Dynamic, Context-Specific SON Management Driven by Operator Objectives

Christoph Frenzel*†, Simon Lohmüller*, Lars Christoph Schmelz†
* Department of Computer Science, University of Augsburg, Germany
{christoph.frenzel, simon.lohmueller}@ds-lab.org
† Nokia Solutions and Networks Research, Network Operations and Architecture, Munich, Germany
christoph.schmelz@nsn.com

***Abstract*—The management and operation of a Self-Organizing Network (SON)-enabled mobile network still requires considerable human effort. On the one hand, SON Functions need to be configured through low-level parameters in order to control the optimization of the network. On the other hand, an operator wants to steer the system with solely technical objectives, and the underlying network should be adapted accordingly. This opens up a gap in network management that is currently closed manually. This paper presents an approach that overcomes the manual gap between technical objectives and SON Functions by choosing the best values for the SON Functions' configurations automatically. Main advantage of this approach is that it allows to manage a system at a high level of abstraction and, at the same time, reduces manual effort. The approach is explained by applying it in a case study in the field of mobile networks with four SON Functions, namely Mobility Load Balancing (MLB), Coverage and Capacity Optimization (CCO), Energy Savings Management (ESM) and Mobility Robustness Optimization (MRO).**

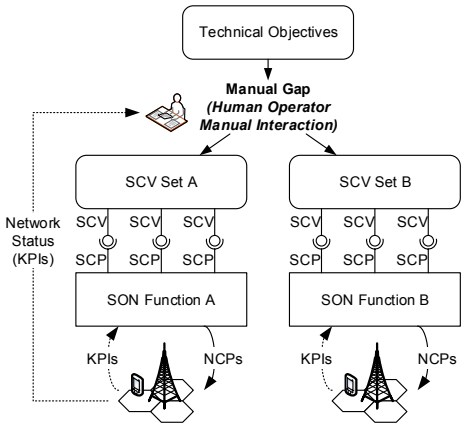

Figure 1. Manual gap between technical objectives and SCV Sets (NCP refers to networks configuration parameter)

I. Introduction

The Self-Organizing Network (SON) paradigm describes a management approach for mobile networks, in which a set of independently acting SON Functions aim at the automation of dedicated tasks in network management, i.e., in the scope of network configuration, network optimization, and failure recovery [1]. Each SON Function represents a closed control loop that adjusts a set of network configuration parameters, e.g., base station transmission power, cell individual offset, handover hysteresis, or time-to-trigger. Through these adjustments, the SON Functions autonomously optimize dedicated Key Performance Indicators (KPIs) of the network such as network capacity, network coverage, call drop rate, handover success rate, or cell load [2] (see Figure 1).

A SON Function can be configured by means of SON Function Configuration Parameters (SCPs) [3]. Depending on the SON Function Configuration Parameter Values (SCVs), the behavior of the SON Function changes, such that it adjusts the network configuration parameters in a different manner. A certain SCV Set, i.e., the collection of SCVs for all SCPs a SON Function has, can modify the SON Function's behavior such that it drives the network KPIs towards a dedicated target value. SCV Sets may be provided by the SON Function manufacturer, and they may be based on operator or project specific requirements. However, often only one default SCV Set for a SON Function is used, and this default SCV Set remains unchanged during network operation, i.e., the SON Functions are usually uniformly configured.

The goal of the mobile network operator is to run the network in such way that it works optimally according to dedicated technical objectives, i.e., the target values that have been defined for the KPIs. The KPI targets may furthermore be dependent on context information such as the time of the day or a certain cell type. Changes to a KPI target or to the context require adjusting the SCV Sets for the SON Functions in order to adapt their behavior in such a way that they contribute to the changed KPI target. In case only default SCV Sets are used, and no adjustment is performed, the SON-enabled network may not operate optimally. If an adjustment of the SCV Sets shall be performed, considerable manual intervention by the human network operator is required in today's mobile networks. This opens up a *manual gap* in the automated operation of a SON-enabled mobile radio network, which is shown in Figure 1.

In this paper, an approach is presented that allows to overcome the manual gap described above. The approach includes a *SON Objective Manager* that performs an automated transformation of KPI targets into an SCV Policy, by creating a state space over all possible contexts, where all KPI targets are mapped to the best suited SCV Sets for the implemented SON Functions. The appropriate SCV Set is then selected by a *Policy System*, which allows to dynamically react on changes in the context, according to the SCV Policy. In sum, the approach facilitates objective-driven control of the SON Functions' behavior.

DOI: 10.1109/NOMS.2014.6838256

II. Manual Gap

In order to motivate the approach presented in this paper, this section introduces a case study which outlines the manual gap. It is deliberately kept simple in order to explain the SON Objective Manager and the Policy System approaches in the following sections along this example.

