How should the human operator be informed about autonomic functionalities? Any kind of free ideas?
    a. *What should the notification informing the human operator be like (e.g., blinking icon, sound, text message, colour coded logs …)?*
    b. *How should the functioning of autonomic functionalities be shown?*
    c. *Should an autonomic functionality suggest operations to be done (for example to fix an error)?*

30. What are the characteristics in autonomic functionalities that would build trust on them? *What does the autonomic functionality need to provide in order to be trusted? [Show everything that is going on?]*
31. What are the shortcomings that would ruin trust on autonomic functionalities? *Would kind of mistakes would you tolerate?*
32. Is there a danger to trust too much the autonomic functionality? In what kind of situations?

Self-configuration in network functioning means automated configuration of components and adaptation of the system to dynamically changing conditions, without any acts on operator part.

33. What would be the pros and cons of self-configuration in your work? *(if you have time to ask: how about pros and cons in general and not only in your work?)*

Self-organisation in network functioning means a capability of the network to change its own organisation without any external or central dedicated control entity.

34. What would be the pros and cons of self-organisation in your work? *(if you have time to ask: how about pros and cons in general and not only in your work*?)

Self-optimisation in network functioning means that a component of system continuously tries to identify opportunities for improving its own performance and efficiency, without any acts on operator part.

35. What would be the pros and cons of self-optimisation in your work? *(if you have time to ask: how about pros and cons in general and not only in your work?)*

Self-healing in network functioning means discovery, diagnosis and actions that prevent disruptions in network functioning, without any acts on operator part.

36. What would be the pros and cons of self-healing in your work? *(if you have time to ask: how about pros and cons in general and not only in your work?)*

Self-protection in network functioning means the capability of the network to anticipate, detect, identify and protect itself against threats, without any acts on operator part.

37. What would be the pros and cons of self-protection in your work? *(if you have time to ask: how about pros and cons in general and not only in your work?)*

Assume a tool to which the human operator could insert high-level business goals and the tool would translate the goals into technology-specific terms autonomously so that the human would not need to deal with or know any technical details. *(Business goals may be related to the introduction of a new application, sets of user classes for the application, sets of Quality of Service (QoS) levels for each user class of the application, etc. This introduction can be related to a specific location, time period, volume of users, etc)*.

38. How would your work be related to this kind of tool? *(e.g. are you currently working with this kind of tasks in a technical level)*
39. What would be the pros and cons of such a tool in your work?
40. What type of input to such a tool would you envisage and what type of output (in terms of results related to your work)?

Annex H: State of the Art for Governance

Governance has appeared as the term to describe the new way (techniques and functionalities) for managing autonomous behaviours instead of being based on the stovepipe type of traditional network management. Prior to putting this in our analysis, we first recall that **policies** are intrinsic to **network governance**. In particular, network governance is tightly interlaced with the concept of policy continuum. Typically, business level policies are defined in the highest level, that is, they express **business objectives**. In the sequel and according to the policy continuum concept these policies are **propagated** to the network going through an arbitrary set of levels (related to different aspects of the management of a communications network) where they are being **transformed** into lower level policies, until they finally reach the element(s) in which to be enforced in terms of low level, technology-specific commands.

The words in bold in the previous paragraph have been highlighted with the purpose of outlining the main research activities for this task. The following subsections provide a brief state of the art of each of them.

Network Governance

Milestone MS24 [131] presented a detailed state of the art of autonomic architectures, frameworks and projects, including their contribution to governance. from the conclusion of that analysis was that many of the examined initiatives worked towards a real governance of self-managed behaviours aiming at resulting into autonomic networks and systems with the ability to dynamically adapt to changes in accordance with high-level business policies [132][133].

The analysis also showed that a great set of the examined architectures adopted traditional management solutions. They covered lower level policy based management with the focus being placed on the network side in particular, but they do not consider high level business goals nor offer a service view. Sometimes the aspect of governance was not even captured as an item of their research agenda [134].

Last but not least, little attention was paid to the novel type of dialogue between a human network operator (HNO) and the envisaged, self-managed network and in particular to the innovations and peculiarities that most probably the HNO will need to handle while migrating to this new type of management.

