EU FP7 ICT STREP SEMAFOUR (316384)

INFSO-ICT-316384 SEMAFOUR

D5.1

Integrated SON Management Requirements and Basic Concepts

Contractual Date of Delivery to the EC:	November 30, 2013
Actual Date of Delivery to the EC:	December 13, 2013
Work Package	WP5 – Integrated SON Management
Participants:	NSN-D, ATE, EAB, FT, TID, TNO, TUBS
Authors	**Sana BEN JEMAA,** Christoph FRENZEL, Dario GÖTZ, Beatriz GONZÁLEZ, Sören HAHN, Ovidiu IACOBOAIEA, Simon LOHMÜLLER, Tanneke OUBOTER, Lars Christoph SCHMELZ, Ana SIERRA, Hans VAN DEN BERG
Reviewers	Zwi ALTMAN, Andreas EISENBLÄTTER, Kathleen SPAEY
Estimated Person Months:	28
Dissemination Level	Public
Nature	Report
Version	1.0
Total number of pages:	50

Abstract: Integrated SON management is an integral part of the SEMAFOUR unified self-management system. This document introduces the basic functional architecture of the four components of integrated SON management: Policy-based SON Management, Operational SON Coordination, Monitoring and Diagnosis, and the Decision Support System. The interworking and interfaces between these components are described.

Keywords: Self-organisation, SON, SON Coordination, Policy-based Management, Monitoring and Diagnosis, Decision Support System

Executive Summary

In spite of the great benefits expected from the SON technology in terms of cost (capital, operational and implementational expenditures) and energy savings as well as the anticipated improvement in the network capacity and in the user experience, at the current state of SON deployment it is sometimes difficult for mobile operators to quantify such advantages. The SON implementation in real networks can be complex as many stakeholders and different processes are affected, besides the fact that many tasks are outsourced to third parties or manufacturers.

The configuration, operation and management of a SON system is a challenging task. Firstly, this is reasoned by the increasing number of SON functions, being operated in a non-coordinated manner. Secondly, these SON functions operate across Radio Access Technologies (RATs), and across layers within these RATs. Thirdly, the SON functions may come from different manufacturers, and may be designed based on different assumptions. To get confidence into an autonomously operating SON system, the network operators require a common means to define business and technical goals, objectives and targets for a SON-enabled network. The operators need a clear visibility of what is being changed at the network elements by the SON functions and the impact of these changes on the network performance. Also the ability to undo modifications that may be negative is of great interest.

The perspective of SEMAFOUR Work Package 5 is therefore to highlight the importance of an integrated management system for SON. SEMAFOUR WP5 positions itself to address the complexity associated with integrating SON into existing heterogeneous mobile networks. WP5 develops concepts, methods and algorithms for an integrated SON management system, consisting of four major functional areas: (i) Policy-based SON Management (PBSM, transformation of operator-defined general network-oriented objectives into operational rules and policies for SON functions); (ii) Operational SON Coordination (SONCO, real-time detection, analysis and resolution of conflicts occurring between operational SON function instances); (iii) Decision Support System (DSS, provisioning of recommendations to the operators to modify and enhance the network); and (iv) Monitoring and Diagnosis (MD, continuous monitoring and analysis of network configuration and performance; providing input to other integrated SON management components).

This deliverable describes the evolution of the functional architecture (initiated in D2.2 [10]) of the four functional areas that constitute the SEMAFOUR integrated SON management. In addition, the interworking and interfaces among these modules are presented. Preliminary results on the simulation, implementation and demonstration activities are included in this document.

List of Contributors

Partner	Name	E-mail
NSN-D	Christoph Frenzel Simon Lohmüller Lars Christoph Schmelz	christoph.frenzel.ext@nsn.com simon.lohmueller.ext@nsn.com christoph.schmelz@nsn.com
ATE	Andreas Eisenblätter Dario Götz	eisenblaetter@atesio.de goetz@atesio.de
iMinds	Kathleen Spaey	kathleen.spaey@uantwerpen.be
FT	Ovidiu Iacoboaiea Sana Ben Jemaa	ovidiu.iacoboaiea@orange.com sana.benjemaa@orange.com
TID	Beatriz González Rodríguez Ana María Sierra Díaz	bgr@tid.es asd@tid.es
TNO	Tanneke Ouboter Hans van den Berg	tanneke.ouboter@tno.nl j.l.vandenberg@tno.nl
TUBS	Sören Hahn	hahn@ifn.ing.tu-bs.de

List of Acronyms and Abbreviations

2G	2nd Generation mobile wireless communication system (GSM, GPRS, EDGE)
3G	3rd Generation mobile wireless communication system (UMTS, HSPA)
3GPP	3rd Generation Partnership Project
ANR	Automatic Neighbour Relations
CAPEX	CAPital EXpenditure
CIO	Cell Individual Offset
CCO	Coverage and Capacity Optimisation
DSS	Decision Support System
ECA	Event-Condition-Action
EDGE	Enhanced Data rates for GSM Evolution
EM	Element Manager
eNodeB	LTE radio base station
E-UTRAN	Evolved UTRAN
GERAN	GSM EDGE Radio Access Network
GPRS	General Packet Radio Service
GSM	Global System for Mobile communication
HetNet	Heterogeneous Networks
HO	Handover Optimisation
HSPA	High Speed Packet Access
ICIC	Inter-Cell Interference Coordination
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IMPEX	IMPlementational EXpenditure
IRAT	Inter-RAT
KPI	Key Performance Indicator
KUI	Key User Indicator
LAN	Local Area Network
LTE	Long Term Evolution
LTE-A	Long Term Evolution – Advanced
MD	Monitoring and Diagnosis
MIMO	Multiple Input Multiple Output
MLB	Mobility Load Balancing
MRO	Mobility Robustness Optimisation
NE	Network Element
NEM	Network Element Manager
NGMN	Next Generation Mobile Networks
NM	Network Management
OAM	Operation, Administration and Maintenance
OPEX	OPerational EXpenditure
PAN	Personal Area Network
PBSM	Policy Based SON Management
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PCI	Physical Cell Identifier
QoE	Quality of Experience
QoS	Quality of Service
RACH	Random Access Channel
RAN	Radio Access Network

RAT	Radio Access Technology
RRC	Radio Resource Control
SCP	SON function Configuration Parameter
SCV	SON function Configuration parameter Value
SeNB	Serving eNodeB
SLA	Service Level Agreement
SON	Self-Organising Network
SONCO	SON Coordinator
TD-LTE	Time Division duplex LTE
TeNB	Target eNodeB
TTT	Time-To-Trigger
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
UTRAN	UMTS Terrestrial Radio Access Network
WLAN	Wireless Local Area Network
WP	Work Package

Table of Contents

1 Introduction

Work Package 5 has the objective to configure, operate and manage individual SON functions, allowing their conflict-free operation according to the high-level, network-wide business and technical goals as defined by the mobile network operators. This objective is represented by the integrated SON management system being an integral part of the SEMAFOUR unified self-management system.

In SEMAFOUR D2.2 [10] the basic technical and business requirements of the four functional areas of integrated SON management (namely, Policy-based SON Management (PBSM), Operational SON Coordination (SONCO), Monitoring and Diagnosis (MD), and the Decision Support System (DSS)) have been defined. Furthermore, an initial high-level functional architecture for integrated SON management has been introduced and described. This deliverable introduces a more detailed functional architecture for the four functional areas of integrated SON management, describes the interfaces between these functional areas, and provides initial concepts on the interworking and the required information exchange between the functional areas, the network operator, and the SON functions. Furthermore, first results on the simulations that have been conducted are explained.

A global view of the functional architecture of the unified self-management system is depicted in Figure 1. The integrated SON management system integrates and coordinates the multitude of multi-RAT and multi-layer SON functions, including those SON functions developed within SEMAFOUR Work Package 4. This allows the mobile network operators to move their operational focus to a higher, more global level, which is more transparent to the specifics of the underlying network technologies and cellular layout. For this purpose, four functional areas have been specified (see also D2.2 [10]):

- *Policy Transformation and Supervision:* The work in this functional area has been labelled Policy-based SON Management (PBSM). High-level objectives and goals describing the desired network and SON system behaviour are transformed into dedicated policies and rules that control the individual SON functions in such a way that the high-level objectives and goals are met. Such objectives are, for example, stated in terms of network performance or user satisfaction.
- *Operational SON Coordination (SONCO):* Different SON functions operating concurrently in the network can interact such that they negatively impact network performance. For example, they may request for conflicting modification of the same configuration parameters with different values. Operational SON Coordination aims at a resolution of such conflicts during run-time of the SON functions.
- *Decision Support System (DSS):* SON functions cannot resolve all problems occurring during the operation of the network. Some tasks require human interaction. The Decision Support System shall advise the human operator such that these tasks can be conducted more efficiently and effectively.
- *Monitoring and Diagnosis (MD):* The functionality of monitoring as described in SEMAFOUR deliverable D2.2 [10] has been enhanced by a diagnosis component. The goal is to provide one functional block to acquire, analyse, and process information from the network and the OAM system (e.g., performance measurements, alarms, and configuration data) in such a way that it can directly be used as input to the other functional areas of WP5, namely, PBSM, SONCO and DSS.

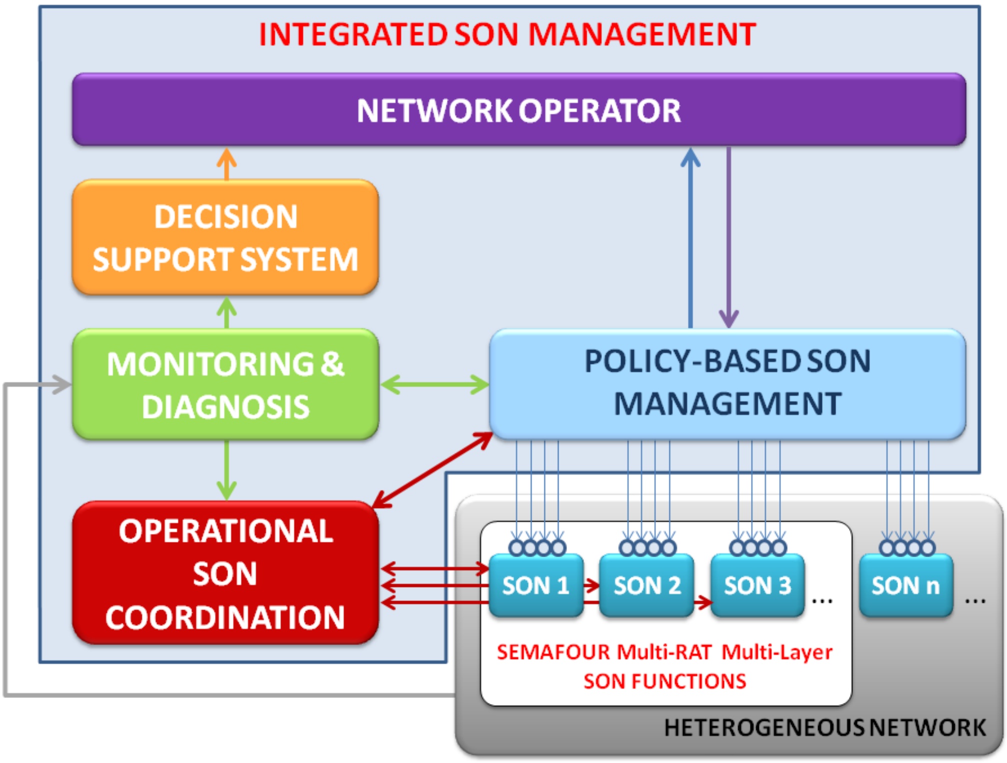

Figure 1: Functional view of SEMAFOUR unified self-management system

On the basis of the four functional areas presented above, the document is organised as follows:

Chapter 2 is devoted to the PBSM and identifies the scope of this functional area, describes its functional architecture, defines the main concepts related to the PBSM and introduces the implementation approach. Also the PBSM external interfaces towards the network operator, the SON functions, and the SON manufacturer, are described.

Chapter 3 focuses on the SONCO. After a summary of the state of the art on SON coordination, the scope of this functional area and the main associated concepts are defined, the functional description including the interfaces is provided, the first coordination scenarios are presented and the implementation options are explained.

Chapter 4 provides the functional architecture of the DSS as well as the description of the initial DSS demonstrator set up in the first year of the project.

Chapter 5 presents the MD, defining its scope and main building blocks, including also some details about the first implementation activities.

Finally, Chapter 6 summarises the work done until now on the PBSM, the SONCO, the DDS and the MD and outlines the next steps planned for the functional areas of integrated SON management.

2 Policy-based SON Management

Policy Based SON Management (PBSM) is one major part of the SEMAFOUR integrated SON management system. The PBSM's role is to contribute to making the network perform efficiently according to the operator’s goals (high-level network-oriented objectives). The PBSM transforms high-level operator’s goals into a SON policy, which consists of a set of dedicated technical rules for the individual components of the SON system, at the level of individual SON functions and their configuration settings. This transformation process is specific to the individual locations, cells, sectors, or sites in the network a SON functions is operating at, and also specific to events and conditions that can trigger SON functions. The SON function instances (i.e., the instantiations of SON function algorithms at specific locations and times, with the respective configuration settings) will then be coordinated by the SON Coordinator (SONCO) when acting on the network. To successfully fulfil its role, the PBSM must perform this transformation process automatically, as part of the whole integrated SON management system. This transformation has to be done in an appropriate way, in order to minimise the potential for conflicts between different instances of SON functions.

A first version of a PBSM functional architecture, together with business, functional and non-functional requirements, have been described in SEMAFOUR deliverable D2.2 [10]. In this document, the focus is on improvements to this first architecture, and a more detailed description of the different components, together with a description of the first simulation results related to PBSM. In Section 2.1 the scope of PBSM is introduced, providing the approach of a layered architecture. Section 2.2 explains the PBSM functional architecture in detail, in particular, the input provided to PBSM by the network operator and the SON function manufacturer, the functional blocks of PBSM, and the interfaces of PBSM to other functional areas of integrated SON management. Finally, in Section 2.3, the approach and results of a preliminary PBSM implementations are described.

2.1 Scope

Deriving the SON policy in support of the mobile operator comprises linking the operator’s goals (business, strategic and technological ones) to the network configuration and performance, identifying relevant SON functions, and deciding on how to instrument them for the intended purposes. Assuming that the operator’s goals are stored in operator specific repositories located in a higher management layer, two distinct logical layers are foreseen within the PBSM to provide contexts for each of the necessary steps (see also Figure 2):

- The *intermediate layer* represents a network view, contains technological objectives and provides a detailed network description based on which the network performance can be assessed. This layer is unaware of SON functions.
- The *bottom layer* represents the SON view, contains the SON policy or SON function configurations for controlling the SON functions acting in the network.

Mapping a given operator goal into a dedicated configuration of a SON function, such that this SON function contributes to the operator goal, comprises relating the given goal to the network configuration and performance, identifying relevant SON functions, and deciding on how to instrument them for the intended purposes. Three logical layers are foreseen to provide contexts for each of the necessary steps (see also Figure 3).

The Top Layer, through which the operator goals and a system model are provided, is part of the network operator domain and not considered being part of PBSM within SEMAFOUR. It is merely seen as a data and information provider for PBSM. The Intermediate Layer (system layer) provides a detailed technical network description with which network performance may be assessed. The Bottom Layer (SON policy) contains the means to control and configure SON functions, or SON function instances, respectively, that act in the network.

The logical layers within PBSM (intermediate and bottom layers) shall be clearly separated such that each of them may stand for itself. This postulate allows for splitting the whole PBSM transformation process into smaller pieces, which can be investigated independently within their respective contexts. It is foreseen to design the layers as being self-contained and independent in order to achieve this separation. The layers are intended to be 'self-contained' in the sense that focusing on one layer, these representations of the operator's goals may be interpreted within the context of that layer without relying on information from the other ones.

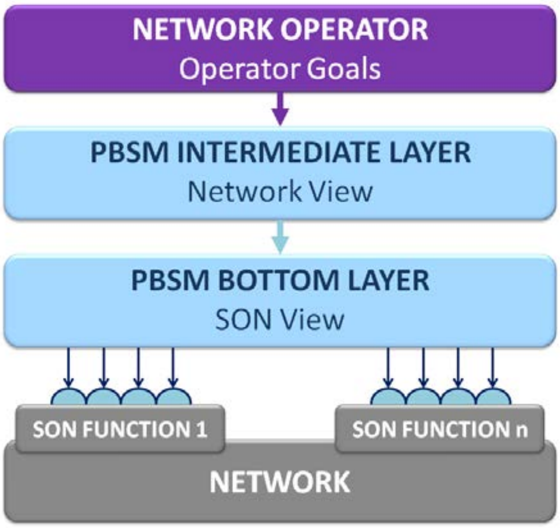

Figure 2: PBSM Layered Structure

Each layer works with information that is processed and held in accordance with layer-specific syntax / semantics. The technical specification of a SON policy, for example, relies on the definition of a generic SON function. The layers shall be 'independent' in their specific syntax / semantics. This means, for instance, that for the processing of information inside the intermediate layer (e.g., information from MD on the KPI composition), it does not matter in which form SON function configurations are handled inside the bottom layer. In particular, it is possible to change or replace one layer without impacting the functioning of the other layers. For example, the semantics of the intermediate layer does not need to be changed if a new SON function is introduced to the system.

Moreover, each layer shall be 'complete' in the sense that, at a given point in time (when all current transformation processes are finished), the layer contains all relevant data necessary to express the operator's goals as well as possible at that layer.

Communication between layers necessitates mappings between the corresponding syntax and semantics. Operator goals or the system model passed down to the intermediate layer need to be transformed before having a meaning within the context of the intermediate layer. Another mapping is needed when the SON policy is created from technological objectives and the SON function models at the intermediate layer. That is, transformation processes play a distinguished role in this setup. Obviously, an understanding of the syntax and the semantics of the source and target layers is necessary when mapping from one layer to another.

Mapping or transformation processes happen between adjacent layers. If the transformation processes between two layers were part of either of the layers, then the logical separation between the layers would become blurred. In particular, the layers would not be independent anymore. Considering, for example, the transformation from the highest layer (at the network operator) to the intermediate layer (in the PBSM) as a part of the latter, then the addition of a new category of operator goals would entail changes to the intermediate layer. Whereas, if the transformation is considered as something between layers, then the transformation process needs an update, but not the intermediate layer.