The primary aim of mobile radio network operations is not the optimization of dedicated single performance indicators at the cell or base station level, but the achievement of dedicated KPI targets. Different KPI targets may be competing with each other, i.e., they are not achievable together. The operator needs to define the importance of the KPI targets in order to trade them off against each other. This importance can be expressed through allocating priorities to the individual KPI targets. Note that a priority here means a precedence of KPI targets and not a weighting. The KPI targets and their priorities may change over time due to changing operator requirements. Furthermore, KPI targets and their priorities may depend on operational or network context, e.g., the time of day, the weekday, the cell location, or the network status. This means that there may be different values assigned to the KPI targets, or a different prioritization between the KPI targets, depending on, e.g., whether the system currently operates in the busy hours or at nighttime, or whether the targeted cell is located in an urban or rural area. In this paper, a context-dependent KPI target and its associated priority is referred to as a *technical objective*.

The following are typical examples for KPI targets:

- Dropped call rate $< 2.5\%$ (indicates the percentage of dropped voice calls due to, e.g., failed handovers or bad radio conditions)
- Cell load $< 90\%$ (indicates the used radio resources per cell or sector)
- Handover success rate $> 99.5\%$ (indicates the percentage of successful handovers between cells or sectors)
- Energy consumption $< 80\%$ (indicates the average consumed energy by the base station compared to the maximum energy consumption)

However, concrete KPI target values have no meaning for the selection of an SCV Set for a SON Function since the configuration of the SON Functions is independent of whether the KPI target value is violated or not. For this reason, in this paper no concrete values for the KPI targets are used, but it is assumed that a KPI target means only to, e.g., minimize the dropped call rate or maximize the handover success rate.

The KPI targets and their priorities may not be the same globally or at all times within a mobile network, but depend on a certain context. Such context can include:

- The time of the day, since the KPI targets and their importance may be different during peak traffic hours and periods with low traffic, e.g., the time period from 08:00 till 17:59, or the time periods from 18:00 till 23:59 and from 00:00 till 07:59.
- The location of the cell, since the KPI targets and their importance may be different in, e.g., urban, suburban, and rural areas, due to user behavior, number of users, or coverage and capacity requirements.
- The cell type, e.g., macro cell, micro outdoor cell, or indoor cell, since the KPI targets and their importance may be different with respect to coverage and capacity requirements, user behavior, or the availability of cells.
- The status of the system based on performance or fault data, e.g., KPI values or alarms.

When combining KPI targets and their priorities with context information, dedicated technical objectives can be derived which build the basis for the operation of the network and, hence, the SON system. In the presented example, the non-formalized technical objectives are the following:

- With a very high priority, the cell load in an urban location during peak hours should be minimized.
- With a high priority, the dropped call rate in an urban location should be minimized.
- With a moderately high priority, the handover success rate during peak hours should be maximized.
- With a moderate priority, energy consumption in a rural location should be minimized.
- With a low priority, the cell load during peak hours should be minimized.
- With a very low priority, energy consumption during periods with low traffic should be maximized.

Based on these technical objectives, the SON-enabled network needs to be configured such that the technical objectives are met. It has been shown in [4] that different SCV Sets for a SON Function can lead to clearly distinguishable network behavior, satisfying specific technical objectives. In other words, the SON Functions can be configured through the SCV Sets to target a particular technical objective. For instance, Mobility Load Balancing (MLB) can be configured with one SCV Set such that it optimizes the network primarily towards a reduced dropped call rate, or with another SCV Set such that it optimizes the network primarily towards a low cell load by balancing the load between neighboring radio base stations [4]. Hence, the technical objectives need to be mapped to specific SCV Sets in order to configure the individual SON Functions such that they contribute to the technical objectives by optimizing single performance measurements or KPIs at the cell or base station level. The mapping from technical objectives to SCV Sets requires technical knowledge about which SCV Set for a SON Function is reasonable for a specific technical objective. This technical knowledge, however, is usually available only within the domain of the SON Function manufacturer, and may not be explicitly formalized.

In the example, four SON Function, namely MLB, Coverage and Capacity Optimization (CCO), Energy Savings Management (ESM) and Mobility Robustness Optimization (MRO) [1], are considered. So, the SCPs of, e.g., the MLB SON Function may include the upper and lower cell individual offset limits, i.e., the virtual cell border defining at which radio reception level a user can be handed over to a neighboring cell [1], [3]. Within these limits MLB can perform changes. For MLB, the cell individual offset thereby also represents the network configuration parameter modified by the SON Function. Further SCPs of MLB are the step size at which

MLB is allowed to modify the cell individual offset, the upper cell load threshold from which MLB becomes active, the lower cell load threshold from which MLB returns to inactive state, and the load averaging time based on which the current cell load is calculated. An example for an MLB SCV Set is:

- Upper cell individual offset limit: +6dB
- Lower cell individual offset limit: -6dB
- Step size: 1dB
- Upper cell load threshold: 50%
- Lower cell load threshold: 30%
- Load averaging time: 60 seconds

Taking the above definition of the technical objectives as context-dependent KPI targets and associated priorities, and the necessity to configure the SON Functions according to these technical objectives, the manual gap can be divided into two major problems (see Figure 1), for which no solutions exist in current systems:

- Technical objectives cannot be interpreted directly by the SON Functions. To enable the operation of the SON-enabled mobile network through technical objectives, an automatic transformation of technical objectives to SCV Sets is necessary (automation gap).
- The SON-enabled mobile network, and thus the operational and network context, may be subject to frequent changes. This in turn requires a dynamic adaptation of the SON Functions' configuration by changing their SCV Sets (dynamics gap).