A more detailed elaboration on the state of the art can be found in the Milestone MS24 document.

Business language

In large-scale, distributed systems such as a production-level autonomic network, an implicit or explicit agreement between a client and a service provider specifies service level objectives, both as expressions of client requirements and as provider's assurances. These objectives are expressed in a high-level, service-, or application-specific manner rather than requiring clients to detail the necessary resources. There have already been some efforts aiming to consolidate high level information in the form of a business model or language.

The Common Information Model (CIM) [157] provides a common definition of management information for systems, networks, applications and services, and allows for vendor extensions. The CIM model is proposed by the Distributed Management Task Force (DMTF), which is an industry organization that develops, maintains and promotes standards for systems management in enterprise IT environments. CIM's common definitions enable vendors to exchange semantically rich management information between systems throughout the network. CIM is composed of a Specification and a Schema. The Schema provides the actual model descriptions, while the Specification defines the details for integration with other management models. In addition, the CIM Policy Model [158] based on the CIM model enables administrators to be able to represent policies in a vendor-independent and device-independent way. Thus, service level and other high- level policy abstractions can be supported, and be translated to device-specific configuration parameters at a lower level, across an aggregate of heterogeneous managed entities.

The Shared Information and Data model (SID) [136] is a set of comprehensive standardized information definitions, developed by the TeleManagement Forum (TMF), acting as the common language for building easy to integrate OSS (Operational Support System) and BSS (Business Support System) solutions. The SID model focuses on what on business entities and associated attribute definitions. The adoption of the SID as the industry's standard information model is growing rapidly, with many service providers, vendors, and systems integrators using the SID as the basis for their development and integration.

Often the business information is formalized in a Service Level Agreement (SLA). It is worth mentioned the standardisation work of Web Services Agreement (WS-Agreement), which defines a protocol and the respective abstract model for linking agreements to services, irrespective of the domain-specific details of contract terms. The EU funded SLA@SOI project, as part of its research agenda, has proposed a syntax grammar to express SLAs [137], including the business terms [139] needed for the relationships with the customers and with third parties.

Translation mechanisms

Network operators, on the other hand, require low-level, resource specific performance criteria that can easily be interpreted and provisioned. As a consequence, a framework that addresses the gap between high-level specification of client performance objectives and existing resource management infrastructures of network operators is traditionally required.

Former European projects have developed some work relative to translation. EFIPSANS project has implemented a mechanism that allows the translation from high-level goals to network policies [140]. A prototype has been developed, based on an ontology model, and implementing the translation by means of SWRL rules. Nevertheless, the generated policies described only the specifications for the network monitoring. That is, only a fraction of the service lifecycle was covered. SLA@SOI project presents a theoretical approach [141] for the translation of SLAs across layers, but an implementation has not yet been achieved.

Semantics & reasoning

Ontologies are formal representation of knowledge within a domain, that is, a description of concepts and relationships between them. The importance of the ontologies comes from the fact that they enable knowledge sharing and reuse [142]. In the last years, several ontology languages have been developed, both proprietary and standards-based.

The WWW Consortium (W3C) developed the Resource Description Framework (RDF) a language for encoding knowledge on Web pages to make it understandable to electronic agents searching for information. The Defense Advanced Research Projects Agency (DARPA), in conjunction with the W3C developed the DARPA Agent Markup Language (DAML) by extending RDF with more expressive constructs aimed at facilitating agent interaction on the Web.

Some years later, the W3C Web Ontology Working Group presented the OWL (Web Ontology Language) [143], which is one of the most popular ontology languages. OWL is a markup language to describe the properties and capabilities of the information in such a way that the descriptions can be interpreted by a computer system in an automated manner. OWL allows applications to automatically discover, compose, and invoke services in a dynamic services-oriented environment. It can be enhanced with an inference engine in order to allow reasoning. In addition, OWL is maintaining as much compatibility as possible with the pre-existing languages, including RDF and DAML. Recently, the W3C OWL Working Group, a follow-on group of W3C Web Ontology Working Group, developed the OWL2. OWL2 is an extension and revision of OWL and it is designed to facilitate ontology development and sharing via the Web, with the ultimate goal of making Web content more accessible to machines.