2.2 Functional Architecture

In this section the preliminary functional architecture of the SEMAFOUR Policy-based SON Management approach is given.

Based on the reasoning behind the PBSM layered structure described in the previous section, the PBSM functional architecture block diagram has been developed and it is depicted in Figure 3.

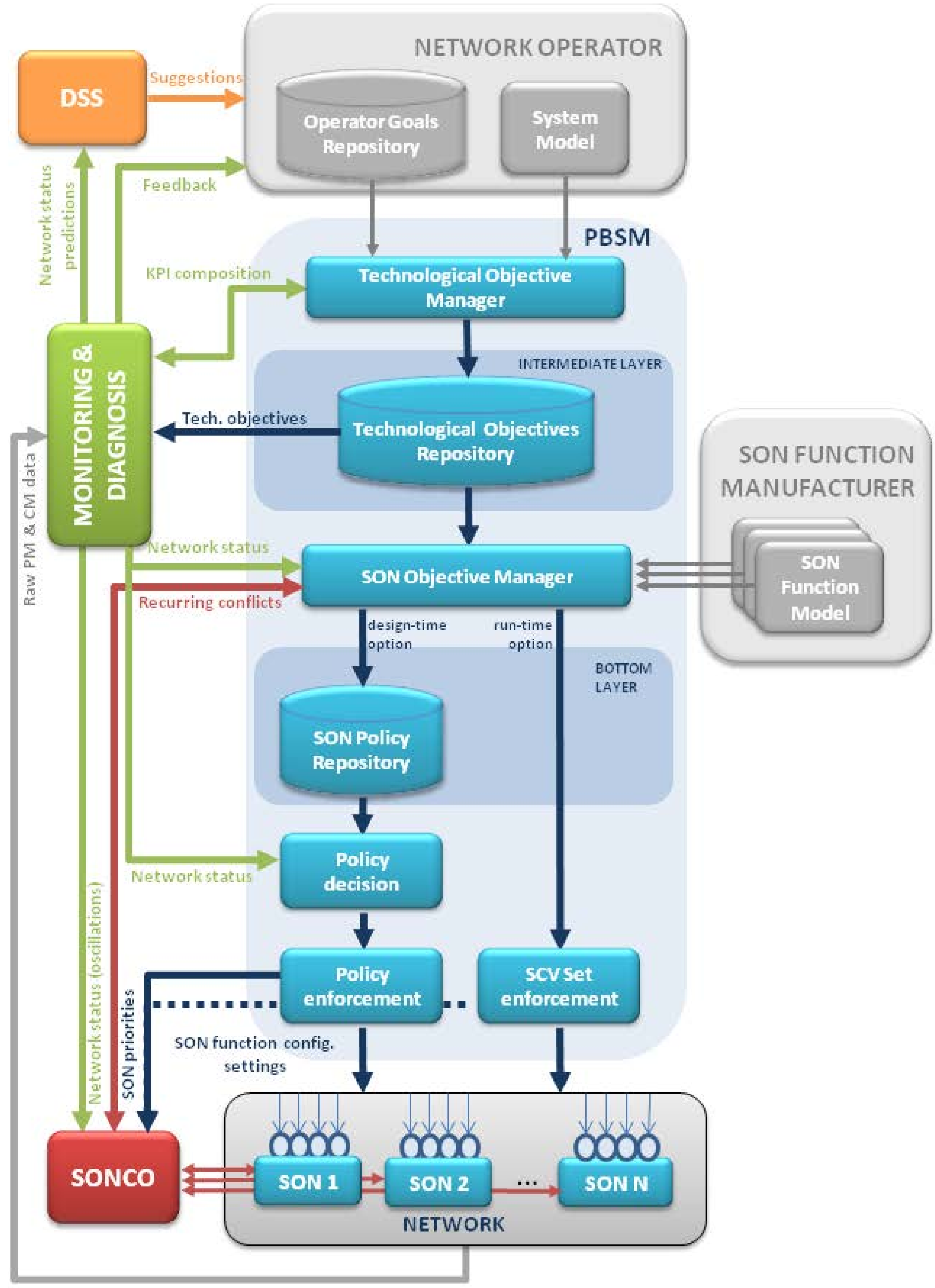

Figure 3: High-level PBSM Functional Architecture

There have been several refinements with respect to the initial PBSM functional architecture as described in D2.2. In particular, it has been decided to move the operator goals repository to the network operator domain, where also a network system model has been explicitly added. This decision based on the fact that operator goals and the system model cannot be intrinsic information of PBSM, but are information required as input to PBSM. The information content of the operator goals and the system model are independent of the actual implementation of PBSM. As input to the SON Objective Manager (denominated as Transformation 2 in D2.2 [10]) SON function models are foreseen in the updated functional architecture. These are necessary to provide information about the SON functions implemented in the network to the PBSM transformation process. Finally, two options for providing new configurations to the SON functions have been developed: a method using a SON policy that decides on the appropriate SON function Configuration Value sets (SCV Sets, see also the explanation of the SON function in Paragraph 2.2.4) to be deployed, and a run-time method directly providing SCV Sets to the SON functions. In the remainder of this section, the refined functional architecture is explained in detail.

2.2.1 Input from the Network Operator

The PBSM requires two different types of information from the network operator in order to perform the selection, configuration and prioritisation of the SON functions that will act on the network to improve its performance when and where it is required: Operator Goals and a System Model.

Operator Goal

An *Operator Goal* describes the targets a mobile network operator has regarding its business, technical and customer strategy, and with respect to issues related to regulatory authorities. Furthermore, operator goals may reflect Service Level Agreements (SLAs) with 3rd party providers of content and services. Within the scope of SEMAFOUR only those operator goals are considered that can be solved or (at least partially) addressed by means of the network environment. Operator goals may be related to:

- Operational, implementation and capital expenditure (OPEX, IMPEX, and CAPEX): the network can contribute to these goals, e.g., through minimal maintenance and energy cost, or appropriate planning and deployment.
- Customer satisfaction: service availability, and swift, reliable service accessibility and delivery are important factors contributing to customer satisfaction. Terms often used in this context are Quality of Service (QoS) and Quality of Experience (QoE).
- Regulatory authority requirements: the regulatory authorities in each country have dedicated requirements or restrictions a network operator has to follow, for example, regarding certain network coverage requirements that need to be achieved, restrictions regarding electromagnetic compatibility and pollution, or performance requirements that need to be achieved in competition rankings.

Operator goals may sometimes be defined concretely regarding certain measurements or timings to be achieved, others may be formulated vaguely. Distinct operator goals may furthermore have a strong interrelation or even depend on each other, and the determination of a ranking between the operator goals may be difficult. The challenge for PBSM, in particular the Technological Objective Manager component, is therefore to extract from this multitude of goals measurable or calculable technical objectives, their priorities, and conditions under which these objectives and priorities apply.

The information described within an operator goal can be structured according to the following information types and categories:

- High-level goal concept: Network characteristics that best suit the high-level goal (e.g. coverage, capacity, interference, mobility robustness, drop calls ratio, …),
- High-level goal impact area: List of location types or network elements installed in the geographical area linked to the high-level goal,
- High-level goal target value: Target threshold associated to the high-level goal concept. Different target values are possible for the same high-level goal, depending on the time-frame or the priority for example,
- High-level goal priority: The mobile network operator can associate different priorities to the high-level goals by means of this indicator,
- High-level goal time-frame: Time period during which the high-level goal must be fulfilled,
- Additional conditions (e.g., related to the weather…) or context information during which the high-level goal should be fulfilled.

System Model

To be able to create technological objectives, i.e., KPI targets, priorities, restrictions, and conditions from the operator goals within PBSM, it is necessary to provide information about the current mobile radio network system deployment to the PBSM transformation process. This *System Model* is typically stored in network-wide configuration management and network planning systems and databases. Some of the information contained in such a system model is listed below:

- Available technologies (e.g., 2G, 3G, 4G, …),
- Available layer or layers for each technology (e.g., macro cell, microcell, picocell, all, …),
- Available frequency band for each technology and layer (e.g., 2600 MHz for LTE macro layer, …),
- Available location types for each technology and layer (e.g., urban, suburban, indoor, outdoor, …),
- Available configuration types for each technology and layer,
- Additional information for further refinement.

2.2.2 Input from the SON Function Manufacturer

The aim of PBSM is to create configuration settings for the SON functions in such a way that they contribute to the operator goals. Therefore, the system needs to be aware of the relationship between different SCV Sets and their impact on KPIs. This relationship information is contained in the *SON Function Model*, which is to be provided by the SON function manufacturer. For each KPI target a SON function can contribute to (note: a SON function does not need to contribute to every possible KPI target) the SON function model provides one dedicated SCV Set, i.e., one dedicated value for each configuration parameter of the SON function that can be modified. The SON function model provided by the SON function manufacturer needs to be described in a network scenario independent way, or the model has to contain SCV Sets for different scenarios (e.g., dependent on the base station type where the SON function instance operates on). The SON function model can thereby be provided in form of a table. An example with three SON function models for Load Balancing (MLB), Mobility Robustness Optimisation (MRO) and Coverage and Capacity Optimisation (CCO) is given in Table 2 in Section 2.3.

2.2.3 PBSM Building Blocks

The PBSM consists of different building blocks that are described below. Firstly, the *Technological Objective Manager* is the transformation function between the network operator and the PBSM intermediate layer. Secondly, the *SON Objective Manager* is the transformation function between the intermediate layer and the bottom layer. For the bottom layer between SON Objective Manager and the SON functions, two different implementation options have been developed (design-time and run-time option). The corresponding building blocks are therefore introduced for each of these implementation options. The input and output information and data for each building block are summarised in Table 1.

Technological Objective Manager

The Technological Objective Manager is the function that automatically derives the technological objectives and their target values from the operator goals and the system model. The operator goals as described above are not usable to operate a SON-enabled network, as they mostly cannot be directly measured, calculated or interpreted by technical entities or processes. For this reason it is necessary to transform the operator goals into numerical expressions, denominated as Technological Objectives. A technological objective can consist of several components, including target values or ranges for measurements and settings in the network (e.g., KPI target or range values), priorities associated with these target values, and context information (e.g., conditions, restrictions) under which these target values apply.

This context information is an important part of technological objectives, as it allows defining different targets for, e.g., different times of the day, locations, network types, cell types, user groups etc. In the following some examples for technological objectives are given:

- With Priority 1 (highest priority) the Drop Call Ratio for premium users in an urban location during peak hours should be minimised.
- With Priority 2 the Handover Success Ratio during peak hours should be maximised.

- With Priority 3 the Cell Load during peak hours should be minimised.
- With Priority 4 (lowest priority) the Energy Consumption of macro cells in urban locations during low traffic hours should be minimised.

For each of the examples given above, concrete numbers can be defined, for example:

- Priority levels from 1 to 4
- Dedicated KPI targets, e.g., Drop Call Ratio < 2,5% or Handover Success Ratio > 98%
- Urban location: all cells within a certain area, or a list of cell identifiers
- Peak hour: from Monday to Thursday from 08:00 till 18:59, Friday from 08:00 till 20.59, Saturday from 09:30 till 20:59, Sunday from 09:30 till 18:59

All technological objectives taken together shall describe the desired network behaviour (either by stating what is desired or what is to be avoided). Each individual objective contributes to achieving this desired behaviour and therefore to the operator goals. When a network state is checked against all technological objectives, then a pointed list of discrepancies between what is desired and what is observed can be derived. These discrepancies can then be used (at the bottom level) to trigger changes to the network by means of SON functions/instances execution.

Note that a technical objective does not include any information on how the targets can be reached since at the intermediate layer there is only information available on the system / network, but not on the means to optimise or configure the network, i.e., the SON functions that are available. So the ultimate step to be able to automatically control and operate the SON functions is to map the technological objectives to the configuration of the SON functions in such a way that the operation of the SON functions (i.e., modifying the network configuration) contributes to the technical objectives. This instrumentation is further elaborated below.

The technological objective manager stores the derived technological objectives, with an association to the operator goals, in the *Technological Objectives Repository*, which represents the intermediate layer of PBSM as described in Section 2.1. Furthermore, the technological objectives are provided as input to the Monitoring and Diagnosis (MD) function, to enable a suitable monitoring and analysis functionality.

SON Objective Manager

By means of performing a reasoning process the SON Objective Manager determines the SON policy (including priorities and associated time frame), or directly the appropriate SCV sets, according to the technological objectives, the current network operational context and past experiences (recurring conflicts between SON function instances, if any). To decide the best SON function configuration settings an additional input to the SON objective manager is needed, namely, the SON function model (cf. Paragraph 2.2.2) It contains information that shows the SON Objective Manager how different SCV Sets of a specific SON function satisfy possible operator objectives.

The SON Objective Manager can be operated in two different modes: a design-time option, and a run-time option. Depending on the option, the reasoning process within the SON Objective Manager and the corresponding output to the underlying building blocks are different.

In the *Design-Time Option* the SON Objective Manager creates a SON policy before the instantiation of the SON functions. The SON policy is thereby defined within the SEMAFOUR project as a set of Event-Condition-Action (ECA) rules. Each of these rules consists of an EVENT on which it is triggered; one or more CONDITIONS which are to be checked, and an ACTION that is taken when the conditions are met. For the SON policy within PBSM, the events and the conditions are derived from the technological objectives and their associated target values respectively. The action part of an ECA rule states instructions. In the case of PBSM, these instructions are the configuration values to be deployed to the SON functions or their instantiations, respectively. Note that there might be ECA rules with an empty condition part, i.e., the action part of the ECA rule is always executed if the triggering event has taken place. Furthermore note that a policy does not need to explicitly state its purpose or goal.

The SON policy hence contains all information required to control the SON system, i.e., the multitude of SON function instances, according to the technological objectives of the network operator, i.e., KPI targets, KPI priorities, conditions (e.g., time, location, network status) under which KPI targets and priorities apply, and additional restrictions that may apply. A single ECA rule describes in detail what is to be done under certain conditions. This single ECA rule thereby has to be in line with the technological objectives, but whether or not this is the case cannot generally be determined from considering this rule in isolation. Therefore, the SON policy including the complete set of SON related ECA rules needs to be consistent in such a way that all technological objectives are addressed. This implies that the SON policy has to be defined conflict free, i.e., conflicting technological objectives need to be resolved when the SON policy is created.

Within the design-time option, the SON policy has to be recomputed in case KPI targets and their prioritisation change, or if the definitions, properties or allocation of conditions and restrictions change. With the design-time option it is necessary to deploy a policy decision and enforcement point since the ECA rules contained in the SON policy need to be evaluated, and the corresponding action enforced to the SON functions. These functional blocks are explained below.

- *SON Policy Repository*: This repository, located at the bottom layer, stores the SON policy that will be enforced on the network by the policy enforcement module.
- *Policy Decision*: This entity (Policy Decision Point, PDP, is located below the bottom layer) listens for trigger events and, once such an event occurs and the associated condition is fulfilled, activates the corresponding policy actions within the SON policy.
- *Policy Enforcement*: This entity (Policy Enforcement Point, PEP, is located below the policy decision block) is responsible for providing the SON functions and the SONCO with the appropriate actions (SON instances with their associated configuration settings) coming from the SON policy.

In the *Run-Time Option* the SON Objective Manager performs not only the reasoning and mapping processes between the technological objectives and the SON function models, but also acts as decision point with respect to selecting the appropriate SCV Sets to be deployed. With the run-time option, the output of the SON Objective Manager does not consist of a SON policy, but directly of SCV Sets that are deployed to, and enforced at the SON functions.

In contrast to the design-time option, the SON Objective Manager has to perform the reasoning and mapping process not only in case the technological objectives or the SON function models change, but also in case conditions or restrictions change, since based on these the appropriated SCV Sets are selected instantaneously, and directly deployed. On the other hand, in the run-time option not always a full mapping between all technological objectives and all SON function models has to be performed, but only these affected by the changes have to be updated and deployed. For this reason, the run-time option is preferable in case of rarely changing conditions and restrictions, since no PDP and PEP have to be implemented, simplifying the implementation.

With the run-time option it is only necessary to deploy an SCV Set enforcement functional block that ensures the deployment of the SCV Sets provided by the SON Objective Manager to the SON functions. The SCV set enforcement block is responsible for providing the SON functions and the SONCO with the selected configuration settings (SCV Sets) coming from the SON Objective Manager. The SCV Set enforcement is only applicable in case the run-time option is implemented.

A summary of the corresponding inputs and outputs for each PBSM building block is presented in the following Table 1:

MODULE	INPUT	OUTPUT
Technological Objective Manager	• High-level goals (from operator's objective repository) • Current network configuration (from operator' system model) • KPI composition (from MD)	Technological objectives with priorities, conditions and restrictions
Technological Objectives Repository	Technological objectives with priorities, conditions and restrictions (from the technological objective manager)	Technological objectives with priorities, conditions and restrictions

SON Objective Manager	• Technological objectives (from technological objectives repository) • SON function models (from SON manufacturers) • Network status (from MD) • Recurring conflicts (from SONCO)	• SON policy (in case of design-time option) • SCV Sets (in case of run-time option)
SON Policy Repository	• SON policy (from the SON objective manager)	SON policy
Policy Decision	• SON policy (from SON policy repository) • Network status (from MD)	Policy evaluation result
Policy Enforcement	Policy evaluation result (from policy decision module)	Policy action, i.e., deployment of appropriate configuration to the SON function instances
SCV Set Enforcement	Selected SCV sets (from SON Objective Manager)	Configuration action, i.e., deployment of appropriate SCV set to each affected SON instance

Table 1: Summary of PBSM inputs and outputs

2.2.4 PBSM External Interfaces

The PBSM interfaces (bi-directionally) with the SON Coordinator, and the Monitoring & Diagnosis functional blocks within integrated SON management. Furthermore, PBSM acquires information from the network operator (Operator Goals and System Model) and the SON function manufacturer (SON function model). Finally, the major output of PBSM is towards the SON functions, or the SON function instances, respectively. In the following, the PBSM interfaces with the different functions are briefly described.