III. Concept Overview

In order to overcome the manual gap, the approach presented in this paper introduces two main components as depicted in Figure 2. On the one hand, the SON Objective Manager overcomes the automation gap by automatically transforming the technical objectives into an SCV Policy. This transformation is performed at design-time, i.e., before the instantiation of SON Functions, in case the technical objectives have been adapted or SON Functions have changed, e.g., if a new SON Function has been deployed or an old one has been removed. On the other hand, the Policy System evaluates the SCV Policy and configures the SON Functions accordingly in order to overcome the dynamics gap. This configuration has to be performed at run-time, i.e., when the SON Functions have already been instantiated. Conceptually, the SCV Policy is the linking artifact between the SON Objective Manager and the Policy System and, thus, bridges the design-time process with the run-time process.

The task of the SON Objective Manager is to transform the technical objectives into an SCV Policy. The SCV Policy defines for each SON Function an SCV Set which steers the SON Function to fulfill the technical objectives under a specific context, hence, the SCV Set that should be applied. Therefore, the SON Objective Manager determines the best SCV Set regarding the technical objectives for all relevant contexts.

The SON Objective Manager requires a machine-readable, formalized model of the technical objectives which contains

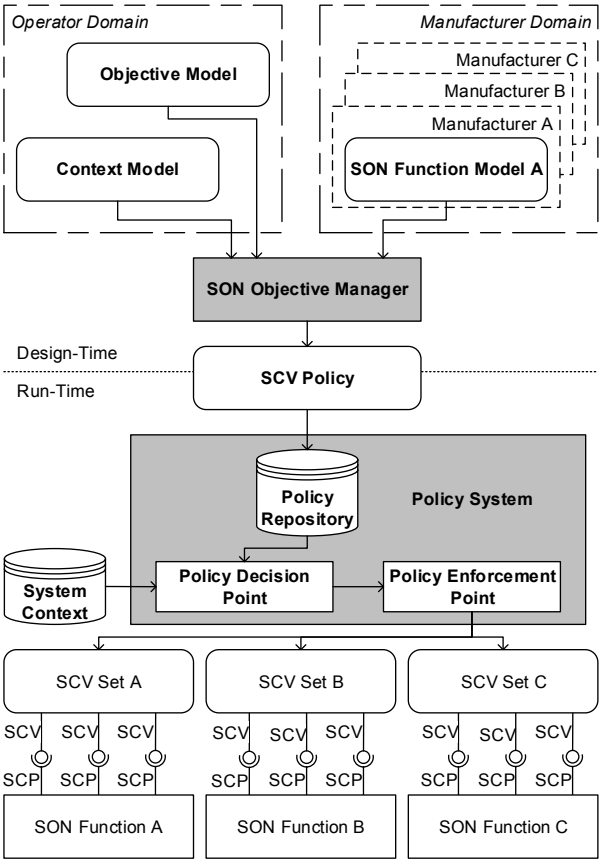

Figure 2. Overview of the SON Objective Manager and Policy System

the context-dependent KPI targets and their priorities. This Objective Model is usually provided by the network operator. Besides enabling automation, the creation of this formal model also supports operators in becoming aware of their technical objectives in the first place.

The SON Objective Manager needs some information about the properties that build up the context (cf. Section II) and their possible values in order to compute the relevant contexts. This information is included in the Context Model which is usually provided by the operator.

In order to determine optimal SCV Sets for the SCV Policy, a machine-readable, formalized description of the SON Functions is required. SON Functions are usually delivered as black boxes by manufacturers, i.e., an operator has no or only few information about the SON Function algorithm or the corresponding mathematical utility function. The SON Function Model allows manufacturers to provide only that information about a SON Function which is required to implement and utilize it properly. Specifically, a SON Function Model contains information on how dedicated SCV Sets for the respective SON Function satisfy specific technical objectives. Such a model is required for each SON Function.

The SCV Policy represents concrete best decisions with respect to which SCV Sets should be applied in order to achieve given technical objectives. Therefore, it contains formalized SCV Policy rules that describe which SCV Set should be applied to a particular SON Function in a specific context. The

Policy System which evaluates the SCV Policy is subdivided into three parts [5]:

- the Policy Repository which stores the SCV Policy,
- the Policy Decision Point (PDP) which evaluates the SCV Policy rules,
- and the Policy Enforcement Point (PEP) which configures the SON Functions with the selected SCV Sets.

Thereby, the Context Database provides the PDP with the current context necessary for SCV Policy evaluation.

IV. System Design

In the following, the details of an implementation of the different parts of the concept are presented. Thereby, the description uses the motivating example introduced in Section II in order to describe the ideas.