SWRL (Semantic Web Rule Language) [144] is a proposal for a Semantic Web rules-language, combining the Ontology Web Language (OWL) and the Rule Markup Language (RuleML) [145]. Both rules and ontologies are necessary for the service descriptions and play complementary roles: while ontologies are useful for representing hierarchical categorisation of services overall and of their inputs and outputs, rules are useful for representing contingent features such as business policies, or the relationship between pre-conditions and post-conditions. SWRL enables to ``build rules on top of ontologies'': it enables rules to have access to ontological definitions for vocabulary primitives (e.g., predicates and individual constants) used by the rules.

Policy model, Policy language & Policy framework

Network Governance heavy relies on a policy based Management model for defining and controlling the network behaviour shifting from classical paradigms focused on individual devices/entities management. Policies are a set of pre-defined rules (defined actions to be triggered when a set of conditions are fulfilled) that

govern resources, including conditions and actions that are established by the administrator with parameters that determine when the policies are to be implemented in the network. In the case of a Telco Operator, policies are defined based on the high-level business objectives of the services on one hand and on the other hand on the SLA agreed with its customers and third party Service Providers. Policies allow changing the behaviour of a system without changing its implementation, creating adaptable systems whose behaviour can be altered dynamically.

Although Policy Based management (PBM [146]) seems to fulfil some of the governance requirements, current architectures need to be enhanced to include the key concept of knowledge and context awareness. Most PBNM (Policy Based Network Management) systems define low-level policies that manage changes in routers, switches or firewalls. The link between the business needs and the configuration of the network resources and services is missing. Novel concepts such as the usage of the network context to determine the modifications in network services and resources are not present in these approaches.
In contrast, the DEN-ng information model [147] includes policy and context sub-models, and has been expressly constructed to facilitate the generation of ontologies, so that reasoning about policies constructed from the model may be done. A policy language can also be derived from the model. This holistic construction of information model, ontologies and policies makes it suitable for the government of management entities. DEN-ng is built on the 3 principles governing the system: capabilities, constraints and context with the following precisions. Capabilities normalize the set of functions available in the same type of managed object made by different vendors. Context defines the current environment, objectives, obligations and policies governing the behaviour of the system. Constraints define which capabilities can be used as a function of a particular context. These 3 principles enable the behaviour of the system to be abstractly modelled.

DEN-ng specific improvement is to define not only the static characteristics of the managed entities but also dynamic ones (behaviour) in a manner independent of any specific type of repository, software usage or access protocol.
Related with DEN-ng is the concept of the Policy Continuum [148], which defines a framework for the development of stratified sets of policy languages, tied together by a common information model. This helps ensure the consistency of the policies deployed across a system and facilitates policy-based analysis processes. This continuum concept seems to be suitable for the achievement of the objectives of Task 2.3.

The DEN-ng information model and the concept of Policy Continuum are realised in the context of FOCALE autonomic architecture [133]. The FOCALE architecture is based on five key concepts:

- The use of a shared information model capable of harmonizing the different data models that are used in Operational and Business Support Systems (OSSs and BSSs).
- The knowledge extracted from information and data models is augmented with the use of ontologies in order to be capable of representing the detailed semantics required to reason about behaviour.
- The existence of an information model able to generate ontologies for governing behaviour.
- The policy model is linked to a context model, so that policies can be written that adapt offered resources and services according to context changes.
- The use of machine learning and reasoning.

The FOCALE autonomic architecture is depicted in Figure 61. The FOCALE architecture by using model-based translation tries to coordinate different management mechanisms and consequently to manage end-to-end services spanning through multiple networks and network technologies. In addition, the system and its environment are continually analysed with respect to business objectives. As depicted in Figure 61 the main adaptive control loops is realized by the interaction between the context manager, policy manager and autonomic manager. The context manager detects changes in the network, user needs and business goals; these context changes in turn trigger a new set of policies to take over control of the autonomic system, which enables the services and resources provided by the autonomic system to adapt to these new needs given that appropriate policies are available for the new context.