Network Operator

The interface with the network operator is required for providing PBSM with the high-level goals and the system model defined by the operator. A standard interface and data formatting are required to get the necessary information from different network operators' repositories.

SON Function Model

The interface with the SON function model provides PBSM with access to detailed information about how different SCV Sets of a specific SON function satisfy certain KPI targets. This interface and the data format can be standardised in order to enable the SON Objective Manager to instrument different SON functions from different SON manufacturers. Alternatively, a SON function model import and normalisation function is required.

Monitoring and Diagnosis

PBSM interfaces at various points with the MD functional block. First, PBSM exchanges information with MD about the composition of KPIs, i.e., how certain KPIs are defined and in how far they reflect high-level operator goals. The introduction of new goals or the modification of existing goals may lead to a changing KPI composition, and additional KPIs that need to be monitored. In the opposite direction, PBSM can use input from MD regarding available or possible KPIs or information about the scope (e.g. timing) of KPIs. This input can be used for the derivation of the technological objectives. Second, the actual technological objectives (KPI target values, KPI priorities, and additional context information) are provided by PBSM to MD in order to allow MD to evaluate the current network state or network operating point according to these targets, and to determine whether the technological objectives are met. Third, MD provides network status information towards PBSM, mainly to the policy decision point, in order to allow the identification of events for the ECA rules in the SON policy, or to check the conditions of ECA rules. Furthermore, the SON Objective Manager may use network status information as input to the creation of the SON policy.

SON Coordinator

A bidirectional interface exists between the PBSM and SONCO. The PBSM and the SONCO exchange information regarding recurring conflicts (from the SONCO to the PBSM), and the identified SON priorities and the SON function configuration settings (from the PBSM to the SONCO). The PBSM uses the incoming information regarding recurring conflicts to implement a different strategy when building the SON policy or when determining the appropriate SCV Sets to be deployed. The PBSM sends the SON priorities and/or the assigned SCV Sets to the decision maker module inside the SONCO to assist it in the conflict solving or avoidance decision process.

SON Functions

From the SON policy system (Policy Enforcement Point) in case of the design-time option (cf. Section 2.2.3), or directly from the SON Objective Manager (run-time option), PBSM sends the configuration values to the SON function instances. Figure 4 depicts a schematic view of the interface between PBSM and a SON function.

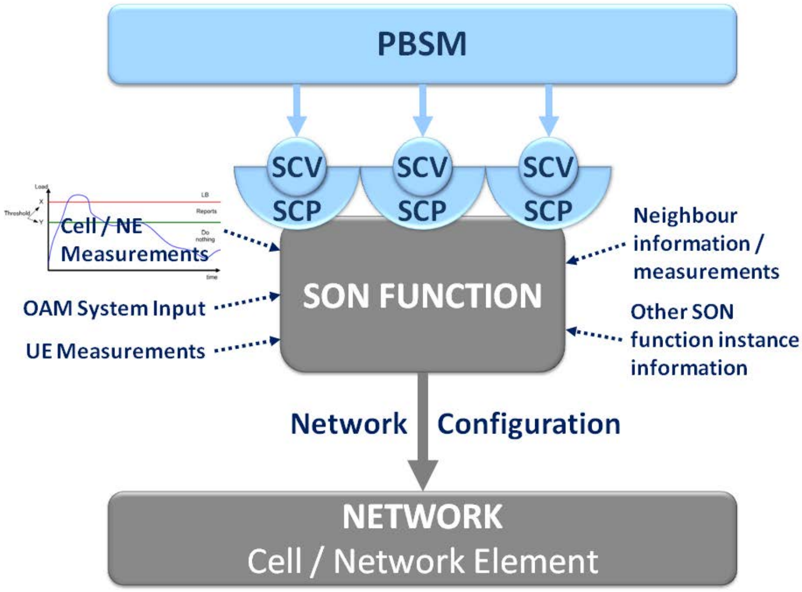

Figure 4: PBSM-SON function Interface

Each SON function has a number of SON function Configuration Parameters (SCP), and dedicated SCP Values (SCV) can be assigned to the SON function instances. Examples for SCP are, e.g., activation thresholds, step size, limits for the output network configuration parameters etc. Different SCVs for the SCPs can influence the behaviour of the SON function instance in such a way that its algorithmic behaviour changes, thereby its influence on the configuration of the cell, network element or the complete network changes, and finally the behaviour of the network changes. A dedicated set of SCVs (SCV Set) for a SON function instance can influence the behaviour of the network in such a way that it works towards dedicated KPI targets, and hence, operator goals. Thus, the appropriate configuration (or instrumentation) of the SON function instances being operational in the network leads to the required network behaviour according to the operator goals.

Standard formats and interfaces are required to provide the necessary information to SON functions from different manufacturers.

2.3 Implementation Approach

During the starting phase of WP5, two different work approaches for PBSM have been considered:

- A top-down approach: Starting at the level of general network-oriented operator objectives, technical objectives and targets for the SON-enabled mobile radio network are derived. These technical objectives and targets are then transformed into policies and rules for the individual SON functions.

- A bottom-up approach. Starting from specific SON functions, identify beneficial configuration settings for these SON functions and map them onto KPI targets of the whole SON-enabled mobile radio network. From these KPI targets technical objectives are derived (i.e., the KPI targets are prioritised and conditioned), which are finally mapped onto general network-oriented operator objectives.

It was decided to pursue both approaches, but first focusing on the bottom-up approach, with the goal to start from the beginning in WP5 with the implementation, simulation and demonstration work. The further descriptions in this section point out the results achieved with implementing the bottom-up approach. Further results from the top-down approach will be described in the coming deliverables of SEMAFOUR Work Package 5, namely, D5.2 and D5.3.

First SON function implementations and testing have been studied in simulation experiments using the scenario data described in D2.4 [11]. The simulation area covers the city centre of Hannover and has a size of 3 km by 5 km, with 123 macrco LTE 1800MHz cells. To observe the effects of the different SON functions, a mobile radio network hosting an average number of 2500 users has been simulated for 30 minutes. The users are divided into different user classes (vehicular, pedestrian, and indoor) with three different required constant bit rates (16, 128 and 512 kbit/s).

Three different SON functions, each of them changing different parameters in the network, have been used in the simulation. These SON functions are Mobility Load Balancing (MLB), Mobility Robustness Optimisation (MRO) and Coverage and Capacity Optimisation (CCO). MLB tries to balance the network load by shifting users from highly loaded cells to less loaded cells. The (cell-) parameter that is modified is the Cell Individual Offset (CIO) to increase and/or decrease the virtual cell boarders. MRO improves the handover performance in the network by changing the hysteresis and Time-To-Trigger (TTT) values for the respective cells. CCO tries to optimise the coverage in the network by changing the antenna tilts.

Each SON function can be modified through a set of SON Function Configuration Parameters (SCPs). SCPs can be a wide range of things, e.g., a method determining how the SON function computes a change in the network, a threshold, a range or a step size. By changing these SCPs the behaviour of the SON functions changes and along with that the network changes accordingly. To see how the behaviour alters, simulation runs have been executed with different SCP Value Sets (SCV Sets). This was done for each and every SON function and also for combinations of SON functions.

In order to observe the changes in the network, six key performance indicators (KPIs) were in focus, namely:

- KPI 1: (Network) load
- KPI 2: (Network) throughput
- KPI 3: (Amount of) unsatisfied users
- KPI 4: (Amount of) executed handovers
- KPI 5: (Amount of) occurred ping-pong handovers
- KPI 6: (Amount of) occurred radio-link-failures (RLFs)

In a first step the different SON functions were tested in a standalone operation. This means that only one SON function, with its varying SCV Sets, was active. Figure 5 shows the effect of varying SCV Sets for the MLB function. As a reference (dark green colour) the system without any SON function has been defined. The number behind the SON name indicates the SCV Sets which were used. One can clearly see that MLB (3) and MLB (4) had a positive effect on the network throughput, namely that the MLB was able to increase the overall throughput.

On the other hand the MLB had an effect on the amount of (overall) RLFs in the system. Figure 6 shows again the results for the MLB function. It is clearly evident that the number of RLFs increased for every simulation where MLB was active, in particular for MLB (3) and MLB (4). Those SCV Sets eventually showed a good network throughput performance. So in this case, the improvement of one KPI comes along with the degradation of another.

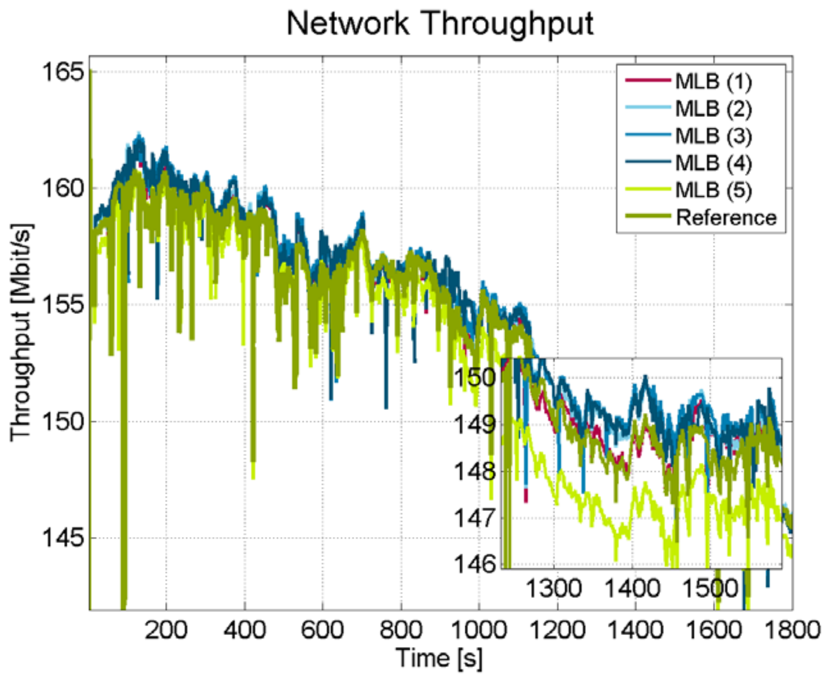

Figure 5: Network Throughput for MLB with different SCV Sets

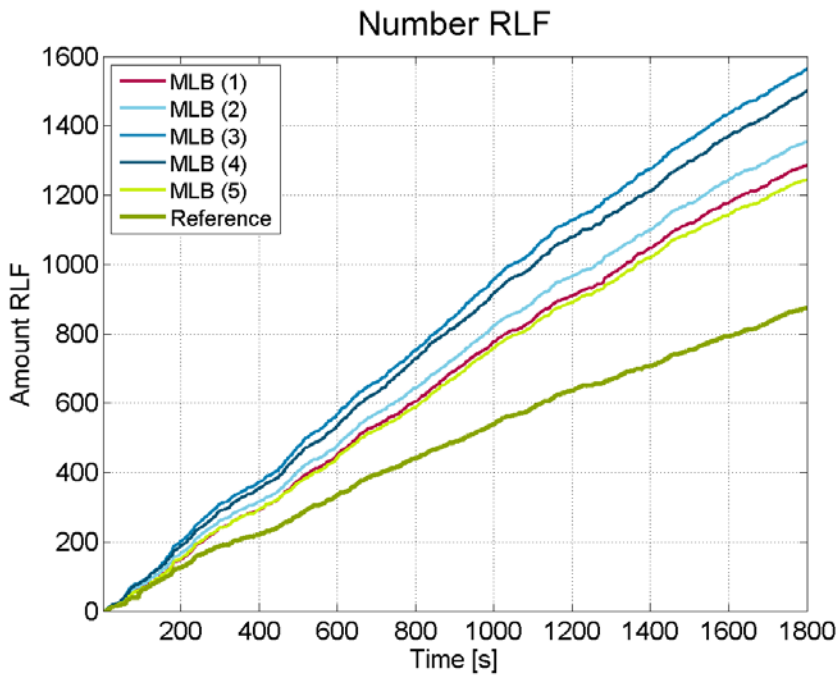

Figure 6: Amount of RLFs for MLB with different SCV Sets

As for the MRO function it was possible to improve the handover performance by decreasing the numbers of RLFs in the system compared to the reference scenario with no SON functions active (again dark green line) as presented in Figure 7. But this entailed that the amount of ping-pong handovers greatly increased as shown in Figure 8.

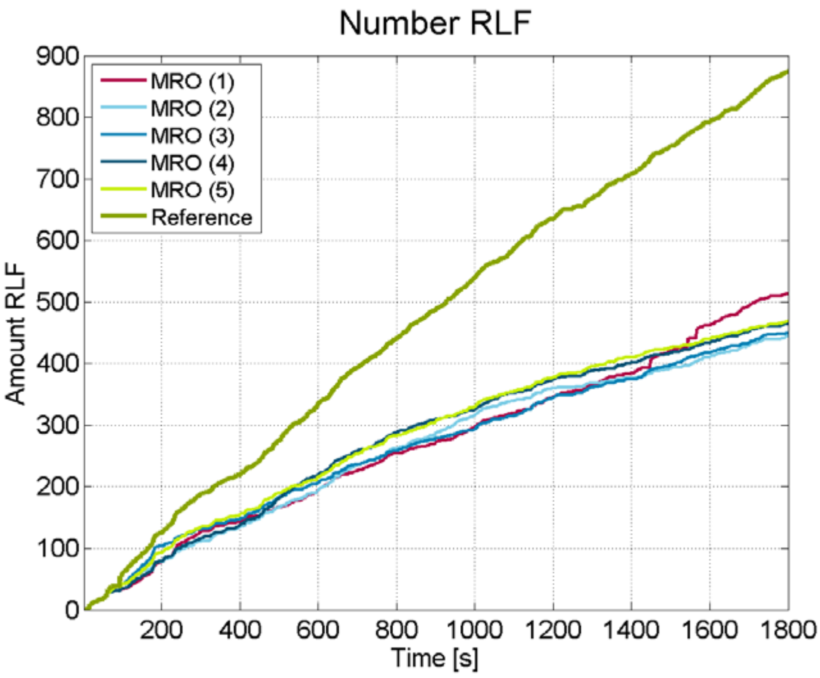

Figure 7: Amount of RLFs for MRO with different SCV Sets

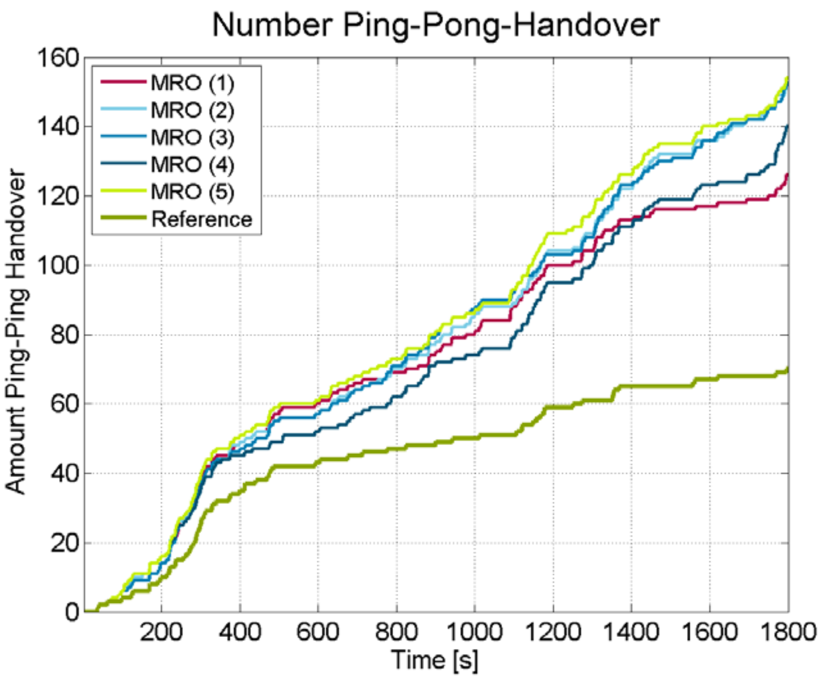

Figure 8: Amount of ping-pong Handovers for MRO with different SCV Sets

As explained above, this analysis was conducted for every individual SON function. In a next step, combinations of such SON functions have been considered. One goal was to see the effects of multiple active SON functions in the network. Another purpose was to correct the negative impact on the network of one SON function with the use of another. For example, some MLB configuration value sets had a positive effect on the throughput, but led to an increasing amount of RLFs. By adding MRO the effect of increasing RLFs should be intercepted such that the number of RLFs would not increase. Results from different combinations of SON function can be seen in Figure 9 and Figure 10. It is possible to reduce the RLFs when MLB (2) is active by adding MRO (6) (dark blue dashed line). It is even possible to reduce the amount a bit further by also adding CCO (2) (light blue dashed line).

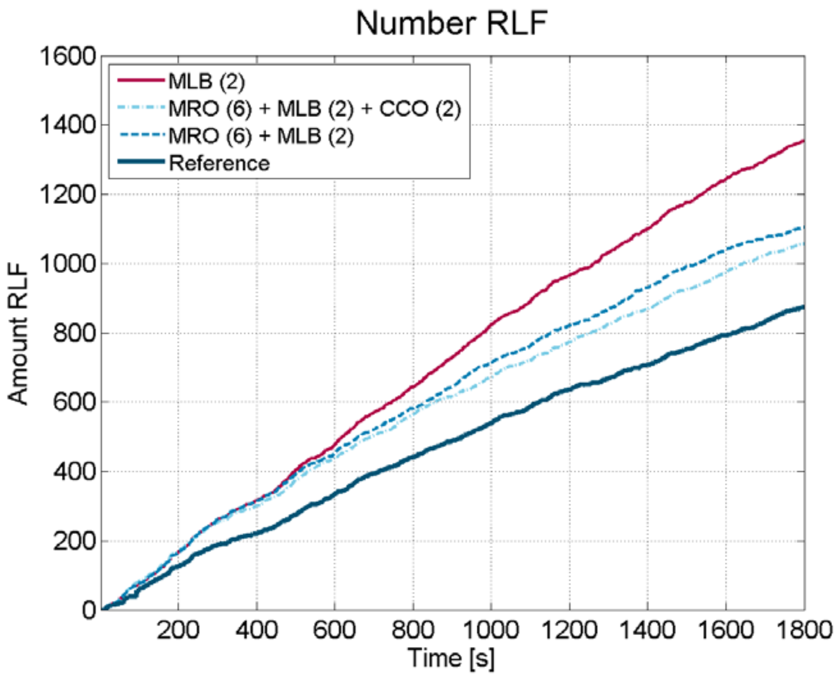

Figure 9: RLFs with Combination of SON functions. Option 1

On the other hand, the amount of RLFs even increases when adding CCO (10) to the SON combination MLB (2) and MRO (6) instead of CCO (2). This behaviour has not yet been fully understood, but first analyses indicate undesired interactions between the SON functions, i.e., SON conflicts. Further simulations are necessary here, in particular with integrating an appropriate SON conflict detection and resolution mechanism into the simulation environment.

