Department of Computer Science

Contributions to Self-Organizing Networks and Network Measurement Data Management

Kasper Apajalahti

Aalto University

DOCTORAL
DISSERTATIONS

Aalto University publication series
DOCTORAL DISSERTATIONS 193/2019

Contributions to Self-Organizing Networks and Network Measurement Data Management

Kasper Apajalahti

A doctoral dissertation completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Science, at a public examination held at the lecture hall TU2 of the school on 15 November 2019 at 12:00.

Aalto University
School of Science
Department of Computer Science
Semantic Computing Research Group

Supervising professor
Professor Eero Hyvönen, Aalto University School of Science, Finland

Thesis advisor
Doctor Vilho Räisänen, Nokia Bell Labs, Finland

Preliminary examiners
Professor Bernhard Bauer, University of Augsburg, Germany
Doctor John Keeney, LM Ericsson, Ireland

Aalto University publication series
DOCTORAL DISSERTATIONS 193/2019

© 2019 Kasper Apajalahti

ISBN 978-952-60-8776-4 (printed)
ISBN 978-952-60-8777-1 (pdf)
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)
http://urn.fi/URN:ISBN:978-952-60-8777-1

Unigrafia Oy
Helsinki 2019

Finland

NORDIC SWAN ECOLABEL
Printed matter
4041-0619

Aalto University
Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

Abstract

Author
Kasper Apajalahti

Name of the doctoral dissertation
Contributions to Self-Organizing Networks and Network Measurement Data Management

Publisher School of Science
Unit Department of Computer Science
Series Aalto University publication series DOCTORAL DISSERTATIONS 193/2019
Field of research Computer Science
Manuscript submitted 9 January 2019 **Date of the defence** 15 November 2019
Permission for public defence granted (date) 27 September 2019 **Language** English
☐ **Monograph** ☒ **Article dissertation** ☐ **Essay dissertation**

Abstract

In the future, the mobile network infrastructure needs to facilitate wireless communication to automate industry processes in many vertical domains. The heterogeneity of domains and use cases need to be addressed by various traffic service types that require new technologies. The management architecture of the upcoming 5G networks should cover flexible cross-platform optimization (both technology and administrative domains) and operator business objectives. The new management aspects combined with the increasing complexity of the mobile network infrastructure denote the necessity of the adaptive automation of operability and management.

Along with the 4G, the Self-Organizing Networks (SON) paradigm has been designed and utilized to automate some network management use cases. The challenge in the future network management is the interplay of cross-platform management functionalities in complex 5G networks. Some of the objectives in the network management for operators in deciding the right context-specific solutions are: 1) comparing similar cross-platform SON functions and their configurations, 2) providing linkages of metrics across platforms, 3) providing graphical user interfaces to understand the decisions and actions of autonomic SON functions, and 4) automating the process of modelling SON functions and their metadata.

This thesis is conducted by designing, implementing, and evaluating frameworks, models, and methods, that address the aforementioned challenges. The research follows the principles of the design science methodology. User interface functionalities with faceted browsing activities are designed in order to provide flexible information exploration for the user. The other user interface design offers an interactive SON function discovery mechanism for a prototype SON service system and the other provides an ontology-based visualization of the functionality of an individual SON function. The thesis presents semantic models for reasoning-based SON function discovery and composition mechanism and for defining metric dependencies. Statistical methods are developed for mining time series-based event patterns as context-specific metadata for SON functions and for matching network metrics across heterogeneous datasets with a correlation-based method. All the contributions are reflected against the related work and discussed from the viewpoint of practical benefits for network management. The contributions are novel in view of adapting methods from other research areas to the SON and network measurement data management.

Keywords
ISBN (printed) 978-952-60-8776-4 **ISBN (pdf)** 978-952-60-8777-1
ISSN (printed) 1799-4934 **ISSN (pdf)** 1799-4942
Location of publisher Helsinki **Location of printing** Helsinki **Year** 2019
Pages 169 **urn** http://urn.fi/URN:ISBN:978-952-60-8777-1

Aalto-yliopisto

Aalto-yliopisto, PL 11000, 00076 Aalto www.aalto.fi

Tiivistelmä

Tekijä
Kasper Apajalahti

Väitöskirjan nimi
Kontribuutioita itseorganisoituviin verkkoihin ja verkkojen mittaustietojen hallintaan

Julkaisija Perustieteiden korkeakoulu

Yksikkö Tietotekniikan laitos

Sarja Aalto University publication series DOCTORAL DISSERTATIONS 193/2019

Tutkimusala Tietotekniikka

Käsikirjoituksen pvm 09.01.2019 **Väitöspäivä** 15.11.2019

Väittelyluvan myöntämispäivä 27.09.2019 **Kieli** Englanti

☐ **Monografia** ☒ **Artikkeliväitöskirja** ☐ **Esseeväitöskirja**

Tiivistelmä

Tulevaisuuden mobiiliverkkoinfrastruktuurin tulee mahdollistaa langaton viestintä automatisoiduille teollisuusprosesseille monella eri toimialalla. Toimialojen ja käyttötapausten heterogeenisyyden vuoksi tarvitaan uusilla teknologioilla toteutettavia tietoliikennetyyppejä. 5G-verkkojen hallinta-arkkitehtuurin tulee tarjota joustavuutta sekä monen toimialan ja teknologian väliseen optimointiin että operaattoreiden liiketoiminnan tavoitteiden saavuttamiseen. Nämä uudet haasteet verkonhallinnassa sekä verkkojen monimutkaistuminen osoittavat tarpeen automatisoida verkonhallintatoimintoja.

4G-verkkojen myötä määriteltiin itseorganisoituvat verkot (SON), joita hyödynnetään verkonhallintatoimintojen automatisoinnissa. Monimutkaisemmissa 5G-verkoissa tulevaisuuden haasteena kuitenkin on monien toimialojen ja teknologioiden välisten verkonhallintatoiminnallisuuksien yhteistoiminta. Jotta operaattori pystyisi valitsemaan oikeat kontekstisidonnaiset SON-pohjaiset ratkaisut, ovat haasteina muun muassa: 1) vertailla samankaltaisia SON-toimintoja ja niiden konfiguraatioita sovellusalustojen välillä, 2) linkittää mittarit sovellusalustojen välillä, 3) tarjota käyttöliittymät, joiden avulla voidaan ymmärtää autonomisten SON-toimintojen päätökset ja suoritukset ja 4) automatisoida SON-toimintojen ja niiden metadatan mallinnus.

Tässä työssä on suunniteltu, toteutettu ja arvioitu viitekehyksiä, malleja ja metodeja, jotka käsittelevät edellä mainittuja haasteita. Tutkimuksessa on noudatettu suunnittelutieteen metodologian periaatteita. Työssä on suunniteltu kaksi fasettihakukäyttöliittymään perustuvaa viitekehystä, jotka tarjoavat käyttäjälle joustavaa ja vuorovaikutteista tiedon tutkimista. Toinen viitekehys tarjoaa vuorovaikutteisen SON-toimintojen hakumekanismin osana kokeellista SON-pohjaista hallintajärjestelmää ja toinen puolestaan vuorovaikutteisen ja ontologiapohjaisen visualisoinnin SON-toiminnolle. Työssä on käytetty semanttisia malleja päättelypohjaiseen SON-toimintojen hakumekanismiin ja määrittelemään riippuvuussuhteita verkonhallinnassa käytettävien mittareiden välillä. Tilastollisia menetelmiä on kehitetty SON-toimintojen metadatan määrittelyä varten (toistuvuuksien louhintaa aikasarjoista) ja vastaavien mittareiden löytämiseen heterogeenisten mittausaineistojen väliltä (korrelaatiopohjainen menetelmä). Työssä esitettyjä kontribuutioita on tarkasteltu suhteessa aiempaan tutkimukseen aiheesta ja pohdittu kontribuutioiden käytännön hyötyjä verkonhallinnassa. Työssä on hyödynnetty muilla tutkimusaloilla käytettyjä menetelmiä ja sovellettu näitä uudella tavalla SON-teknologiaan ja siihen liittyvään mittaustietojen hallintaan.

Avainsanat

ISBN (painettu) 978-952-60-8776-4 **ISBN (pdf)** 978-952-60-8777-1

ISSN (painettu) 1799-4934 **ISSN (pdf)** 1799-4942

Julkaisupaikka Helsinki **Painopaikka** Helsinki **Vuosi** 2019

Sivumäärä 169 **urn** http://urn.fi/URN:ISBN:978-952-60-8777-1

Preface

During the past few years, I have learned the principles of science and challenged myself. Indeed, it has been a challenging yet rewarding journey. I owe many thanks to numerous people who have been part of this process and without whom the thesis would not have seen the daylight. I am most grateful for my thesis instructor Vilho Räisänen and supervisor Eero Hyvönen. Thank you for pushing me further, giving me guidance, and helping to keep me on the track. I thank my co-others, with whom I have been fortunate to collaborate with: Vilho Räisänen, Eero Hyvönen, Haitao Tang, Kaj Stenberg, Juha Niiranen, Shubham Kapoor, Ermias Walelgne, and Jukka Manner. Thank you for your contributions and valuable help.

I am very thankful for Nokia Bell Labs that have enabled the thesis in the first place. Thank you for supporting the thesis and creating the conditions for the research. Especially, I thank Vilho Räisänen, Haitao Tang, Kimmo Hätönen, and Kari Aaltonen for your support and collaboration.

I thank The Helsinki Doctoral Education Network in Information and Communications Technology (HICT) for funding my conference trips and HPY Research Foundation for supporting my research. I am also grateful for the people in the Semantic Computing Research Group (SeCo) that I have had opportunity to work with and that have given me some perspective of digital humanity.

Finally, I want to express my deepest and warmest gratitudes for all my friends, family, and Taru for being there.

In Espoo, October 5, 2019,

Kasper Apajalahti

Contents

Preface **1**

Contents **3**

List of Publications **7**

Author's Contribution **9**

1. Introduction **11**

1.1 Motivation . . . 11

1.2 Self-Organizing Networks (SON) . . . 12

1.3 Conceptual View, Objectives, and Research Questions . . . 13

1.3.1 Conceptual View . . . 13

1.3.2 Objectives . . . 15

1.3.3 Research Questions . . . 17

1.4 Research Process and Dissertation Structure . . . 18

2. Theoretical Foundation **21**

2.1 Autonomic Computing . . . 21

2.1.1 Overview . . . 21

2.1.2 Service-Oriented Computing . . . 22

2.1.3 Network Management . . . 23

2.2 Characterizing SON Functions by Analyzing and Evaluating Their Performance . . . 25

2.3 Semantic Modelling and Reasoning in Network- and Measurement Data Management . . . 27

2.4 GUIs for Providing Better Understanding of Autonomic Network and Service Management Functionalities . . . 29

2.5 Measurement-Based Statistical Methods in Network Management . . . 31

2.5.1 Pattern Mining in Network Management . . . 31

2.5.2 Mapping Metrics between Measurement Datasets and Platforms 32

3. Results 35

3.1 A Semantic Model and Reasoning Method for SON Function Discovery and Composition 37
3.1.1 Overview 37
3.1.2 Evaluation 38
3.2 A Semantic Model and Reasoning Method for Inferring Dependencies about and between LTE Metrics 41
3.2.1 Overview 41
3.2.2 Evaluation 43
3.3 A Framework and GUI Design for Interactive SON Function Discovery 45
3.3.1 Functionalities for the GUI 46
3.3.2 Demonstration 47
3.4 An Ontology-Based Framework and GUI Design for User Interaction and for Detailed Exploration of a Specific SON Function 49
3.4.1 MLN Model 50
3.4.2 Functionalities for the GUI 50
3.4.3 Semantic Model for the GUI 51
3.4.4 Demonstration 52
3.5 A Statistical Method Using Time Series-Based Event Pattern Mining for Creating Context-Specific Metadata about SON Functions 55
3.5.1 Overview 55
3.5.2 Event Detection with CuSum Algorithm 56
3.5.3 Temporal Pattern Mining with Association Rule Learning Method 56
3.5.4 Evaluation 57
3.6 A Correlation-Based Method for Creating Linkages between Measurement Metrics across Different but Related Platforms/-Datasets 59
3.6.1 Overview 59
3.6.2 Evaluation 61
3.7 Summary of the Contributions 64

4. Discussion 67

4.1 Theoretical Implications 67
4.1.1 Characterizing SON Functions by Analyzing and Evaluating Their Performance 67
4.1.2 Semantic Modelling and Reasoning in Network- and Measurement Data Management 68

4.1.3 GUIs for Providing Better Understanding of Autonomic Network and Service Management Functionalities 69
4.1.4 Measurement-Based Statistical Methods in Network Management 70
4.1.5 Summary 71
4.2 Practical Implications 72
4.2.1 SON Function Management Across Networks and Platforms 72
4.2.2 Semantic Modelling of SON Functions and Network Context 72
4.2.3 Interactive GUIs for Understanding and Discovering SON Functions 73
4.2.4 Characterizing SON Functions with Measurement-Based Patterns 73
4.2.5 Sharing Operational Experiences Across Domains 74
4.3 Reliability and Validity 74
4.3.1 Reliability 74
4.3.2 Internal Validity 74
4.3.3 External Validity 76
4.4 Recommendations for Future Work 77

References **79**

Publications **91**

List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.

I Vilho Räisänen and Kasper Apajalahti. Reasoning in Agent-Based Network Management. In *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, IEEE, Taipei, Taiwan, pages 1–7, DOI https://doi.org/10.1109/NOMS.2018.8406171, April 2018.

II Kasper Apajalahti, Juha Niiranen, Shubham Kapoor, and Vilho Räisänen. Sharing Performance Measurement Events Across Domains. In *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, IEEE, Lisbon, Portugal, pages 463–469, DOI https://doi.org/10.23919/INM.2017.7987313, May 2017.

III Haitao Tang, Kaj Stenberg, Kasper Apajalahti, Juha Niiranen, and Vilho Räisänen. Automatic Definition and Application of Similarity Measures for Self-Operation of Network. In *Agüero Ramón, Zaki Yasir, Wenning Bernd-Ludwig, Förster Anna, Timm-Giel Andreas (editors) Mobile Networks and Management. MONAMI 2016. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, Springer, Cham, Abu Dhabi, volume 191, pages 206–219, DOI https://doi.org/10.1007/978-3-319-52712-3_15, October 2016.

IV Kasper Apajalahti, Eero Hyvönen, Juha Niiranen, and Vilho Räisänen. Combining Ontological Modelling and Probabilistic Reasoning for Network Management. *Journal of Ambient Intelligence and Smart Environments*, IOS Press, volume 9, number 1, pages 63–76, DOI https://doi.org/10.3233/AIS-160419, January 2017.

V Kasper Apajalahti. Creating Time Series-Based Metadata for Semantic IoT Web Services. In *Database and Expert Systems Applications. DEXA 2018.*, Springer, Cham, Regensburg, Germany, pages 417–427 DOI https://doi.org/10.1007/978-3-319-98812-2_38, September 2018.

VI Kasper Apajalahti, Ermias Walelgne, Jukka Manner, and Eero Hyvönen. Correlation-Based Feature Mapping of Crowdsourced LTE Data. In *2018 IEEE 29th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, IEEE, Bologna, Italy, September 2018.

Author's Contribution

Publication I: "Reasoning in Agent-Based Network Management"

The author designed and implemented the ontology-based SON function service discovery mechanism that maps requests to SON function operations. The author created and simulated the test cases for evaluating the request-operation mapping. The author also designed and implemented the majority of the ontology model for storing performance and configuration data and queries for accessing the data.

Publication II: "Sharing Performance Measurement Events Across Domains"

The author is the first author of the publication. The author designed and implemented the ontology and the integration of adding statistical data into the ontology.

Publication III: "Automatic Definition and Application of Similarity Measures for Self-Operation of Network"

The author was one of the primary developers of the system and wrote the user interface-related experiments. The author partly implemented and designed the dashboard user interface and REST services for data management purposes. The author also implemented and designed some of the REST interactions between the dashboard and backend services.

Publication IV: "Combining Ontological Modelling and Probabilistic Reasoning for Network Management"

The author is the first author of the publication. The author designed and implemented the ontology of the system. The author designed and implemented the ontology-based user interface in order to to visualize, monitor, and control the system behaviour.

Publication V: "Creating Time Series-Based Metadata for Semantic IoT Web Services"

The author wrote the publication and designed, implemented, and evaluated the process of creating measurement-based metadata from SON functions operating in the network (simulated).

Publication VI: "Correlation-Based Feature Mapping of Crowdsourced LTE Data"

The author is the first author of the publication. The author designed, implemented, and evaluated the feature mapping method (matching metrics across datasets). The author analysed and adapted the datasets to the system.

1. Introduction

1.1 Motivation

Mobile networks are part of a critical infrastructure, facilitating wireless access to Internet with all its services. Recent forecasts predict drastic growth in mobile network traffic and amount of devices [30, 101], which is due to technologies in Internet of Things (IoT), Machine-to-Machine (M2M) communications, cloud computing, and network virtualization. Especially, mobile network infrastructure needs to facilitate wireless communication in order to automate industries and industry processes in many vertical domains, such as automotive, manufacturing, energy, healthcare, and entertainment [101, 1]. Thereby, the heterogeneity of IoT needs to be accommodated by various traffic service types [1]. These require new technologies which in most cases need to co-exist with legacy access technologies.

The fifth generation of mobile technology (5G) is positioned to address the changing environment in the mobile networks with new features [101]. For example, network slicing will provide flexible network services to customers with respect to customized requirements for example in bandwidth or latency [1] and Network Service Orchestration (NSO) will provide automatic selection and control of virtualized network resources and services [118]. Taken together with the increasing complexity of radio access, flexibility is needed for network management since future needs cannot be fully predicted. The role of network management needs to be re-assessed in the new architecture, and it can be concluded that 5G management and orchestration should cover flexible cross-domain optimization (such as technology and administrative domains) and operator business objectives [1]. The new 5G features with previously noted drivers imply the necessity of the adaptive automation of operability and management for future networks.

1.2 Self-Organizing Networks (SON)

The Next Generation Mobile Networks (NGMN) and 3rd Generation Partnership Project (3GPP) have already provided an automation framework for the Long Term Evolution (LTE) network management in the form of Self-Organizing Networks (SON) [99, 61]. The SON paradigm consists of SON functions, that are closed-loop automated agents designed for particular network management use cases categorized as self-configuration, -optimization, and -healing. For example, common self-optimization functions are:

- **Capacity and Coverage Optimization (CCO)** to detect coverage and capacity problems and to reconfigure the transmission powers and/or antenna tilts of cells
- **Mobility Load Balancing (MLB)** to balance traffic between heavily loaded cells to neighbours by reconfiguring the cell-specific handover parameters
- **Mobility Robustness Optimization (MRO)** to improve handover performance by optimizing the handover parameters
- **Energy Saving Management (ESM)** to save energy by turning off unnecessary cells with respect to current capacity and coverage situation

In a typical realization of SON functions, fixed rule bases are used for defining the behaviour of SON functions. The definition and governance of rule bases are expensive [61] which puts a price tag on optimizing closed-loop automation on per-cell level. Complexity arises from the contextual diversity (such as spatial, temporal, and technological diversity) on a cell level of radio access networks. For example, some challenges arise from the fact that in future networks end-users may have their own femtocells (small and low-power cells) deployed which in turn requires better automated coordination and self-optimization from the network management layer [9]. Another challenge is that traditional SON functions are reactively detecting and recovering from network faults instead of proactively prevent faults occurring [11, 159]. In addition to the limitations associated with current SON management use cases, more automation is expected to be needed in many network management tasks in emerging 5G scenarios [101, 3].

SON is also seen as a fundamental part of 5G networks. On the one hand, the ongoing work in improving machine learning capabilities of SON control loops for both the existing and upcoming use cases will have a great emphasis also in the future research. [95, 1, 159] On the other hand, the interplay and interaction of management functionalities in a multi-domain environment is also seen as a key component in the virtualized and service-oriented 5G network management [111, 3, 1, 122].

From the perspective of utilizing SON-related machine learning models effectively, some fundamental issues need to be addressed. First, the increasing number of machine learning models for SON use cases implies a crucial need to define how to select the suitable algorithms with respect to different objectives [159]. Second, the cross-domain (e.g., across different technologies and industries) adaptation of machine learning models needs to be addressed in the future SON research [95].

In summary, both from the machine learning view and architectural view of service-oriented SON management, lots of technological solutions are yet undefined for the service-oriented SON management. This thesis covers some detailed aspects in the utilization of cross-platform (across network management systems) SON functionalities.

1.3 Conceptual View, Objectives, and Research Questions

1.3.1 Conceptual View

The overall goal of this thesis is to provide novel contributions for SON-based network management that facilitate the automatic SON function selection and usage in order to achieve operator objectives in context-specific problems. Particularly, this thesis aims to enable cross-platform utilization of autonomic SON functions so that appropriate solutions from different platforms (such as different local networks of the same operator) could be shared and deployed. For this purpose, the overall goal is addressed by adapting the principles and practices of service-oriented computing to the SON-based management. In service-oriented computing, services are defined as self-describing, platform-agnostic computational elements [104]. Services perform functions from simple requests to complicated business processes. With this general definition of a service, SON functions can be seen as services performing some network management activities. In general, SON management should consider information exchange in a heterogeneous environment with multiple management platforms. In a high-level, the interoperability in a such environment can be achieved with a unified representation of the data elements and with a service-oriented modelling of the SON functions.

The conceptual view of the service sharing mechanism is shown in Figure 1.1 depicting how the SON solutions of already running heterogeneous network management platforms could be utilized across these platforms. The figure depicts the discovery mechanism of services leaving aside the deployments and executions of the found cross-domain SON functions. It illustrates the overall vision of the service-oriented SON management which forms the framework for the objectives, research questions, and contributions of this thesis.

The figure combines two simplified architectures: 1) local networks that are managed with SON functions and 2) the cornerstone model of the web service architecture allowing finding and publishing service information in a global service registry [20]. The figure depicts that from one network (requester), one can query and find SON-related information, such as SON algorithms and their configurations, that are published by another network (provider). In practice, all networks would act as requesters and providers, depending on the situation.