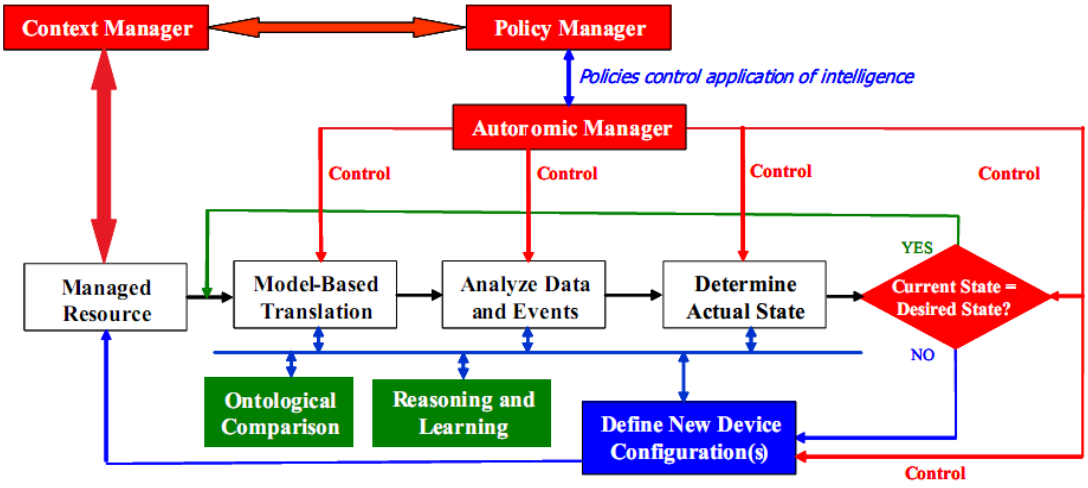


Figure 61. FOCALE autonomic architecture.

Finally, it is worth mention here the commercial solution PCRF (Policy and Charging Rules Function), defined as a node that at runtime determines the policy rules to be applied in a multimedia network. Currently different implementations are available from different vendors, such Huawei, Openet, Camiant/Tekelec.

Conflict resolution

The use of a policy based management system may potentially lead to the appearance of conflicts between policies. These conflicts may be resolved informally by human managers, but the goal of the Network Governance Framework is to provide an automated mechanism to recognise them and resolve them. A general purpose algorithm for policy conflict detection and harmonization has been implemented in KaOS project [149]. Using DAML policy ontologies, the method is based on an assignment of priorities to policies. In case of conflict, the numeric priority and the update times are the criteria used to determine the precedence.

[150] and [151] present an exhaustive work where the use of information models and ontologies that represent relationships between policy components facilitate the detection of conflicts. The approach includes the design of an analysis algorithm for the policy continuum concept that could be used to analyse policies at the different levels of the continuum.

A different approach is introduced in [152], which describes a framework for policy analysis, conflict detection and resolution applied to the QoS management for DiffServ networks. Event Calculus is used as the underlying forma representation of policies, systems behaviour and rules to detect the presence of conflicts.

Distribution & enforcement mechanisms

The different approaches have found different solutions to the problem of distributing the policies to the managed elements. EFIPSANS project [134] has defined its own mechanism for the communications between managed entities, the ONIX (Overlay Network for Information Exchange). ONIX is a distributed system of servers that supports publish/subscribe, query and find type of services for Information and Knowledge such as Capabilities of network elements, Profiles, Goals and Policies of the autonomic network, pointers to resources and Data, and other types of Information/Knowledge.

DEN-ng defines a hierarchy for policy management application [153]. It splits the functionality of a Policy Enforcement Point into a Policy Execution Point (PXP – that implements specified policy actions) and a Policy Verification Point (PVP – that ensures that the policy actions were executed correctly and with the expected results). A Policy Decision Point (PDP) distributes various levels of decision-making amount global and local scopes. DEN-ng defines a PolicyServer as consisting of at least one PDP, PXP and PVP, while a Policy Broker is the entity that enables multiple Policy Servers to negotiate and exchange policies.