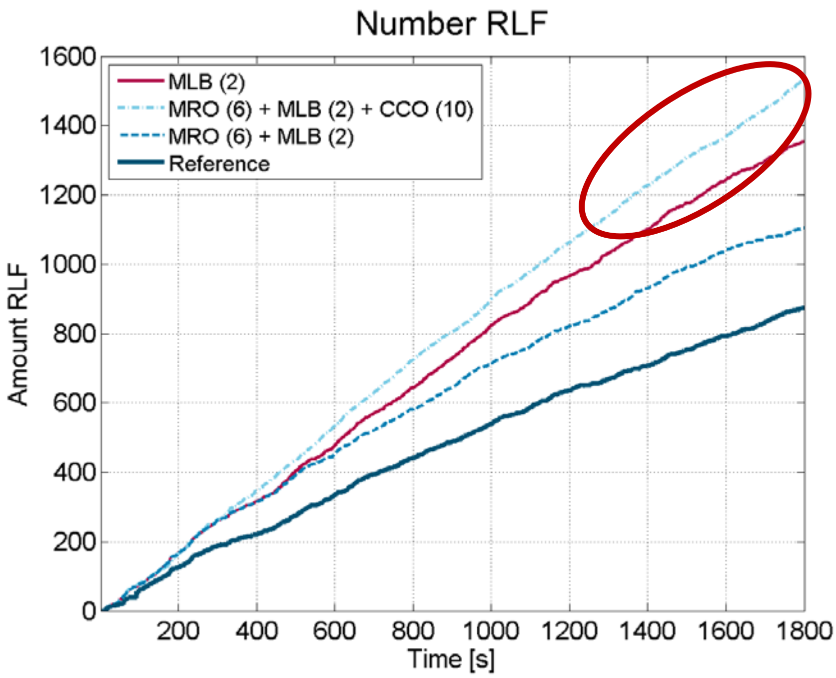

Figure 10: RLFs with Combination of SON functions. Option 2

All simulation results have been collected and examined in order to find appropriate SCV Sets that are in line with high-level operator goals. Exemplary results are presented in form of SON function models for MRO, MLB and CCO in Table 2. The effects on the network are reflected as grades from 1 (excellent) to 6 (very bad).

SON	SON Function Configuration Parameters					KPIs (cf. Definition on page 17)					
	SCP1	SCP2	SCP3	SCP4		KPI1	KPI2	KPI3	KPI4	KPI5	KPI6
MRO	Window	60	--	[1 0,5 2]		2	3	3	5	3	3
MRO	Window	120	--	[1 0,5 2]		3	3	3	6	4	1
MRO	Window	180	--	[1 0,5 2]		3	3	3	6	4	1
MRO	Events	--	10	[1 0,5 2]		3	3	2	6	4	2
MRO	Events	--	25	[1 0,5 2]		3	2	2	6	4	2
MRO	Events	--	50	[1 0,5 2]		1	4	3	5	3	4
	SCP1	SCP2	SCP3	SCP4	SCP5	KPI1	KPI2	KPI3	KPI4	KPI5	KPI6
MLB	100	90	80	6	2	6	4	5	1	4	6
MLB	100	90	80	6	1	6	4	5	1	3	6
MLB	100	90	80	6	0,5	6	4	5	2	3	5
MLB	90	80	70	6	2	6	3	4	1	3	6
MLB	90	80	70	6	1	6	3	4	1	3	6
MLB	90	80	70	6	0,5	4	4	1	2	4	5
	SCP1	SCP2	SCP3	SCP4	SCP5	KPI1	KPI2	KPI3	KPI4	KPI5	KPI6
CCO	Window	60	--	-120	-6,5	2	5	4	3	1	1
CCO	Window	120	--	-120	-6,5	2	6	5	4	1	2
CCO	Window	180	--	-120	-6,5	2	6	4	4	1	2
CCO	Events	--	50	-120	-6,5	2	6	5	4	1	2
CCO	Events	--	100	-120	-6,5	2	6	4	3	1	1
CCO	Events	--	200	-120	-6,5	2	5	4	3	1	3

Legend	
1	excellent
2	very good
3	good
4	fair
5	bad
6	very bad

Table 2: Example for SON Function Models

MRO	SCP1 = MRO Method
	SCP2 = Performance Window [s]
	SCP3 = Evaluated Events
	SCP4 = Operator (Handover) Policy
MLB	SCP1 = Maximum Load [%]
	SCP2 = Maximum Load at SeNB [%]
	SCP3 = Maximum Load at TeNB [%]
	SCP4 = Maximum Cell Individual Offset [dB]
	SCP5 = Step Size [dB]
CCO	SCP1 = CCO Method
	SCP2 = Performance Window [s]
	SCP3 = Evaluated Events
	SCP4 = Minimum RSRP [dBm]
	SCP5 = Minimum SINR [dB]

Table 3: SON Function Configuration Parameters

3 Operational SON Coordination

The operational SON coordination function (SONCO) is one component of the integrated SON management system defined in SEMAFOUR. A first version of a SONCO functional architecture, together with business, functional and non-functional requirements, has been described in SEMAFOUR deliverable D2.2 [10]. In this document, the focus is on improvements to this first architecture, together with a description of the first simulation results related to SONCO. In Section 3.1 an overview about the state-of-the-art in SON coordination is provided, briefly introducing the definitions and approaches from the European research projects SOCRATES [19] and UniverSelf [20], and giving an overview about the technical specifications released by 3GPP standardisation. Subsequently, in Section 3.2, an update of the SONCO related definitions fixed within SEMAFOUR is given, followed by an updated functional description in Section 3.3, both recalling and complementing the SONCO definitions, functional description and architecture that have been introduced in D2.2 [10]. Section 3.4 introduces the first simulation scenarios that are currently studied. Finally, in Section 3.5, different options for SONCO implementation in real network are elaborated.

3.1 State of the Art on SON Coordination

To the best of our knowledge, apart from SEMAFOUR, the coordination of SON functions has been studied in two other European projects: SOCRATES [19] and UniverSelf [20]. The SON coordination discussion started at the end of the SOCRATES project as a study item. The UniverSelf project addresses the coordination problem as a part of a larger research activity in the automatic networking research field. Hence the SON coordination at the radio network level is a particular use case studied in the UniverSelf project.

Considering current 3GPP standardisation activities, the coordination of SON functions has been discussed in 3GPP/SA5 working group in the 3GPP Release 11 timeframe.

The following sections give an overview on the way the SON coordination problem is considered in the context of these research projects and the standardisation. The analysis of these research projects, but also the ongoing work in 3GPP standardisation, influenced the decision on the SEMAFOUR SONCO focus areas as they are defined and described in Sections 3.2 and 3.3.

3.1.1 SOCRATES Coordination Concept

The SOCRATES project proposed a general framework for the coordination of multiple self-organised functions. As described in SOCRATES Deliverable D5.9 [8] and in [6], the SON coordination function includes (but is not limited to) the detection and correction of unexpected and undesirable network behaviour caused by SON functions. The SON Coordinator's role in SOCRATES is to ensure that the control parameter changes of the SON system are harmonised towards the Operator policy, expressing the applicable performance objectives. Two different types of harmonisation are introduced: *heading harmonisation*, which aims at conflict avoidance, and *tailing harmonisation*, which aims at conflict resolution.

Heading harmonisation can be achieved by appropriate alignment of the SON function-specific policies, i.e., ensuring that the policies assigned to the different SON functions in a cell are not in conflict, such that the SON functions do ideally not produce conflicting parameter changes. This may be achieved by working towards non-conflicting targets and sharing applied performance metrics. The *degree* of heading harmonisation that can be achieved, i.e., the extent to which conflicts can be foreseen or prevented, depends on the number of implemented SON functions, the risk for conflicts between the SON functions (e.g., if different SON functions operate on an orthogonal or entwined set of control parameters), and the dynamics of the network system and the SON functions.

Tailing harmonisation aims at resolving conflicts that may occur in case heading harmonisation and thus conflict avoidance could not prevent all interdependencies between SON functions. The output parameter change requests of each SON function are checked to see if there is a conflict with the objectives and control parameter requests from other SON functions. If required, one or several requests are changed or combined.

Heading and tailing harmonisation complement each other. This can be expressed such that, the more harmonisation is performed through the policies, the less harmonisation is required through conflict resolution.

To achieve the harmonisation of SON functions as described above, the following functional roles of the SON Coordinator are defined:

- The *Policy function*, which converts the operator's high-level performance objectives into SON function specific policies and provides the interface between operator and SON system;
- The *Alignment function*, which is responsible for conflict detection and resolution;
- The *Guard function* that detects extreme or undesirable network behaviour and triggers countermeasures;
- The *Autognostics function* that collects and processes performance, fault and configuration data as input to the SON system.

3.1.2 UniverSelf Coordination Approaches

In the UniverSelf project, several coordination mechanisms are introduced with different levels of intelligence and complexity, and depending of the characteristics of the considered SON functions[1]. The role of the coordination mechanism is to manage/avoid these three types of situations which are regarded as giving rise to potential conflicts:

- parameter value conflict,
- metric value conflict, and
- loops or oscillations

The following coordination strategies are defined in the UniverSelf project:

- *Pre-deployment conflict avoidance*: The instance descriptions of a SON function in a given set of functions is used to build a graph that identifies conflicts (parameter-conflict, metric value conflict and loops). Conflict avoidance is triggered before deploying a new SON in the network. If a conflict is identified, a run-time coordination solution is triggered (see below).
- *Hierarchical optimisation*: Hierarchical optimisation deals with SON functions that operate at different time scales. The idea here is to set "slower" SONs as "leaders" and the "faster" SONs as "followers", and have SONs optimise their parameters independently with the faster SONs seeing the configurations enforced by the slower SONs as "semi-static". All SONs in this approach keep their individual objectives intact and they can operate in parallel at different time scales though.
- *Synchronous control theory:* Synchronous control theory approaches deal with SONs that operate at the same time scale and are synchronised. Each SON optimises its own objective and takes into account the influence of other SONs. A two phase approach is utilised. In the first phase, a coordination matrix is computed once for a given set of SON functions to be activated. This phase can be implemented at the management plane. The matrix adapts (or corrects) jointly the parameters of all SONs in the set and enforces stability and connectivity constraints. In a second phase, each deployed SON receives the corresponding line of the coordination matrix which is used during run-time. The same approach can also be generalised and applied for SONs that are not necessarily synchronised in time but have the same average frequency of parameter changes.
- *Separation in time strategy* (*or token based SON selection*): This strategy allows one SON at a time to enforce a parameter change. Each SON calculates an estimated improvement brought about to the network performance (utility) if it was selected, and the SON with the highest utility is then triggered. This means that SONs must be able to predict the effect of their actions.

[1] In UniverSelf, a SON function is a particular network empowerment mechanism (NEM).

- *Centralised multi-objective optimisation:* This strategy deals with SONs that operate at the same time scale and tries to optimise a global/weighted utility function which combines the utility functions of all considered SONs. In particular, the coordination function knows the explicit utility/objective function of each SON and is capable of modifying directly this utility function and/or the inherent optimisation algorithm of each SON. By considering the problem as a global optimisation one, the coordination is implicitly resolved. This approach is relevant for management plane (off-line, with low time reactivity) solutions.

It is noted that the on-line coordination problem is of high complexity. Important R&D work is still needed, e.g. to provide convergence proofs in certain cases, scalability in other cases, and extend their applicability to recent technological evolutions.

3.1.3 3GPP on-going Work on Coordination in 3GPP SA5

SON coordination is part of the standardisation efforts conducted in the 3GPP SA5 working group. This section gives an overview on the SON coordination related definitions provided by this group, based on the technical specifications TS 28.627 [12] and TS 28.628 [13]. The specification documents provide the following definitions:

- The *"SON functions in conflict"* situation is defined as follows: "When multiple SON functions attempt to change some (same or associated) network configuration parameters of some (same or associated) nodes, one or more of these SON functions may not be able to achieve the operator's specified SON target(s) (for individual SON function) since they may have conflicting demands on network resources"
- "*Associated network configuration parameters*" include parameters within the same network element or parameters of different network elements with impact between each other.
- *"SON coordination"* means preventing or resolving conflicts or negative influences between SON functions to make SON functions comply with operator's policy.

Note that conflicts between SON functions and non-SON manual configuration actions are also considered in 3GPP SA5.

According to the 3GPP specifications, the SON coordination function has two tasks: conflict prevention and conflict resolution.

Conflict prevention

To prevent conflicts between the SON functions, the SON functions may ask the SON Coordination function for permission before changing some specific configuration parameters. The SON Coordination Function should check the SON coordination interdependency policy between this requesting SON function and other SON function(s) upon receiving the permission request from the SON function. SON coordination interdependency policy which is pre-configured can help the SON Coordination Function to find which SON function(s) possibly conflict with this requesting SON function.

As a basis for decisions, the SON Coordination Function will typically use one or some of the following inputs received from the SON function(s):

- Which SON functions are modifying configuration parameters (including information about vendor, release etc.)
- Configuration parameters intended to be changed and/or their existing and proposed new values
- The time duration how long the configuration parameter should not be interfered with ("impact time")
- The state of SON functions
- The SON targets which are the justification for the configuration change
- Possible impact of a parameter change on other objects ("impact area")
- The state of certain managed objects

- Possible impact of the parameter change on Key Performance Indicators
- Priority of SON functions, which can be used to determine the execution order of requests from different SON functions in case of conflicts
- SON coordination policies

The SON Coordination Function sends the decision back to the requesting SON function; the decision may be confirmation or suspension or rejection of the SON executing request, or other actions like configuration of specific parameters with specific values. After the SON function executes action, the SON Coordination Function is then informed about the result (successful/unsuccessful, parameter changes) of the executed SON action. The SON Coordination Function may prevent parameter changes by one or more SON functions for a specified time period after the same parameter has been changed by another SON function. The SON Coordination Function may inform a SON function of a managed object state change which may impact the calculation of KPIs.

Conflict resolution

For this purpose, the SON Coordination Function should detect conflicts and attempt to resolve the conflicts. To detect conflicts, the SON Coordination Function will typically analyse one or some of the following types of data:

- Key Performance Indicators which indicate if SON functions are meeting their targets or improving network performance
- Measurements which indicate if SON functions are meeting their targets or improving network performance
- Inacceptable oscillations in configuration parameters

To resolve conflicts, the SON Coordination Function will typically use one or some of the following methods.

- Enabling/disabling/suspending certain SON functions
- Stopping/suspending/modifying certain SON actions
- Modifying certain configuration parameters

The standard gives some examples of coordination policies:

- Prioritising SON functions in case of conflicts
- Assigning weights to SON targets
- Prohibiting further changes of a parameter for a certain amount of time
- Selecting preferred value ranges

3.2 Definitions and Scope

This section summarises the so-far agreed definitions related to SON coordination within SEMAFOUR. These definitions, as well as the scope of the SON coordination scenarios that will be studied in the project are likely to evolve all along the project timeframe.

SONCO definition and role

When two or more SON function instances are simultaneously active or operational in the network and they interact among themselves, these SON functions are coordinated by the SON coordinator (SONCO). Taking into consideration the SON implementations in current networks, but also the current development of new SON functions, led to the assumption that SON functions usually operate as ‘black boxes’. For such a ‘black box’ it is known which measurements are used as input, and which network configuration parameters are modified, but the SON algorithm itself is unknown (see also Section 2.2.2). Furthermore, the analysis of existing SON functions, and the current work conducted in SEMAFOUR Work Package 4, has led to the finding that SON functions usually work in a ‘fire-and-forget’ mode, i.e., the SON algorithms do not keep track of their own history of changes to the network configuration, but only work based on the measurements acquired from the network. Both together, the ‘black box’ and the ‘fire-and-forget’ are seen as pre-conditions for developing the

SEMAFOUR SONCO concept and functional architecture, which will hence focus on SON coordination during operation (or 'tailing coordination' according to the SOCRATES definition as provided in Section 3.1.1).

As defined within SEMAFOUR, the tasks of the SONCO are:

- *Conflict detection*: SONCO shall detect conflicts between simultaneously running SON functions;
- *Conflict resolution/prevention at run-time*: The SONCO shall decide on how a conflict is to be avoided prior to enforcing the corresponding conflicting actions into the network;
- *Undo*: The SONCO can undo action(s) it recently enforced;
- *Priority handling*: The SONCO acts in line with the priorities assigned to SON functions by the PBSM;
- *Repeated conflicts handling*: The SONCO shall inform the PBSM on repeating conflicts between two or more SON functions in order to advise the PBSM to improve the SON policy definition.

Conflict definition

According to the definition in D2.2 [10], three different conflict types are distinguished: configuration conflict, measurement conflict, and characteristic conflict.

- *Configuration conflict:* A conflict induced by changes to a configuration parameter.
  - Direct/output parameter conflict: This may occur when two SON functions act on the same parameter.
  - Input parameter conflict: When a SON function computes a new configuration it may use for this computation the values of parameters that are not configured by this SON function (e.g. neighbouring cell parameters such as PCI values may be taken into account for the update of the own PCI value). If these parameters are modified by another SON function during the calculation, then we have an input parameter conflict.
- *Measurement conflict:* A conflict induced by the change to a measurement. A measurement conflict can occur when parameter changes made by a SON function influence the measurements that are used by another SON function, either to trigger the SON function algorithm execution, or to serve as input to the SON function algorithm in order to evaluate the current state of the system and deduce appropriate configurations / actions.
- *Characteristic conflict:* A conflict induced by the change of a cell's characteristics, for instance the cell size.