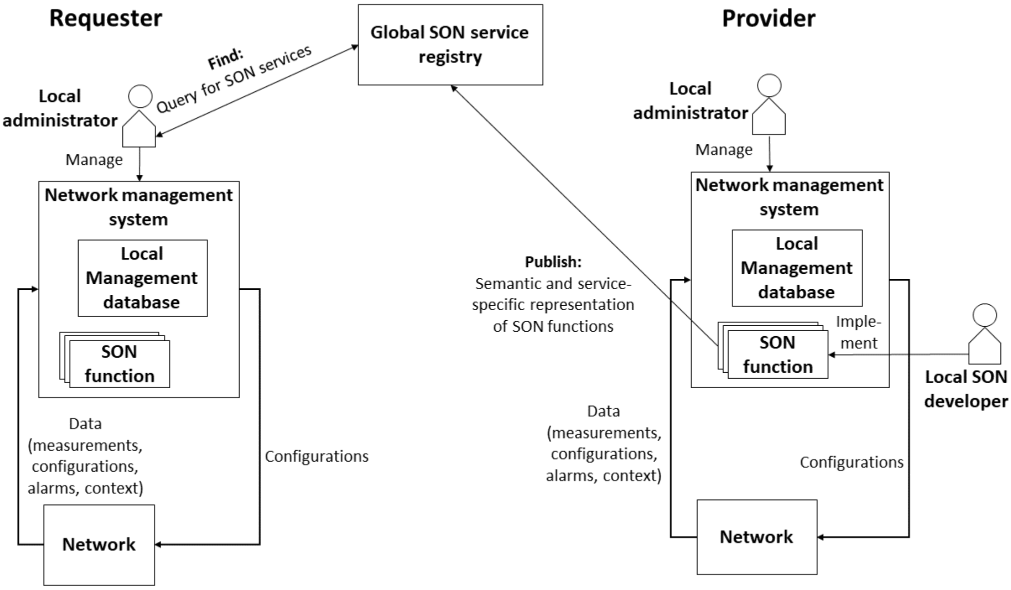


Figure 1.1. A conceptual architecture illustrating the integration of local SON management into a global SON service system.

Service-Oriented Modelling in SON Management

The usage of service-oriented modelling in cross-domain SON management has many benefits. Generally, service-oriented architecture has some fundamental characteristics [147, 104] that are analogically important in the cross-domain SON management:

- **Modularity:** Flexible reuse of existing services (SON functions) and alignment of new functionalities among with existing ones.
- **Loose coupling:** Services (SON functions) can be treated as black boxes with only a few dependencies to the system. This enables flexible reuse and replacement of services (SON functions).
- **Location transparency:** Services (SON functions) are discoverable regardless of their location.
- **Composability:** Complex requests may be solved by running services in parallel or in a sequence (SON function coordination).

- **Self-healing capability:** Recovering autonomously from service errors (such as SON function conflicts).

Semantic Modelling and Reasoning in the Service System
The information exchange between local networks would benefit from the semantic modelling of network context and SON functions. With shared semantics, the service system would make SON functions queryable across heterogeneous local systems. Semantic representations of network context and SON functions could be stored as an ontology that is a document especially for 1) defining classes of objects and relations among them (e.g. taxonomy) and 2) rules for reasoning of the defined data [18]. For example, a reasoning rule in a network context- and SON function-specific ontology could be that "heavy rain causes degradation to the signal quality". Assuming that semantics of terms "heavy rain", "interference", and "signal quality" are defined, it could be deduced that SON functions are unlikely capable of reacting to the root cause and enhance the network performance during the temporary weather condition.

1.3.2 Objectives

The presented conceptual view of a service-oriented SON system works as a basis for this thesis. The contributions in this thesis aim to reduce barriers to build such a complex system by studying some important aspects that are already recognized in the earlier research related to the development and management of a cross-platform service system. The objectives of the thesis can be divided in to three categories, regarding the following perspectives: 1) semantic interoperability between the SON systems, 2) comprehensive Graphical User Interfaces (GUIs) for SON Administrators, and 3) statistical methods for facilitating the modelling of the SON systems. The SON functions and datasets analyzed in this thesis are LTE-related and therefore, also the objectives concern LTE networks.

Semantic Interoperability Between the SON Systems
With respect to the cross-platform interoperability of SON-based management systems, one objective of this thesis is to create a discovery mechanism to find the most suitable SON functions for specific operator objectives in context-specific problems (**OBJ1.1**). This would require that the SON function-specific context metadata, such as objectives, thresholds, and targeted parameters, are modelled and understood across platforms. Interoperability of multiple management systems and their elements is recognized as a key research challenge in IoT research [12, 35, 122] as well as in the network management [108, 70]. For example, in the virtualization of wireless networks, the semantic abstraction of network resources and applications would provide interoperability across heterogeneous local networks [163]. In view of the cross-platform discovery of SON functions, network management platforms and systems should be semantically

interconnected at the data level. This is crucial in order to discover suitable SON functions possibly from different sources by defining the SON functions and problem context semantically.

Another objective of the thesis is to provide linkages of LTE network measurement data across platforms in order to discover the similarities of the metrics (**OBJ1.2**). Useful linkages could be defined semantically (modelled by an expert), statistically (such as correlation analysis), or both. The heterogeneity of measurement and observation data in the management systems is an issue that needs semantic linking between network monitoring and measurement systems [80, 6]. A concrete example of the need to understand the measurement data semantics relates to a project for Mapping of Broadband Services in Europe [139]. The main challenge in the project has been to present the variety of data in one mapping application. Currently, the application shows separate datasets by country level. Similar challenges can be seen in the cross-platform SON function management that also utilizes measurement data from various data sources. Particularly, SON functions monitor and analyze measurement data in order run control actions (e.g. increase the transmission power of an antenna).

GUIs for SON Administrators to Provide Better Understanding of Autonomic SON Functions

This thesis has objectives to design GUI functionalities that facilitate: **OBJ2.1)** browsing, discovering, and comparing multiple SON functions in context-specific situations and **OBJ2.2)** understanding the behaviour and characteristics of an individual SON function. A motivation for addressing these objectives is that new autonomic functionalities brought to network management systems imply new use cases and interfaces for human-computer interaction. The challenges and objectives that autonomic computing and -network management brings to the operators (administrators) have been discussed in several studies [120, 123, 98, 141, 142]. In these researches, two common aspects are recognized in the human-computer interaction of autonomic systems: 1) operators should be able to define high-level goals that can be translated into low-level technical policies in the system and 2) operators should understand the decisions and actions of autonomic agents.

Statistical Methods to Facilitate and Assist the Modelling of SON Systems

Two objectives of this thesis are related to implementing and adapting existing statistical methods in order to **OBJ3.1)** create metadata for SON functions and **OBJ3.2)** produce linkages between LTE metrics. These objectives are derived from the issue that new high level autonomic systems require lots of manual work in the development and deployment phase. The development of a service-oriented system that combines data from multiple management systems is cost-intensive, especially due to the heterogeneity of the data and applications [92, 130, 125, 58, 15]. Web services share the same challenge and, for example, the creation and maintenance of metadata, which is crucial for discovering services, is cost-intensive [164, 117, 153, 62]. The potential development costs also lead to a cold-start problem among the system modellers: a service-oriented system cannot recommend suitable SON functions before the developers of different networks have used their resources to manually model them. Thus, there is a need to lower the threshold of participating in such a complex system.

1.3.3 Research Questions

With respect to the objectives of this thesis, following research questions are derived from them:

- **RQ1.1) How does semantic reasoning facilitate the mapping of context-specific objectives to SON functions in a SON function discovery mechanism?** For this research question, it is assumed that SON function actions are monitored with measurement-, network parameter- and context-specific metadata. Moreover, the SON function- and metadata-specific data should be shareable across platforms in order to allow the semantic reasoning method to infer mappings. The research question is related to the objective of creating a discovery mechanism to find the most suitable SON functions for specific operator objectives in context-specific problems (**OBJ1.1**)

- **RQ1.2) How does semantic reasoning facilitate the cross-platform inference of statistical and human-defined dependencies between network metrics?** An assumption of RQ1.2 is that measurement data from different network platforms are available and shareable, as the semantic reasoning could be used for generating new semantic linkages of metrics across platforms (**OBJ1.2**).

- **RQ2.1) What network- and SON-related GUI functionalities should be provided when using case-based reasoning for discovering suitable SON functions and configurations in specific problem contexts?** In addition to RQ1.1, also this research question assumes that SON function actions are recorded in the management system with proper metadata, such as measurements, network parameters, and problem context metadata. Moreover, it is assumed that SON-based management systems may have multiple similar SON functions (with multiple sets of configurations) in the system from which to select the most suitable one for a particular situation. The research question addresses the **OBJ2.1**.

- **RQ2.2) What network- and SON-related information models and GUI functionalities should be used in order to provide SON administrators a metadata-based GUI as a tool for understanding the behaviour and characteristics of a single SON function?** This research question focuses on a customized capacity and coverage optimizing SON function model and addresses **OBJ2.2**. The custom SON function is based on an Markov Logic Networks (MLN) model [114] that uses probabilistic reasoning.

- **RQ3.1) How statistical methods can be used to facilitate and assist the creation of LTE measurement-based and context-specific metadata for SON functions?** The research question assumes that the performance of the target cells can be monitored by analyzing time series of the metrics before and after a SON function has been operated in the network. Moreover, context metadata should be available in order to distinguish problem contexts from each other. The research question focuses on the objective of automating the process of modelling SON functions by implementing and utilizing existing statistical methods for producing information about the dependencies between LTE metrics (**OBJ3.1**).
- **RQ3.2) How statistical methods can be used to facilitate and assist the mappings between LTE metrics across platforms?** An assumption is that measurement data from different LTE platforms/datasets are available and shareable. The research question addresses the objective of providing statistical linkages of LTE network measurement data across platforms in order to discover the similarities of the metrics (**OBJ3.2**).

Table 1.1 summarizes the relations between defined objectives and research questions.

The objectives are addressed and research questions answered with the contributions presented in Publications I–VI. Table 1.2 shows which research questions the individual publications contribute to. The contributions of the publications are presented in Chapter 3.

1.4 Research Process and Dissertation Structure

In this work, the principles of design science research [89, 63, 106, 105] have been applied to ensure a systematic research process. The design science is a technology-oriented problem-solving paradigm that has the target of creating innovative artifacts that serve human purposes [89] and of producing solutions for unsolved problems [63]. The artifacts can be either constructs (concept vocabularies, assertions, or syntaxes), models (abstract representations using formal notations or languages), frameworks (conceptual guides and structures), methods (algorithms and/or practices), or instantiations (implementations) [63, 105]. The design science research process contains the following steps: 1) motivating and identifying the problem, 2) defining objectives for the solution, 3) designing and developing the artifact, 4) demonstrating the artifact, 5) evaluating the artifact, and 6) communicating the results to the audience [106].

Table 1.1. The relationships between the objectives and research questions.

Objective	Research question
Semantic interoperability between SON systems:	
OBJ1.1: Create a discovery mechanism to find the most suitable SON functions for specific operator objectives in context-specific problems.	**RQ1.1**
OBJ1.2: Provide linkages of LTE network measurement data across platforms in order to discover the similarities of the metrics.	**RQ1.2**
GUIs for SON administrators to provide better understanding of autonomic SON functions:	
OBJ2.1: Design GUI functionalities that facilitate browsing, discovering, and comparing multiple SON functions in context-specific situations.	**RQ2.1**
OBJ2.2: Design GUI functionalities that facilitate understanding the behaviours and characteristics of an individual SON function.	**RQ2.2**
Statistical methods for facilitating and assisting the modelling of SON systems:	
OBJ3.1: Implement and adapt existing statistical methods for creating metadata for SON functions.	**RQ3.1**
OBJ3.2: Implement and adapt existing statistical methods for producing linkages between LTE metrics.	**RQ3.2**

The research contributions and artifacts of this thesis are produced by: 1) identifying the problems from the related research literature and technical reports, 2) deriving objectives and research questions from the problems, 3) designing and implementing artifacts to answer the research questions, 4) demonstrating and 5) evaluating the artifacts, and 6) communicating the solutions to the research community in peer-reviewed journals and conferences. The artifacts developed and presented in this thesis are either models (semantic models), frameworks (conceptual guides and structures for GUIs), or methods (algorithms, processes, and sets of reasoning rules). This thesis discusses the designed artifacts systematically by reflecting their significance, validity, and reliability in the research and development communities relevant to future mobile network management.

This thesis is organized as follows. The background and theory of the work are presented in Chapter 2. The results of the thesis are summarized in Chapter 3. Finally, the implications of the results, the validity of the work, and further research are discussed in Chapter 4.

Table 1.2. The relationships between the research questions and publications.

Research question	Publication
RQ1.1	**PI**
RQ1.2	**PII**
RQ2.1	**PIII**
RQ2.2	**PIV**
RQ3.1	**PV**
RQ3.2	**PVI**

2. Theoretical Foundation

This section elaborates the theoretical foundation for this thesis. First, the background for SON management is introduced in terms of the autonomic computing paradigm and how it is related both to the autonomic network management and service-oriented management. The autonomic computing paradigm has had impact in the evolution of SON and autonomic network management. At the same time, service-oriented computing has been seen as an appropriate paradigm to address particular aspects in autonomic computing [75]. Consequently, selected topics clarify the background of SON management and service-oriented computing in terms of the overall goal of this thesis: contributions to the multi-domain and cross-platform utilization of autonomic SON functions by means of the principles of service-oriented computing. After discussing the theoretical background of the overall goal, previous efforts related to the contributions and research questions of this thesis are presented. The related works are discussed from two view points: 1) possible earlier solutions in the SON management and 2) solutions in the nearby research domains, such as sensor networks and service-oriented computing, that match more accurately to the methodology used in this thesis.

2.1 Autonomic Computing

2.1.1 Overview

Horn [65] has stated that one of the most important challenges in Information Technology (IT) industry is the problem of the increasing complexity in IT systems. He conceptualized the autonomic computing as a solution; computing systems should have autonomic capabilities in order to make systems dynamically adaptable and flexible. They should allow users to focus on high-level management of the system, such as defining goals that the system should achieve. In later research, the autonomic computing concept is extended and adapted in several research fields including the mobile network management.

Kephart and Chess [75] has addressed the autonomic computing by presenting the ingredients of the concept in more details. They have presented the architecture and process of managing autonomic elements with the Monitor, Analyse, Plan, and Execute with shared Knowledge (MAPE-K) loop. Figure 2.1 depicts the MAPE-K closed-control loop process. The process consists of managed elements that can be controlled with an element manager. The functionality of the element manager is defined with four phases: 1) monitoring the element and its environment, 2) analyzing the monitored data, 3) planning actions in order to achieve with respect to the analysis and high-level goals defined for the manager, and 4) executing the plan for the element. The fifth building block of the manager is knowledge that highlights the need for sharing the data and understanding the full process between the phases. Also, the knowledge-layer contains information about the overall situation in the environment and inter-dependencies to other elements and element managers.

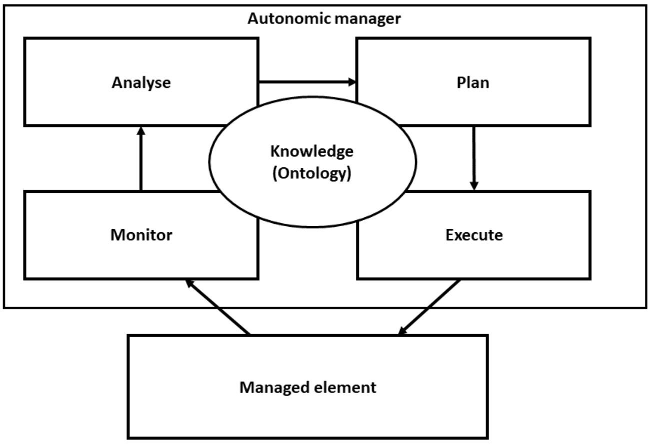


Figure 2.1. Monitor, analyze, plan, and execute concept [75]

2.1.2 Service-Oriented Computing

In this thesis, the concept of service discovery is adapted to the autonomic self-management of mobile networks as depicted in Figure 1.1. Service-oriented computing is seen closely related to the autonomic computing in view of autonomic elements providing services that may be requested and discovered by humans or other machines [75]. One of the ideas behind the autonomic computing is to utilize standard service ontologies in order to represent the capabilities and metadata of the elements with a unified language in order to enable communication between the elements as well as with human administrators [75].

Recent works have addressed the relation between service-oriented applications and autonomic computing. For example, Souza et al. [133] have discussed the dynamic service-oriented computing (service discovery and service level agreement establishment and monitoring) that has adapted architectural models from the autonomic computing. Also, Vargas-Santiago et al. [151] have proposed a checkpointing mechanism for web services that are implemented as MAPE-K-control loops. In the field of cloud service management, Druagan et al. [43] have discussed the benefits of self-management principles and of the MAPE-K loop in order to autonomously manage service-oriented heterogeneous cloud environment [43]. Also, web service implementation has been used together with autonomic computing in order to provide a flexible self-adaptation of robotic arm functionalities in a human-machine collaboration [113]. These studies show the relevance of using web services in the autonomic computing and in self-management environments.

2.1.3 Network Management

The autonomic network management is a general concept used in the network management community to reduce the complexity of managing network infrastructures [71, 124, 94]. The autonomic network management adapts and maps the self-management principles of autonomic computing to the network management domain. Moreover, the MAPE-K loop is utilized as a control loop for an autonomic management process. For example, management of sensor networks can be facilitated by applying algorithms for analyzing and planning phases while network actuators execute actions to the network and sensors producing the measurement data [124, 94]. The autonomic management presented in the 5G architectural view [1] is also relying on the MAPE-K control loop which is integrated with technologies such as Network Functions Virtualization (NFV) [44], Software-Defined Networking (SDN) [52], Policy-Based Network Management (PBNM) [134], and Lambda Architecture [90] (a framework for designing big data applications).

Especially in the mobile network management, SON functions are seen as autonomic managers and network cells as the managed elements [94, 143, 53]. In SON management, three categories of self-management principles are considered [61]: 1) self-configuration of new cells as part of the network, 2) self-optimization of cell parameters with respect to objectives, and 3) self-healing for detecting, diagnosing, and recovering from failure situations.

In the past, developing the SON technology further has gained lots of interest. For example, several multiyear research projects (such as the SELF-NET [127], CogNet [32], SEMAFOUR [128], COMMUNE [33], Univerself [146], SOCRATES [131]) have addressed the autonomic and cognitive mobile network management by bringing together stakeholders from industry and academia and by designing management frameworks for SON.

Some individual papers have focused on the autonomic network management from different view points. Zhao et al. [161] have researched the autonomic computing in the SDN/NFV-based-networks (the virtualization of the mobile network management). They have recognized some challenges that need attention in the future research of autonomic network management: 1) machine learning algorithms and their combinations should be examined for specific optimization scenarios and 2) decentralized and interoperable Operations Support System (OSS) (including network management systems) should be evaluated and policy integration among them should be redesigned. Van der Meer et al. [149] have also discussed the management aspects in the upcoming 5G environment and have pointed out the necessity of having autonomic functionalities in the management. They have emphasized the need for PBNM in the autonomic management of 5G networks and mentioned that policy systems should be less domain-specific and support reuse and multi-domain policies which also lend itself to the semantic modelling of policies.

Some studies have especially addressed the cross-domain network management, where multiple stakeholders and environments share and utilize information from different sources in order to achieve automation in the management tasks. Mannweiler et al. [88] have proposed an architecture for cross-domain 5G network management system between cellular and industrial networks. Khan et al. [76] have also presented a network configuration platform for a shared infrastructure in cellular networks. Their solution aggregates multi-domain resources and translates requests between several control planes and NMSs.

Meshkova et al. [91] have proposed an autonomic a self-optimizing resource management system for wireless home networks. They concluded that the system provides good performance in a dynamic cross-domain environment with multiple stakeholders and stated that the system could be adapted to similar wireless networks, such as LTE small cell networks. Their experiments in the agent-based system indicate the importance of the well-defined interface models both between agents and humans. They concluded that more research is needed to understand the benefits and drawbacks of alternative agent implementation in different contexts.

In the field of federated network management, Feeney et al. [50] have proposed a layered federation model depicting the level of dependencies between stakeholders in the federated management. They have mentioned that shared semantics of resources, policies, and services would facilitate interoperability between the networks and provide the reuse of suitable solutions and configurations. Famaey et al. [48] have reviewed the state-of-the-art research in the federated network management and concluded that one of the most important future research directions is to provide better internal semantics for the collaborative network domains. They have also emphasized the importance of having better cross-domain discovery mechanisms which can dynamically handle large amount of resource types and attributes.

Generally in the IoT research, conceptual studies have been made to combine ontological modelling and sensor measurement analysis to provide information representation in a heterogeneous and cognitive IoT networks [156, 116, 135].

Altogether, the autonomic computing in the network management domain has gained interest along with the upcoming 5G technology and IoT paradigm, as they increase the complexity of the networks and their management. Many of the presented conceptual works in the autonomic network management have pointed out the need for a semantic representation of domain-specific management information and studying the context-specific performance of the network agents, such as SON functions. These challenges are relevant also for this thesis.

2.2 Characterizing SON Functions by Analyzing and Evaluating Their Performance

Many of the contributions and research questions of this thesis have the need for characterizing SON functions by analyzing and evaluating their performance. Especially, it is important that SON function actions are monitored and characterized with measurement-, network parameter- and context-specific metadata in order to provide better context-specific discovery of SON functions and their configurations and better understanding of the context-specific behaviours of SON functions.

This thesis addresses the SON function characterization by 1) evaluating SON function performances in various context-specific problems (addressing RQ1.1) and 2) analyzing contexts and measurement patterns of SON function actions (addressing RQ3.1).

Some previous efforts have had the similar effort of addressing the characterization of network configurations and sharing them with other networks. Bosneag et al. [21] have proposed a system for the global information exchange of measurements in LTE environment. They have analyzed cell-specific performances, configurations, and contextual data and combined these as signatures to characterize network situations. The signatures are aggregated in a global signature dataset that can be accessed and queried across domains. They have evaluated the system by learning baseline behaviours of cells with respect to performance data and contextual data, and stated that the approach also helps the operator in discovering good network configurations and that they will focus on this aspect in the future research. Also, Ciocarlie et al. [29] have proposed a model for sharing diagnosis knowledge across cellular networks. They have used topic modeling and MLN [114] to analyze cell-specific performance data and to share the topics of KPI measurements and MLN analysis across other networks.

In the SON environment, the characterization and verification of self-optimization functions usually estimate 1) the characterization of SON function effects or 2) the impact of the joint effects of concurrently operating SON functions.

In view of characterizing SON function effects, Frenzel et al. [55] have presented a model to map operator-defined technical objectives to SON function configurations. They have considered context-specific KPI targets with priority levels in order to map the request to correct configurations. Lohmüller et al. [85] have presented an adaptive SON function management mechanism that analyzes context-specific measurements in order to define the effect of a configuration set on the network. Moreover, Lohmüller et al. [84] have studied whether the unknown configuration sets of SON functions can be estimated reliably.

Some verification studies of SON functions have specifically focused on estimating their impact in order to avoid anomalous behaviour. Tang et al. [137] have used case-based reasoning to verify and estimate the impact of a configuration on the network. Tsvetko et al. [145] have addressed the verification with a process where the network is divided into verification areas (configured cell and its neighborhood) which are then analyzed, and if anomalies occur, some configurations are rollbacked. The authors have also mentioned the potential risk in verification when an external issue affects the verification area by causing an anomaly and interrupting the optimization. Ciocarlie et al. [28] have addressed the verification of SON functions and other configuration changes by detecting and diagnosing anomalies after configurations have been made.