A. Objective Model

The Objective Model is implemented as a set of rules, since this is a simple and well-known approach which can be easily understood [6]. Thereby, each of these *objective rules* determines the priority of a KPI target in a specific context. Consequently, the technical objectives from the example are modeled as shown in Listing 1.

```
1  IF time in [08:00, 17:59] AND
       location = urban
3          THEN CL_MIN WITH 1
   IF location=urban
5          THEN DCR_MIN WITH 2
   IF time in [08:00, 17:59]
7          THEN HOSR_MAX WITH 3
   IF location=rural
9          THEN EC_MIN WITH 4
   IF time in [08:00, 17:59]
11         THEN CL_MIN WITH 5
   IF time in [00:00, 07:59] OR
13     time in [18:00, 23:59]
           THEN EC_MIN WITH 6
```

Listing 1. Objective Model example (CL_MIN refers to minimization of the cell load, DCR_MIN refers to minimization of the dropped call rate, EC_MIN refers to minimization of the energy consumption, HOSR_MAX refers to maximization of the handover success rate)

The objective rules have the following general form:

```
IF condition THEN KPI target WITH priority
```

So, they consist of three parts:

- The `condition` part is a logical formula over predicates, which evaluates context properties and, thereby, determines the applicability of the objective rule in a specific context. This allows to specify under which condition, e.g., time periods or cell locations, a KPI target is active and which priority it has. Note that the condition can be empty, indicated by the logical formula `true`, which leads to a general objective rule that is always applicable.
- The `KPI target` defines the KPI that the system should optimize. For instance, in Listing 1 the `CL_MIN` refers to the minimization of the cell load.
- The `priority` encodes the importance of the KPI target to the operator. The KPI target with the highest importance is indicated with a priority of 1, decreasing importance is indicated with priority 2, 3, 4, etc.

Note that some important facts apply for this implementation of the Objective Model. First, the priorities do not need to be unambiguous in some specific context, i.e., it can be the case that one KPI target has two different assigned priorities. This can happen if two objective rules with overlapping conditions and the same KPI target are triggered. An overlap thereby means that at least one specific context exists in which both conditions are true. In such cases, this conflict is resolved by solely considering the higher priority.

Second, it should never be the case that two different KPI targets have an equal priority in a certain situation, since this would mean that it does not make a difference to the operator which KPI target is pursued. In such a situation, the system can not make a deterministic decision. Instead, the triggered KPI targets must be in a total, strict order regarding the priorities in every context. This requirement makes the development more complex, however, the SON Objective Manager provides support for validation and verification of the Objective Model.

Third, the Objective Model does not need to be complete, i.e., not all KPI targets need to be defined in all contexts. As presented later, this might result in the selection of a default configuration for some SON Functions.

Using rules for modeling the technical objectives is only one possible option. An alternative could be to allow the operator to define a utility function which maps the contexts to utilities for the KPI targets. Whereas priorities only allow to rank the KPI targets according to their importance, these utilities would allow to make a trade-off between the satisfaction of different KPI targets. This is especially useful if there are conflicting KPI targets like the minimization of the energy consumption and the minimization of the cell load. However, the elicitation of the utility function requires much more effort than the writing of objective rules.

In this approach, technical objectives are at a low-level of abstraction, i.e., close to the technical details of the system like KPIs. In a realistic scenario, an operator may plan and operate the network in terms of high-level goals which are closer to the business view on the network, e.g., coverage, capacity, or quality of service. Hence, these high-level goals need to be transformed, i.e., refined, into low-level technical objectives. There are promising approaches for such a refinement, e.g., [7], however, this is not in the scope of this paper.

B. Context Model

The Context Model provides a description of the context properties that can be used in the condition part of the objective rules. More precisely, it defines the domain, i.e., possible values, of the context properties that can be used in the predicates of the conditions of an objective rule. The Context Model for the example is:

```
time : [00:00, 23:59]
```

```
MLB Model
        CL_MIN   : MLB_loadEqual
        EC_MIN   : MLB_loadUnequal
        HOSR_MAX : MLB_handover
        default  : MLB_handover

CCO Model
        DCR_MIN  : CCO_on
        CL_MIN   : CCO_off
        default  : CCO_off

ESM Model
        EC_MIN   : ESM_aggressive
        CL_MIN   : ESM_passive
        default  : ESM_passive

MRO Model
        HOSR_MAX : MRO_maxSensitive
        DCR_MIN  : MRO_maxSensitive
        CL_MIN   : MRO_minSensitive
        default  : MRO_minSensitive
```

Listing 2. SON Function Models (CL_MIN refers to minimization of the cell load, DCR_MIN refers to minimization of the dropped call rate, EC_MIN refers to minimization of the energy consumption, HOSR_MAX refers to maximization of the handover success rate)

```
location : {rural, urban}
```

Hereby, the `time` is defined in the range `[00:00, 23:59]`, whereas `location` is an element from the set {`rural`, `urban`}.

C. SON Function Model

A SON Function Model is responsible for encoding a functional description of a specific SON Function. That is, the model describes which KPI targets the SON Function can pursue and how to configure the SON Function accordingly. This knowledge can be expressed in simple mappings from KPI targets to SCV Sets. The four SON Functions in the example are provided with the SON Function Models shown in Listing 2. As can be seen, a mapping in the SON Function Model links a KPI target to a single SCV Set. Note that SCV Set names, e.g., `MLB_handover` in Line 5, are visual placeholders for concrete SCV Sets as shown in Section II.