Temporal dependencies

In [1] and [7], the authors present an analysis of the temporal dependencies between SON functions that should be taken into account by the SONCO. This book and the paper define the impact time components of a SON functions as follows:

- *Enforcement time*: Is defined as the time interval between the configuration request by a SON function and the acknowledged enforcement of the parameter changes at the NE level.
- *Visibility delay*: Is defined based on the average time required for changes of a particular SON function to become visible in the network.
- *Protection time*: This component is only specified for a SON function if its input data is aggregated over a time interval. This protection time guarantees that over the complete aggregation time of the input, all previously performed changes are fully reflected.
- *Relevance time*: Is defined for a SON function with respect to another SON function. It specifies the time interval during which a configuration change performed by a SON function may impact another SON function.

These definitions are given as a starting point for a deeper investigation in the project. With respect to simulating and implementing SONCO, the temporal dependencies of a SON function are an important input to the SONCO algorithm.

Coordination scenarios classification

The SON coordination scenarios can be classified according to the temporal and/or spatial nature of the SON functions' interactions as follows:

- *Coordination of identical SON functionalities:* The same SON functionality is active in neighbouring cells (the nature of the SON interactions is spatial). The integrated SON management framework covers several instances of the same or similar SON functions over a zone of neighbouring cells. A typical example could be the network-wide deployment of MLB, where it is necessary to control/coordinate the interactions/conflicts between the MLBs of neighbouring cells. Note that such a situation is more prominent in multi-vendor environments, where a design-phase coordination/interworking between these different SON instances from different vendors cannot be expected.
- *Local coordination and management of distinct/different SON functionalities:* A single cell having distinct/different SON functions is considered. It is assumed that neighbouring cells do not activate SONs.
- *Coordination of distinct/different SON functionalities:* Several neighbouring cells are considered, where in each cell two or more SON functions are in operation. The integrated SON management covers several instances of the same SON function for more than one SON. A typical example is the network-wide deployment of MRO + MLB.

3.3 SONCO Functional Description

This paragraph recalls and complements the functional description of the SONCO described in D2.2 [10].

3.3.1 SONCO Functional Architecture

The SONCO functional architecture consists of the following building blocks as depicted in Figure 11:

- *Information acquisition and processing*, which stores all required information for decision making, comprising the memory of the previous internal states of the coordinator (that might be needed for undo operations, for detecting recurring conflicts and for enhancing the SONCO decisions through learning), the SON functions requests, and post-processed KPIs. This function receives metrics and indicators from the MD and post processes the collected information to become an input for both Conflict Detection and the Decision Maker.
- *Conflict detection function* that detects conflicts based on the information processed by the acquisition and processing function. The conflict detection informs the Decision Maker (see below) on the detected conflict. This function stores the description and occurrence of past conflicts and if a conflict occurs too frequently, then it is reported to the PBSM.
- *Decision maker*: Conflict resolution or avoidance decisions are made by this "intelligent" block. Depending of the implementation choice, this decision is sent to the SON functions or directly enforced to the network. The Decision maker receives information from PBSM to orient its decisions, such as SON priorities.

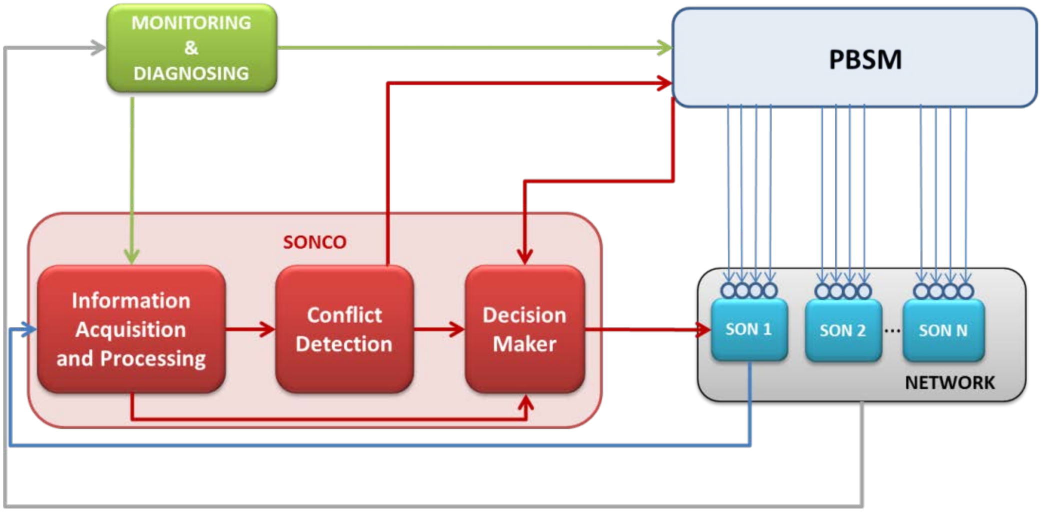

Figure 11: SONCO functional architecture

3.3.2 SONCO Interfaces

SONCO as a part of the integrated SON management system

SONCO exchanges information with MD and PBSM functions via the corresponding interfaces:

- *SONCO/PBSM interface*: PBSM provides the SONCO with information on the SON functions' priorities and SON function configuration settings (e.g. configuration parameter ranges, temporal granularity, …). In the opposite direction, SONCO provides a feedback to the PBSM about recurring conflicts to the PBSM, to improve the PBSM SON policies' definition. For instance, a conflict occurring frequently and that cannot be handled by the SONCO may be due to a bad policy design. Standard formats/interfaces between these two functions are required.
- *SONCO/MD interface*: The MD provides the SONCO with the KPIs and metrics that the SONCO needs. This information can be processed by the "Information Acquisition and Processing" block, and serve as input to the "Conflict Detection" block. For instance, measurement conflicts or characteristic conflicts can be detected by observing and post-processing some measurements and metrics that are reported by the MD to the SONCO. These measurements or observations can be requested by the SONCO to the MD or by the PBSM to the MD at run time.

Interaction between SON functions and the SONCO

The minimal set of information exchanged between the SON functions and the SONCO are the following:

- SON functions send requests to the SONCO to modify parameters;
- The SONCO sends accept/deny feedback to the SON functions.

The information exchanged between the SONCO and the SON functions may depend on the coordination scenario and the SON functions capabilities. With respect to the interaction between the SONCO and the SON functions, two different implementation options are possible, that also affect the actual method of enforcing configuration parameter changes at the network element:

- SONCO as *extra function*: Before a SON function changes a configuration parameter it has to check with SONCO if this is allowed. Once the SON function has the approval/validation of the SONCO, the SON function itself enforces the parameter change to the network. This approach assumes that SON functions are aware of the SONCO and are able to react to its decisions. In case of failure of the SONCO system, the SON functions are still able to function in the network. The SONCO has no absolute control over changes applied to the network.

- SONCO as *intermediate layer*: SONCO acts in an overlay layer to the NE configuration interface. Configuration parameter changes to the NE performed by the SON function can be forwarded (acknowledged), discarded (rejected), or cumulated by SONCO. SONCO is in charge of enforcing the parameter changes and may or may not inform SON functions about the changes performed to the NE (or network) configuration. SON functions cannot avoid being coordinated and may not even be aware of the coordination. In particular, the SONCO holds absolute control over changes applied to the network.

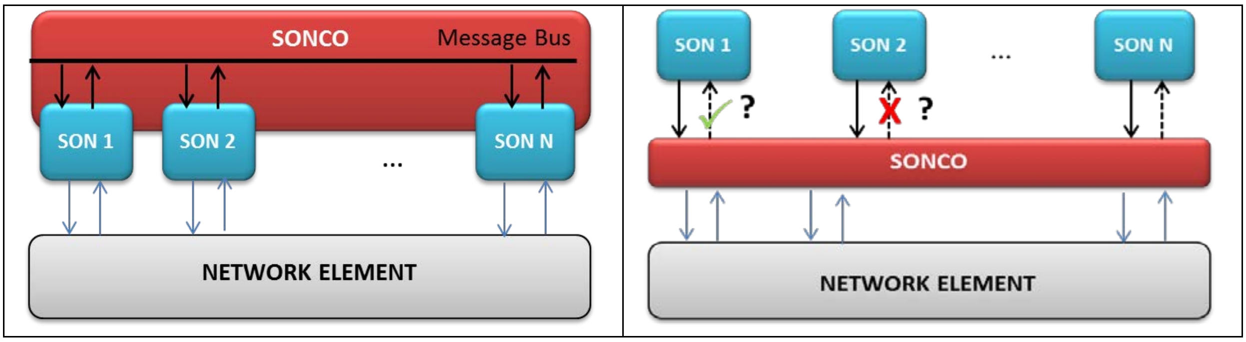

Figure 12: SONCO implementation options

While the first approach takes benefit from an explicit interface for the communication between the SONCO and the SON functions, the second approach can simply replicate the control interface of network elements (managers). The actual implementation is likely to be a hybrid version blending aspects of both extremes.

The network implementation and design of the SONCO have not been studied yet in detail in the SEMAFOUR project. Some initial thoughts are described in Section 3.5.

Considering the simulation framework, we assume that the SONCO will have an explicit interface for SON functions. This interface will allow for exchanging information between the SONCO and the SON functions.

The particular case of "undo" action

Considering the "Undo" action that could be decided by the SONCO, the SONCO has to know whether a SON function is capable of executing an "undo" action. This capability includes the fact that the SON function keeps in memory the previous states/configurations. Hence, the number of previous states that the SON function is able to "remember" is an additional parameter, namely the "undo range" that the SONCO has to know to be able to decide an "undo" action. The PBSM should inform the SONCO about the undo capability and the undo range of a SON function as a part of the SON Configuration settings sent by PBSM to SONCO.

SONCO sends a specific "undo" request to the targeted SON function. This request contains the number of actions that the SON function has to undo. This request can also contain the target configuration, if the SON function is stateless and if the SONCO keeps in memory the previous states of the SON functions. This second option supposes that the SONCO can force a configuration change either by imposing to a SON function to enforce a particular configuration or by directly enforcing a parameter configuration to the network.

3.4 First SONCO Simulation Scenarios

This section introduces the first SONCO simulation scenarios that are currently studied, namely the coordination of different instances of the same Mobility Load Balancing (MLB) function, and the coordination of MLB and mobility robustness optimisation (MRO) SON functions. The following general assumptions are taken:

- Synchronous coordination. The SONCO receives coordination requests in a synchronous manner with the same time periodicity. This implicitly means that the coordinated SON functions have the same time granularity. It is supposed also that this period is long enough to reach steady state on the measurements of interest, meaning that the impact of the previous parameter configuration changes can be seen.
- The SON functions are considered to be black boxes: the SONCO is not aware of their internal implementation (optimisation algorithm, utility function).
- The SONCO does not analyse directly the KPIs, and does not use any information coming from the MD. The SONCO relies only on the information exchanged with the SON functions.

The current simulations are conducted for an LTE-only network scenario with hexagonal cell layout. The next steps in the simulations will use the same network scenario as it is used for PBSM (see Section 2.3). The proposed SONCO algorithm is based on reinforcement learning. Indeed, the SONCO is not able to predict the impact of the coordination decisions that it takes. The reinforcement learning allows the SONCO to learn from its past experience, and to enhance its decisions accordingly.

3.4.1 Coordination of MLB Instances

In this scenario the SONCO coordinates different instances of the same MLB SON function. Each instance is located at an LTE base station (eNodeB) and is in charge of optimising the setting of the cell individual offset (CIO) of this eNodeB in a way that the load of the corresponding cell remains under a given threshold. For an overloaded cell, decreasing the CIO has as a consequence to offload the cell traffic on its neighbouring cells, as handovers initiated by this cell will be triggered earlier. It may happen that a neighbouring cell that receives this traffic becomes overloaded. Hence the change of configuration of one MLB impacts the KPI input of another MLB: This is a *measurement conflict*.

In the proposed coordination framework, the SONCO entity is located on an architectural layer above the MLBs and it has the same temporal granularity as the MLBs. It is assumed that each MLB sends a CIO change request to the SONCO in a synchronous manner. The SONCO decides which of these requests to accept and which to deny. As shown in Figure 13, the simulation scenario consists of three MLBs implemented on 3 neighbouring eNodeBs. The conflict occurs mainly due to the traffic hotspot (HS) close to the boundary between two neighbouring eNodeBs.

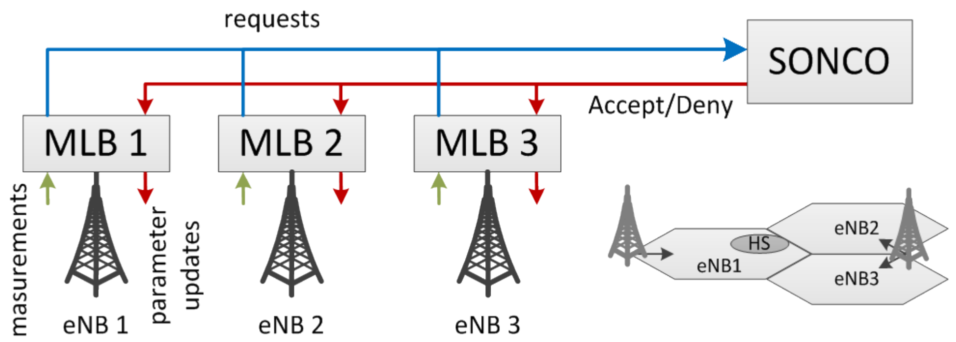

Figure 13: MLB instances coordination, simulation scenario

It is assumed that the SONCO knows only the current requests of the MLB instances together with the current CIO values of the eNodeBs. The objective of the SONCO is to reach a state where all the MLBs are "satisfied" meaning that they do not send any request anymore. For this purpose, the SONCO is implemented based on reinforcement learning: the SONCO enhances its decisions based on the current MLBs requests, the MLBs past requests and on its past decisions.

3.4.2 Coordination of MRO and MLB SON Functions

In this scenario, the SONCO coordinates several instances of two different SON functions: MLB and MRO. The MRO enhances the mobility robustness by minimising the number of “too early” handovers, “too late” handovers and the number of ping pongs. This function impacts two parameters: the handover hysteresis and the CIO. It may happen that the MRO and the MLB instances of the same eNodeB request to change the CIO parameter in opposite directions. This case corresponds to a *parameter conflict* that should be avoided and/or handled by the SONCO.

The SONCO is assumed to have the same time granularity as the MLB and MRO instances. It receives the requests from all the running SON instances at the same time and decides which requests it will accept and which it will deny. As shown in Figure 14, the simulation scenario consists of four sites with a total number of seven eNodeBs. The central cell contains a hotspot, and needs to offload the traffic on its neighbouring cells. This may lead to a high number of too late handovers on a neighbouring cell and needs to change the CIO in the opposite direction. As the two SON functions are requesting opposite changes on the same parameter, the SONCO has to arbitrate and to decide which of the changes is the most appropriate and leads to a reasonable trade-off between the two SON functions.

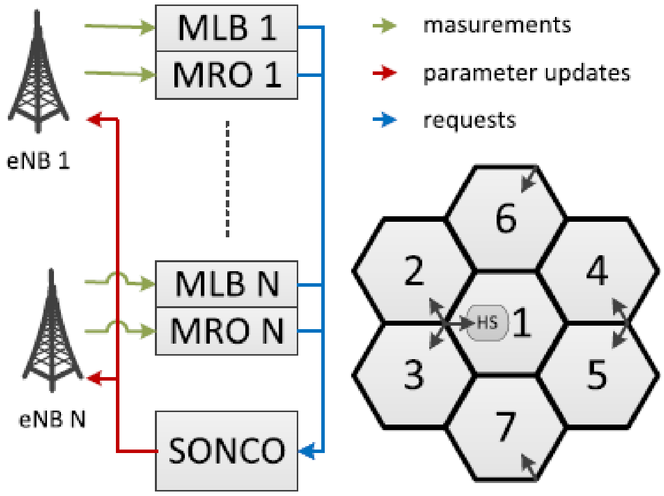

Figure 14: MRO-MLB coordination simulation scenario

In the previous coordination scenario, if two MLB instances send the same request, this corresponds to the same “radio” state of the cell (overloaded or not), as they both are optimising the same load metric. In our case, the situation is different, as MLB and MRO have completely different and even contrary objectives. For example, MLB aims at reducing the load of a cell by handing over users to a neighbouring cell, whereas MRO aims at the reduction of handovers in general. The role of the SONCO is to find as soon as possible a situation that corresponds to the best trade-off between these objectives. For this purpose, additional information is introduced which is to be sent by the SON functions to the SONCO together with the configuration change request: the indication on how critical the request is, which corresponds to the level of satisfaction of the SON function with the current configuration setting. The SONCO will then find the optimal trade-off between the two SON functions taking into account their satisfaction level. Furthermore the possibility is taken into consideration that these two SON functions are not given the same priority by the PBSM as they are impacting different KPIs. To each SON function a corresponding weight is attributed that should be taken into account by the SONCO. The Reinforcement learning algorithm combines the weights of the SON functions and their level of happiness to evaluate the reward corresponding to a given state and to steer the SONCO decisions towards the best trade-off.

3.5 SONCO Implementation Options

The SON Coordination Function is a logical function that can be implemented as a separate function entity within the SON system or as part of SON functions. The implementation impacts the architecture and the information flow of the SON system and may also impact the OAM standardisation, i.e., the Itf-N interface (cf. Section 3.1.3).

When the SON Coordination Function is implemented as a separate function entity (cf. Figure 12), all the SON functions send the necessary information to the SON Coordination Function, and the SON Coordination Function coordinates these SON functions as a centralised control point. This centralised coordination approach fulfils the requirements of SON coordination in a large area scope, for example, the coordination between centrally operated SON functions (e.g., at Network Management Layer) and distributed operating SON functions (e.g., at the Network Elements).

In some other situations, coordination is only needed in a smaller area, for example, the coordination between distributed SON functions inside one domain. Then the SON Coordination Function can be implemented as part of each SON function. The necessary coordination information can be interchanged between each SON function to achieve coordination as well.