The characterization of joint SON function effects has been studied in order to coordinate them dynamically. Hahn et al. [59] have simulated different combinations of SON functions in the network and have examined their context-specific results by classifying cells with respect to context attributes. Iacoboaiea et al. [68] have proposed a reinforcement learning-based algorithm for SON coordination in order to decide which SON functions may be active simultaneously.

Bandh et al. [14] have presented a policy-based approach to handle SON functions that simultaneously operate in the same area by prioritizing them. They have evaluated their method by coordinating CCO-based functions and monitored capacity- and coverage-related performance measurements. Frenzel et al. [54] have researched coordination between SON functions with an objective-driven approach. Their objective-driven method coordinates the executions of SON functions with respect to the operator objectives, such as maintaining a certain KPI level in a specific context.

In summary, previous works have characterized SON functions and configurations in order to identify and share information about suitable solutions in context-specific network problems. However, none of the earlier contributions specifically 1) characterize SON functions with context-specific and measurement-based metadata or 2) semantically model problem contexts in networks and experiment SON functions in those.

2.3 Semantic Modelling and Reasoning in Network- and Measurement Data Management

This thesis uses semantic reasoning to map context-specific objectives to SON functions (addressing RQ1.1) and to infer dependencies on measurement data (addressing RQ1.2). As illustrated in Figure 1.1, mapping objectives to SON functions may analogically be described by mapping requests to services. This thesis utilizes the analogy between services and SON functions in the semantic modelling.

In SON research, semantic and logical reasoning have been used for self-optimization tasks, for example for the conflict handling of SON functions and for defining the inner logic of a SON function. Van der Meer et al. [148] have addressed the reasoning of SON policies. They propose using semantic reasoning to resolve inconsistencies between policies and business goals. Räisänen and Tang [109] have also used semantic reasoning to detect conflicts among SON functions. They have described network- and SON function-related information, such as function input, output, type, and context metadata, and use ontology rules to identify conflicts, for example, if two functions have overlapping impact areas but have conflicting objectives.

Munoz et al. [96, 97] have focused on analyzing the handover mechanism in SON and use fuzzy logic to adapt configuration parameters to optimize the handover performance. They have introduced rules where changes in the handover configuration parameters are derived from the fuzzy values of the chosen Key Performance Indicator (KPIs). Also, Cardoso et al. [24] have used fuzzy reasoning to infer handover configuration changes by using relevant parameters, KPIs, and device status. Celdran et al. [26] have implemented a semantic-based energy saving functionality for mobile networks. They have used Semantic Web Rule Language (SWRL) rules [67] for energy saving policies and have analyzed measurements and context attributes in order to infer which cells should be turned on or off.

In addition to the SON research, the ontology- and reasoning-based optimization of networks have gained interest more generally in the mobile network management domain. Al-Saadi et al. [7] have proposed a cognitive framework for heterogeneous wireless networks. They have utilized ontologies and semantic reasoning to abstract and separate the network infrastructure from the control system. They have demonstrated the framework with simulations in three separate networks: the wireless mesh network, LTE network, and vehicular ad-hoc network. They have used semantic inference with fuzzy reasoning to populate ontology with instances, such as QoS metrics, and to select the most suitable network for a specific data transmission task.

Semantic reasoning has also been applied to analyze measurements in the OSS. For example, Sotelo Monge et al. [132] have presented a framework that combines metrics discovery, pattern recognition, prediction methods and rule-based reasoning to infer anomalies in the 5G networks. Su et al. [136]

have compared semantic reasoning on IoT edge and cloud environment. Their objective has been to identify events from moving vehicles with respect to the fuzzified sensor observations, such as velocity and direction. Keeney et al. [74] have presented a model of using RDF stream processing to infer event correlations in the network data.

Generally in the sensor network management and IoT research, semantic reasoning has facilitated autonomic computing in various use cases. Lam et al. [79] have addressed the autonomic management and interoperability of heterogeneous IoT environments with MAPE-K model and have used ontologies in the knowledge base of the model. They have proposed using reasoning and inference rules in the planning phase in order to generate an action plan for a problematic situation. For example, they have considered reasoning scenarios where service disruption needs to be handled and where a service needs to be substituted to a similar (with respect to context attributes) but more suitable service (with respect to context-specific performance).

Alaya et al. [8] have pointed out that M2M standards provide interoperability for IoT devices at communication level, but not at the semantic level. They have proposed an ontology-based approach to this issue. They have mentioned using inference rules to enable the discovery and matching mechanism of relevant resources, such as binding devices, actuators, and sensors in heterogeneous M2M environments. Moreover, they have utilized reasoning to enrich the ontology instances with new individuals and relationships in order to characterize applications and services. Rana et al. [112] have used semantic modelling to facilitate policy-based management of home area networks. They have applied inference rules that combine human-defined ontological knowledge and monitored network events in order to infer configurations for the network, such as prioritizing certain users or allocating bandwidth for certain applications. Fallon et al. [47, 46] have presented applications to automatically optimize end-user service delivery in home area and telecommunication networks. The authors have implemented MAPE-K control loop for the optimization task and they use ontologies and semantic reasoning to infer information related to end-user context, provided services, and service operations. For example, they have inferred the priority of a service session with respect to the service and user information. Vergara et al. [40, 39] have used an ontology-based approach to provide autonomy for the service management of home area networks. Their semantic reasoning tasks include inferring relationships between users, devices, and services. For example, the reasoning engine discovers and advertises services to users with respect to the context, such as user profiles and device capabilities.

As a final notion, some relevant general models have been presented for semantic modelling and reasoning of measurements. QUDT ontology [64] contains a general upper ontology to describe quantities, measurement units, dimensions, and data types. Rijgersberg et al. [115] have presented an ontology for the measurement units focusing on Web Ontology Language (OWL) [93] axioms that facilitate their five pre-defined use cases: 1) the representation of observations, 2) the representation of formulas, 3) manual annotation, 4) automatic annotation, and 5) unit conversions. To mention more targeted measurement-centric ontologies, some authors have designed semantic models of business performance indicators and properties of them, such as causality, aggregation, and correlation [42, 41, 107]. All these works have defined inference rules for the metrics, either presented in Prolog [31] or SWRL.

This thesis presents semantic models for metric dependencies in SON management systems and a semantic reasoning-based SON function discovery and composition mechanism. The proposed solutions differ from the earlier work by combining statistical and semantic information about effects (value changes) and combining contextual metadata with it. The context-specific and metric effect-based metadata is also at the core of the novel SON function discovery and composition mechanism presented in this thesis.

2.4 GUIs for Providing Better Understanding of Autonomic Network and Service Management Functionalities

This thesis presents designed GUI functionalities for: 1) browsing, discovering, and comparing suitable SON functions with respect to context-specific objectives (addressing RQ2.1) and 2) monitoring and understanding the behaviour of a single SON function (addressing RQ2.2).

The SELFNET project has proposed a comprehensive architecture to manage 5G network autonomically [127]. On top of the architecture, a GUI component is presented for administrators to interact, understand, and configure the system. It is mentioned that the GUI should enable the administrator to intervene in autonomically performed operations [51]. The GUI is sketched and prototyped to involve several panels to, for example, browse and control a list of services, management of tenants (service customers), network topology visualizations, monitoring system metrics, and problem diagnosis.

Caraguay et al. [23] have implemented and presented a monitoring GUI for the SELFNET framework. They have focused on gathering and storing the information network measurements in different data sources, such as LTE, sensors, SDN environment, and cloud environment. The GUI visualizes the measurement information of heterogeneous objects with respect to their type, such as physical devices, virtual instances, LTE devices, sensors, and flow statistics.

In the earlier SON management research, several studies have implemented experimental GUIs to demonstrate SON functionality. Koutsouris et al. [77] have introduced a coordinator system for autonomic control loops, especially for SON functions. Their GUI shows network problems and actions taken by the SON functions. The GUI also provides the administrator with an interface to be part of the decision making when needed. Bajzik et al. [13] have implemented a demonstrator of an agent-based SON system in mobile backhaul and an experimental dashboard GUI. Their GUI includes features such as system status, anomalies detected, network topology visualization, monitoring of cell-level metrics, and presenting the actions. Moreover, Sharm et al. [129] have presented a GUI to dynamically create 5G network slices with respect to performance and context parameters. Schmelz, Lohmüller et al. [126, 83] have demonstrated their SON function management functionalities with GUIs including management panels for example to define technical objectives (target values for KPIs) that are mapped to the particular configuration sets of SON functions.

In web service research, GUIs have been utilized to interactively explore and compare discovered services. For example, there have been many efforts to implement these GUIs for web map services [157, 57, 34]. Authors of these articles have implemented GUI panels to compare and display suitable web services among various providers with respect to quality metrics, such as response time and reliability score, and context attributes of the services. These implementations provide faceted search capabilities to interactively explore search results. Moreover, users are provided with a detailed plot views in which user can compare service performances from alternative visualizations.

To point out relevant works addressing the visualisation and presentation of single machine learning agents, Yet et al. [160] have presented a GUI for browsing the evidence and output of a Bayesian Networks (BN) implementation. Their framework helps health care professionals to understand the evidence that is used in the decision process of a BN reasoning. For this purpose, they have used an ontology to organize the evidence and a browser GUI to present the semantically structured evidence. Also, Zheng et al. [162] have used BN for medical decision support. They have utilized the ontology to represent the uncertain information relevant to the clinical practice guidelines. The user may input evidence and examine the risk probability of an activity analyzed by the BN agent. Kulesza et al. [78] have built an interactive machine learning [45] system where the text classification predictions of their multinomial Bayes model are explained to users with an interactive GUI. Users may modify the algorithm parameters, for example by adding, removing, or adjusting features. Their results indicate that the GUI increased both the users' understanding about the system behaviour as well as the classification scores in the system.

In summary, previous works have presented GUIs for monitoring and controlling SON functions, but no GUIs for interactive exploration and discovery of SON functions have been presented. However, in other research domains, such as web service research, interactive service discovery GUIs have been proposed. Also, in the machine learning research GUIs have been proposed to provide better understanding of complex machine learning agents to human users. This thesis proposes novel GUIs to interactively discover, monitor, and understand SON functions.

2.5 Measurement-Based Statistical Methods in Network Management

2.5.1 Pattern Mining in Network Management

In this thesis the automation of SON management is improved by creating measurement-based metadata for SON functions with a method that combines event detection with a Cumulative Sum algorithm (CuSum) [102] and association rule learning [60] (addressing RQ3.1).

In the SON-related management, measurement patterns are learned either for analyzing and predicting the causes of actions, classifying symptoms, or detecting and predicting anomalies. Imran et al. [69] presented a conceptual framework to empower SON functionality with big data on the 5G environment. Their model included the automatic learning of context-specific associations between KPIs and network parameters in order to predict system behaviour. For this purpose, they have proposed methods such as association mining, dimensionality analysis, and manifold learning. Lohmüller et al. [84] have clustered similarly behaving network cells with respect to a set of KPIs while specific SON functions has been active. Ciocarlie et al. [29] have presented a topic modelling approach to post-action analysis. Their objective has been to learn and share measurement patterns between local networks in situations where new cells are added to a network in order to prevent anomalous configurations in the future. Jiang et al. [72] have proposed a SON decision-making framework for 5G environment and machine learning algorithms to analyze the measurement patterns of unidentified problems in order to decide proper actions for them. Torres et al. [140] have addressed the SON management by learning measurement patterns with regression forecasting methods to predict cell congestions that need to be targeted with some SON functions. Xu et al. [158] have learned regularities from the mobile traffic data and utilized that information in order to forecast upcoming traffic patterns for individual cells. As a use case of the traffic pattern prediction method, they have mentioned of targeting traffic-specific configuration algorithms, such as load balancing.

Generally in the field of IoT networks, association rule learning [60] has been used together with ontologies in order to learn and utilize patterns from sensor-based measurement data. Bytyçi et al. [22] have proposed a method that combines association rule learning and ontologies to mine patterns from water quality measurement data. They enriched the mining results by first populating the context ontology with sensor data and then using the ontology as an input to the association rule learning. Fan et al. [49] have used association rule learning for sensor-based constructions to find contextual patterns from sensor measurements [49]. The experiments showed that temporal patterns can be identified with respect to time metadata, such as a public holiday, weekday, or weekend.

In summary, the related work of measurement-based pattern mining in SON management focuses on analyzing and predicting the causes of actions, classifying symptoms, and detecting and predicting anomalies. In related research fields, association rule learning has been used for measurement-based sensor data. However, none of the earlier works learn association rules from measurements in order to characterize SON functions.

2.5.2 Mapping Metrics between Measurement Datasets and Platforms

This thesis provides a correlation-based method for linking metrics across crowdsourced LTE measurement platforms (addressing RQ3.2).

The LTE-related research in mapping QoS and other network-level metrics across data sources has been focused on using human expertise in order to map the metrics across datasets. In 2016, a project for Mapping of Broadband Services in Europe [139] was launched and as they mention, the key challenge in this project is to present the variety of data in one mapping application. Such projects would also benefit from a metric matching method that facilitates the data integration from multiple stakeholders. Lipenbergs et al. [82] have analyzed the data representation of the broadband mapping of European operators. Moreover, Li et al. [81] have mapped QoS parameters across LTE network components, such as Evolved Universal Terrestrial Radio Access Network (E-UTRAN), Backhaul transport network, and Evolved Packet Core (EPC) network. Malandrino et al. [86] have merged two crowdsourced LTE measurement datasets to analyze mobile traffic traces. All of the aforementioned papers propose models where the cross-domain mapping of metrics is defined by human.

More generally in the field of wireless and sensor networks research, there have been approaches to map features across data sources with statistical and machine learning methods. These methods have relied either on manually defined mappings or on classified data where labels describe the measurements and feature values. For example, Manco-Vásquez et al. [87] have used a Kernel Canonical Correlation Analysis (KCCA) method for spectrum sensing in the cognitive radio environment. Pan et al. [103] have presented a transfer component analysis method that learns a cross-domain feature space for indoor WiFi localization. Their method addresses a supervised learning task where the feature mapping is trained with respect to labels (locations) in the training set. The mapping of features (metrics) have also been studied in the human activity recognition domain. However, these works define either manually the sensor metadata [150, 27] or activities (classification labels) [154] in order to enable mappings between the features.

Altogether, analyzing mappings between metrics have gained interest in the related research fields, but most of the previous efforts rely either on manually defined mappings or on classified data where labels describe the measurements and metric values. To the best of author's knowledge, no earlier work for mapping

LTE-related metrics automatically exists yet.

3. Results

This section presents the contributions of the thesis. These are summarized in Table 3.1 by introducing them briefly and mapping them to publications and research questions. With respect to the design science methodology, the contributions are models, frameworks, and/or methods. Most of the contributions include SON function experiments performed with an internal LTE simulator made by Nokia. It has been widely used in SON research [54, 145, 144, 84, 137]. The simulator setup used in this thesis comprises 20 LTE base stations with 32 LTE macro cells covering an urban area with a radius of 5 km and with 1000–5000 users depending on the scenario. The simulator creates performance management data reports that contain cell level KPIs. The simulator enables experiments with SON functions as the cell parameters, such as Transmission Power (TXP), antenna tilt, and handover parameters, can be modified in the run-time.

The contributions are explained in more details in the following sections. The contents of the upcoming sections are based on the publications included in this thesis.

Table 3.1. The contributions of the thesis and their relation to publications and research questions.

Contributions (CONTRIB)	**Publication**	**Research question**
CONTRIB1.1) A semantic model and a reasoning method for SON function discovery and composition. The contribution includes modelling of problem contexts and experimenting SON functions in those. The semantic reasoning method infers metadata about relationships about and between SON function effects and operator objectives.	I	RQ1.1
CONTRIB1.2) A semantic model and a reasoning method for inferring dependencies about and between LTE metrics. The dependencies (human-defined and statistical) are used to describe effects (e.g. SON functions) and objectives in the network.	II	RQ1.2
CONTRIB2.1) A framework and GUI design for interactive SON function discovery. The background system provides search capability to match function metadata to a given objective by using case-based reasoning. The designed GUI functionalities assist in browsing, finding, and applying appropriate functions for different objectives in different contexts.	III	RQ2.1
CONTRIB2.2) An ontology-based framework and GUI design for user interaction and for detailed exploration of a specific SON function. The specific SON function is an experimental CCO SON function that uses MLN reasoning for analysing the network status. The designed ontology and GUI functionalities assist in describing, monitoring, and browsing the behaviour of the function.	IV	RQ2.2
CONTRIB3.1) A statistical method using time series-based event pattern mining for creating context-specific metadata about SON functions. The method mines measurement event patterns that characterize SON functions and make them discoverable with respect to metadata.	V	RQ3.1
CONTRIB3.2) A correlation-based method for creating linkages between measurement metrics across different but related platforms/datasets. The method assists in integrating and merging separate LTE measurement datasets and their metrics together.	VI	RQ3.2

3.1 A Semantic Model and Reasoning Method for SON Function Discovery and Composition

CONTRIB1.1 is presented in Publication I. The main findings of the contribution can be summarized as:

1. Presenting a semantic model for describing SON functions (agents), operator-defined objectives/problem contexts (requests), and measurement events (effects).
2. Presenting a semantic reasoning rule (based on the semantic model) discovering and composing SON function operations that satisfy given requests.
3. Evaluating the semantic reasoning-based discovery model and method with illustrative scenarios by analyzing SON function performances in LTE simulator. The output of the evaluation is that the mapping works, but the semantic classification needs to be done carefully in order to avoid (partly) misclassified problem contexts.

3.1.1 Overview

In this contribution, semantic reasoning has been used for discovering and composing SON function operations to complex requests. The semantic model is shown in Figure 3.1. It is used for describing SON functions (agents), operator-defined objectives/problem contexts (requests), and measurement events (effects) in order to map context-specific requests to SON function operations. The arrows depict *"hasElement/elementOf"* relations. The idea is adapted from a simple semantic web service model ontology, WSMO-lite [152]. An agent/SON function (analogous to a service in WSMO-lite) has operations that monitor or change the status of a network cell. Operations have effects that represent the desired impact on the target. Furthermore, operations have metadata, for example, operation area (the part of the network where the SON function is operating) and temporal range.

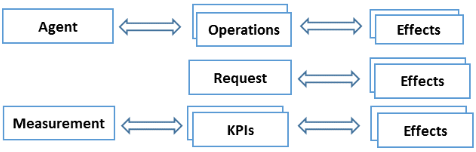


Figure 3.1. Ontology constructs for agent (top), request (middle), and network measurements (bottom).

The ontology is an OWL ontology that uses description logic [66] for knowledge representation and SWRL for representing specific logical reasoning rules in the system. Reasoner maps operations to requested effects with Equation 3.1. The semantic reasoning-based mapping is implemented as a service discovery mechanism: a request is mapped to SON function operations that match to the context and effects of it. The rule indicates that if the effect of an operation ($?oe$) is dependent on the effect of a request ($?re$), the operation satisfies (is able to produce) $?re$.

$$\begin{gathered} Operation(?op) \sqcap Request(?req) \sqcap \\ hasEffect(?op, ?oe) \sqcap hasEffect(?req, ?re) \sqcap \\ hasDependency(?re, ?oe) \Rightarrow satisfies(?op, ?re) \end{gathered} \tag{3.1}$$

The mapping mechanism is defined so that on the one hand, single requested effect might be solved with various competing operations. On the other hand, single operation may solve various requested effects. Hence, the model provides flexibility in modelling and customizing SON functions. For example, one may implement a SON function including both anomaly detection and reaction to it, but another may have separated these functionalities into two distinct algorithms that could be executed in a sequence. The reasoning-based mapping mechanism outputs a response including a list of operation combinations that satisfies the requested context and effects. From the list of results, one combination needs to be selected either by an operator or automatically with respect to the context- and configuration-specific SON function evaluation.

In the selection of the best combination, one needs to define a scoring function which considers the evaluated scores of individual operations, as the performance of two disjoint SON functions (such as a network optimizer or anomaly detector) are evaluated with different criteria. As an example, SON functions may be characterized with 1) the average level of QoS improvement in the network, 2) the precision/recall of detecting anomalies, or 3) success ratio. This question is left for future work.

3.1.2 Evaluation

OBJ1.1 was "Create a discovery mechanism to find the most suitable SON functions for specific operator objectives in context-specific problems." For this purpose, some SON functions need to be experimented how they behave in different problem contexts. The hypothesis is that by recording and analyzing the problem contexts and SON functions operating in those, the SON function discovery method could recommend suitable SON functions and configurations. Thus, three simulation scenarios were created in order to evaluate the suitability of the contribution. Table 3.2 presents the simulated context-specific network problems where users demand higher throughput levels in different circumstances: the coverage problem, local overload (capacity problem), and mobile overload (a group of mobile users causing abrupt load peaks). Each of the network contexts is built with different settings (slightly modified parameter values) in order to examine the generalization of the context. Moreover, three simple SON function operations were deployed and tested multiple times in each scenario: CCO-based function increasing the transmission power (CCO-TXP), CCO-based function performing downtilt operations (CCO-RET), and MLB function balancing the load between cells by modifying handover parameters.

Table 3.2. Simulation scenarios

Scenario	Objective	Default solution
Coverage problem	Increase throughput	Increase signal power (CCO-TXP)
Local overload	Increase throughput	Downtilt (CCO-RET)
Mobile overload	Increase throughput	Balance load (MLB)

The results from two different simulation setups reported in Publication I is shown in Figures 3.2 and 3.3. Figures present the relative changes of the throughput with standard deviation margins (simulations with the same setup are repeated multiple times) before and after SON function actions on every scenario. With respect to the different simulation setups between the figures, SON functions and cell positions are the same, but the problem contexts differ from each other by the number of the users and/or by the size, shape and movement direction of the simulated user groups.

Simulation results from Figure 3.2 are used as learned cases in order to suggest suitable SON functions for upcoming network issues in similar contexts. The simulation results in Figure 3.3 in turn demonstrate new network issues that should be managed with respect to the learned cases.

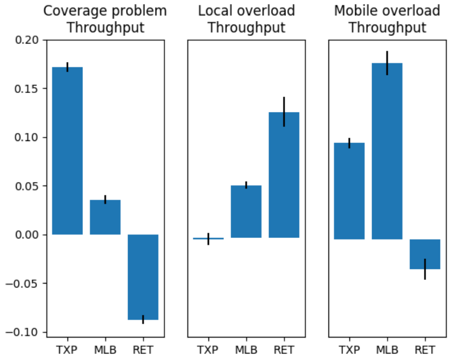


Figure 3.2. Scenario-specific relative changes of the throughput values after actions were made. These results are used for learning how SON functions behave in different problem contexts.