A SON Function Model must be unambiguous, i.e., for each KPI target there can be at most one SCV Set defined. Otherwise, the SON Objective Manager would not know which SCV Set to use. Furthermore, each SON Function Model needs to provide a default mapping (see Line 5) defining an SCV Set if no matching KPI target is relevant to the operator. This can be, e.g., a balanced configuration of the SON Function which trades off different KPI targets.

A possible extension of the SON Function Model is to make the SCV Sets context-dependent like the Objective Model. This would allow to express different SCV Sets for each SON Function, for example, whether the cell on which ESM is active overlaps with other cells or not, given a so called Heterogeneous Networks scenario [8]. However, this increases modeling complexity because it has to be ensured that the rules of the SON Function Model are conflict-free.

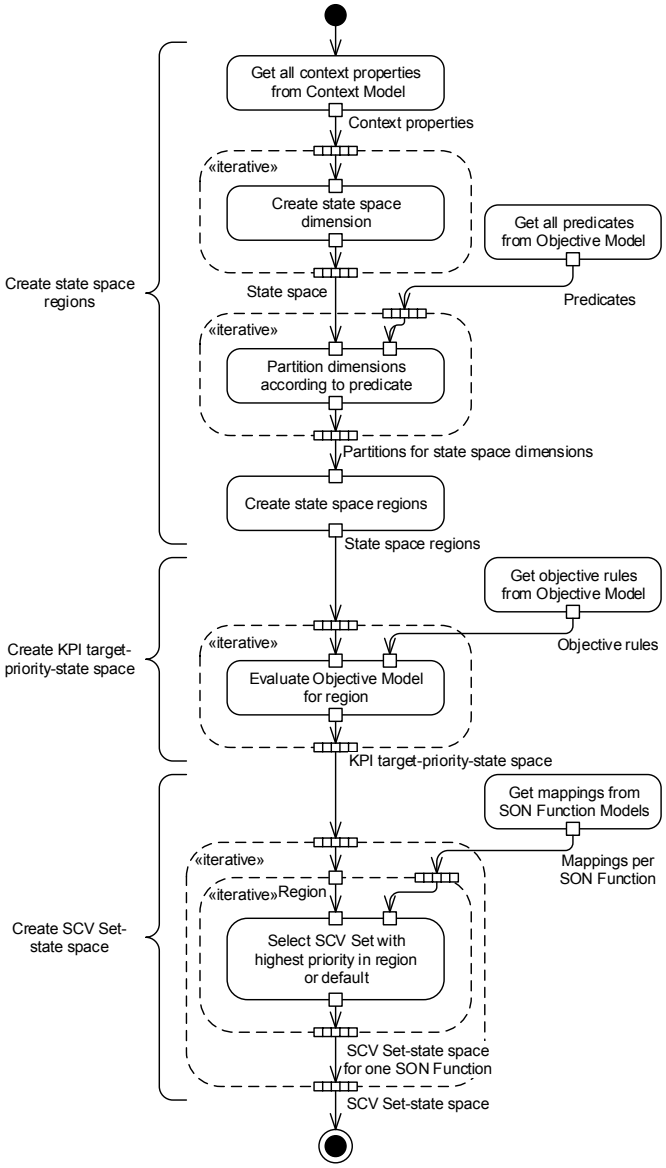

Figure 3. SCV Policy derivation algorithm (noted in UML 2)

For the presentation of the SON Objective Manager in this paper, it is assumed that the KPI targets used in the Objective Model and the SON Function Model match each other, i.e., they have the same name and meaning. This simplifies the explanation but might be too inflexible for a realistic scenario. However, this assumption is not a limitation of the general approach since a translation model can provide a mapping between the KPI targets of both models.

D. SON Objective Manager

Based on the three previously introduced models, the SON Objective Manager derives the SCV Policy according to the algorithm depicted in Figure 3. In principle, it determines the best SCV Sets with respect to the technical objectives in all possible contexts and subsequently creates SCV Policy rules from this information.

In the first step, the system builds up a space of all possible contexts the system could be in, referred to as *state space*. Therefore, it analyzes the Context Model: each context property represents a dimension in the state space and the domain refers to the scale of this dimension. In the example there are two dimensions: the `time` with the continuous scale `[00:00, 23:59]` and the `location` with a discrete scale over the values `rural` and `urban`.

Since the state space can be infinitely large, i.e., it contains an infinite number of states, the system needs to reduce the state space. Therefore, the algorithm divides the state space into a finite number of state space regions with respect to the technical objectives. Specifically, a region is a set of adjacent states which have the same KPI targets and priorities, i.e., in which the SON system should be configured equally. The regions can be computed by analyzing the conditions of the objective rules: for each predicate p, the dimension of the context property in p is partitioned according to the value in p. For instance, consider the following predicate `time in [08:00, 17:59]`. Here, the dimension for the context property `time` would be split into three partitions: `[00:00, 07:59]`, `[08:00, 17:59]`, and `[18:00, 23:59]`. After partitioning the dimensions for all objective rules, the state space regions are defined as the elements of the cross product of the partitions of all dimensions. For instance, one region in the example is defined by the tuple (`time in [18:00, 23:59]`, `location = urban`). Note that the number of regions grows exponentially. For instance, a Context Model with ten parameters and one threshold for each parameter results in 2^{10} regions.