Figure 15 shows an example of various SON functions located at different levels and a SON Coordination Function located above the 3GPP-standardised Itf-N management interface, i.e. in the Network Management layer. Such a solution based on a centralised SON Coordination Function requires every SON function (i.e. all SON functions in the Network Management Layer, Element Management Systems and Network Elements) to ask for permission from one central SON Coordination Function before making any configuration change.

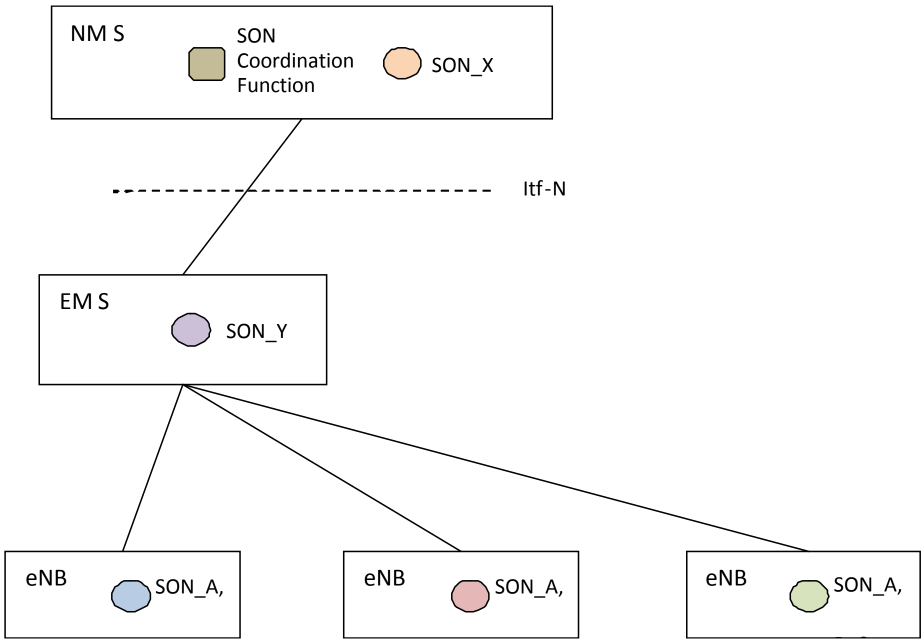

Figure 15: SON Coordination Function above Itf-N

In some other situations, e.g., when coordination is only needed in a smaller area, the SON Coordination Function can be implemented in a distributed way, as part of each eNodeB. The necessary coordination information would then have to be interchanged between all SON functions to achieve coordination. In this implementation, a SON Coordination Function is co-located with SON functions (see Figure 16). With such a distributed SON Coordination Function, each SON function requests permission from the local SON Coordination Function before modifying configuration parameters.

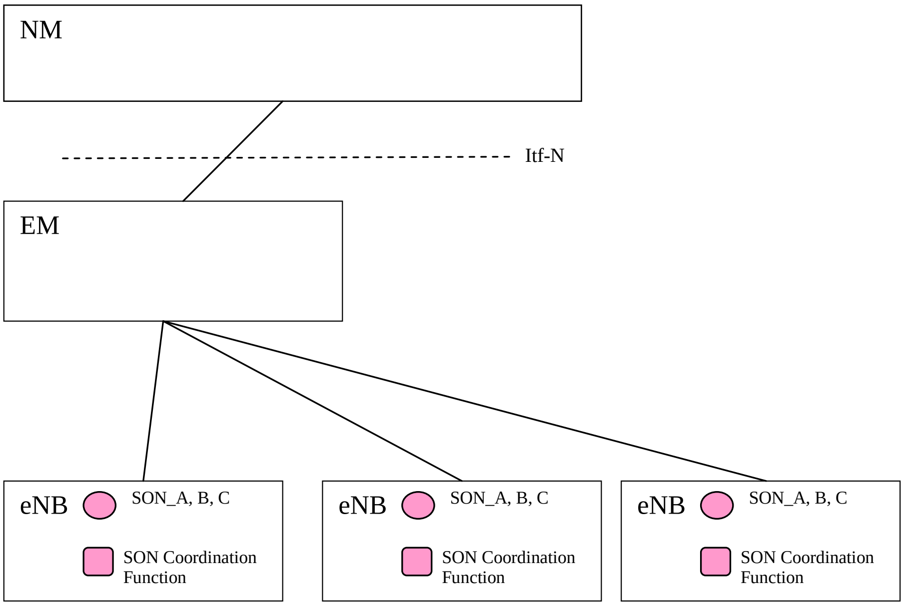

Figure 16: Distributed SON Coordination Function

Since various architectures can be used simultaneously to deploy SON functions (distributed, EM-Centralised, NM-Centralised) the coordination between these SON functions should adapt to the architecture of the coordinated SON functions and shall sometimes be performed in a centralised manner. For distributed SON functions, pushing the SON coordination at the lowest possible level in the network, i.e., at the base station, has the advantage of higher efficiency and higher reactivity. However, in this case the SON coordination remains local and may be less consistent than a centralised or global SON coordination. Moreover, designing the SON coordination as a part of the SON functions is likely to result in proprietary solutions for SON coordination which raises the same concern of coordinating these different SON coordinators in a multi-vendor network or at the boundary between a shared RAN and a non-shared RAN.

The implementation approach is an important part of the SEMAFOUR SON coordination concept, in particular with respect to the requirement to develop a solution which is clearly implementable into real systems. The current SONCO functional architecture allows a centralised as well as a distributed implementation, or a hybrid implementation combining both. The development of an implementation concept is part of Activity 5.3 on operational SON coordination, and the results will be documented in SEMAFOUR Deliverable D 5.3.

4 Decision Support System

In this chapter, firstly the functional architecture of the Decision Support System (DSS) as it has been defined in D2.2 [10] is briefly recalled. Secondly, an overview of the initial DSS demonstrator is provided. This demonstrator has been developed based on the elaborations of the functional architecture in D2.2, and it has been presented at the ICT 2013 conference in Vilnius, 6-8 November 2013. Besides exemplifying some specific functionality of DSS use cases, the development of the initial demonstrator is also helpful for gaining insights in the DSS requirements and challenges. These insights will be used for the enhancement and generalisation of our solutions to be implemented in future upgrades of the demonstrator planned for 2014/2015.

4.1 Functional Architecture

Figure 17 shows the functional architecture of DSS in relation to the (other) functionalities of the integrated SON management system. In this figure the DSS part covers in particular the DSS use cases ‘Spectrum and Technology Management’ (DSS-STM) and ‘Network Evolution’ (DSS-NE), which are described in detail in D2.1 [9] and D2.2 [10]. For these use cases, the DSS basically comprises two key tasks, or stages, see Figure 17:

- *Stage 1 “Triggering of DSS analysis”*. The first DSS stage identifies when and for which area recommendations for network upgrades are to be derived. This is a continuous process, which will be executed every day, based on the most ‘up-to-date’ data.
- *Stage 2 “Derivation of recommendations for network upgrades”*. The second stage assesses a list of potential network upgrades, derives a shortlist of most suitable candidates and reports this to the network operator, along with a prediction of the performance impact over time.

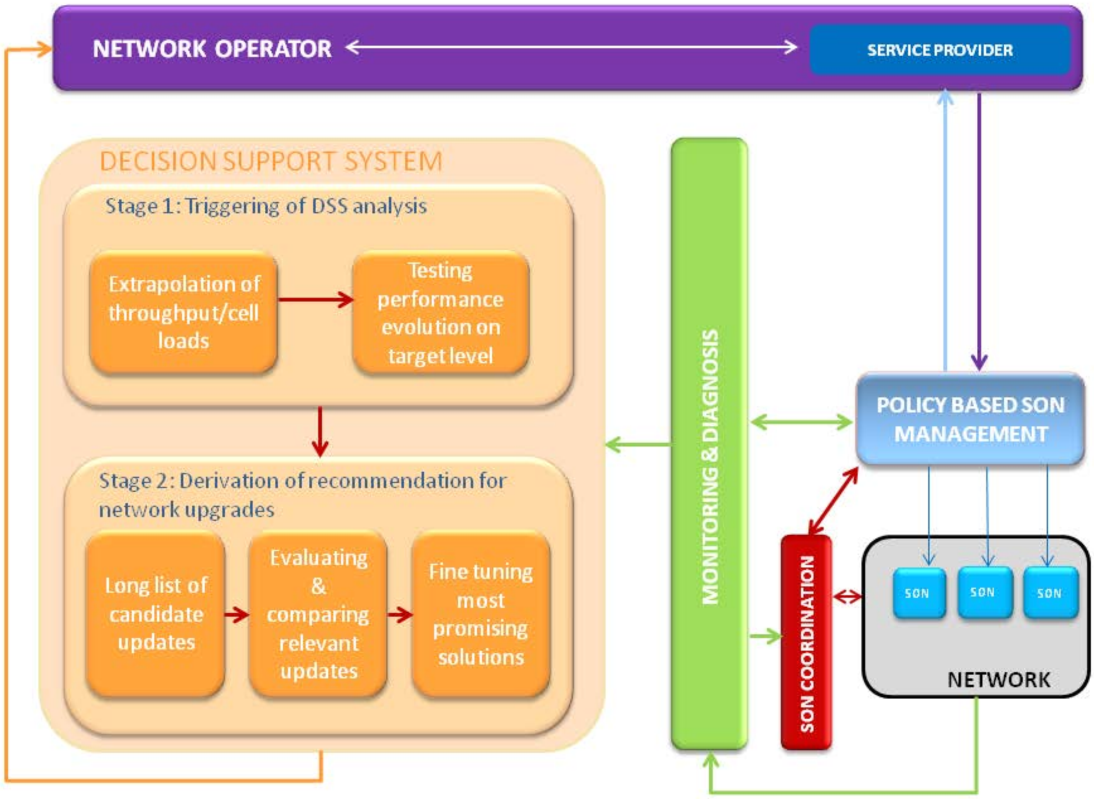

Figure 17: Functional architecture of DSS, detailed for the STM and NE use cases

An overview of the main parameters and information streams involved in the execution of both DSS stages is provided in Figure 18. Extensive elaborations of these key DSS tasks 'Triggering' (Stage 1) and 'Derivation of recommendations for network upgrades' (Stage 2) are given in Section 2.3 of D2.2, which will not be repeated here. Based on the work from D2.2, an initial DSS demonstrator has been built, which will be briefly described in the next section.

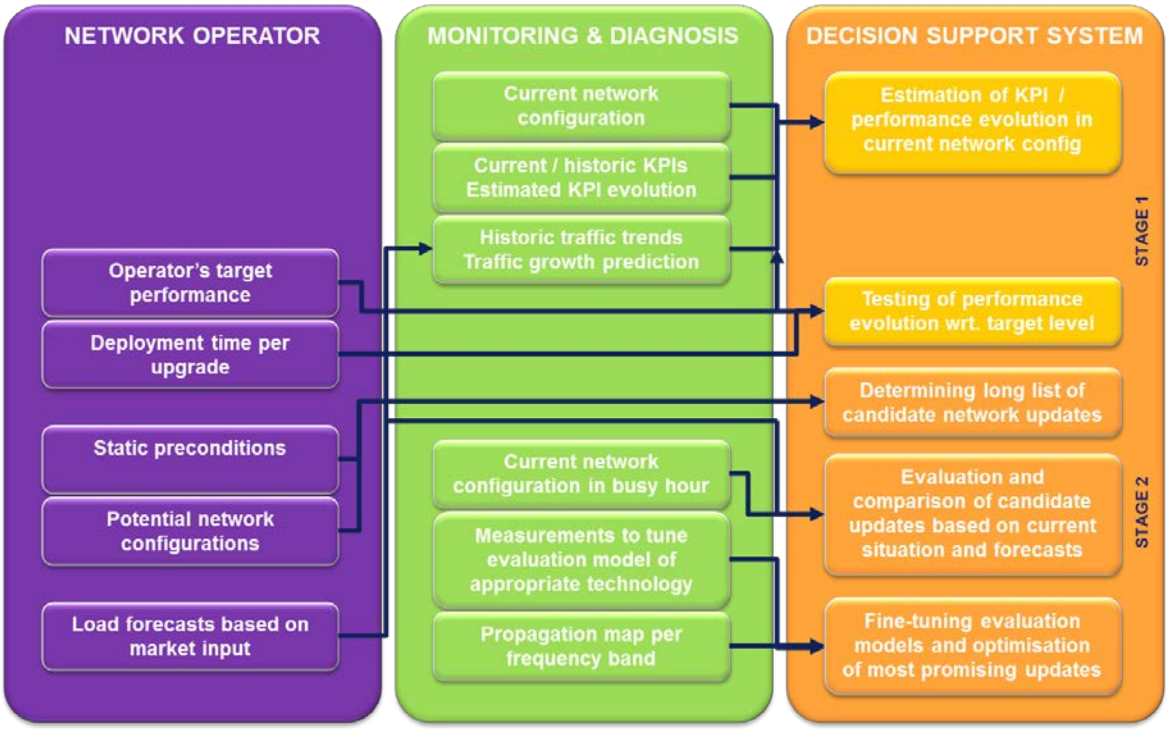

Figure 18: Overview of main DSS input parameters and their origin (for STM and NE use cases)

Note that the DSS functional architecture defined in this section represents a first concept, considering the requirements of the two use cases DSS-STM and DSS-NE. In the further work on the DSS, with developing and implementing the third use case that has been proposed in D2.1 [9], namely, 'Resource costs of QoS as input for SLA management' (DSS-SLA), the functional architecture may need to be refined or enhanced to address specific DSS-SLA requirements. DSS-SLA deals with supporting the operator in making trade-offs in (re)negotiations on Service Level Agreements (SLAs) with service providers. Thus, it requires insight into the additionally required resources (costs) needed to provide a higher QoS level to a service provider. This should be obtained from traffic, performance and resource usage measurements to be provided by the MD function, but will also require information from the network operator on the SLA related goals.

4.2 Initial DSS Demonstrator

First, in Subsection 4.2.1, a short overall description of the initial demonstrator is given. Then, the following subsections zoom in on several specific aspects of the demonstrator. Finally, the main 'storyline' for the demonstrations is described.

4.2.1 Overall Set-up

The demonstrator uses a realistic mobile network environment ("Hannover area") as baseline scenario, including realistically chosen traffic loads (for more details about the generation of traffic loads, see Section 4.2.2 below). The involved LTE multi-layer access network comprises macro cells only. To speed up the demonstrations and make them practically feasible emulation of the 'real' network behaviour over time (e.g., several months) is done through evaluation of network performance at a *n-daily/weekly* basis taking into account a given traffic growth function (details are given below). The initial version of the demonstrator comprises a limited set of possible network upgrades from the DSS use cases DSS-STM and DSS-NE.

4.2.2 Traffic Load Scenarios

A realistic initial traffic load scenario has been determined taking into account the various land-use classes (roads, buildings, commercial, industrial, green space, etc., see Figure 19) for the Hannover area. In addition, various traffic growth scenarios over longer time periods (typically 6-12 months) have been defined. A traffic growth scenario uses spatially different growth rates depending on the geographical location and land-use class. In more detail, per land-use class/geographical location a growth rate plus some random 'noise' according to a lognormal distribution is used, in order to account for 'realistic' fluctuations in the traffic growth. An example for a traffic intensity map for the Hannover scenario is depicted in Figure 20.

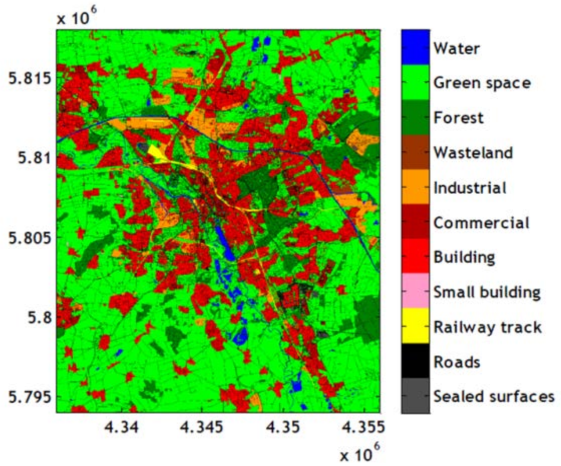

Figure 19: Land-use classes for the city part of Hannover

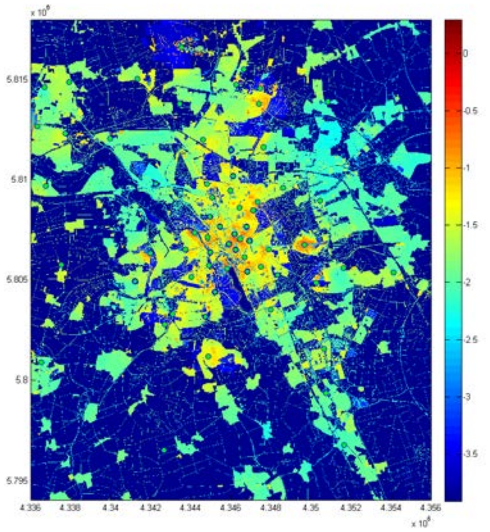

Figure 20: Traffic intensity map (example) for the area of Hannover

4.2.3 Possible Network Upgrades

The initial demonstrator provides two options for network upgrades: (i) a site upgrade in terms of extending the number of sectors (from 3 to 6 sectors) and (ii) addition of a new site. Regarding the addition of a new site there is a limited, predefined set of candidate locations from which the DSS should select the best one(s) as possible solutions to be suggested to the operator.

4.2.4 Network Simulation/Evaluation

Network simulation/evaluation tools are needed both for emulating the 'real' network behaviour over time and for evaluating/comparing possible solutions for future bottlenecks identified in the triggering phase. For both cases the mobile network simulation environment SONLAB[2] is used. Note that, as mentioned before, the network behaviour over time is represented through the traffic load and network performance (to be evaluated by SONLAB) during the 'busy hour/day' (per 'area') of each n-th day/week.

4.2.5 Algorithms for Triggering and Upgrading

Detection of possible future bottlenecks and determining a shortlist of recommendations for network upgrades that will remove these bottlenecks is done along the lines presented in the previous section (see also D2.2). Note that triggers (and upgrades) are derived per 'area' (cell), i.e. future network performance/utilisation in the network is checked for each 'area'.