Based on the relative changes in the throughput values, Table 3.3 describes the classification and mapping performance of the SON functions. The learned "case base" data (Figure 3.2) defines a predicted classification of the SON functions and the new simulations the "true" classes. The first row depicts the results when SON functions are classified with thresholds to four groups. For example,

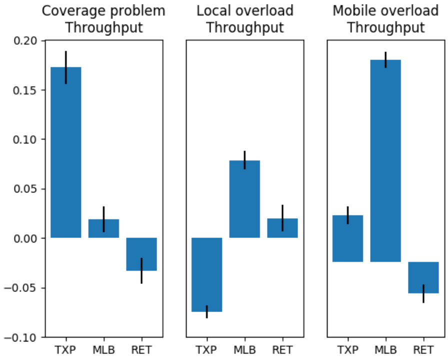


Figure 3.3. Scenario-specific relative changes of the throughput for the new simulation scenarios. These results are used for evaluating the learned SON function experiences based on earlier simulations in Figure 3.2.

the SON function performance is classified as neutral if the relative change of the throughput is between 0 % and +5 %. The second row depicts a mapping task in which +5 % is set as the threshold for mapping results (hits). Last row shows a SON function mapping task when only the best SON functions are considered (those improving the throughput more than +15 %).

As the classification and retrieval metrics indicate, the accuracy of the classification improves when the threshold for "good" performance is higher. The recall values for the mapping tasks indicate high sensitivity in identifying suitable SON functions, as all possible solutions are present in the search results. Precision values show that some false positive cases are also retrieved. With respect to Figures 3.2 and 3.3 the most probable explanation for the false positive hits is the CCO-RET function in the local overload scenario and the CCO-TXP function in the mobile overload scenario. Clearly, the new local overload scenario is not solved with the downtilt, because the user group is located in a wider area around the base station. Finally, the F_1 score, which is the combination of precision and recall, indicates that the overall performance of the classification is good in this demonstrated use case.

Classes	Thresholds	Precision	Recall	Acc.	F_1
{bad,neutral, ok,good}	0 %, +5 %, +15 %	0.74	0.74	0.74	0.74
{reject,hit}	+5 %	0.73	1.0	0.85	0.85
{reject,hit}	+15 %	0.83	1.0	0.96	0.91

Table 3.3. Statistics of the SON function mapping task. With respect to mapping SON functions to problem contexts, the labels "hit", "ok", and "good" refer to suggesting SON function as a suitable solution .

The expected results was that each of the SON functions outperforms the others in different contexts, especially when the problem context is obvious. However, the results also pointed out a situation in which the crisp classification of a problem situation may not give the optimal SON function as a solution. Another simulation setup for local overload problem demonstrated this risk as the downtilt was not the optimal solution when users were more spread out in a wider area from the issued cell when a downtilt operation did not improve the performance. Another important finding was that all scenarios had more than one SON function that improved the QoS (throughput level) in the network, even though the SON function might not have been targeted to the network problem. For example, among with the MLB (the assumed solution), also CCO-TXP improved the QoS during the mobile overload problem. This shows that the service system with a flexible use of different SON functions enables discovering new situations where one can utilize existing functions.

3.2 A Semantic Model and Reasoning Method for Inferring Dependencies about and between LTE Metrics

CONTRIB1.2 is presented in Publication II. The main output of the contribution is:

1. Presenting semantic model and logical axioms for statistical and human-defined dependencies between LTE metric effects (measurement events).
2. Presenting reasoning rules to infer new cross-platform dependencies between LTE metrics.
3. Evaluating and demonstrating the suitability of the model and method with illustrative scenario by analyzing statistical correlations in LTE simulator and test network. The output of the evaluation is that although more experiments are needed, the suitability of the model and inference rules are illustrated with the given scenarios.

3.2.1 Overview

CONTRIB1.2 focuses on defining context- and metric-specific effects and dependencies between the effects. The metric effect modelling complements **CONTRIB1.1** as the effects are used for SON function discovery. Table 3.4 presents the metadata attributes and links that are part of the effect model/ontology:

As this model is described at an abstract and conceptual level, the presented context attributes are assumed to expand over time, when measurement scenarios need more sophisticated definitions. For example, when a statistical data analysis shows that a high positive correlation occurs between some metrics only at subway stations and in the mornings, then these attributes should be added as new spatial and temporal attributes.

Attribute/link type	Example instances/description
Metric	Unified Resource Identifier (URI) as an identifier
Impact direction	Increase or decrease
Context attributes:	
Spatial attribute	E.g. urban, suburban, rural, or highway
Temporal attribute	E.g. night, day, or rush hour
Status of the metric	E.g. low, medium, or high
Network technology	E.g. LTE, Hetnet, UMTS, GSM
Dependency links:	
Statistical	Between metrics (and their effects) having a certain level of correlation coefficient
Strong statistical	Between metrics (and their effects) having a high correlation coefficient
Human-defined	Between metrics (and their effects) mapped by a network expert
Subsumption of	A one-way link from a sub-effect to its parent effect
Transitive	A parent property for strong dependencies, such as human-defined, strong statistical dependency, or subsumption.
Contradiction link	Between effects that are not achievable at the same time

Table 3.4. Attributes and links of the effect model.

Axioms and Reasoning Rules to Infer Dependencies Between Metrics and Effects

In order to combine both human-defined and statistically analyzed metric dependencies, this contribution defines simple equivalence axioms for them. With respect to the transitivity, following links are defined as subproperties of a *hasTransitiveDependency* property: *hasStrongDependency* (statistical), *subsumptionOf*, and *hasLogicalDependency* (human-defined mapping). Moreover, *hasStrongDependency* and *hasLogicalDependency* are defined as symmetric properties, whereas *subsumptionOf* as a non-symmetric property.

Due to these definitions, SWRL rule (Equation 3.2) can be utilized for generating dependencies between effects that are transitively connected with two *hasTransitiveDependency* properties. Using the general superproperty, the reasoner may infer transitive links whether they are statistical, human-defined, or a mix of them. The resulting property is a general *hasDependency*, as the transitivity cannot be guaranteed for it (see Section III.A in Publication II for more details). The notation of the equation describes an SWRL rule in the ontology. The rule indicates that if the ontology contains a transitive dependency between effects $?x$, $?y$, and between $?y$, $?z$, then a dependency link is generated between $?x$ and $?z$.

$$\begin{gathered} hasTransitiveDependency(?x,?y) \sqcap \\ hasTransitiveDependency(?y,?z) \\ \Rightarrow hasDependency(?x,?z) \end{gathered} \tag{3.2}$$

In addition to the dependencies, also contradiction is defined as a symmetric and transitive property in situations where effects cannot occur at the same time. Equation 3.3 shows a rule that infers a contradiction between effects $?x$ and $?y$, because they have the same metric $?metric$, but their impact directions $?impX$ and $?impY$ have different types ($Decrease$ and $Increase$).

$$\begin{gathered} hasMetric(?x,?metric) \sqcap hasMetric(?y,?metric) \sqcap \\ hasImpact(?x,?impX) \sqcap hasImpact(?y,?impY) \sqcap \\ Decrease(?impX) \sqcap Increase(?impY) \\ \Rightarrow hasContradiction(?x,?y) \end{gathered} \tag{3.3}$$

With respect to the SWRL rule above and to *hasStrongDependency* property between effect instances, the semantic reasoner can infer new contradictions with a rule defined in Equation 3.4. The rule defines that if effects $?x$, $?y$ contradict and $?y$, $?z$ have a transitive dependency, then a contradiction link is generated between $?x$ and $?z$.

$$\begin{gathered} hasContradiction(?x,?y) \sqcap \\ hasTransitiveDependency(?y,?z) \\ \Rightarrow hasContradiction(?x,?z) \end{gathered} \tag{3.4}$$

Inferring contradictions is useful for determining which SON functions can operate parallel in the same network and context without having conflicts.

3.2.2 Evaluation

The usage of the semantic model and inference rules were experimented by analyzing Pearson's correlations in two environments: LTE simulator and LTE test network called Netleap, which is functional in Aalto University and University of Helsinki in Finland. In the simulator, the measurement scenario consists of a 2 GHz LTE network with 32 macro cells covering an urban area with a diameter of 5 km and 2000 terminals. The test network is a live LTE network for research purposes. The network operates at 2.6 GHz and comprises 20 LTE base stations with 36 LTE cells. The test network can host up to 200 real and simulated LTE users. As the simulation scenario had high throughput and load levels, also test network data was selected from cells and time periods having at least medium level of throughput and capacity.

The threshold for labelling two metrics to have dependency is set as |0.5| for demonstration purposes. In order to have a transitive strong dependency (used in Equation 3.2) between the metrics, the threshold value of a correlation coefficient is $|\frac{\sqrt{3}}{2}| \approx |0.866|$.

From the simulator, following metrics were analyzed: individual Channel Quality Indicator (CQI) classes, average CQI (CQI_Avg), Radio Link Failure (RLF), terminals per cell (CUEs), and average Reference Signal Received Power (RSRP). Table 3.5 shows correlation coefficients between the metrics. As the table indicates, CQI class 1 (CQI_1) has a strong correlation with RLF and the CQI_Avg has a strong negative correlation with the RSRP.

	CQI_1	CQI_ Avg	RLF	CUEs	RSRP
CQI_1	1	0.19	**0.87**	0.69	-0.44
CQI_Avg	0.19	1	0.07	0.46	**-0.93**
RLF	**0.87**	0.07	1	0.78	-0.27
CUEs	0.69	0.46	0.78	1	-0.58
RSRP	-0.44	**-0.93**	-0.27	-0.58	1

Table 3.5. Pearson's correlations in the simulator.

The following per-cell metrics were analyzed in the test network: the Signal to Inference and Noise Ratio (SINR), Received Signal Strength Indicator (RSSI), Uplink (UL), and Downlink (DL). All the cells had similar behaviour and correlations for given metrics. Correlations are reported in Table 3.6, which presents coefficients between selected metrics. From these metrics, the RSSI correlated strongly with the SINR and the DL with the UL.

	RSSI	SINR	DL	UL
RSSI	1	**0.88**	0.49	0.31
SINR	**0.88**	1	0.66	0.44
DL	0.49	0.66	1	**0.90**
UL	0.31	0.44	**0.90**	1

Table 3.6. Pearson's correlations in the test network.

With respect to the correlations, the semantic reasoner may now infer contradictions between metric effects. The obvious contradictions can be found with respect to the axiomatic rule in Equation 3.3 (opposite impacts of the same metric cannot occur at the same time). In addition to these, contradiction can be inferred with respect to the rule and Equation 3.4 and earlier presented correlation data (Tables 3.5 and 3.6). For example, the RSRP and CQI_Avg cannot increase or decrease at the same time (Table 3.5), and the RSSI cannot increase when SINR decreases and vice versa (Table 3.6).

Finally, the usage of Equation 3.2 and the semantic model is demonstrated with an example utilizing the strong correlations between the RSSI and SINR and between the RSRP and CQI_Avg. By mapping the RSSI and RSRP between the test network and simulator (the mapping is hypothetical as the metrics may not correlate in all contexts), the semantic reasoner may also infer relations between the CQI_Avg and SINR (when the CQI_Avg decreases, SINR increases and vice versa).

The reasoning rules presented in **CONTRIB1.2** complement the SON function discovery method presented in **CONTRIB1.1** as the inference of effect dependencies enables the discovery of SON function operations for a wider range of requests. In the earlier example, an operator might request an increase for the SINR. By means of the effect inference and SON function discovery methods, the request could be mapped to a SON function that has been observed to decrease the CQI_Avg in a similar context.

As a final notion, Publication II also mentions that more experiments are needed in various context-specific scenarios and data sources in order to define proper levels for labelling correlations, examining effects, and to extract relevant sets of context attributes. However, these aspects are considered as future work. Thus, it can be concluded that the suitability of the model and inference rules are illustrated with the given scenarios.

3.3 A Framework and GUI Design for Interactive SON Function Discovery

CONTRIB2.1 is presented in Publication III. The contribution includes:

1. Presenting an operator objective (input for the system) with a goal, location (a set of cells), and time range to allow a context-specific definition of an objective.

2. Presenting search results (SON functions and their configurations) with both SON function-specific metadata (KPI values and network status attributes) and case-based reasoning-specific metadata (the statistics of the search results). The SON function-specific metadata of the search results is constructed as facets and facet values.

3. Designing a faceted search interface and GUI functionalities to interactively select and compare SON function operations with respect to the metadata values of individual cases.

4. Demonstrating the GUI functionalities with a prototype implementation

and experimenting its suitability with an illustrative scenario in the LTE simulator. The output of the evaluation is that the designed framework with the described facet- and metadata-related functionalities is suitable when using case-based reasoning for discovering SON functions and configurations.

3.3.1 Functionalities for the GUI

Publication III presents a case-based reasoning mechanism for learning and storing the past experiences of SON function operations and applying them as historical cases to solve similar problems in the future. The basic idea of the mechanism is that by giving a context-specific objective as input, the system retrieves cases with similar objective from the database. Each case contains a SON function operation (SON function with specific configurations and action) and context metadata for it, such as a set of performance, network, and function attributes.

In addition to the case-based reasoning mechanism itself, an important aspect is that the human operator understands why a certain SON function and its configurations match a certain objective and context. Also, the operator should have the possibility to impact to the search results whenever it is necessary to finetune the context definition. For these reasons, a GUI is designed as part of the publication and presented as a contribution of this thesis (**CONTRIB2.1**). In order to refine the context attributes, the GUI is designed as a faceted search interface as it enables interactive exploration of the search results.

With respect to the GUI design, following phases are proposed where operator/administrator can define input for the system:

Phase 1) Defining context-specific objective and parameters for the case-based reasoner. The case-based reasoner returns matching SON functions and their performance statistics based on the analyzed cases.

Phase 2) Refining the results (matching SON functions based on cases) by specifying the context with attributes defined in the case base.

Phase 3) Selecting a SON function from the search results in order to retrieve its configurations. The case-based reasoner returns matching configurations of the selected SON function with the performance statistics based on the analyzed cases.

Phase 4) Refining the results (matching configurations based on cases) by defining context attribute values as filters.

Phase 5) Selecting the SON function configuration.

In Phase 1, the operator makes the initial request (similarity search) resulting in SON function operations fulfilling the objective. The operator defines a request consisting of an objective (e.g., optimize coverage), spatial metadata (cells included), and temporal metadata (time of a day). Moreover, before invoking the case-based reasoner, the operator may also set threshold parameters as an input to the case-based reasoner: minimum amount of cases that matches a specific SON function configuration and a confidence level (the success ratio of the retrieved cases). The performance statistics of the matching SON functions are: amount of cases, successful cases, confidence score (the success ratio of the cases), and proportion (of all retrieved cases) related to the given objective.

In Phase 2, the operator may investigate and refine the search results with respect to SON function-specific facet values/filters (KPI values and network status attributes). For each case, KPI values are stored before and after SON function operations. Thus, separate facets are constructed for pre- and post-conditions of the SON function operations. Moreover facets are constructed for relevant network parameters (e.g. tilt angle or transmission power) and for network status values (e.g. cell type, antenna type, and antenna elevation).

In Phase 3, the operator may select a SON function in order to retrieve its configurations. The case-based reasoner calculates the performance statistics for the configuration sets of the SON function.

In Phase 4, the operator may again refine the search results by defining context attribute values as facet values/filters. The action updates the performance statistics of the configurations.

In Phase 5, the administrator may select a specific set of configurations for the SON function in order to activate it in the network.

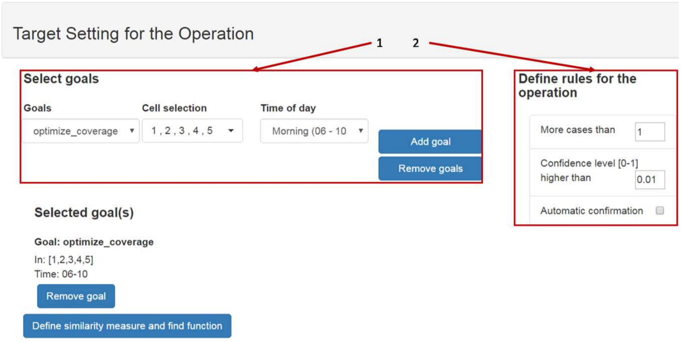
In view of the case-based reasoning system, all interaction phases except Phase 1 are optional as the system may recommend a SON function and its configurations by default with respect to the highest confidence levels.

3.3.2 Demonstration

The system is demonstrated with a prototype implementation of the case-based reasoner and a GUI for it. The interactions between the simulator, reasoning system, and the administrator are implemented with Representational State Transfer (REST) services. The system contains two SON functions, CCO and ESM, which are further implemented as context-specific instances, for example targeted for surrounded and hotspot cells. Every SON function has been also simulated with multiple sets of configurations.

With respect to the GUI design, Figures 3.4 and 3.5 present a full dashboard GUI for making a request (similarity search) resulting in SON function operations (configurations) fulfilling the objective. In Figure 3.4 can be seen how Phase 1 is implemented by providing the administrator GUI functionality to input an objective with the defined context attributes (Wizard 1), and parameters for the case-based reasoner (Wizard 2).

In Figure 3.5 is shown Phases 2, 3, and 4. Wizard 3 presents the matching SON functions in a tabular view with the summary of their performance history



Select goals

Goals: optimize_coverage

Cell selection: 1 , 2 , 3 , 4 , 5

Time of day: Morning (06 - 10

Define rules for the operation

More cases than: 1

Confidence level [0-1] higher than: 0.01

Automatic confirmation ☐

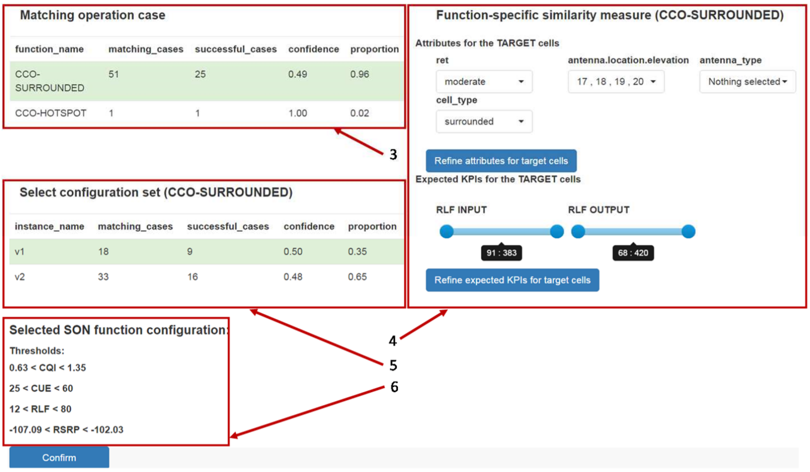
Selected goal(s)

Goal: optimize_coverage
In: [1,2,3,4,5]
Time: 06-10

Figure 3.4. A screenshot of the prototype GUI demonstrating the definition of the objective with a goal, location (set of cells), and time range.

in the network. With respect to Phase 2, Wizard 4 shows the SON function-specific facets, such as antenna type, cell type, RLF input range (the values of the cases before actions), and RLF output range (the values of the cases after actions). RLF is defined as a relevant KPI for the CCO-based SON function, as it aims to reduce the number of RLFs.

When the administrator selects a function in the table (Phase 3), the configurations are loaded and shown in a separate table shown in the Wizard 5. Again, the administrator may refine the SON function-specific facet values in order to investigate the changes in the search results (Phase 4). Finally, a specific configuration set can be selected (Phase 5). The parameter values of the selected configuration set can be seen in the Wizard 6.



Matching operation case

function_name	matching_cases	successful_cases	confidence	proportion
CCO-SURROUNDED	51	25	0.49	0.96
CCO-HOTSPOT	1	1	1.00	0.02

Function-specific similarity measure (CCO-SURROUNDED)

Attributes for the TARGET cells

ret: moderate

antenna.location.elevation: 17 , 18 , 19 , 20

antenna_type: Nothing selected

cell_type: surrounded

Expected KPIs for the TARGET cells

RLF INPUT: 91 : 383

RLF OUTPUT: 68 : 420

Select configuration set (CCO-SURROUNDED)

instance_name	matching_cases	successful_cases	confidence	proportion
v1	18	9	0.50	0.35
v2	33	16	0.48	0.65

Selected SON function configuration:

Thresholds:

$0.63 < CQI < 1.35$

$25 < CUE < 60$

$12 < RLF < 80$

$-107.09 < RSRP < -102.03$

Figure 3.5. A screenshot of the prototype GUI demonstrating the search results with metadata (KPI values and network status attributes) represented as facets.

Figure 3.6 demonstrates the suitability of the GUI design with an example use case. In the use case, user has decided to refine the results shown in Figure 3.5 by reducing the value range of the "RLF OUTPUT" shown in Entry A and the elevation value shown in Entry B. As the figure presents, the amount of cases has reduced (with respect to Figure 3.5) for the CCO-SURROUNDED function and for its configurations v1 and v2, shown in Entry C and Entry D. As a result of this context definition, the confidence levels of the instances have increased.

Matching operation case

function_name	matching_cases	successful_cases	confidence	proportion
CCO-SURROUNDED	15	9	0.60	0.88
CCO-HOTSPOT	1	1	1.00	0.06

Select configuration set (CCO-SURROUNDED)

instance_name	matching_cases	successful_cases	confidence	proportion
v2	4	3	0.75	0.27
v1	11	6	0.55	0.73

Selected SON function configuration:

Thresholds:
0.63 < CQI < 1.35
15 < CUE < 40
20 < RLF < 80
-107.09 < RSRP < -102.03

Function-specific similarity measure (CCO-SURROUNDED)

Attributes for the TARGET cells
ret: moderate
cell_type: surrounded
antenna.location.elevation: 17
antenna_type: Nothing selected

Expected KPIs for the TARGET cells
RLF INPUT: 91 : 383
RLF OUTPUT: 68 : 185

Figure 3.6. A screenshot of the prototype GUI demonstrating the change in the search results after human interaction (refining the facet values).

Altogether, the GUI presents and demonstrates how the administrator is involved in a system using case-based reasoning for discovering suitable SON functions for context-specific objectives. The designed framework with the described facet- and metadata-related functionalities is suitable when using case-based reasoning for discovering SON functions and configurations.

3.4 An Ontology-Based Framework and GUI Design for User Interaction and for Detailed Exploration of a Specific SON Function

CONTRIB2.2 is presented in Publication IV. The contribution is an interactive framework for monitoring a specific SON function. It is introduced along with a custom SON function based on an MLN model [114] that uses probabilistic reasoning.

The output of the contribution can be summarized as:

1. Defining GUI functionalities for monitoring and understanding the behaviour of the MLN-based SON function.

2. Presenting a semantic model for SON- and network-related concepts and cell-specific network performance values. These are used in a faceted search interface in order to allow interactive monitoring of the SON function.