In the second step of the algorithm, the SON Objective Manager determines the KPI targets and their priorities in each region. Since all contexts in a region trigger the same objective rules, this can be done by picking a random state from the region and evaluating the Objective Model for it. The result of doing this for all regions is a KPI target-priority-state space as shown in Figure 4 for the example. For instance, in the region (`time in [18:00-23:59]`, `location = urban`) only the objective rules in Line 4 and Line 12 in Listing 1 apply, thus, defining the KPI target `DCR_MIN`, i.e., the minimization of the dropped call rate, with priority 2 and `EC_MIN`, i.e., the minimization of the energy consumption, with 6. Note that it is possible that a KPI target appears several times in a region with different priorities.

The KPI target-priority-state space is not just an intermediate product of the algorithm but can also be used for validation and verification of the Objective Model. On the one hand, the users of the system can inspect the KPI targets and their priorities for all regions and validate that the objective rules correctly represent their requirements. On the other hand, the system can verify that there are no two KPI targets with the same priority within a region, i.e., there is no confusion in the priority order of the KPI targets.

In the third step of the algorithm, the system determines the SCV Sets for each region based on the KPI target-priority-state space. This is an iterative mapping process for each region r and each SON Function f: from the SCV Sets in the SON Function Model for f, the system selects the one whose KPI target has the highest priority in r. If none of the KPI targets in f's SON Function Model matches any

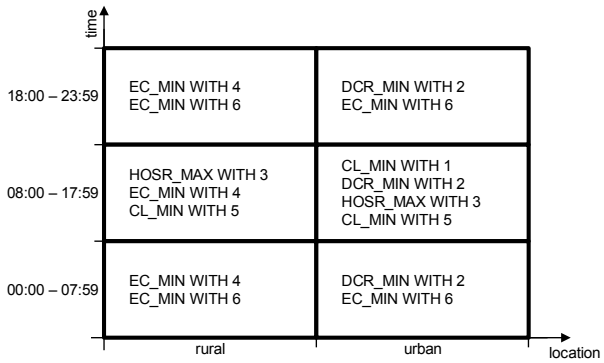

Figure 4. KPI target-priority-state space example (CL_MIN refers to minimization of the cell load, DCR_MIN refers to minimization of the dropped call rate, EC_MIN refers to minimization of the energy consumption, HOSR_MAX refers to maximization of the handover success rate)

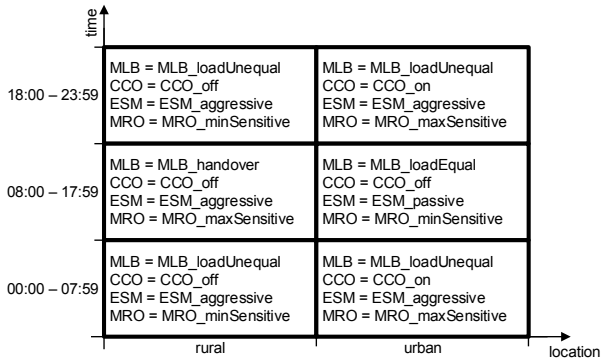

Figure 5. SCV Set-state space example

KPI target in r then the system selects the default SCV Set. The result of this process is an SCV Set-state space as shown in Figure 5 of the example. For instance, in the region (`time in [18:00-23:59]`, `location = urban`) the SCV Set for MLB is `MLB_loadUnequal` because the KPI target with the highest priority in the SON Function Model is the minimization of the energy consumption. Similarly, the SCV Set for MRO is `MRO_minSensitive` because no KPI target in the SON Function Model matches the KPI targets in the region and, so, the default configuration is selected.

Based on the SCV Set-state space, the algorithm can finally compile the SCV Policy as shown in Listing 3 of the example. As it can be seen, the SCV Policy is a set of IF-THEN rules, referred to as SCV Policy rules, which, based on some condition over the context, define SCV Sets for the SON Functions. A simple approach to build up the SCV Policy is to create an SCV Policy rule for each region and each SON Function which sets the corresponding SCV Set. Thereby, the components of the region tuples are translated into the conjunctive condition of the SCV Policy rule. This, of course, results in a large number of SCV Policy rules. An approach, which overcomes this shortcoming and has been used in the example, creates the SCV Policy rules by combining neighboring regions with equal SCV Sets. Note that the SCV Policy is complete and conflict-free because the SON Objective Model has a strict order of

```
1  IF time in [00:00, 07:59] OR
       time in [18:00, 23:59]
3          THEN MLB = MLB_loadUnequal
   IF time in [08:00, 17:59] AND
5      location = rural
           THEN MLB = MLB_handover
7  IF time in [08:00, 17:59] AND
       location = urban
9          THEN MLB = MLB_loadEqual
   IF (time in [00:00, 07:59] OR
11      time in [18:00, 23:59]) AND
       location = urban
13         THEN CCO = CCO_on
   IF time in [08:00, 17:59] OR
15     location = rural
           THEN CCO = CCO_off
17 IF time in [00:00, 07:59] OR
       time in [18:00, 23:59] OR
19     location = rural
           THEN ESM = ESM_aggressive
21 IF time in [08:00, 17:59] AND
       location = urban
23         THEN ESM = ESM_passive
   IF ((time in [00:00, 07:59] OR
25       time in [18:00, 23:59]) AND
        location = urban) OR
27     (time in [08:00, 17:59] AND
        location = rural)
29         THEN MRO = MRO_maxSensitive
   IF ((time in [00:00, 07:59] OR
31       time in [18:00, 23:59]) AND
        location = rural) OR
33     (time in [08:00, 17:59] AND
        location = urban)
35         THEN MRO = MRO_minSensitive
```