[2] SONLAB is a common radio network simulation platform used within SEMAFOUR, see also [30]

Initial, simplified algorithms for triggering and upgrading have been developed. Triggering is currently based on simple extrapolations of the observed trends in traffic loads and the achieved performance at cell level. Regarding upgrading all 'locally possible' upgrades (i.e., upgrade of a bottleneck site from 3 to 6 sectors, or addition of a new site at predefined locations in the neighbourhood of the bottleneck site) are scanned, evaluated (using SONLAB, see Section 2.3) and compared. After the demonstration at the ICT 2013 conference these algorithms will be extended and enhanced for use in future versions of the demonstrator.

4.2.6 Storyline

The storyline for the demonstrations is as follows. First, the main role of the DSS is explained. In particular, it is emphasised that the unified self-management system cannot resolve/prevent all performance problems. Sometimes, e.g. when the traffic keeps growing, more 'drastical' changes of the network configuration are needed that require manual action by the human operator. That is where the DSS comes into play. Then the DSS demonstration runs as follows (cf. the snapshot of the demonstrator in Figure 21):

- The simulator is started to get to the instant when the performance/utilisation forecasts fire a trigger for DSS evaluations. During this period, a performance/utilisation pixel map is shown in the network panel and the evolution of the KPIs / Key User Indicators (KUIs) on the KPI panel.
- At this point the DSS (behind the scenes) creates and assesses a list of upgrades. What the demonstration audience sees is a DSS report named 'Recommendations for network upgrades' to be indicated on the DSS panel.

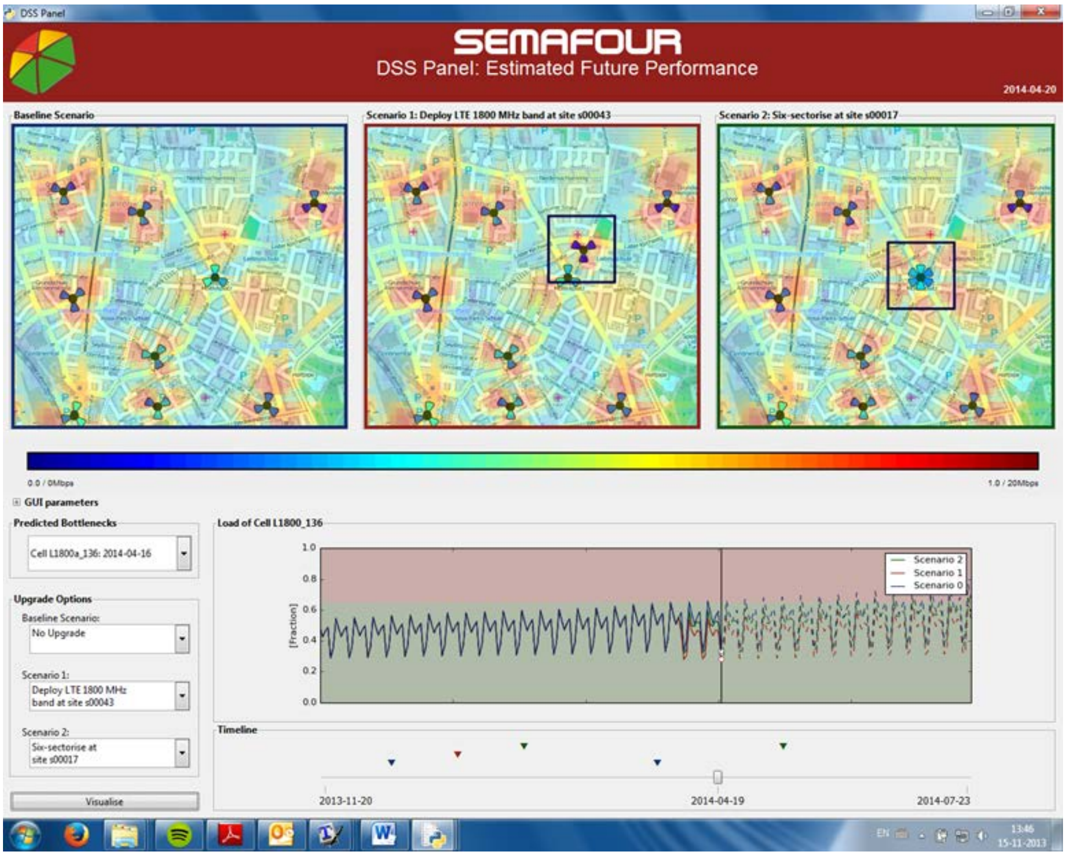

Figure 21: Snapshot of DSS Demonstrator

- With selecting this report, two options are provided:

  Option 1: Open report (PDF file), which presents the analyses (forecast of bottleneck, evaluation of upgrades).

  Option 2: Visualisation of performance upgrades. If this option is activated,

  - A new DSS panel is opened up. A red banner on the panel is used to clarify that this is not reality anymore, yet a prediction of future behaviour.
  - On this panel, a zoomed-in display of the current pixel map from the network panel is shown.
  - A list of upgrade options that the DSS has analysed and computed can be seen. The demo-giver can checkmark which of the options he wants to be visualised.
  - When starting the visualisation of the checkmarked options, the zoomed-in display is copied as many times as options have been checked, next to the first copy (the default option of doing nothing).
  - A KPI/KUI chart is displayed below this series of zoomed-in displays.
  - All three pixel maps and also the KPI/KUI chart start ‘running’ synchronously, showing the predicted performance/utilisation.
  - Based on the option’s deployment time, the actual network upgrade will be visualised.
  - The performance/utilisation evolutions are in correspondence with the network upgrades and their respective deployment times.
  - The predicted evolutions will run until for each option performance is too low.
  - Conclusions are driven about the degree to which the options resolve the performance issues, and how future-proof they are.

5 Monitoring and Diagnosis

The Monitoring and Diagnosis (MD) function aims at delivering a fully automated way of providing network intelligence, such that entities of the integrated SON management system have a quick and reliable access to network context and performance information. As such, the MD function can be viewed as an infrastructure component of the integrated SON management system.

The functionalities of the MD function span the provision of performance measurements and corresponding network configuration data, informing other entities about relevant events occurring within the network, the evaluation of performance metrics corresponding to operator-defined goals, and analysis of historical KPI data such as extraction of trend information or computing data based predictions.

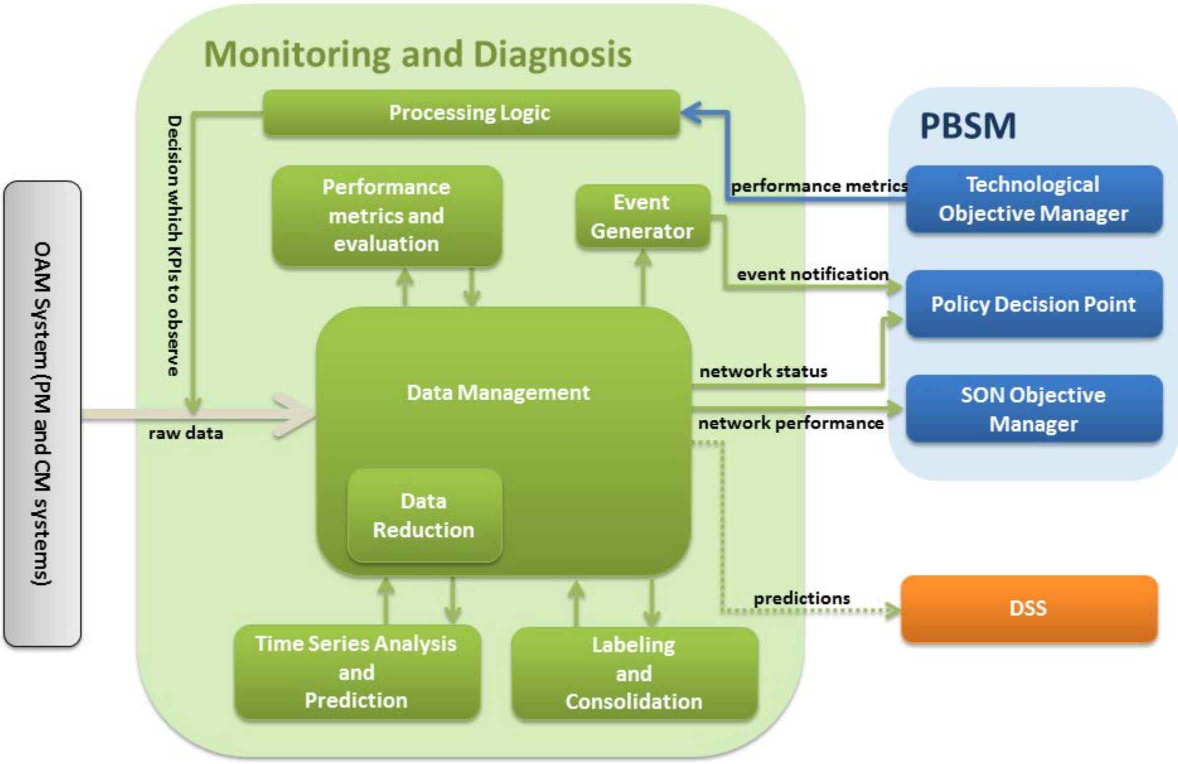

Figure 22: Building Blocks of the MD function

The major clients of the MD function are the Policy Based SON Management, in particular the Policy Decision Point and the SON Objective Manager, and the Decision Support System. Major data sources are performance and configuration management systems. The Technological Objective Manager within the PBSM provides definitions of performance metrics and instructions for their evaluation.

In D2.2 [10], the main functional and non-functional requirements have been identified. Three different forms for the MD function have been introduced varying in complexity and the amount of tasks it is responsible for. The “light-weight version” comprises the basic functionalities for providing network status information from PM- and CM-systems. For the “medium version”, statistical processing of the acquired data is foreseen. This includes labelling and reconstruction of missing and corrupt measurements, time-series analysis, and data reduction mechanisms. The “heavy-weight version” would additionally provide performance assessment functionalities. It was decided to aim for the heavy-weight version of the MD function in the project.

Various statistical mechanisms are foreseen to be evaluated in the context of the MD function. For the time-series analysis, they comprise moving averages, Kalman filters, linear regression, and Gaussian regression; see for example [2] and [3]. For the labelling and reconstruction functionality, several detection mechanisms may be evaluated. A survey of state-of-the-art outlier detection mechanisms can be found in [4]. The envisioned data reduction functionality of the MD is planned to be achieved by the so-called principal component analysis. This method is able to reduce the data dimensionality while retaining most of the statistical features. An extensive introduction to the principal component analysis may be found in [5].

5.1 Scope

The MD function is envisaged as a database that is able to statistically process and evaluate its stored data. Clients can access data within the MD in a similar way as for usual databases, whether it is raw data or the result of statistical processing. This ensures a simple and versatile interface between the MD and its clients. In the background, the MD takes care that the available data is kept up-to-date and may statistically process, consolidate, and label the data.

The “raw” data collected from the performance and configuration management systems may be faulty or incomplete. For this reason, the MD function performs reliability checks whenever possible by cross correlating KPI values or comparing with historical data, if available. Measurements which are missing or deviate strongly from expected values are flagged appropriately and may be estimated using statistical tools.

In order to cope with the large amount of available performance measurements, the MD function implements means to ensure efficient working. This includes a processing logic, which determines the data that is necessary to perform the required tasks on the one side, and on the other side a data reduction mechanism, which compresses the data stored within the MD.

The MD works in the context of the intermediate (technological) layer as described in Section 2.2. This layer contains technological KPIs that are measurable and “provide a detailed technical network description on which the network performance can be assessed”. At this layer, the MD provides network context information as well as performance assessment in line with the technological objectives stored in the Technological Objective Repository. In the following, the relation of the MD function with its major clients is outlined.

Policy Decision Point

The Policy Decision Point within the PBSM (or the SON Objective Manager itself in case of the run-time option) is required to timely react to changing network situations and activate ECA rules suitable for the given network status. The activation process relies on the up-to-date network context, potentially including network configuration parameters or performance metric values. This information is provided by the MD function, which is designed to acquire the underlying data and keep them up-to-date. In case of performance measurements, the data may be inspected in terms of timeliness and reliability and is flagged in case of irregularities. There are different possible ways to receive the measurement results. One method is to actively access the MD database and fetch the required information. Another possibility is for the MD to generate event notifications in case certain predefined situations involving the observed metrics occur, e.g., thresholds for KPI values are exceeded.

SON Objective Manager

In order to optimise the policy derivation process performed by the SON Objective Manager within the PBSM, policies are evaluated in terms of their achieved network performance. The assessment of network performance, however, is required to be in line with the operator goals. The MD function receives the definition of performance metrics by the PBSM as well as instructions how they are to be evaluated. This enables it to determine underperforming parts of the network and hint at improvable SON policies. Network performance is inherently dependent on configuration parameters, and those parameters may change frequently in the presence of SON functions. To be able to interpret given performance measurements it may therefore be necessary to record configuration parameter changes alongside performance data.

Decision Support System

The Decision Support System is planned to identify performance problems within the network and evaluate various upgrade options. Such evaluations may involve sophisticated simulations and are usually computationally expensive, such that the use of simpler heuristics may be helpful. Data based prediction of KPI values and performance metrics may provide acceptable approximations for hinting at possible future bottlenecks. The MD function holds much of the relevant performance data, such that one may consider having such prediction capabilities reside within the MD function. Since the development of the DSS use cases is still in an early stage, it has yet to be decided if data based prediction algorithms are a viable option and which part of the integrated SON management system would be responsible for it.

5.2 Building Blocks

Even though the MD should appear to its clients as simple as a database, there are several internal functionalities that ensure the required data quality and functionality. In the following, the major building blocks of the MD functions will be described in more detail to give a clearer picture of the MD's intended functionalities.

5.2.1 Data Management

The network data that is being gathered from PM- and CM-systems as well as the results from statistical analyses will be stored within the MD function. The Data Management building block of the MD function is responsible for the organization, accessibility, and storage of the network information including data reduction mechanisms.

Storage

The MD function collects raw data required for the assessment of network performance and the provision of network context from performance and configuration management systems and stores them. The specifics of the collection process, such as expected granularity, time delays, and resolution, are determined by the processing logic of the MD.

Data from performance and configuration management functions within existing Operation, Administration and Maintenance (OAM) systems can often only be made available with considerable delay, such that it has yet to be determined whether modern operation and maintenance systems fulfil the actuality requirements of the MD function. Depending on their capabilities it may be necessary to rely on additional data sources, e.g., direct access of network elements or SON functions. For example, performance data is usually collected only within certain granularity periods by the OAM system, and these granularity periods can last from several minutes up to hours or days. Such a delay, however, is not practicable for PBSM or SONCO tasks. Another problem can be the up-to-dateness of configuration data within the OAM system in case SON functions can perform local changes at the NEs without updating this data at the OAM configuration database.

Depending on the specifications in the corresponding monitoring job, the data is then statistically processed and evaluated. Both the result of the processing as well as the underlying raw data is kept within the data storage.

Data Reduction

Although the data collection by the MD is limited to only those parts that are required as determined by the processing logic, the large amount of performance and configuration data to be stored leads to the question of data compression. As described in D2.2 [10], in current performance management systems, data points are aggregated (e.g., averaged or re-sampled) and eventually discarded as they age, following several stages of a compression hierarchy. This method, however may lead to data being purged too early such that important statistical features get lost. A more promising alternative compression strategy may be accomplished by applying the so-called Principle Component Analysis (PCA), see [5]. After analysing statistical correlation, the stored data is transformed into artificial, mutually uncorrelated quantities. By discarding those quantities of low statistical relevance, a compression of the data set can be achieved while retaining most of the statistical features.

Due to the lack of available test data and infrastructure, the PCA methodology could not yet be evaluated within the project.

5.2.2 Labelling and Consolidation

After the raw data have been collected from the performance and configuration management systems and stored in the database, the new data may be checked for reliability and labelled accordingly. The statistical tools that are employed for the data consolidation have already been introduced in D2.2. Due to the lack of available test data and infrastructure, they could, however, not yet be tested and evaluated. The intended mechanism is briefly described below for the sake of completeness.

The raw data that the MD receives from the operation and maintenance systems is often incomplete or inconsistent. If, for example, due to large communication overhead or other system failures, a series of measurements could not get reported from the network elements to the performance management systems, they will arrive at the MD function with a certain delay, if at all. When a client requests the current status of this KPI, only old, eventually outdated, data is available.

The activation decision of the Policy Decision Point, for example, relies on temporal and contextual network information. Incorrect activation of ECA rules based on an outdated network status may lead to bad network performance and should be avoided. In that situation it is advantageous to know whether the currently available data is up-to-date or not.

Through the information provided by its processing logic, the MD is aware of expected granularity and time delays for the data that is to be collected. It is therefore possible to recognise unnaturally long delays and, in case, flag the available data as “outdated”. If new measurements that correspond to later times are collected, the missing parts may be labelled as “missing”. This may be important, for example, if the required metric demands the computation of averages from several data points. If parts of the data are missing, the averages are less reliable.

Missing data may be estimated with the help of statistical imputation methods or time series analysis. Since such estimations would still give a distorted picture, these would be labelled as “estimate”.

Apart from estimating missing data points, it is possible to employ these methods to analyse measurements to detect unusual behaviour. This can be achieved by using the estimation algorithms, which in general reproduce “regular behaviour”, and comparing the estimation results with the actual measurements. A large discrepancy hints at unusual behaviour, such that the measurement may be labelled as “suspect” and further analysis may be initiated.

5.2.3 Time Series Analysis and Prediction

The data that is stored within the MD may be statistically analysed in order to extract relevant information and to understand the dynamics of the measurements. Moreover, performance metrics may directly depend on statistical features of the measurement series, e.g. statistical variance may indicate oscillatory network behaviour. The time series analysis has been introduced in SEMAFOUR D2.2 [10]. A short overview over the intended methods is given below.

By analysing the series of measurements, it is possible to determine various factors that influence their dynamics. These factors may be trends (or growth rates), seasonal behaviour, the effect of holidays or weekends, daily fluctuations, or abnormal effects. Knowing these factors enables the MD to discern “regular behaviour” from “unusual events”. Such information may be used to classify and label measurements.