3. Presenting a semantic model for the inner logic of the SON function. The model is also used in a faceted search interface in order to explore the inner logic and configuration of the SON function.

4. Demonstrating the GUI functionalities and the semantic model with a prototype implementation and experimenting its suitability with an illustrative scenario in the LTE simulator. The output of the evaluation is that the designed framework with the described facet- and metadata-related functionalities is suitable for the introduced MLN-based custom SON function.

3.4.1 MLN Model

The underlying MLN model is a probabilistic reasoning model that is capable of coping with complexity and adaptability. Instead of using a rule or case base, the MLN-based SON system is driven by weighted first order logic formulae for reasoning performed over evidence about system status. Every formula in the knowledge base has a structure described in Equation 3.5.

$$Context \Rightarrow (Objective \Leftrightarrow Action). \tag{3.5}$$

$Context$ (current fuzzy KPI values), $Objective$ (changes in KPI values), and $Action$ (changes in network parameters) are sets of one or more predicates, respectively. The MLN model presented in Publication IV has similar objectives to a standard CCO-based SON function. Thus, the analyzed KPIs and network parameters are also related to CCO use case.

The model learns weights for each formula by analyzing historical data (cell-specific performance data and actions). With respect to new evidence (context and objective data) and the weighted formula base, the model may propose actions for cells with probabilities that indicate the "confidences" of the actions. See Section 4 in Publication IV for more details about the MLN model.

3.4.2 Functionalities for the GUI

Following functionalities are defined for the user interaction with the system:

1) The administrator can monitor the current status of the cells where the SON function (MLN model) operates. The presented data is SON function-specific and comprised from following classes: relevant KPI values, cell metadata (e.g. amount of neighbors), SON function objectives (e.g. in-

creasing/decreasing cell-specific KPI values), and action proposals (increasing/decreasing a network parameter).

2) The administrator can examine the inner logic of the SON function and may modify its parameters. With respect to the MLN-based, this requires a presentation of the MLN-based rules/formulae and functionality to modify the rule weights.

3) The administrator can interactively explore the metadata and refine the results (displayed cells and rules) by filtering them with facets extracted from the metadata.

In order to provide flexible information exploration for the functionalities, two simple ontologies with SPARQL interfaces (a specific query language for semantic data) are designed for the SON function model and GUI.

3.4.3 Semantic Model for the GUI

Although the MLN model presented in Publication IV is targeted for CCO use case with particular settings (e.g. set of KPIs and network parameters), the same MLN model framework can be used to deploy various SON functions for different use cases with different configurations. For this purpose, semantic modelling facilitates the data management as it eases the adaptation of new MLN models to the system. The semantic modelling is done by using OWL and it contains mobile network concepts such as KPIs and cells on the one hand, and MLN model concepts like rules, actions, and parameters on the other. The MLN model supports reasoning of most effective actions to achieve a particular goal in a specific network context. The results of the reasoning (output) as well as evidence (input) are reflected in the instances of the ontology.

In Figure 3.7, relevant mobile network concepts are described together with the MLN evidence and action proposals. The "Cell" is the most fundamental class in the model and has properties "hasKPI" to its performance metrics (instances of the class "CellKPI") and "hasParameter" to its configuration parameters (instances of the class "CellParameter"). The "CellKPI" has a fuzzy description for its value, such as low, moderate, or high, which is defined in the MLN model. Also, according to the MLN model, a KPI might have a "KPIObjective" with an increasing or decreasing "EventImpact". The "CellParameter" can have an "ActionProposal", if the parameter needs to be adjusted with respect to the MLN inference. The "ActionProposal" has an "EventImpact" describing its impact direction.

In the 3.8 the weighted formulae of the MLN model are represented with concepts and mapped to mobile network concepts in the semantic model. The "rule" term is used as a synonym for a formula. The "MLNRule" class defines a formula that has a numerical value "hasRuleWeight" defining its weight and relations to formula classes "RuleContext", "RuleObjective", and "RuleAction". The figure also depicts that the formula classes are bound to network classes

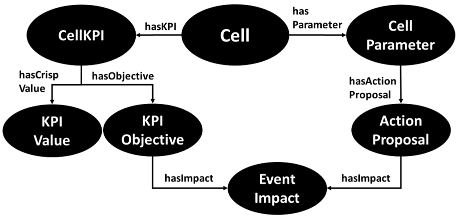


Figure 3.7. MLN evidence, action proposals, and their cell-related concepts in the semantic model.

"CellParameter" and "CellKPI". A "RuleAction" has a relation to a "CellParameter" (such as Txp) whereas "RuleObjective" and "RuleContext" have relations to "CellKPI" instances in order to express predicates.

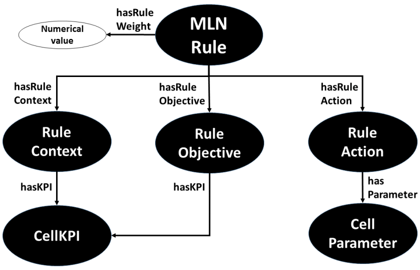


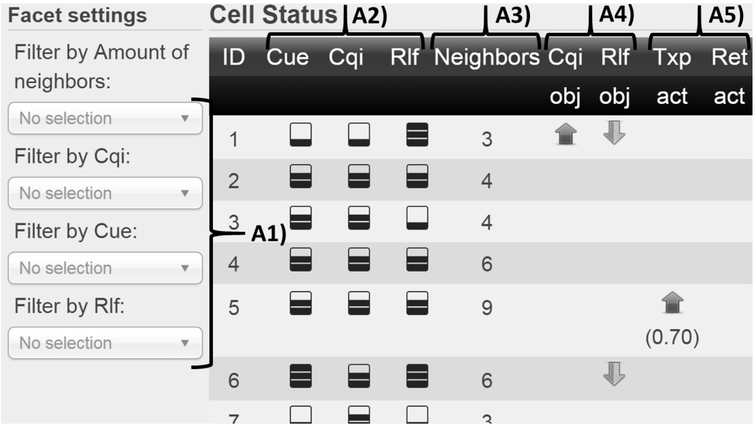
Figure 3.8. Rules (same as formulae in the MLN model) and their cell-related concepts in the semantic model.

The semantically modelled network and MLN-based SON function data can be queried with SPARQL queries which return cell- and rule-specific items fulfilling the query criteria. More detailed information about the data processing (data sequence in the system) is in Section 3.3 and about the SPARQL queries in Section 5.3 in Publication IV.

3.4.4 Demonstration

The MLN-based SON system is demonstrated with a prototype implementation that visualizes the semantically modelled data and supports exploration of the MLN-based SON function. The purpose of the GUI is to provide the end-user with informative and interactive tools for evaluating the MLN functionality. Thus, views are implemented to present the SON function-related cell states and MLN formulae.

Figure 3.9 shows cell states in a tabular visualisation. With this view, the administrator examines how MLN evidence (KPI values and objectives) affects the MLN reasoning outcome (action proposals). The rows depict cell instances and the columns their attributes, such as classified KPI values (A2), amount of neighbors (A3), KPI objectives (A4), and action proposals (A5). The data describe the current states of cells and thus are based on the latest PM report from the simulator. The administrator can interactively browse cells with similar states by selecting facet values (A1), such as the amount of neighbors and/or classified KPI values.



Facet settings

Filter by Amount of neighbors: No selection

Filter by Cqi: No selection

Filter by Cue: No selection

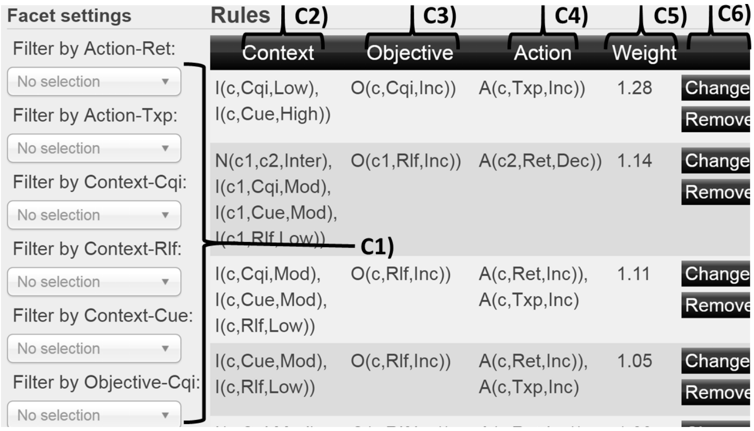
Filter by Rlf: No selection

Cell Status

ID	Cue	Cqi	Rlf	Neighbors	Cqi obj	Rlf obj	Txp act	Ret act
1	▂	▂	█	3	⬆	⬇		
2	▅	▅	▅	4				
3	▅	▅	▂	4				
4	▅	▅	▅	6				
5	▅	▅	▅	9			⬆ (0.70)	
6	█	▅	█	6		⬇		

Figure 3.9. A faceted view for a cell-specific tabular visualization

Figure 3.10 depicts the weighted formulae in a tabular view by dividing each formula into a formula weight (C5) and the formula classes defined earlier: context (C2), objective (C3), and action (C4). The administrator examines this view to learn the contents of the formulae and may modify or create formulae in order to change the behaviour of the MLN reasoner. For example, modification can be done by removing a formula or by changing its weight (C6). Facets (C1) in this view are generated as a combination of formula classes (contexts, objectives, and actions) and their objects (CQI, RLF, CUE, TXP, and RET).



Facet settings

Filter by Action-Ret: No selection

Filter by Action-Txp: No selection

Filter by Context-Cqi: No selection

Filter by Context-Rlf: No selection

Filter by Context-Cue: No selection

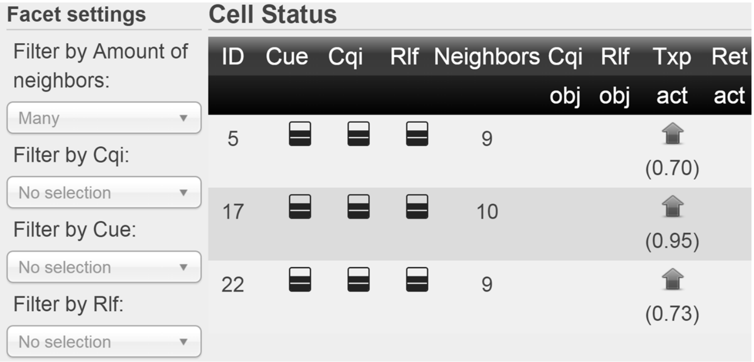
Filter by Objective-Cqi: No selection

Rules

Context	Objective	Action	Weight	
I(c,Cqi,Low), I(c,Cue,High))	O(c,Cqi,Inc))	A(c,Txp,Inc))	1.28	Change Remov
N(c1,c2,Inter), I(c1,Cqi,Mod), I(c1,Cue,Mod), I(c1,Rlf,Low))	O(c1,Rlf,Inc))	A(c2,Ret,Dec))	1.14	Change Remov
I(c,Cqi,Mod), I(c,Cue,Mod), I(c,Rlf,Low))	O(c,Rlf,Inc))	A(c,Ret,Inc)), A(c,Txp,Inc)	1.11	Change Remov
I(c,Cue,Mod), I(c,Rlf,Low))	O(c,Rlf,Inc))	A(c,Ret,Inc)), A(c,Txp,Inc)	1.05	Change Remov

Figure 3.10. Faceted view for MLN formulae.

In Publication IV is also demonstrated how the selection of facet values may help in understanding the behaviour of the SON function. For example, Figure 3.11 illustrates that the administrator can find out that the most central cells (most neighbours) are configured (Txp increased) in a specific situation.



Facet settings

Filter by Amount of neighbors: Many

Filter by Cqi: No selection

Filter by Cue: No selection

Filter by Rlf: No selection

Cell Status

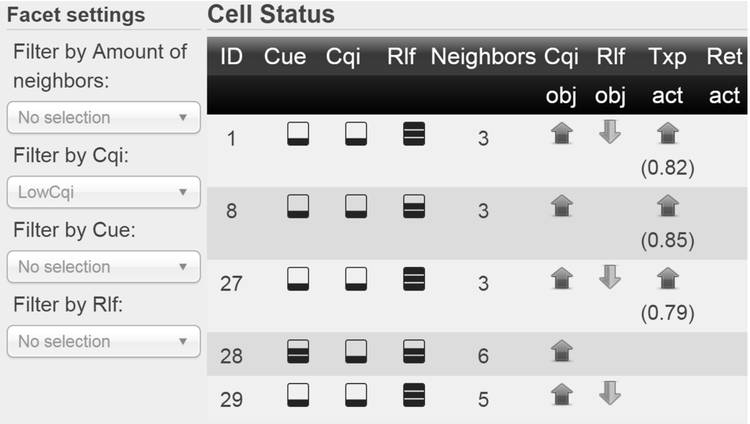
ID	Cue	Cqi	Rlf	Neighbors	Cqi obj	Rlf obj	Txp act	Ret act
5	▄	▄	▄	9			⇧ (0.70)	
17	▄	▄	▄	10			⇧ (0.95)	
22	▄	▄	▄	9			⇧ (0.73)	

Figure 3.11. Cells having many neighbors.

Also for demonstration purposes, a weight of a formula in the MLN model is modified from 0.41 to 4 (Equation 3.6).

$$Context(c,Cqi,Low) \Rightarrow (Objective(c,Cqi,Inc) \iff Action(c,Txp,Inc) \tag{3.6}$$

Figure 3.12 shows cell states for cells having low CQI values after MLN reasoner has recalculated the action probabilities. The change in the formula base has generated new action proposals for cells 1, 8, and 27, which implies that the weight update had an effect on the functionality of the reasoner.



Facet settings

Filter by Amount of neighbors: No selection

Filter by Cqi: LowCqi

Filter by Cue: No selection

Filter by Rlf: No selection

Cell Status

ID	Cue	Cqi	Rlf	Neighbors	Cqi obj	Rlf obj	Txp act	Ret act
1	▁	▁	█	3	⇧	⇩	⇧ (0.82)	
8	▁	▁	▄	3	⇧		⇧ (0.85)	
27	▁	▁	█	3	⇧	⇩	⇧ (0.79)	
28	▄	▁	▄	6	⇧			
29	▁	▁	█	5	⇧	⇩		

Figure 3.12. Updated states (actions re-inferred after model update) for cells having low CQI.

Altogether the GUI demonstrates that in addition to a dashboard GUI presented in **CONTRIB2.1**, it is also important to provide a detailed view for individual SON functions. As the design and demonstration of the framework presents, the administrator may extract relevant attributes for the SON function presentation, such as objectives, action proposals, and labelled KPI values. Moreover the administrator may search, explore, and modify the MLN formulae in order to understand the inner logic of the MLN SON function. Although the presented GUI framework is specific for the MLN model, it demonstrates the idea of presenting the inner functionality of a SON function and understanding its behaviour by modifying its parameters and interactively monitoring it. The same constructs of the framework may be used for other SON functions also in order to present them in their own GUI panels.

3.5 A Statistical Method Using Time Series-Based Event Pattern Mining for Creating Context-Specific Metadata about SON Functions

CONTRIB3.1 is presented in Publication V. The main output of the contribution is:

1. Describing the overview and components of the event pattern mining method. The components are existing statistical methods used for: 1) detecting events from time series and 2) mining patterns among multivariate data.
2. Demonstrating and evaluating the applicability of the method with simulated LTE data. The evaluation shows that the method facilitate the SON function modelling by generating meaningful metadata.

3.5.1 Overview

The contribution is a time series-based event pattern mining method that creates context-specific metadata about SON functions. The metadata can be used in SON function modelling to characterize SON functions and to make them discoverable with respect to context-specific objectives and requests (as described in **CONTRIB1.1**).

The implemented method is a combination of event detection and pattern mining and it has following phases:

Phase 1) The method takes the actions of a SON function operation as an input.

Phase 2) The method takes the performance data of the network cells as an input.

Phase 3) From the actions, the method analyzes and detects measurement events with a cumulative sum algorithm (CuSum) [102].

Phase 4) From the detected events, association rule learning component [60] finds dependencies among the events (e.g. whether two KPIs have increasing events at the same time). The association rule learning uses an open-source implementation of a well-known Apriori algorithm [4].

Phase 5) Finally, the associated events are sent to the SON function discovery-related ontology (described in **CONTRIB1.1**) and the SON function is characterized with these events. The events are used as "operation effects" in the ontology.

3.5.2 Event Detection with CuSum Algorithm

CuSum algorithm [102] is a statistical quality control method that can be used to detect value changes in a time series. The basic concept is to cumulatively sum up changes between data points and a comparison value, and flag a change if the sum exceeds a predefined threshold value. The Equation 3.7 describes how to detect increasing event in the system. The equation contains a max of zero and the cumulative sum of value s_h, the data point x_t, and the combined comparison value of mean and standard deviation, μ and σ, calculated from the time series. σ is used as a threshold sum value for increasing trends.

$$s_h = max(0, s_h + x_t - \mu - \sigma) \tag{3.7}$$

For analyzing decreasing trends in a time series, Equation 3.8 is used instead. Compared with the earlier equation, now a min operator is used and CuSum contains a positive sign for the σ. The threshold sum for detecting a decreasing trend is $-\sigma$.

$$s_l = min(0, s_l + x_t - \mu + \sigma) \tag{3.8}$$

When CuSum is executed for all operation-specific actions, the outcome is a dataset where each row depicts a single action having a list of measurement events it produced. From this dataset, SON function operation-specific event patterns can be learned.

3.5.3 Temporal Pattern Mining with Association Rule Learning Method

Association rule learning is a data mining method that learns rules between the sets of items in a database. The idea is to analyze the co-occurrence of items in a database row and to use some measure and threshold to find out relevant rules. The simplest measure is the support, which is calculated as a proportion of the database rows containing the given set of items. [60] In this method, the support is calculated as the proportion of detection timestamps containing a set of metric events. Thus, it indicates the frequency of the events occurring simultaneously in the given context.

In addition to support, confidence is another measure to determine associations between items. The Equation 3.9 shows the definition of the confidence. It can be interpreted as an if/then pattern: if set of events X occurs, then set of events Y also occurs. As it can be seen, the measure indicates the proportion of X (the support of X) that also contains Y (the support of X and Y). [60]

$$conf(X \rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)} \tag{3.9}$$

In this system, the objective is to learn support and confidence values for measurement events occurred during a set of actions made by an agent. For this purpose, an open-source Python implementation [10] that of a well-known Apriori algorithm (see [5] for further details) is used.

3.5.4 Evaluation

The method was evaluated with the LTE simulator with a similar setup described in Table 3.2. Thus, there were three SON functions: CCO-TXP with a TXP increase, CCO-RET with a downtilt, and MLB with a load balance operation). These were tested in three problem scenarios: coverage problem, local overload, and mobile overload. Several metric time series were analyzed, such as CQI, Physical Resource Block (PRB), RLF, RSRP, and throughput.

First, the support values of every operation on every problem scenario were evaluated. Figure 3.13 presents the SON function operation-specific support values in different scenarios and KPIs. The figure presents one subplot for each scenario and each subplot presents KPI-specific support values for each operation. Positive support value indicates a support measurement for an increase and negative a decrease. For example, the first five bars show support values for the increasing and decreasing events for the KPIs when no action has been taken in the coverage problem scenario. The first bar shows that increasing events for CQI has been measured with a support value of 0.12 and decreasing events with a value of 0.09. With respect to these experiments, a threshold level of ±0.15 (marked with two dashed lines) is suitable for labelling SON function operation-specific KPI effects.

Especially the increased throughput values in Figure 3.13 show that the best SON functions in every scenario also enhance the performance in the network, which is the desired outcome. Also, the fact that the number of false positive support values (values when no action is taken) is low, indicates an adequate performance of CuSum method.

In addition to support values, confidence values were also processed and evaluated. Confidence value indicates the frequency of a rule occurring in the event set. A rule may be, for example that "whenever throughput increases, RLFs decrease" ($IncTHR \rightarrow DecRLF$). The association rules for every scenario-specific action were generated with a minimum support level of 0.15 and confidence level of 0.70. Figure 3.14 shows the quantities of associations learned among the recorded events. With the given parameters, the best SON functions also

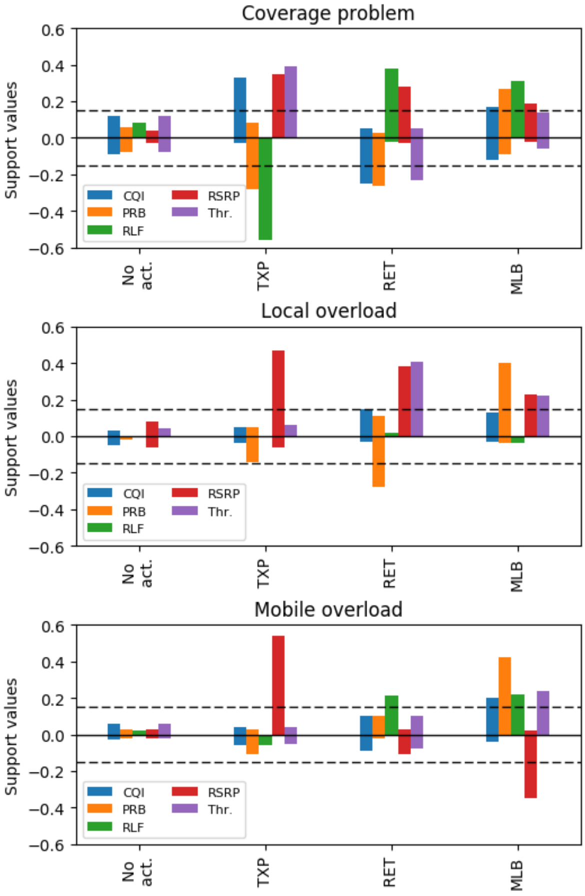


Figure 3.13. Support values for action- and KPI-specific events in three scenarios

generate most of the associations between the KPI effects. This is a desired outcome as the goal is to characterize the best matching SON functions in different contexts.

Finally, the associations based on confidence rules were analyzed by presenting the unique sets of associations that distinguish the scenario-SON function combinations from each other (Table 3.7). From these results it can be concluded that the suitable SON functions have their unique context-specific sets of associations which make also them discoverable in the SON service system. One could request to have these as desired objectives and would discover only the given scenario-specific SON function. Altogether, with respect to the simulator setup, all suitable context-specific SON functions could be characterized with the metadata generated by the described method.

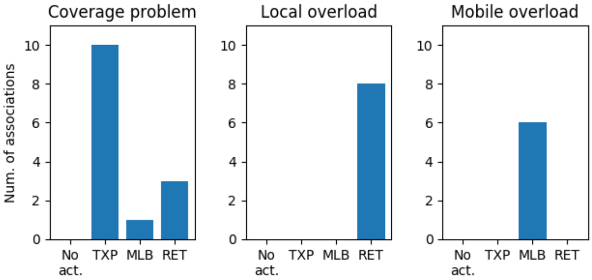


Figure 3.14. Scenario-specific quantities of associations for SON functions. Threshold for support is 0.15 and for confidence 0.70.