Listing 3. SCV Policy example

the priorities and the SON Function Model is unambiguous. In other words, there is always exactly one possible SCV Set for each SON Function in every context defined.

E. Policy System

Policy systems are, in contrast to the SON Objective Manager, an already known approach for network management [6]. Several implementations are available, e.g., JBoss Drools [9]. The Policy System in this approach (cf. Figure 2) evaluates the SCV Policy, which is stored in the Policy Repository, at run-time and dynamically configures the SON Functions accordingly. Therefore, an external component is required which triggers the execution of the Policy System. For example, a timer can trigger the Policy System at fixed time intervals like every 5 minutes.

The decision, which SCV Policy rules must be applied, is taken by the PDP component. Therefore, the current context is needed which is stored in the Context Database. Using this context, the PDP can evaluate the conditions of the rules in the SCV Policy, i.e., the `IF` parts, and gather the applicable SCV Sets for the SON Functions. Since the SCV Policy is complete and conflict-free, there is exactly one SCV Set for each SON Function. For instance, in the context (`time = 18:00`,`location = urban`) the SCV Policy rules in Line 1, Line 10, Line 17, and Line 24 as depicted in Listing 3 are applicable.

The PEP is responsible for the execution of the `THEN` part of the SCV Policy rules selected by the PDP, i.e., it configures the SON Functions with the respective SCV Sets. So, for the selected SCV Policy rules from the example, the following SCV Sets have to be applied:

```
MLB = MLB_loadUnequal
CCO = CCO_on
ESM = ESM_aggressive
MRO = MRO_maxSensitive
```

For each SCV Set, the PEP determines whether the respective SON Function is already configured accordingly or, otherwise, deploys the SCV Set to the SON Function. Note that `MLB_loadUnequal`, `CCO_on`, `ESM_aggressive` and `MRO_maxSensitive` thereby represent concrete SCVs Sets.

V. Related Work

There is some work related to the dynamics gap identified in the paper. These approaches propose the usage of policies to dynamically configure network elements. Thereby, they derive low-level policies from high-level policies. However, since the high-level policies are an abstract description of the system behavior rather than a definition of the operator's goals, theses approaches do not fill the automation gap.

For instance, the authors of [10] present the possibility to enable several constituencies to describe policies at five levels of abstraction, referred to as Policy Continuum. High-level policies are transformed into low-level policies through a process called policy refinement. In [11], a similar policy-based framework is presented that, however, defines three layers of abstraction on which policies are described. Here, a process called policy translation transforms a policy on one level into a policy on a lower level. Another related approach is presented in [7]. This work actually presents a case study from SON management which configures the SON Functions by defining their algorithm in form of a policy.

In contrast to the dynamics gap, there is only little work related to the automation gap. The authors of [12] present an approach that transforms goals into low-level policies, i.e., actions to take in response to some event, in a two step process. Thereby, goals are a high-level description of the expected system state after some event occurrence. First, a high-level goal is manually elaborated into more detailed sub-goals. Second, a sequence of concrete actions which achieve the goals are inferred through a process called abduction. This latter step requires a detailed semantic description of the actions in form of pre- and postconditions. A similar approach is presented in [13]. In contrast to the previous concept, it defines the semantics of the actions in form of forecast functions which estimate the system state after the execution of some action. Thereby, these functions can be learned. The disadvantage of the two approaches is the need for a formal, detailed action model which requires SON function manufacturers to reveal the details of their SON Functions.

In [3], the authors present an idea which can potentially fill the manual gap in SON management and operation. This

idea describes the refinement of operator policies into SON Function specific policies in order to configure the SON Functions in a way that their behavior is aligned towards a common goal. However, it has never been described how this could be accomplished.

VI. Conclusion

In this paper, an approach has been presented that allows to overcome the manual gap between network operator-defined technical objectives for the operation and management of a Self-Organizing Network (SON)-enabled mobile radio network, and the configuration of SON Functions that aim at optimizing network configuration parameters in order to fulfill these technical objectives. The manual gap comprises, on the one hand, the missing availability of an automated transformation between technical objectives and configurations of SON Functions (automation gap), and, on the other hand, the lacking capabilities to dynamically adapt the SON Functions' configurations to a changing operational and network context (dynamics gap).