Also, trend and seasonal information can be used to compute and evaluate performance metrics. A technological objective within the PBSM could, for example, not only demand that a call drop rate threshold is not exceeded, but also that the average call drop rate for a certain area should not be showing a growing trend on the long run.

Linear regression and sliding averages provide suitable methods to extract this kind of information. A more sophisticated analysis technique is the so-called Gaussian regression, see [3]. It allows using complex models, also implementing additional contextual information, for the regression of measurement series. In addition to that, the Gaussian regression may be used to estimate missing measurements, thus supporting the “Labelling and Consolidation” building block of the MD.

Another possible use case for the time series analysis by means of Gaussian regression is data based prediction of future data values. Together with statistical inference methods such as regression trees, this method could be employed to support the DSS in the detection of future performance bottlenecks. Due to the early stage in the DSS development, it will have to be discussed whether this method is a viable option and whether it should be considered as a part of the MD.

5.2.4 Performance Metrics and Evaluation

Additionally to providing network context information to the Policy Decision Point within the PBSM, the MD is responsible for the assessment of network performance. The results of the performance assessment are in particular used for the optimisation of the policy derivation process within the SON Objective Manager by evaluating and comparing the network performance for various policy variants.

The Technological Objective Manager within the PBSM identifies suitable performance metrics with which the operator’s goals may be expressed as closely as possible at the intermediate (technological) layer. These performance metrics depend on measurable KPI values and counters such that they can be computed by the MD function for any network state.

Values of the performance metrics by themselves do not necessarily need to admit an evaluation of the network performance. For example, a cell load value of 0.6 may be considered as acceptable or bad network performance depending on the technological objectives. Therefore, the MD function additionally receives evaluation instructions for registered performance metrics.

5.2.5 Processing Logic

Due to the large amount of data available in mobile networks, it is necessary to limit the investigated and stored data to those parts that are relevant for the MD’s clients. Moreover, the MD function should maintain the flexibility to work in various network environments and with different clients. For these reasons, a processing logic is provided that enables the clients to specify the MD’s behaviour and define the data to be stored.

The clients of the MD provide information about the data they expect the MD to collect, observe, and process. This may include the composition of KPIs, whether they should be processed in any way (e.g., computing sliding averages), the area (e.g., a set of cells) in which measurements shall be taken and how these are to be aggregated (e.g., maximum, minimum, average, percentiles), the expected time granularity of measurements, an expiration date, and other information. Furthermore, it may include the instruction to generate event notifications for certain situations based on the KPI measurements.

The processing logic is responsible that the requested data is stored and accessible in the MD’s database and, in case of demand for event notifications, informs the event generator of the requested tasks. Depending on the requirements that have been specified by the client, the processing logic triggers the various processing capabilities within the MD.

As an example, consider the Policy Decision Point within the PBSM. It decides which ECA rules are to be activated based on contextual network information. Apart from standalone quantities such as time, the decision may depend on performance and configuration data, e.g. average cell load levels for a given area. For these, the Policy Decision Point instructs the MD to keep track of the corresponding KPIs and generate an event notification in case, a certain situation occurs, e.g. the sliding average of measurements during two consecutive hours of the average cell load in a given urban area surpasses sixty per cent. The processing logic triggers the collection and storage of the base cell load values for the cells in question and the computation of the sliding averages. The event generator is then instructed to test the resulting values against the threshold of sixty per cent and inform the Policy Decision Point in case the test succeeds.

5.2.6 First Implementation

First efforts have been made to implement and test functionalities of the MD function. In particular, the Time Series Analysis and Prediction part, see Section 5.2.3, has been investigated in the course of the development of the initial DSS demonstrator, see Section 4.2. However, with the further development of the DSS functional architecture, and the implementation of additional DSS use cases, these tests may need to be repeated.

The scenario that has been developed for the initial DSS demonstrator covers a time span of six to twelve months and includes snapshots for each n-th day, typically the busy hour. For these snapshots, network performance data, in particular traffic volume per cell and cell loads, is computed by the radio simulation platform SONLAB[3] and serves as input for the Time Series Analysis and Prediction.

Considering a fixed timestamp of the simulation as “now”, the historical performance data that has been collected up to that point in time is analysed and then used to predict the evolution of the performance in the future. Depending on the granularity of the historical data and the intended purpose of the predictions, the prediction length can vary between hours and months. In particular in the context of the Initial DSS Demonstrator, a prediction length of four months has been chosen.

For this first approach, two steps have been implemented to compute the prediction results. First, the growth rate of the historical data is determined and a suitable average function is computed. As a second step, a Gaussian regression model is trained on the remaining residuals and then evaluated to compute the corresponding predictions. Regular fluctuations, such as different performance profiles for weekends, are taken into account by the regression model.

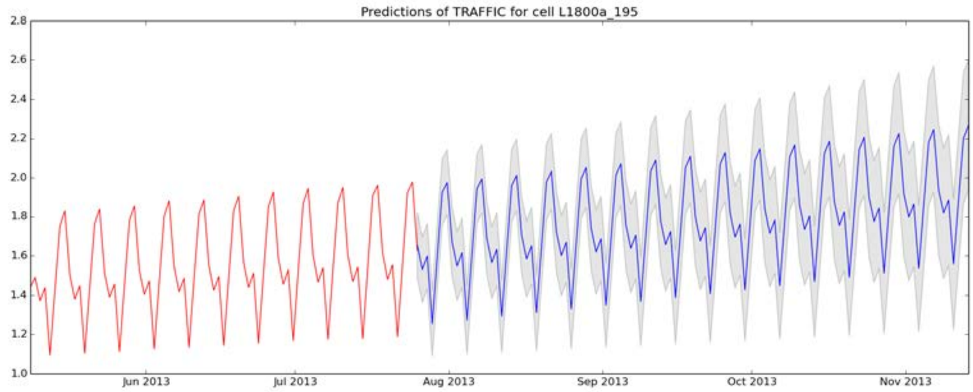

Figure 23: Example for a traffic prediction

In Figure 23 the result of a traffic prediction for one cell of the scenario of the Initial DSS Demonstrator is illustrated. The historical data up to “now” is shown in red colour and the corresponding prediction is depicted in blue colour. The grey area in the background of the prediction shows the standard deviation corridor.

The figure is based on daily traffic measurements at 4pm. One can see that the prediction follows the growing trend of the historical data. Moreover, the regression model is able to capture the weekly traffic profile with lower traffic values during weekends.

[3] SONLAB is a common radio network simulation platform used within SEMAFOUR, see also [30]

6 Summary and Next Steps

During the initial project phase the efforts of the project devoted to ***Policy-based SON Management*** (PBSM) have focused on devising the global functional architecture and the PBSM associated business, functional and non-functional requirements (included in Deliverable 2.2 [10]). In parallel the main operators' goals have been identified and studied. As the second step it was decided to focus on the SON function instrumentation, and for this purpose extensive simulations with well-known SON functions have been conducted (cf. Section 2.3). The outcome of this simulation work was the finding that different configurations of a SON functions lead to a clearly visible different behaviour of the SON functions in such a way that their impact on the network changes with regard to certain KPIs. These findings contributed to the development of the concept of the SON Objective Manager, which automatically transforms operator-defined KPI targets, which may be prioritised and conditioned, into appropriate configurations for the SON functions. As further input to the SON Objective Manager a standardised description of the behaviour of the implemented SON functions is used.

The implementation of the SON Objective Manager, and a connected SON policy environment, is still ongoing. Simulations and testing results will be used to improve the SON Objective Manager concept, in particular regarding the automatic adaptation of SON function models according to network analysis results provided by the Monitoring and Diagnosis function. Furthermore, the future research work will focus on the process to transform the high-level operator goals into technological objectives (i.e., the identification of target values for KPIs/metrics, priorities and additional conditions). In addition, the interconnection between the operational SON coordination and PBSM plays an important role, with respect to identifying the appropriate configuration of operational SON coordination from PBSM, and to include feedback information from the SON coordination into PBSM. The PBSM interfacing with SON functions on the one side, and the network operator on the other side, may both require standardised interfaces to allow multi-vendor implementations. This topic is dedicated to the concluding activities within WP5.

The initial work on ***Operational SON Coordination*** (SONCO) focused on the analysis of existing concepts, in particular, the results from the European funded projects SOCRATES and UNIVERSELF. Based on this analysis it was decided to put the SONCO focus on the detection and resolution of conflicts between concurrently operating SON functions, while the conflict avoidance (or harmonisation between SON functions) shall be part of PBSM. This led to the definition of the functional architecture of SONCO, where also a decision on the implementation approach has been made – with the assumption that SON functions work in a fire-and-forget mode, i.e., they do not require feedback from SONCO on the coordination decision. It became also clear that the existing SON coordination concepts, which mainly build on a uniquely defined, non-adaptive interaction of SON functions, may not be optimal with respect to the performance of conflict resolution. A first step to overcome these restrictions has been made with the concept of adding an additional factor in the coordination decision making that reflects the degree of satisfaction of the coordinated SON functions.

There are still several issues to be solved with respect to SONCO. A challenge which has not been sufficiently addressed up to now is the implementation of SONCO in real systems, where SON functions with a different scope (e.g., single cell, single base station, or network-wide) operate concurrently, and neither a centralised nor a distributed SONCO implementation fulfils all requirements. Furthermore, the interaction with the PBSM is an important issue to be solved, with respect to the configuration and instrumentation of the SONCO function according to the operator goals. The conflict detection task of the SONCO needs to be further investigated. Besides, as the SONCO detects conflicts by observing and analysing the behaviour of the network configuration parameters and KPIs, the interaction with the MD needs to be studied. A solution is furthermore required for the avoidance of undesired behaviour of interacting SON functions, for example, oscillating changes to the network configuration which cannot be prevented by the SONCO, but need to be resolved by PBSM.

The initial work on the ***Decision Support System*** (DSS) concentrated on defining use cases and a functional architecture, but particularly on implementing the DSS approach into the SEMAFOUR demonstrator. The first feedback on the initial version of the DSS is promising, in particular with respect to the feedback from the SEMAFOUR Advisory Board members and the visitors of various demonstrator presentations. However, the actual conceptual and implementation work is still to be done. Thereafter, in the remainder of the project the work on DSS will be extended in several ways. In particular, regarding the DSS-STM and DSS-NE use cases considered, the set of possible network upgrades (see Section 4.2.3) will be considerably extended (see the plans provided in D2.1 [9]), which requires major generalisations/modifications of the initial algorithm concepts for triggering and upgrading that have been developed for the current version of the demonstrator. Besides extension of the present work on the DSS-STM and DSS-NE use cases, also methods and algorithms for the third DSS use case proposed in D2.1 [9] will be developed and implemented. This use case ‘Resource costs of QoS as input for SLA management’ (DSS-SLA) deals with supporting the operator in making trade-offs in (re)negotiations on Service Level Agreements (SLAs) with service providers. Thus, it requires insight into the additionally required resources (costs) needed to provide a higher QoS level to a service provider. This should be obtained from traffic, performance and resource usage measurements to be provided by the MD function.

For the ***Monitoring and Diagnosis*** (MD) function, the first studies concentrated on identifying its main functionalities and role within the framework of the integrated SON management. As an infrastructure component, its main task is to provide network intelligence at the level of the intermediate, technological layer. It appears as a database for network configuration and performance data that is extended by various data consolidation and analysis functionalities. Some of the tools are intended to work in the background, such as reliability checks, data labelling, or data reduction and some may be triggered externally, such as prediction algorithms. Based on technological objectives derived within the PBSM and corresponding performance metrics, the MD is supposed to determine to which extent the operator’s goals are met. In relation to the Policy Decision Point within the PBSM, the MD function may generate event notifications, given certain predefined situations are met, e.g., thresholds for KPIs are exceeded. First implementations of prediction functionalities are available.

For the next steps the algorithms used in the implementation of the prediction tool need to be evaluated for different types of data and, in case of insufficiencies, alternatives need to be considered. Their relation to the data labelling mechanisms will be investigated and other methods considered. Detailed concepts for the performance analysis are to be developed and corresponding algorithms identified.

7 References

[1] Seppo Hämäläinen, Henning Sanneck, Cinzia Sartori (Editors), 'LTE Self-organizing Networks (SON): Network Management Automation for Operational Efficiency', John Wiley & Sons, 2012

[2] J. D. Hamilton, 'Time Series Analysis', Princeton University Press, 1994

[3] C. E. Rasmussen, C. K. I. Williams, 'Gaussian Processes for Machine Learning', MIT Press, 2006

[4] V. Hodge, J. Austin, 'A survey of outlier detection methodologies', in Artificial Intelligence Review, Vol. 22, No.2, pp. 85-126, October 2004

[5] I. T. Jolliffe, 'Principal Component Analysis', Second Edition, Springer Series in Statistics, 2002

[6] L.C. Schmelz et al, 'A Coordination Framework for Self-Organisation in LTE Networks', 12th IEEE/IFIP International Symposium on Integrated Network Management (IM 2012), Dublin, Ireland, May 23-27, 2011

[7] T. Bandh, L.C. Schmelz, 'Impact-time Concept for SON-Function Coordination', in proceedings of the 2012 International Symposium on Wireless Communication Systems (ISWCS), pp. 16-20, Paris, France, August 2012

[8] EU FP7 project SOCRATES (Self-optimisation and self-configuration in wireless networks), Deliverable D5.9: 'Final report on self-organisation and its implications in wireless access networks', January 2011

[9] EU FP7 project SEMAFOUR (Self-management for unified heterogeneous radio access networks), Deliverable 2.1, 'Definition of self-management use cases', March 2013, available at http://www.fp7-semafour.eu

[10] EU FP7 project SEMAFOUR (Self-management for unified heterogeneous radio access networks), Deliverable D2.2: 'Definition of Requirements for a Unified Self-Management System', July 2013, available at http://www.fp7-semafour.eu

[11] EU FP7 project SEMAFOUR (Self-management for unified heterogeneous radio access networks), Deliverable 2.4, 'Definition of Reference Scenarios, Modelling Assumptions and Methodologies', June 2013, http://www.fp7-semafour.eu

[12] 3GPP TS 28.627, 'Telecommunication management; Self-Organizing Networks (SON) Policy Network Resource Model (NRM) Integration Reference Point (IRP); Requirements', version 11.0.0, December 2012 (http://www.3gpp.org/ftp/Specs/html-info/28627.htm)

[13] 3GPP TS 28.628, 'Telecommunication management; Self-Organizing Networks (SON) Policy Network Resource Model (NRM) Integration Reference Point (IRP); Information Service (IS)', Technical Specification (http://www.3gpp.org/ftp/Specs/html-info/28628.htm); version 11.0.0, December 2012

[14] 3GPP TS 36.300, 'Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2'

[15] NGMN Alliance, 'NGMN Use Cases related to Self Organising Network, Overall Description', Deliverable, May 2007

[16] NGMN Alliance, 'NGMN Informative List of SON Use Cases', Annex deliverable, April 2007

[17] NGMN Alliance, 'NGMN Recommendation on SON and O&M Requirements', Requirement specification, December 2008

[18] NGMN Alliance, 'NGMN Top OPE Recommendations', Deliverable, September 2010

[19] EU FP7 SOCRATES project website: http://www.fp7-socrates.eu/

[20] EU FP7 UniverSelf project website: http://www.univerself-project.eu/

Appendix A Glossary

There are a couple of definitions used within this document that need to be clarified in advance.

- **Design-time**: The time during which the SON function is built and can still be modified in the development environment.
- **Key Performance Indicator (KPI)**: A KPI is defined as one, or a set of, measurements, counters, timers etc. being collected from the network elements, or calculated from performance information collected from the network elements. A KPI usually represents a certain property of the network element or the network that is of high interest for the operator.
- **KPI target**: In the technological goals the operator defines reference values for KPIs, i.e., the KPI targets the system should achieve. The purpose of the SON functions in then to configure the network elements such that these KPI targets are achieved.
- **Policy**: In the context of SEMAFOUR, a policy describes a set of rules, in particular, Event – Condition – Action rules. This means that, on the occurrence of a certain event, conditions are checked, and if conditions apply, a certain event is created which may, e.g., cause a SON function to start. Note that a policy may in an extreme case consist of only one rule. An example for a policy is an MLB policy that contains all ECA rules required to configure the MLB function.
- **SON coordination**: The SON coordination is a function in the integrated SON management system that is in charge of coordinating the SON functions, whenever coordination is needed. The SON coordination function comprises detecting conflicts, predicting conflicting situations and avoiding them.
- **SON Function Configuration Parameter (SCP)**: A SON function can be configured by means of SCPs, which thereby represent the configuration interface of a SON function. Such parameters may include, for example, thresholds within which the SON function is allowed to operate, step sizes for the network element configuration the SON functions modifies, or activation thresholds for the SON function algorithm.
- **SON Function Configuration ParameterValue (SCV)**: These are the actual values for the SCPs. A dedicated SON function configuration is represented by an SCV Set, i.e., one collection of SCVs for all SCPs a SON function has. By applying a certain SCV Set, a certain behaviour of the SON function can be achieved. Therefore, there may be different SCV Sets for one SON function, in order to configure it according to different technological objectives.
- **SON function instance**: A SON function itself only represents an algorithm or software program. A SON function instance is the actual run-time instantiation of this algorithm or program. Depending on the scope of a SON function, the algorithm or program can be instantiated multiple times, i.e., several instances are running simultaneously and concurrently in the network.
- **Operator Goal**: An Operator Goal escribes targets in the operator's business or customer strategies, which are to be solved / addressed by means of the network. For example, operator goals can include customer satisfaction, operational expenditure, network coverage, service quality, service availability etc. Operator goals may not be described in such way that they can be directly interpreted by the network, i.e., they may not be described in concrete numbers.
- **Run-time**: The time during which the SON function is executed in the network. The software code of the SON function can no more be modified by the designer.
- **Technological Objective**: A Technological Objective describes a target for a system KPI (KPI target), which may have a certain priority. Thereby, different technical objectives can have different priorities allowing the operator to define focus areas for the functioning of the SON system. Furthermore, technical objectives (and the priorities) may depend on certain conditions or context information (e.g., time of the day, location, cell type, network status), allowing the operator to define different priorities or values for a KPI target. For example, different target values for the handover failure rate during busy hour or during night time.