Table 3.7. Unique set of rules that characterize suitable SON functions in every scenario.

Scenario	Action	Matching rules
Coverage problem	TXP	$IncTHR \rightarrow DecRLF, DecRLF \rightarrow DecPRB$
Local overload	RET	$IncTHR \rightarrow DecPRB, IncRSRP \rightarrow DecPRB$
Mobile overload	MLB	$IncTHR \rightarrow IncPRB, DecRSRP \rightarrow IncPRB$

3.6 A Correlation-Based Method for Creating Linkages between Measurement Metrics across Different but Related Platforms/Datasets

CONTRIB3.2 is presented in Publication VI. The output of the contribution is:

1. Describing the overview of the correlation-based cross-platform metric mapping methodology.
2. Presenting the results of the mapping method based on the correlations calculated from the datasets. Our results show that the method maps the common features with high confidence scores (between 0.78 to 1.0 depending on the amount of features). The average mapping score increases when more similar features are involved.

3.6.1 Overview

In this contribution, the idea is to find statistical dependencies in local networks and utilize them in linking metrics across platforms. Instead of considering manual analysis and linkages across platforms, this work has the focus of facilitating the mapping of corresponding metrics across networks which might not always be straightforward. For example, potentially linked metrics might have: 1) similar names but be different metrics (such as "download speed" depicting either the throughput or average bit rate), 2) different names but be similar metrics (such as "latency" and "ping duration"), or 3) the same metrics in general but have differences in the underlying methodology (such as latencies measured with different protocols). This problem is addressed with a mapping method that is based on correlations among local metrics, and the algorithm is solving a maximum constraint satisfaction problem (CSP). In maximum CSP, the goal is to maximize the similarity of correlation patterns between the columns (metrics) of measurement datasets. The hypothesis of the method is that similar metrics across platforms have similar correlations between each other. For example, regardless of the network, the correlation between latency and downlink throughput should be less than the correlation between downlink and uplink.

The platform-specific correlation patterns are stored as inequality clauses and used as an input in the cross-platform mapping. Figure 3.15 shows an overall how the metrics can be mapped across the platforms. Both source and target platforms have their own sets of inequality clauses between correlation coefficients. From the source platform, it needs to be learned which of the coefficient inequalities are more regular than others regarding the monthly-divided N subdatasets (step 1). An inequality $r(f_x, f_y) < r(f_y, f_z)$ between correlation coefficients of metrics f_x, f_y, and f_y,f_z is added into the constraint base, if the inequality occurs in majority (more than 0.5 times) of the subdatasets. After this, the constraint base is used to find similar patterns from the target platform (step 2).

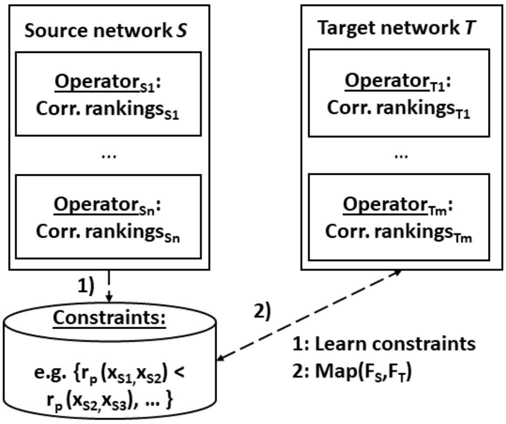


Figure 3.15. Steps in the mapping procedure: 1) learning the constraints from the source platform and 2) mapping those to correlation coefficients from the target platform.

Algorithm 1 demonstrates at a high level how the maximum CSP is adapted to the feature mapping. In CSP, values need to be assigned to variables so that given constraints are satisfied [121]. In this case, the constraints are the inequalities that are learned between the correlation coefficients of the source platform and variables the feature pairs of the source features F_S. The problem is then to assign feature pairs from the target platform (features F_T) as variable values to the constraint base (replacing F_S with F_T) so that the assigned feature pairs maximize the number of truth statements when comparing the constraint base with the target datasets. Assuming that a set of features would have a similar ranking of the coefficient values across platforms, the solution of the maximum CSP problem would then also be a mapping of features between F_T and F_S.

Basically, the algorithm tries every possible mapping combination between a set of target and source features, and tries to maximize the truth statements that the assignments (mapping) in the constraint base will produce while comparing the assigned constraints with the correlation data from the target platform.

Algorithm 1 Pseudoalgorithm demonstrating the functionality of the feature mapping.

```
for each possible mapping $Map_i(F_T,F_S)$ do
    Assign features $F_T$ to the constraint base wrt. $Map_i$
    for each monthly-based correlation matrix in the target platform do
        Count, how many times constraints are satisfied in the target platform
        with the current assignment.
    end for
end for
return Mappings having the highest count of truth statements
```

The algorithm returns a list of possible mappings, that have the highest satisfiability count. For every possible mapping between features $f_T i$ and $f_S j$, a mapping score is defined. The score is a portion of their occurrence in the returned list. For example, let us consider a mapping case where the problem is to map three features between platforms S and T: $F_s = \{x,y,z\}$ and $F_T = \{a,b,c\}$. The algorithm returns two lists of mappings: $Map_1\{(x,a),(y,b),(z,c)\}$ and $Map_2\{(x,a),(y,c),(z,b)\}$. For this example case, the mapping scores would be: $(x,a) = 1.0$ and 0.5 for $(y,b),(y,c),(z,b)$ and (z,c). The scores indicate that x and a could be mapped with each other while other mappings can not be deduced from these results. Generally, as the method requires inequalities between coefficients, at least three features from both platforms are required at minimum and a higher number of features would provide a richer set of constraints for the analysis.

3.6.2 Evaluation

The method is tested with two crowdsourced LTE measurement datasets, Netradar [2] and RTR Nettest [119]. Netradar data is collected from Helsinki and Nettest data from Vienna. To evaluate the feature mapping method, five features are included from the RTR Nettest (upload_kbit, download_kbit, ping_ms, lte_rsrp and lte_rsrq) and eight from the Netradar: uplink, downlink, latency, RSRP, Reference Signal Received Quality (RSRQ), Reference Signal to Noise Ratio (RSSNR), battery_level, and speed. The assumption was that there are five pairs of common metrics across the datasets: uplink-upload_kbit, downlink-download_kbit, latency-ping_ms, RSRP-lte_rsrp, and RSRQ-lte_rsrq.

Average Mapping Scores of the Method

The overall performance of the mapping is evaluated by generating all possible mapping combinations between the source and target features in order to examine how the method catches the different levels of similarities between the feature sets. The overall results are presented in plots that show average mapping scores as a function of the ratio of common features (the ratio of true positive mappings). This means that all mapping cases that have the same ratio of common features are grouped and the average mapping score over those are calculated. For example, the mapping of three features means that there are 560 mapping combinations ($\binom{5}{3} \times \binom{8}{3}$) in the plot and the point where the common ratio is 3/3, the value is averaged over 10 ($\binom{5}{3}$) different mapping combinations.

Figure 3.16 shows the average scores of the feature mapping while RTR Nettest is the source and Netradar the target platform. The figure has three subplots separating the mapping results between the mapping of three, four, and five features. All the subplots show that the average scores of perfect mappings (common ratio is 1.0) can clearly be distinguished from mappings having more false positives (common ratio is lower than 1.0). False positives include all imperfect mappings that do not refer to the same metric. Moreover, the increasing trend of the scores as a function of the common ratio can be noticed. This shows the desired outcome that the feature mapping method gives better scores when a higher portion of common features are mapped. The different correlation methods, Pearson, Spearman, and their combination (both of them are used in the constraint base), give rather similar results, meaning that linear relations can be used as well as non-linear.

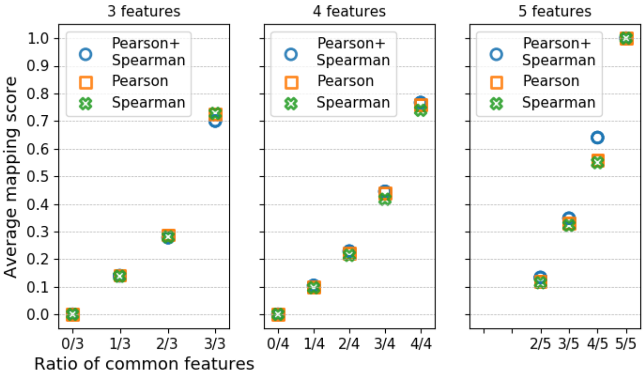


Figure 3.16. Average mapping scores when mapping from RTR Nettest to Netradar. The score increases as a function of the common ratio. The perfect mappings (ratio of 1.0) outperforms the incomplete mappings.

Figure 3.17 shows the mapping results when Netradar is the source, and RTR the target platform and the same features are included in the evaluation. Apart from a small variation, the scores are rather equal to the earlier mapping case shown in Figure 3.16.

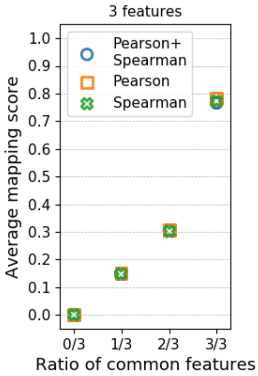


Figure 3.17. Average mapping scores when mapping from Netradar to RTR Nettest. The score increases as a function of the common ratio. The perfect mappings (ratio of 1.0) outperforms the incomplete mappings.

The average scores show that the feature mapping method is able to distinguish incomplete mappings from the perfect mappings. Regardless of the amount of mapped features, there is a remarkable gap in the scores between the incorrect mappings and correct mappings (between the scores having a common ratio of 1.0 compared with lower ratios). Moreover, the scores of the incorrect mappings give an insight of how many correct features there might be between the platforms, as the average score clearly correlates with the common ratio.

Feature-Specific Scores

The method is also evaluated from the perspective of the common features in order to report the differences between the features. A feature-specific mapping score is defined as an average over all mapping cases, in which the common ratio is 1.0. There are 16 such cases ($\binom{5}{3} + \binom{5}{4} + \binom{5}{5}$), the feature-specific score of the ith common feature (f_{Ti} and f_{Si} respectively) is $\frac{1}{16}\sum_{j=0}^{16} score(f_{Tij}, f_{Sij})$.

Feature-specific scores of the mappings can be seen in Figure 3.18. This plot shows that there are more variations between the correlation methods and between the feature scores than that could be seen from the earlier overall results. The highest difference in the performance is in the uplink scores; uplink gets a score of 0.8 while RTR Nettest is the source platform, but 0.96 while Netradar is the source platform.

The latency outperforms all other features having the highest possible mapping score of 1.0. This result is expected, because latencies in both platforms had the lowest correlations with all the other features, which makes it easy to distinguish it with respect to this method. Another finding of these plots is that RSRP and RSRQ are more difficult to map than the other features, as their scores are lower. A closer look to the individual mapping cases shows that RSRP and RSRQ were sometimes mixed up together when only three features were mapped.

Altogether, the feature-specific figure scores are high enough to make correct mappings between all features, but there are some variations between the feature scores. For example, RSRP and RSRQ have lower scores, whereas latency clearly has the highest scores of 1.0. All scores are acceptable, as any score higher than 0.5 for a common feature implies that on average a correct mapping is selected. Moreover, it should be noted that a random guess would have a mapping score of 0.33, 0.25, or 0.2, depending of the amount of features ($1/N$, in general).

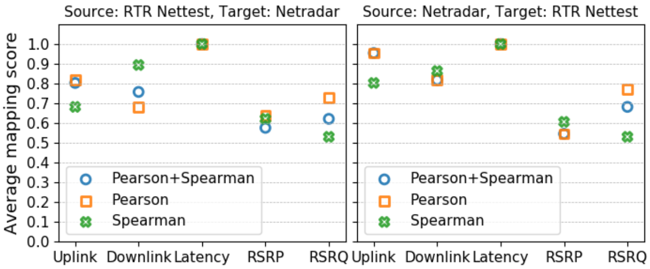


Figure 3.18. Feature-specific mapping scores. The scores are high enough to make correct mappings for all features but there is variation between the features.

3.7 Summary of the Contributions

This section summarizes the contributions by presenting briefly their main content and then answering related research questions.

CONTRIB1.1 is a semantic reasoning-based SON function discovery and composition model and method. The contribution 1) presents a semantic model relevant to mapping SON functions to network problem contexts, 2) presents a semantic reasoning rule for the mapping, and 3) evaluates the model and method by simulating SON functions and problem contexts in the LTE simulator. The contribution answers RQ1.1 which was *"How does semantic reasoning facilitate the mapping of context-specific objectives to SON functions in a SON function discovery mechanism?"* The contribution shows that by using the presented semantic model and reasoning rule, the contribution facilitates the mapping of requests (context-specific objectives) to SON functions. The semantic reasoning mechanism infers mappings between effects of requests and suitable SON functions (with respect to analyzed historical data).

CONTRIB1.2 contains a semantic model and reasoning method for LTE metric dependencies. The contribution 1) presents a semantic model and logical axioms for dependencies between LTE metric effects, 2) presents reasoning rules for inferring new cross-platform dependencies between LTE metrics, and 3) evaluates the suitability of the model and method with an illustrative scenario by analyzing statistical correlations in the LTE simulator and test network. The contribution answers RQ1.2 which was *"How does semantic reasoning facilitate the cross-platform inference of statistical and human-defined dependencies between network metrics?"* The presented and evaluated semantic model and reasoning rules facilitate the cross-platform inference of metric dependencies by generating new dependency linkages between LTE metric effects.

CONTRIB2.1 is a SON service system with a framework for an interactive SON function discovery. The designed framework 1) defines a context-specific administrator objective (input for the system) with relevant metadata, 2) represents the search results (SON functions and their configurations) with case-based reasoning-specific statistics and with SON function-specific metadata, 3) provides the administrator with an interactive faceted search interface for exploring the metadata, and 4) demonstrates the framework with a prototype implementation. This contribution answers RQ2.1 which was *"What network- and SON-related GUI functionalities should be provided when using case-based reasoning for discovering suitable SON functions and configurations in specific problem contexts?"* The contribution shows that the designed framework with the described facet- and metadata-related functionalities is suitable when using case-based reasoning for discovering SON functions and configurations.

CONTRIB2.2 is an ontology-based framework for the user interaction and detailed exploration of a specific SON function. It 1) presents GUI functionalities for monitoring and understanding the behaviour of the MLN-based SON function, 2) presents a semantic model/ontology for SON- and network concepts, 3) presents a semantic model/ontology for the inner logic of the MLN-based SON function, and 4) demonstrates the framework with a prototype implementation. The contribution answers RQ2.2 which was *"What network- and SON-related information models and GUI functionalities should be used in order to provide SON administrators a metadata-based GUI as a tool for understanding the behaviour and characteristics of a single SON function?"* The contribution shows that the framework with the presented semantic model and GUI functionalities is a suitable tool for the administrator when the behaviour of a specific MLN-based SON function needs to be explored.

CONTRIB3.1 is a time series-based event pattern mining method that creates context-specific metadata about SON functions. The method mines measurement event patterns that characterize SON functions and make them discoverable with respect to the metadata. The contribution 1) presents the overview and components of the method and 2) demonstrates and then evaluates the applicability of the method with simulated LTE data. The contribution answers RQ3.1 which was *"How statistical methods can be used to facilitate and assist the creation of LTE measurement-based and context-specific metadata for SON functions?"* Meaningful metadata can be produced for SON functions by combining event detection- and association rule learning-based methods and applying those to network and SON management data. The produced metadata characterizes the set of SON functions and enables the discovery of suitable SON functions for specific problem contexts.

CONTRIB3.2 Correlation-based method for creating linkages between LTE metrics across different but related platforms/datasets. The method assists in integrating and merging separate LTE measurement datasets and their metrics together. The contribution 1) presents the overview of the correlation-based cross-platform metric mapping methodology and 2) evaluates the mapping method based on the correlations calculated from two crowdsourced LTE measurement datasets. It provides an answer to RQ3.2 which was *"How statistical methods can be used to facilitate and assist the mappings between LTE metrics across platforms?"* Cross-platform mappings of LTE metrics can be facilitated by analyzing correlation patterns among the sets of LTE metrics inside the platforms and then identifying corresponding metrics across these platforms. This can be done with the algorithm solving a maximum constraint satisfaction problem (CSP).

4. Discussion

A comparative evaluation of the frameworks, models, and methods included in this thesis is difficult, as the systems provide novel approaches that are targeted for future use cases. These use cases are realized when heterogeneous SON management platforms along with new network infrastructures are implemented. This section discusses how the designed contributions have impacted the theories and practices in the SON management and related domains. Following criteria have been used to evaluate the work:

1. **Theoretical implications**: Reflection against the previous literature
2. **Practical implications**: Implications for society, companies, and/or organisations
3. **Reliability**: Evaluation of the clearness of research questions and basic constructs. Discussion about contribution descriptions and personal biases.
4. **Validity (internal and external)**: Plausability of the contributions with respect to stated objectives, alternative methods, and limitations that are found (internal validity). Generalisation of the results and congruency with earlier theory (external validity).

4.1 Theoretical Implications

The theoretical implications of the thesis relate to the same topics presented in Sections 2.2-2.5. The novelties of the contributions are reflected against the earlier theories of these topics.

4.1.1 Characterizing SON Functions by Analyzing and Evaluating Their Performance

In order to characterize SON functions, **CONTRIB1.1** and **-3.1** emphasize the modelling of contextual metadata regarding the SON function environment and the evaluation of SON functions in context-specific scenarios. These contributions can be used to discover SON functions based on their metadata and verify their functionality in various contexts.

The future perspectives of autonomic network management consider SON management as an important building block [95, 1, 159]. Characterization of SON functions has been studied in view of the dynamic mapping of operator goals for SON function configurations [55, 85]. Also, some approaches have enriched the predefined SON function models by gathering experience of their actual performance and impact on the network [85, 84]. Other SON-related studies have specifically focused on the verification and coordination of the SON functions, either to avoid degradations [137, 145, 28] or to analyze joint effects [59, 68, 14, 54] of the functions. In view of cross-platform information exchange, some approaches in the mobile network management have been proposed to share local configurations and operation models across networks [21, 29].

This thesis presents two complementary solutions to previous works and novel approaches which test existing theories from other research fields to assist future SON systems. **CONTRIB1.1** provides a semantic modelling of SON functions and network context in view of SON function discovery across local networks. This solution is inspired by semantic web service discovery and composition mechanisms [58]. **CONTRIB3.1** provides measurement patterns as context- and configuration-specific SON function metadata. The solution utilizes the theories of CuSum [102] and association rule learning [60].

4.1.2 Semantic Modelling and Reasoning in Network- and Measurement Data Management

Reasoning use cases considered in this thesis include the mapping of requests and SON function operations (**CONTRIB1.1**) and inference of LTE metric effect dependencies (**CONTRIB1.2**).

With respect to the request-operation mapping, some previous efforts have focused on policy-based network management in SON environment [55, 85, 36, 37, 56]. Although these works share the idea of mapping human-defined objectives to SON function configurations, they neither consider semantic modelling nor logical reasoning that would facilitate the discovery and composition of SON function operations.

Particularly related to the reasoning in SON research, earlier efforts do not address directly same use cases as this thesis. Some works define SON function policies with logical rules [96, 97, 24, 26] and are targeting directly to model the inner logic of single SON functions. The works addressing conflict detection between SON functions with semantic modelling and reasoning [148, 109] are partially related to this work as one objective of the effect dependency inference is to reveal conflicts between SON function effects. However, related studies either consider conflicts in configuration parameters [148] or conflicts in the same metric [109], whereas the effect dependency inference presented in this thesis detects indirect conflicts between metrics that are semantically different but correlate in a specific context. To the best of author's knowledge, the semantic inference of metric dependencies and SON function discovery has not yet been presented in the SON-specific research.

In a wider picture, semantic inference use cases in the wireless network research have been studied for service discovery [79, 8], selection [6], recommendation [40] and priority definition [47]. In **CONTRIB1.1**, request-operation mapping tasks differ from the previously presented papers by not only discovering services (analogical to SON functions) but also providing a detailed description of how requests may include a set of sub-requests and how the service discovery mechanism infers and composes a set of services that fulfil a complex request.

With respect to **CONTRIB1.2**, semantic inference of metric dependencies is not explicitly presented in the wireless network research. Some works infer knowledge from measurement data [132, 136, 47], but no inference rules or descriptions of reasoning about metric dependencies are mentioned. Generally, there have been efforts to define metric ontologies including correlations, aggregations, and hierarchies among metrics [115, 42, 41, 107]. Similar ideas are applied to the semantic modelling and inference in this work by especially targeted to SON functions and context-specific network measurements related to those.

4.1.3 GUIs for Providing Better Understanding of Autonomic Network and Service Management Functionalities

This thesis presents user interaction in 1) the discovery and selection of SON functions and their configurations with respect to operator objectives (**CONTRIB2.1**) and 2) the visualization of SON function-specific functionality and characteristics (**CONTRIB2.2**).

In the earlier SON management research, some GUI demonstrators have been built to visualize and present network problems and to show how SON functions have reacted to the problems [77, 13]. Some implementations have also presented interfaces to define operator objectives that are then mapped to particular SON function configurations or other management services in order to fulfil the objectives [129, 126, 83]. Unlike earlier works, the contributions of this thesis present exploratory faceted search interaction based on context and performance attributes. The contributions provide both context-specific interactive comparison of multiple SON functions and their configurations (**CONTRIB2.1**) and exploration of an individual MLN-based SON function behaviour (**CONTRIB2.2**).

In the design of the GUIs, existing theories from other research fields have been applied. In view of the interactive comparison of SON functions (**CONTRIB2.1**), similar efforts can be found from the earlier web service research to compare services. The comparisons are based on contextual metadata and filtering mechanisms to allow users to dynamically examine the performances [157, 57, 34]. Moreover, similar design for interactive exploration and semantically structurized representation of a single agent behaviour (as presented in **CONTRIB2.2**) has been earlier applied for machine learning agents in order to make humans understand their decisions [160, 162, 78].

4.1.4 Measurement-Based Statistical Methods in Network Management

In this thesis, the methods for facilitating the development process of autonomic network management are focused on creating measurement-based metadata for SON function operations (**CONTRIB3.1**) and to map metrics between the measurement datasets of different platforms (**CONTRIB3.2**).