Several models have been introduced that allow the normalized and machine-readable description of the information required to operate a SON-enabled mobile radio network. The Objective Model, provided by the network operator, describes the Key Performance Indicator (KPI) targets, priorities associated with the KPI targets, and the conditions under which these KPI targets and priorities shall apply. The SON Function Model, provided by the SON Function manufacturer, describes which KPI targets the SON Function can pursue and the according SON Function Configuration Parameter Value (SCV) Sets to configure the SON Function. The Context Model provides a description of the properties of operational and network status context information.

A SON Objective Manager has been introduced that creates an SCV Policy using the information provided in the three models. This SCV Policy maps a context to an SCV Set per SON Function configuring the SON Function to pursue the technical objectives. In this way, the SON Objective Manager closes the automation gap. Furthermore, a Policy System has been proposed which evaluates the SCV Policy according to a specific context and configures the SON accordingly. This allows to dynamically react on changing network and operational conditions, thus closing the dynamics gap.

The presented approach represents an important step towards automated network operation, by shifting the responsibility of the human network operator from the repetitive configuration and operation of individual SON Functions towards the definition of technical objectives according to which the mobile network shall operate. The approach thereby represents a means for an objective-driven control of a SON-enabled mobile radio network.

In order to achieve the vision of an automated network operation with an objective-driven control, the presented approach needs to be extended. On the one hand, the level of abstraction of the objectives must be raised from a technical level towards a business level focusing on strategic goals. On the other hand, the level of automation needs to be extended by autonomously creating and continuously improving the SON Function Model using machine learning techniques. Furthermore, the exponential growth of the state space is a problem that has to be solved. That means, a method has to be found to significantly reduce the number of regions within the state space. Another important issue within SON is the conflict-free simultaneous execution of SON Functions, known as SON coordination, which requires to be included in the described approach.

Acknowledgement

The presented work was carried out in cooperation with the FP7 SEMAFOUR project [14], which is partially funded by the Commission of the European Union.

References

[1] S. Hämäläinen, H. Sanneck, and C. Sartori, Eds., *LTE Self-Organising Networks (SON): Network Management Automation for Operational Efficiency*. Chichester, UK: John Wiley & Sons, Dec. 2011.

[2] 3GPP, "Telecommunication management; Self-Organizing Networks (SON) Policy Network Resource Model (NRM) Integration Reference Point (IRP); Information Service (IS)," 3rd Generation Partnership Project (3GPP), Technical Specification 32.522 V11.7.0, Sep. 2013.

[3] T. Kürner, M. Amirijoo, I. Balan, H. van den Berg *et al.*, "Final Report on Self-Organisation and its Implications in Wireless Access Networks," Self-optimisation and self-configuration in wireless networks (SOCRATES), Deliverable D5.9, Jan. 2010.

[4] S. Ben Jemaa, C. Frenzel, D. Götz, B. González *et al.*, "Integrated SON Management - Requirements and Basic Concepts," SEMAFOUR, Deliverable D5.1, Dec. 2013.

[5] A. Westerinen, J. Schnizlein, J. Strassner, M. Scherling *et al.*, "Terminology for Policy-Based Management Status," Internet Engineering Task Force, Request for Comments 3198, Nov. 2001.

[6] J. Strassner, *Policy-Based Network Management: Solutions for the Next Generation*. Amsterdam, Netherlands: Morgan Kaufmann, Aug. 2003.

[7] R. Romeikat and B. Bauer, "Automated Refinement of Policies for Network Management," in *17th Asia-Pacific Conference on Communications (APCC 2011)*, Kota Kinabalu, Malaysia, Oct. 2011, pp. 439–444.

[8] 4G Americas, "Developing and Integrating a High Performance HETNET," 4G Americas, Whitepaper, Oct. 2012.

[9] JBoss Community, "Drools Expert." [Online]. Available: http://www.jboss.org/drools/drools-expert

[10] S. van der Meer, A. Davy, S. Davy, R. Carroll *et al.*, "Autonomic Networking: Prototype Implementation of the Policy Continuum," in *1st IEEE International Workshop on Broadband Convergence Networks (BcN 2006)*, Vancouver, Canada, Apr. 2006, pp. 1–10.

[11] A. Galani, N. Koutsouris, K. Tsagkaris, P. Demestichas *et al.*, "A policy based framework for governing Future networks," in *IEEE Globecom Workshops (GC Wkshps 2012)*, Anaheim, CA, USA, Dec. 2012, pp. 802–806.

[12] A. Bandara, E. Lupu, J. Moffett, and A. Russo, "A goal-based approach to policy refinement," in *5th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2004)*, Yorktown Heights, NY, USA, Jun. 2004, pp. 229–239.

[13] N. Samaan and A. Karmouch, "An automated policy-based management framework for differentiated communication systems," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 12, pp. 2236–2247, Dec. 2005.

[14] SEMAFOUR, "The SEMAFOUR Project." [Online]. Available: www.fp7-semafour.eu