Measurement-Based Pattern Mining
The time series-based metadata creation method of SON functions is a combination of event detection and association rule learning among detected events. With respect to measurement-based pattern mining, Lohmüller et al. [84] presented a method in order to characterize cells while single SON functions were active. Ciocarlie et al. [29] stored measurement-based patterns in order to characterize topology changes (the addition of new cells) in the network [29]. Contrary to **CONTRIB3.1**, the related works do not mine patterns to produce metadata for SON functions, which is in the centre of this thesis.

CONTRIB3.1 is novel in view of using measurement-based pattern mining to characterize SON functions and network contexts. Instead, other SON-related and measurement-based pattern mining methods address anomaly detection in order to trigger suitable SON functions and other algorithms with targeted configurations [72, 140, 158]. Particularly related to the association rule learning, some earlier methods have been developed in the field of IoT and sensor networks [22, 49]. Although these works are neither related to mobile networks nor automated agents operating in the networks, they show the applicability of association rule learning in the sensor time series in order to learn contextual patterns from those.

Automatic Methods to Map Metrics Across Data Sources
In the SON function research, statistical methods to cross-dataset metric mapping could not be found. The usual setup of a SON research study is a single-domain simulator system where KPIs and their semantics are known. In analyzing end-user-based LTE measurements, the need for matching metrics across measurement platforms has been recognized, for example to compare the performance of network providers [82, 81, 86].

Compared with **CONTRIB3.2**, the previous works rely on human-defined mappings of LTE metrics whereas the method presented in the contribution can map certain metrics automatically. Generally in the wireless and sensor network management, there has been interest in matching measurement features across data sources with methods that are partially related to the one presented in this thesis [87, 103, 154, 27, 150]. These works use manually defined mappings or classified data where labels describe the data samples. The presented contribution has the objective of matching the similar sets of metrics, without a need to manually process the dataset contents. Generally, the method shares the idea of statistically analyzing the features in order to learn common features across datasets [19, 38, 103].

4.1.5 Summary

Contributions and their theoretical implications are summarized in this subsection.

CONTRIB1.1 is a semantic model and reasoning method for SON function discovery and composition. The novelty is in providing semantic modelling for SON functions and network contexts in view of discovering SON functions for specific problems. Moreover, a novel solution is to apply the principles of service discovery and composition mechanisms (reasoning capabilities) to the SON system.

CONTRIB1.2 is a semantic model and reasoning method for inferring dependencies about and between LTE metrics. It presents a new semantic model for inferring network metric dependencies in order to understand similar and conflicting effects in network performance data.

CONTRIB2.1 is a framework and GUI design for interactive SON function discovery. A novel solution is to combine user interaction with case-based reasoning in order to characterize and discover SON functions. Also, the design of the GUI framework includes exploratory faceted search capability in order to compare SON functions and their configurations in specific contexts.

CONTRIB2.2 is an ontology-based framework and GUI design for user interaction and for a detailed exploration of a specific SON function. The novelty of the contribution is in applying exploratory faceted search capabilities and semantically structurized representation of a single agent into SON system in order to understand the behaviour (such as decisions and actions) of an MLN-based SON function.

CONTRIB3.1 is a statistical method using time series-based event pattern mining for creating context-specific metadata about SON functions. The solution produces association rule-based measurement patterns as context- and configuration-specific SON function metadata.

CONTRIB3.2 is a correlation-based method for creating linkages between measurement metrics across different but related platforms/datasets. The method is novel in view of matching similar sets of LTE metrics without a need to manually process the dataset contents.

4.2 Practical Implications

The contributions of the thesis were designed and targeted for multi-domain SON management environment integrating SON function data and experiments from multiple heterogeneous platforms. In general, following stakeholders may have practical benefits from the contributions presented in this thesis:

- an operator or vendor having numerous heterogeneous local networks and siloed management platforms in, for example, multiple countries
- operators and/or vendors managing the same federated network
- operators and/or vendors who are willing to collaborate and share SON operational experiences with each other (for example, operating in different countries or industry domains)

Following subsections elaborate the practical benefits of the solutions in more details.

4.2.1 SON Function Management Across Networks and Platforms

SON paradigm will be an important building block in the 5G network management [127] and thus, new methods that improve the automation and performance of SON function management would be beneficial. The contributions of this thesis are related to the utilization of cross-platform SON function management by defining and creating contextual metadata (**CONTRIB1.1** and **-3.1**) and providing the administrator with a tool to interactively verify the context-specific performances of SON function operations (**CONTRIB2.1**). The results would benefit SON function and context modelling tasks in cross-platform management systems as semantic definitions of context attributes require effort and collaboration in order to ensure seamless information exchange. Moreover, in the 5G environment, it is likely that the number of SON use cases increase significantly [138, 110] and more competing algorithms are developed for existing use cases [159, 95]. These two aspects make the context-specific evaluation of the higher number of SON function operations even more important.

4.2.2 Semantic Modelling of SON Functions and Network Context

Currently, SON functions are rigorously designed and targeted for some predefined use cases so that every problem context could be solved with distinct SON functions [100]. Even in the current situation, configurations of every function need to be adjusted with respect to the context in the different parts of a local network. Moreover, SON function implementations may slightly vary among organizations. The proposed ontology and inference rules (**CONTRIB1.2**) would help an operator and/or vendor who monitor numerous of counters and KPI in situations where contextually similar local solutions could be shared across networks. When the number of SON use cases and function implementations are increasing, the proposed semantic modelling and automated inference for SON function discovery (**CONTRIB1.1**) would be beneficial. As an example of an upcoming scenario that could be solved with semantic reasoning would be adapting suitable SON functions from cellular networks to industrial wireless networks [88].

4.2.3 Interactive GUIs for Understanding and Discovering SON Functions

The increasing number of SON use cases and implementations will also imply practical significance for an interactive SON function discovery GUI (**CONTRIB2.1**) that is presented in this thesis. The need for comparing both the SON functions and their various configuration sets increase. At the same time, administrators should have tools for both selecting SON functions and understanding automatically performed selections of the most suitable SON functions. Especially, the administrator should understand: 1) why the function is the best option in a specific context and 2) which context attributes affect to the discovery results. Moreover, the interactive SON function discovery GUI has a concrete practical implication as it is adapted as part of further research [137], which is not part of this thesis.

Furthermore, it is crucial to provide operators with comprehensive visualizations of the 5G network status and its autonomic functionalities. This aspect is motivated also in the design of the SELFNET GUI component [51]. In this thesis, the interactive GUI for an individual SON function (**CONTRIB2.2**) provides important experiments for this purpose. In order to visualize functionalities and to enable parameter modifications to a diverse set of SON functions, the proposed ontology-based model provides flexibility in designing function-specific panel views.

4.2.4 Characterizing SON Functions with Measurement-Based Patterns

In a cross-domain network environment, the association rule learning-based metadata creation method (**CONTRIB3.1**) eases the deployment of semantically defined autonomic SON functions in order to distinguish SON functions and their configuration sets from each other with respect to metadata. As a practical implication, the method benefits in situations where a context-specific problem scenario needs to be solved by selecting the most suitable operation among various SON functions and configuration sets. In this case, past experiences of SON function operations should be stored with appropriate contextual metadata.

4.2.5 Sharing Operational Experiences Across Domains

SON functions may be developed by different service providers but still semantically have similar objectives and effects on the network, even though they are analyzing syntactically different metrics. In this case, finding and matching the correspondences between metrics (**CONTRIB3.2**) are essential aspects in order to share the operational experiences across domains. To avoid cost-intensive manual work, automatic metric matching would benefit especially those who have either various data sources to integrate together or huge amount of metrics to match between datasets, or both.

4.3 Reliability and Validity

4.3.1 Reliability

The reliability refers to the quality control of the research process and whether it has been consistent and reasonable over time. The overall goal, research objectives, and research questions were defined in Chapter 1. The overall goal of the thesis was to provide novel contributions to the multi-domain and cross-platform utilization of autonomic SON functions by adapting the principles and practices of service-oriented computing to the SON-based management. Objectives were derived from the overall goal and the relations between objectives and research questions were presented in Table 1.1.

The previous works were gathered and grouped with respect to research questions in Chapter 2. The overview and evaluation of the contributions were presented in Chapter 3 and explained in more details in the individual publications included in the thesis. Subsection 3.7 summarizes the contributions and shows that each of them answers one research question of the thesis. Altogether, all research questions are answered with respect to the presented publications.

The work is done by following the principles of design science. The objectivity is ensured by systematically recognizing and solving unsolved research problems and by reporting the findings to the research community. Although the research is mostly done in collaboration with Nokia, it is rather influenced by general long-term visions (of previous literature) of SON and 5G. Thus, the author does not recognize personal biases that might have impacted the process.

4.3.2 Internal Validity

Internal validity refers to the plausibility of the developed methods and models with respect to the stated objectives, alternative methods, and limitations that were found. Table 4.1 show that objectives have been addressed with the corresponding contributions.

Table 4.1. The relationships between the objectives and contributions.

Objective and contribution
OBJ1.1: Create a discovery mechanism to find the most suitable SON functions for specific operator objectives in context-specific problems. **CONTRIB1.1**: A semantic model and reasoning method for SON function discovery and composition.
OBJ1.2: Provide linkages of LTE network measurement data across platforms in order to discover the similarities of the metrics. **CONTRIB1.2**: A semantic model and reasoning method for inferring dependencies about and between LTE metrics.
OBJ2.1: Design GUI functionalities that facilitate browsing, discovering, and comparing multiple SON functions in context-specific situations. **CONTRIB2.1**: A framework and GUI design for interactive SON function discovery.
OBJ2.2: Design GUI functionalities that facilitate understanding the behaviours and characteristics of an individual SON function. **CONTRIB2.2**: An ontology-based framework and GUI design for user interaction and for detailed exploration of a specific SON function.
OBJ3.1: Implement and adapt existing statistical methods for creating metadata for SON functions. **CONTRIB3.1**: A statistical method using time series-based event pattern mining for creating context-specific metadata about SON functions.
OBJ3.2: Implement and adapt existing statistical methods for producing linkages between LTE metrics. **CONTRIB3.2**: A correlation-based method for creating linkages between measurement metrics across different but related platforms/datasets.

As the research questions were formed from these objectives and answered with evaluated contributions in Chapter 3, it can be concluded that this work meets its objectives. Alternative methods and theories were discussed and compared in Section 4.1. A general limitation in the internal validity is the availability of real data, which is a recognized issue in SON research [95, 159, 73]. Getting SON-related data from multiple sources is difficult as they usually face privacy and confidentiality issues. The evaluations related to SON functionality were done in a simulator and other network-level measurements were analyzed from the test network (Netleap) and crowdsourced measurement datasets [119, 2].

4.3.3 External Validity

External validity refers to the generalizability of the findings and their congruence with earlier theory. The generalizations of the contributions (listed below) are made within SON function and mobile network management scope as the evaluations of contributions has been made with LTE-related performance data from network simulator, test network, and crowdsourced platforms.

- The faceted search interface for retrieving and comparing multiple agents (**CONTRIB2.1**) combined with the semantic model for SON function discovery (**CONTRIB1.1**) could be applied to cross-platform SON management where contextually similar SON functions operate in other networks.
- The reasoning of metric effect dependencies (**CONTRIB1.2**) could be utilized generally in (not only SON-related) cross-platform LTE- and future mobile network management where metric and KPI dependencies are not known.
- The ontology-based GUI for exploring individual SON functions interactively (**CONTRIB2.2**) could be adapted to a diverse set of SON functions with machine learning use cases where the inputs, outputs, contexts, and inner functionalities of complex algorithms need to be explored.
- The automated measurement-based metadata creation method (**CONTRIB3.1**) could be used generally for time series in the LTE- and future mobile network management systems in order to distinguish contextual situations with measurement event patterns.
- The automatic matching of measurement-based features between data sources (**CONTRIB3.2**) could have external validity generally in mobile network management systems where the integration of heterogeneous measurement data is beneficial and where features are assumed to have similar context-specific correlations with each other.

With respect to the congruency to the earlier studies, all the contributions of this thesis have been developed by considering the previous works and challenges in SON management. Moreover, many of the methods have been inspired from related research fields, such as exploratory (faceted) search interface [155], ontological modelling of metrics [115], semantic web service discovery and composition mechanisms [58], association rules learning [60], and common feature representation learning [19, 38, 103]. These paradigms and methods are utilized as a basis to implement use case-specific methods and systems in the service-oriented SON management in order to address the objectives and to answer the research questions of this thesis.

4.4 Recommendations for Future Work

The findings of this thesis imply future work in several directions related SON management. In order to gain more experiments about the cross-platform SON function discovery mechanism in practice, several heterogeneous SON datasets and simulators should be used. As there are practical difficulties in getting real data, especially including both SON-specific configurations and performance data, the next step would be setting up similar environments with multiple simulators. With experiments from multiple simulators, one could develop and adapt experimental SON algorithms from earlier research and for example investigate, how a machine learning-based SON algorithm targeted for one simulator, could be shifted to another simulator with respect to contextual attributes and semantically defined metrics, which may slightly vary among simulators. This research direction also lends itself to the transfer learning [25] and domain adaptation [17] paradigms in the machine learning.

The human-computer interaction could be further studied by integrating the two presented GUI frameworks in Publications III and IV together so that the administrator may select a SON function and its configuration in the search results and examine the inner functionality of the individual object. This research would give more insights of the exploratory interactive GUI design in order to understand behaviours of individual SON function configurations. Moreover, it would give more experiments of how semantic modelling could facilitate and save the development costs in the presentation of different individual SON functions in similar GUI panels.

Another future research direction is to analyze context-specific performance in the LTE and 5G networks. In view of cross-platform management, the relation between a set of context attributes and measurement patterns should be learned in the real-world network scenarios. Moreover, understanding the relation between contexts and measurements, it could be also beneficial to research integration of third party data to management systems. As an example, weather forecasts might imply estimations about signal quality and information about mass events would provide predictions about traffic patterns. The relevant context attributes and their effects on metrics should be modelled semantically in order to utilize them across platforms.

As a final suggestion for future work, the earlier work done in the policy- and objective-based SON management [55, 53, 85, 36, 37, 56] as well as the intent-based networking concept [16] could be extended with semantic definitions of context-specific high-level goals in the networks. This research direction would also require a semantic modelling of context attributes and their relation to measurement patterns in order to define, what impacts a high-level goal in a certain situation will have. For example, by combining measurement pattern mining and semantic models of contexts and high-level goals, a management system may infer that an improvement in the customer satisfaction and signal quality is unlike to achieve outdoors during a heavy rain due to signal interference.

References

[1] 5GPPP Architecture Working Group. View on 5G architecture. Technical report, 5GPPP, 12 2017.

[2] Aalto University, the Department of Communications and Networking. Netradar measurement platform. `https://www.nettitutka.fi/`. Accessed: July 9, 2018.

[3] Mamta Agiwal, Abhishek Roy, and Navrati Saxena. Next generation 5g wireless networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 18(3):1617–1655, 2016.

[4] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. *SIGMOD Rec.*, 22(2):207–216, June 1993.

[5] Rakesh Agrawal, Ramakrishnan Srikant, et al. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487–499, 1994.

[6] Yahya Al-Hazmi and Thomas Magedanz. Towards semantic monitoring data collection and representation in federated infrastructures. In *Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on*, pages 17–24. IEEE, 2015.

[7] Ahmed Al-Saadi, Rossitza Setchi, and Yulia Hicks. Semantic reasoning in cognitive networks for heterogeneous wireless mesh systems. *IEEE Transactions on Cognitive Communications and Networking*, 3(3):374–389, 2017.

[8] Mahdi Ben Alaya, Samir Medjiah, Thierry Monteil, and Khalil Drira. Toward semantic interoperability in onem2m architecture. *IEEE Communications Magazine*, 53(12):35–41, 2015.

[9] Jeffrey G Andrews, Stefano Buzzi, Wan Choi, Stephen V Hanly, Angel Lozano, Anthony CK Soong, and Jianzhong Charlie Zhang. What will 5g be? *IEEE Journal on selected areas in communications*, 32(6):1065–1082, 2014.

[10] Anonymous. Python implementation of apriori algorithm. `https://github.com/asaini/Apriori`. Accessed: July 9, 2018.

[11] Muhammad Zeeshan Asghar, Paavo Nieminen, Seppo Hämäläinen, Tapani Ristaniemi, Muhammad Ali Imran, and Timo Hämäläinen. Towards proactive context-aware self-healing for 5g networks. *Computer Networks*, 128:5–13, 2017.

[12] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010.

[13] Lajos Bajzik, Csaba Deák, Tamás Kárász, Péter Szilágyi, Zoltán Vincze, and Csaba Vulkán. Qoe driven son for mobile backhaul demo. In *Personal, Indoor, and Mobile Radio Communications (PIMRC), 2016 IEEE 27th Annual International Symposium on*, pages 1–2. IEEE, 2016.

[14] Tobias Bandh, Raphael Romeikat, Henning Sanneck, and Haitao Tang. Policy-based coordination and management of son functions. In *Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium on*, pages 827–840. IEEE, 2011.

[15] Debasis Bandyopadhyay and Jaydip Sen. Internet of things: Applications and challenges in technology and standardization. *Wireless Personal Communications*, 58(1):49–69, 2011.

[16] M. Behringer, M. Pritikin, S. Bjarnason, A. Clemm, B. Carpenter, S. Jiang, and L. Ciavaglia. Autonomic networking: Definitions and design goals. RFC 7575, RFC Editor, June 2015.

[17] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

[18] Tim Berners-Lee, James Hendler, Ora Lassila, et al. The semantic web. *Scientific american*, 284(5):28–37, 2001.

[19] John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 120–128. Association for Computational Linguistics, 2006.

[20] David Booth, Francis McCabe, Eric Newcomer, Michael Champion, Chris Ferris, and David Orchard. Web Services Architecture. Technical report, W3C, 2004. Accessed: December 9, 2018.

[21] Anne-Marie Bosneag, Sidath Handurukande, and James O'Sullivan. Automatic discovery of sub-optimal radio performance in LTE RAN networks. In *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, pages 36–43, April 2016.

[22] Eliot Bytyçi, Lule Ahmedi, and Arianit Kurti. Association rule mining with context ontologies: An application to mobile sensing of water quality. In *Research Conference on Metadata and Semantics Research*, pages 67–78. Springer, 2016.

[23] Ángel Leonardo Valdivieso Caraguay and Luis Javier García Villalba. Monitoring and discovery for self-organized network management in virtualized and software defined networks. *Sensors*, 17(4):731, 2017.

[24] Evelin Cardoso, Ketyllen Silva, and Renato Francês. Intelligent handover procedure for heterogeneous lte networks using fuzzy logic. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2017 13th International*, pages 2163–2168. IEEE, 2017.

[25] Rich Caruana. Multitask learning. In *Learning to learn*, pages 95–133. Springer, 1998.

[26] Alberto Huertas Celdrán, Manuel Gil Pérez, Félix J García Clemente, and Gregorio Martínez Pérez. Policy-based management for green mobile networks through software-defined networking. *Mobile Networks and Applications*, pages 1–10, 2016.

[27] Yi-ting Chiang and Jane Yung-jen Hsu. Knowledge transfer in activity recognition using sensor profile. In *2012 9th International Conference on Ubiquitous Intelligence and Computing and 9th International Conference on Autonomic and Trusted Computing*, pages 180–187, Sept 2012.

[28] Gabriela F Ciocarlie, Christopher Connolly, Chih-Chieh Cheng, Ulf Lindqvist, Szabolcs Nováczki, Henning Sanneck, and Muhammad Naseer-ul Islam. Anomaly detection and diagnosis for automatic radio network verification. In *International Conference on Mobile Networks and Management*, pages 163–176. Springer, 2014.

[29] Gabriela F Ciocarlie, Chéri ta Corbett, Eric Yeh, Christopher Connolly, Henning Sanneck, Muhammad Naseer-Ul-Islam, Borislava Gajic, Szabolcs Nováczki, and Kimmo Hatonen. Diagnosis cloud: Sharing knowledge across cellular networks. In *Network and Service Management (CNSM), 2016 12th International Conference on*, pages 228–232. IEEE, 2016.

[30] Cisco. Cisco Visual Networking Index: Forecast and Methodology, 2016–2021 White Paper. Technical report, Cisco, 9 2017.

[31] William F Clocksin and Christopher S Mellish. *Programming in Prolog: Using the ISO standard*. Springer Science & Business Media, 2012.

[32] CogNet. CogNet – Building an Intelligent System of Insights and Action for 5G Network Management. `http://www.cognet.5g-ppp.eu`, 2015. Accessed: December 9, 2018.

[33] COMMUNE. The Commune (COgnitive network ManageMent under UNcErtainty)). `https://www.celticplus.eu/project-commune/`, 2011. Accessed: December 9, 2018.

[34] Paolo Corti, Benjamin Lewis, and Athanasios Tom Kralidis. Hypermap registry: an open source, standards-based geospatial registry and search platform. *Open Geospatial Data, Software and Standards*, 3(1):8, 2018.

[35] Li Da Xu, Wu He, and Shancang Li. Internet of things in industries: A survey. *IEEE Transactions on industrial informatics*, 10(4):2233–2243, 2014.

[36] Tony Daher, Sana Ben Jemaa, and Laurent Decreusefond. Cognitive management of self—organized radio networks based on multi armed bandit. In *Personal, Indoor, and Mobile Radio Communications (PIMRC), 2017 IEEE 28th Annual International Symposium on*, pages 1–5. IEEE, 2017.

[37] Tony Daher, Sana Ben Jemaa, and Laurent Decreusefond. Softwarized and distributed learning for son management systems. In *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–7. IEEE, 2018.

[38] Hal Daume III. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, 2007.

[39] Jorge E López De Vergara, Antonio Guerrero, Víctor A Villagrá, and Julio Berrocal. Ontology-based network management: study cases and lessons learned. *Journal of Network and Systems Management*, 17(3):234–254, 2009.

[40] Jorge E López de Vergara, Víctor A Villagrá, Carlos Fadón, Juan M González, José A Lozano, and Manuel Álvarez-Campana. An autonomic approach to offer services in osgi-based home gateways. *Computer Communications*, 31(13):3049–3058, 2008.

[41] Adela Del-Río-Ortega, Manuel Resinas, and Antonio Ruiz-Cortés. Defining process performance indicators: An ontological approach. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 555–572. Springer, 2010.

[42] Claudia Diamantini, Laura Genga, Domenico Potena, and Emanuele Storti. Collaborative building of an ontology of key performance indicators. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 148–165. Springer, 2014.

[43] Ioan Drăgan, Teodor-Florin Fortiş, Gabriel Iuhasz, Marian Neagul, and Dana Petcu. Applying self-* principles in heterogeneous cloud environments. In *Cloud Computing*, pages 255–274. Springer, 2017.

[44] ETSI. Network Functions Virtualisation – Introductory White Paper. Technical report, ETSI, 2016. Accessed: December 9, 2018.

[45] Jerry Alan Fails and Dan R Olsen Jr. Interactive machine learning. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 39–45. ACM, 2003.

[46] Liam Fallon, John Keeney, and Declan O'Sullivan. Applying semantics to optimize end-user services in telecommunication networks. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 718–726. Springer, 2014.

[47] Liam Fallon and Declan O'Sullivan. The aesop approach for semantic-based end-user service optimization. *IEEE Transactions on Network and Service Management*, 11(2):220–234, 2014.

[48] Jeroen Famaey and Filip De Turck. Federated management of the future internet: Status and challenges. *International Journal of Network Management*, 22(6):508–528, 2012.

[49] Cheng Fan, Fu Xiao, and Chengchu Yan. A framework for knowledge discovery in massive building automation data and its application in building diagnostics. *Automation in Construction*, 50:81 – 90, 2015.

[50] Kevin Feeney, Rob Brennan, John Keeney, Hendrik Thomas, Dave Lewis, Aidan Boran, and Declan O'Sullivan. Enabling decentralised management through federation. *Computer Networks*, 54(16):2825–2839, 2010.

[51] Filipe Ferreira, Ricardo Preto, Tiago Batista, Tiago Teixeira, Fabrizio Ippoliti, and et al. D7.1 Report and Prototypical Implementation of the Graphical User Interface. Technical report, SELFNET, 2017. Accessed: December 9, 2018.

[52] Open Networking Foundation. Software-Defined Networking: The New Norm for Networks. Technical report, ONF, 2012. Accessed: December 9, 2018.

[53] Christoph Frenzel. *Objective-Driven Operations of Self-Organizing Networks.* PhD thesis, University of Augsburg, Germany, 2016.

[54] Christoph Frenzel, Bernhard Bauer, Christoph Schmelz, and Henning Sanneck. Objective-driven coordination in self-organizing networks. In *Personal, Indoor, and Mobile Radio Communications (PIMRC), 2015 IEEE 26th Annual International Symposium on*, pages 1453–1458. IEEE, 2015.

[55] Christoph Frenzel, Simon Lohmüller, and Lars Christoph Schmelz. Dynamic, context-specific son management driven by operator objectives. In *Network Operations and Management Symposium (NOMS), 2014 IEEE*, pages 1–8. IEEE, 2014.

[56] Dario Götz and et al. Integrated son management - policy transformation and operational son coordination (final results). Technical report, EU FP7 project SEMAFOUR, 2016. Accessed: December 9, 2018.

[57] Zhipeng Gui, Chaowei Yang, Jizhe Xia, Kai Liu, Chen Xu, Jing Li, and Peter Lostritto. A performance, semantic and service quality-enhanced distributed search engine for improving geospatial resource discovery. *International Journal of Geographical Information Science*, 27(6):1109–1132, 2013.

[58] Amelie Gyrard, Martin Serrano, and Ghislain A Atemezing. Semantic web methodologies, best practices and ontology engineering applied to internet of things. In *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*, pages 412–417. IEEE, 2015.

[59] Sören Hahn, Michael Schweins, and Thomas Kürner. Impact of son function combinations on the kpi behaviour in realistic mobile network scenarios. In *Wireless Communications and Networking Conference Workshops (WCNCW), 2018 IEEE*, pages 1–6. IEEE, 2018.

[60] Michael Hahsler, Bettina Grün, and Kurt Hornik. arules - a computational environment for mining association rules and frequent item sets. *Journal of Statistical Software, Articles*, 14(15):1–25, 2005.

[61] Seppo Hämäläinen, Henning Sanneck, and Cinzia Sartori. *LTE Self-Organising Networks (SON): Network Management Automation for Operational Efficiency*. Wiley Online Library, 1st edition, 2012.

[62] Andreas Heß and Nicholas Kushmerick. Learning to attach semantic metadata to web services. In *International Semantic Web Conference*, pages 258–273. Springer, 2003.

[63] Alan R Hevner, Salvatore T March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *MIS Quarterly*, 28(1):75–105, 2004.

[64] Ralph Hodgson, Paul J Keller, Jack Hodges, and Jack Spivak. Qudt-quantities, units, dimensions and data types ontologies. *USA Available http://qudt. org March*, 2014.

[65] Paul Horn. Autonomic computing: IBM's Perspective on the State of Information Technology, 2001.

[66] Ian Horrocks et al. Daml+oil: A description logic for the semantic web. *IEEE Data Eng. Bull.*, 25(1):4–9, 2002.

[67] Ian Horrocks, Peter F Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, Mike Dean, et al. Swrl: A semantic web rule language combining owl and ruleml. *W3C Member submission*, 21:79, 2004.

[68] Ovidiu-Constantin Iacoboaiea, Berna Sayrac, Sana Ben Jemaa, and Pascal Bianchi. Son coordination in heterogeneous networks: A reinforcement learning framework. *IEEE Transactions on Wireless Communications*, 15(9):5835–5847, 2016.

[69] Ali Imran, Ahmed Zoha, and Adnan Abu-Dayya. Challenges in 5g: how to empower son with big data for enabling 5g. *IEEE Network*, 28(6):27–33, Nov 2014.

[70] Brendan Jennings, Rob Brennan, William Donnelly, Simon N Foley, Dave Lewis, Declan O'Sullivan, John Strassner, and Sven van der Meer. Challenges for federated, autonomic network management in the future internet. In *Integrated Network Management-Workshops, 2009. IM'09. IFIP/IEEE International Symposium on*, pages 87–92. IEEE, 2009.

[71] Brendan Jennings, Sven Van Der Meer, Sasitharan Balasubramaniam, Dmitri Botvich, Mícheál Ó Foghlú, William Donnelly, and John Strassner. Towards autonomic management of communications networks. *IEEE Communications Magazine*, 45(10):112–121, 2007.

[72] Wei Jiang, Mathias Strufe, and Hans D Schotten. A son decision-making framework for intelligent management in 5g mobile networks. In *Computer and Communications (ICCC), 2017 3rd IEEE International Conference on*, pages 1158–1162. IEEE, 2017.

[73] Ljupco Jorguseski, Adrian Pais, Fredrik Gunnarsson, Angelo Centonza, and Colin Willcock. Self-organizing networks in 3gpp: standardization and future trends. *IEEE Communications Magazine*, 52(12):28–34, 2014.

[74] John Keeney, Liam Fallon, Wei Tai, and Declan O'Sullivan. Towards composite semantic reasoning for realtime network management data enrichment. In *Network and Service Management (CNSM), 2015 11th International Conference on*, pages 246–250. IEEE, 2015.

[75] Jeffrey O Kephart and David M Chess. The vision of autonomic computing. *Computer*, 36(1):41–50, 2003.

[76] Ashiq Khan, Wolfgang Kellerer, Kazuyuki Kozu, and Masami Yabusaki. Network sharing in the next mobile network: TCO reduction, management flexibility, and operational independence. *IEEE Communications Magazine*, 49(10):134–142, 2011.

[77] Nikolaos Koutsouris, Kostas Tsagkaris, Panagiotis Demestichas, Zwi Altman, Richard Combes, Pierre Peloso, Laurent Ciavaglia, Lefteris Mamatas, Stuart Clayman, and Alex Galis. Conflict free coordination of son functions in a unified management framework: Demonstration of a proof of concept prototyping platform. In *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, pages 1092–1093. IEEE, 2013.

[78] Todd Kulesza, Margaret Burnett, Weng-Keen Wong, and Simone Stumpf. Principles of explanatory debugging to personalize interactive machine learning. In *Proceedings of the 20th international conference on intelligent user interfaces*, pages 126–137. ACM, 2015.

[79] An Ngoc Lam and Øystein Haugen. Supporting iot semantic interoperability with autonomic computing. In *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*, pages 761–767. IEEE, 2018.

[80] Jorge Lanza, Luis Sanchez, David Gomez, Tarek Elsaleh, Ronald Steinke, and Flavio Cirillo. A proof-of-concept for semantically interoperable federation of iot experimentation facilities. *Sensors*, 16(7):1006, 2016.

[81] Li Li and Subin Shen. End-to-end qos performance management across lte networks. In *Network Operations and Management Symposium (APNOMS), 2011 13th Asia-Pacific*, pages 1–4. IEEE, 2011.

[82] Elmars Lipenbergs, Alina Stafecka, Girts Ivanovs, and Inga Smirnova. Quality of service measurements and service mapping for the mobile internet access. In *2017 Progress In Electromagnetics Research Symposium - Spring (PIERS)*, pages 2526–2532, May 2017.

[83] Simon Lohmüller, Andreas Eisenblatter, Christoph Frenzel, Dario Gotz, Soren Hahn, Thomas Kurner, Remco Litjens, Andreas Lobinger, Bart Sas, Lars Christoph Schmelz, et al. Policy-based son management demonstrator. In *Vehicular Technology Conference (VTC Spring), 2015 IEEE 81st*, pages 1–2. IEEE, 2015.

[84] Simon Lohmüller, Fabian Rabe, Andrea Fendt, Bernhard Bauer, and Lars Christoph Schmelz. Son function performance prediction in a cognitive son management system. In *Wireless Communications and Networking Conference Workshops (WCNCW), 2018 IEEE*, pages 13–18. IEEE, 2018.

[85] Simon Lohmüller, Lars Christoph Schmelz, and Sören Hahn. Adaptive SON management using KPI measurements. In *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, pages 625–631, April 2016.

[86] Francesco Malandrino, Carla-Fabiana Chiasserini, and Scott Kirkpatrick. Understanding the present and future of cellular networks through crowdsourced traces. In *A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2017 IEEE 18th International Symposium on*, pages 1–9. IEEE, 2017.

[87] J Manco-Vásquez, S Van Vaerenbergh, J Vía, and Ignacio Santamaría. Kernel canonical correlation analysis for robust cooperative spectrum sensing in cognitive radio networks. *Transactions on Emerging Telecommunications Technologies*, 28(1), 2017.

[88] Christian Mannweiler, Lars Christoph Schmelz, Simon Lohmüller, and Bernhard Bauer. Cross-domain 5g network management for seamless industrial communications. In *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, pages 868–872, April 2016.

[89] Salvatore T March and Gerald F Smith. Design and natural science research on information technology. *Decision support systems*, 15(4):251–266, 1995.

[90] Nathan Marz and James Warren. *Big Data: Principles and best practices of scalable real-time data systems*. New York; Manning Publications Co., 2015.

[91] Elena Meshkova, Zhou Wang, Krisakorn Rerkrai, Junaid Ansari, Jad Nasreddine, Daniel Denkovski, Tim Farnham, Janne Riihijärvi, Liljana Gavrilovska, and Petri Maehoenen. Designing a self-optimization system for cognitive wireless home networks. *IEEE Transactions on Cognitive Communications and Networking*, 3(4):684–702, 2017.

[92] Rashid Mijumbi, Joan Serrat, Juan-Luis Gorricho, Steven Latré, Marinos Charalambides, and Diego Lopez. Management and orchestration challenges in network functions virtualization. *IEEE Communications Magazine*, 54(1):98–105, 2016.

[93] Boris Motik, Peter F Patel-Schneider, Bijan Parsia, Conrad Bock, Achille Fokoue, Peter Haase, Rinke Hoekstra, Ian Horrocks, Alan Ruttenberg, Uli Sattler, et al. Owl 2 web ontology language: Structural specification and functional-style syntax. *W3C recommendation*, 27(65):159, 2009.

[94] Zeinab Movahedi, Mouna Ayari, Rami Langar, and Guy Pujolle. A survey of autonomic network architectures and evaluation criteria. *IEEE Communications Surveys & Tutorials*, 14(2):464–490, 2012.

[95] Jessica Moysen and Lorenza Giupponi. From 4g to 5g: Self-organized network management meets machine learning. *Computer Communications*, 2018.

[96] P Muñoz, Raquel Barco, and Isabel de la Bandera. Load balancing and handover joint optimization in lte networks using fuzzy logic and reinforcement learning. *Computer Networks*, 76:112–125, 2015.

[97] Pablo Muñoz, Raquel Barco, and Isabel de la Bandera. On the potential of handover parameter optimization for self-organizing networks. *IEEE Transactions on Vehicular Technology*, 62(5):1895–1905, 2013.

[98] Pedro Neves, Rui Calé, Mário Rui Costa, Carlos Parada, Bruno Parreira, Jose Alcaraz-Calero, Qi Wang, James Nightingale, Enrique Chirivella-Perez, Wei Jiang, et al. The selfnet approach for autonomic management in an nfv/sdn networking paradigm. *International Journal of Distributed Sensor Networks*, 12(2):2897479, 2016.

[99] NGMN. Ngmn recommendation on son and o&m requirements. Technical report, NGMN, 2008. Accessed: December 9, 2018.

[100] NGMN. Recommended Practices for Multi-vendor SON Deployment. *Next Generation Mobile Networks, White paper*, 2013.

[101] NGMN. 5G white paper. *Next Generation Mobile Networks, White paper*, 2015.

[102] Ewan S Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954.

[103] Sinno Jialin Pan, Ivor W. Tsang, James T. Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, Feb 2011.

[104] Mike P Papazoglou. Service-oriented computing: Concepts, characteristics and directions. In *Web Information Systems Engineering, 2003. WISE 2003. Proceedings of the Fourth International Conference on*, pages 3–12. IEEE, 2003.

[105] Ken Peffers, Marcus Rothenberger, Tuure Tuunanen, and Reza Vaezi. Design science research evaluation. In *Proceedings of the 7th International Conference on Design Science Research in Information Systems: Advances in Theory and Practice*, DESRIST'12, pages 398–410, Berlin, Heidelberg, 2012. Springer-Verlag.

[106] Ken Peffers, Tuure Tuunanen, Marcus A Rothenberger, and Samir Chatterjee. A design science research methodology for information systems research. *Journal of management information systems*, 24(3):45–77, 2007.

[107] Viara Popova and Alexei Sharpanskykh. Modeling organizational performance indicators. *Information systems*, 35(4):505–527, 2010.

[108] A. Pras, J. Schonwalder, M. Burgess, O. Festor, G. M. Perez, R. Stadler, and B. Stiller. Key research challenges in network management. *IEEE Communications Magazine*, 45(10):104–110, October 2007.

[109] Vilho Räisänen and Haitao Tang. Knowledge modeling for conflict detection in self-organized networks. In *International Conference on Mobile Networks and Management*, pages 107–119. Springer, 2011.

[110] Deepa Ramachandran. Son will define the future with 5g, 2016. Accessed: December 9, 2018.

[111] Carlos Ramirez-Perez and Victor Ramos. Sdn meets sdr in self-organizing networks: fitting the pieces of network management. *IEEE Communications Magazine*, 54(1):48–57, 2016.

[112] Annie Ibrahim Rana and Brendan Jennings. Semantic aware processing of user defined inference rules to manage home networks. *Journal of Network and Computer Applications*, 79:68–87, 2017.

[113] João Reis, Rui Pinto, and Gil Gonçalves. Human-centered application using cyber-physical production system. In *Industrial Electronics Society, IECON 2017-43rd Annual Conference of the IEEE*, pages 8634–8639. IEEE, 2017.

[114] Matthew Richardson and Pedro Domingos. Markov Logic Networks. *Machine Learning*, 62(1-2):107–136, February 2006.

[115] Hajo Rijgersberg, Mark van Assem, and Jan Top. Ontology of units of measure and related concepts. *Semantic Web*, 4(1):3–13, 2013.

[116] Fernando Roda and Estanislao Musulin. An ontology-based framework to support intelligent data analysis of sensor measurements. *Expert Systems with Applications*, 41(17):7914–7926, 2014.

[117] Marko A Rodriguez, Johan Bollen, and Herbert Van De Sompel. Automatic metadata generation using associative networks. *ACM Transactions on Information Systems (TOIS)*, 27(2):7, 2009.

[118] Charalampos Rotsos, Daniel King, Arsham Farshad, Jamie Bird, Lyndon Fawcett, Nektarios Georgalas, Matthias Gunkel, Kohei Shiomoto, Aijun Wang, Andreas Mauthe, et al. Network service orchestration standardization: A technology survey. *Computer Standards & Interfaces*, 54:203–215, 2017.

[119] Rundfunk und Telekom Regulierungs-GmbH (RTR). RTR Nettest measurement platform. urlhttps://www.netztest.at/en/. Accessed: July 9, 2018.

[120] Daniel M Russell, Paul P Maglio, Rowan Dordick, and Chalapathy Neti. Dealing with ghosts: Managing the user experience of autonomic computing. *IBM Systems Journal*, 42(1):177–188, 2003.

[121] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Prentice Hall, 1995.

[122] Doruk Sahinel, Cem Akpolat, Manzoor A Khan, Fikret Sivrikaya, and Sahin Albayrak. Beyond 5g vision for iolite community. *IEEE Communications Magazine*, 55(1):41–47, 2017.

[123] Mazeiar Salehie and Ladan Tahvildari. Self-adaptive software: Landscape and research challenges. *ACM transactions on autonomous and adaptive systems (TAAS)*, 4(2):14, 2009.

[124] Nancy Samaan and Ahmed Karmouch. Towards autonomic network management: an analysis of current and future research directions. *IEEE Communications Surveys & Tutorials*, 11(3), 2009.

[125] Zohreh Sanaei, Saeid Abolfazli, Abdullah Gani, and Rajkumar Buyya. Heterogeneity in mobile cloud computing: taxonomy and open challenges. *IEEE Communications Surveys & Tutorials*, 16(1):369–392, 2014.

[126] Lars Christoph Schmelz, Andreas Eisenblätter, Sören Hahn, Thomas Kürner, Remco Litjens, Andreas Lobinger, Simon Lohmüller, Bart Sas, and Ulrich Türke. Demonstrator for objective driven son operation. In *Wireless Communications Systems (ISWCS), 2014 11th International Symposium on*, pages 506–507. IEEE, 2014.

[127] SELFNET. Selfnet: A framework for self-organized network management in virtualized and software defined networks. `https://selfnet-5g.eu/`. Accessed: December 9, 2018.

[128] SEMAFOUR. The FP7 SEMAFOUR (Self-Management for Unified Heterogeneous Radio Access Networks). `http://www.fp7-semafour.eu/`, 2012. Accessed: December 9, 2018.

[129] Sameerkumar Sharma, Raymond Miller, and Andrea Francini. A cloud-native approach to 5g network slicing. *IEEE Communications Magazine*, 55(8):120–127, 2017.

[130] Sukhpal Singh and Inderveer Chana. Qos-aware autonomic resource management in cloud computing: A systematic review. *ACM Comput. Surv.*, 48(3):42:1–42:46, December 2015.

[131] SOCRATES. The SOCRATES (Self-Optimisation and self-ConfiguRATion in wirelEss networkS). `http://www.fp7-socrates.eu/`, 2008. Accessed: December 9, 2018.

[132] Marco Antonio Sotelo Monge, Jorge Maestre Vidal, and Luis Javier García Villalba. Reasoning and knowledge acquisition framework for 5g network analytics. *Sensors*, 17(10):2405, 2017.

[133] Fabio Souza, Tarcisio Coutinho, and Nelson Rosa. Monitoring solution in a dynamic service-oriented platform. *Computers & Electrical Engineering*, 63:289 – 307, 2017.

[134] John Strassner. *Policy-based network management: solutions for the next generation*. Elsevier, 2003.

[135] John Strassner and Wael William Diab. A semantic interoperability architecture for internet of things data sharing and computing. In *Internet of Things (WF-IoT), 2016 IEEE 3rd World Forum on*, pages 609–614. IEEE, 2016.

[136] Xiang Su, Pingjiang Li, Jukka Riekki, Xiaoli Liu, Jussi Kiljander, Juha-Pekka Soininen, Christian Prehofer, Huber Flores, and Yuhong Li. Distribution of semantic reasoning on the edge of internet of things. In *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 1–9. IEEE, 2018.

[137] Haitao Tang, Kaj Stenberg, Vilho Räisänen, and Joonas Kylliäinen. Avoidance of improper configuration based on self-operation. In *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–8. IEEE, 2018.

[138] Telus and Huawei. White paper: Next generation SON for 5G. Technical report, Telus and Huawei, 2016. Accessed: December 9, 2018.

[139] The European Commission. Mapping of Broadband Services in Europe. `https://ec.europa.eu/digital-single-market/en/broadband-and-infrastructure-mapping-project`. Accessed: July 9, 2018.

[140] Pedro Torres, Paulo Marques, Hugo Marques, Rogério Dionísio, Tiago Alves, Luis Pereira, and Jorge Ribeiro. Data analytics for forecasting cell congestion on lte networks. In *Network Traffic Measurement and Analysis Conference (TMA), 2017*, pages 1–6. IEEE, 2017.

[141] Kostas Tsagkaris and et. al. Unified management framework (umf) specifications release 2. *Deliverable d2.4*, 2, 2012.

[142] Kostas Tsagkaris and et. al. Unified management framework (umf) specifications release 3. *Deliverable d2.2*, 4, 2013.

[143] Kostas Tsagkaris, Panagiotis Vlacheas, George Athanasiou, Vera Stavroulaki, Stanislav Filin, Hiroshi Harada, Jens Gebert, and Markus Mueck. Autonomics in wireless network management: Advances in standards and further challenges. *IEEE Network*, 25(6), 2011.

[144] Tsvetko Tsvetkov, Janne Ali-Tolppa, Henning Sanneck, and Georg Carle. A minimum spanning tree-based approach for reducing verification collisions in self-organizing networks. In *Network Operations and Management Symposium (NOMS), 2016 IEEE/IFIP*, pages 137–145. IEEE, 2016.

[145] Tsvetko Tsvetkov, Janne Ali-Tolppa, Henning Sanneck, and Georg Carle. Verification of configuration management changes in self-organizing networks. *IEEE Trans. Network and Service Management*, 13(4):885–898, 2016.

[146] Univerself. UniverSelf, realizing autonomics for Future Networks. `http://www.univerself-project.eu`, 2010. Accessed: December 9, 2018.

[147] Mohammad Hadi Valipour, Bavar AmirZafari, Khashayar Niki Maleki, and Negin Daneshpour. A brief survey of software architecture concepts and service oriented architecture. In *Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on*, pages 34–38. IEEE, 2009.

[148] Sven van der Meer, John Keeney, and Liam Fallon. Dynamically adaptive policies for dynamically adaptive telecommunications networks. In *Network and Service Management (CNSM), 2015 11th International Conference on*, pages 182–186. IEEE, 2015.

[149] Sven van der Meer, John Keeney, and Liam Fallon. 5g networks must be autonomic! In *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–5. IEEE, 2018.

[150] Tim van Kasteren, Gwenn Englebienne, Ben JA Kröse, et al. Recognizing activities in multiple contexts using transfer learning. In *AAAI Fall Symposium: AI in Eldercare: New Solutions to Old Problems*, 2008.

[151] Mariano Vargas-Santiago, Luis Morales-Rosales, Saul Pomares-Hernandez, and Khalil Drira. Autonomic web services enhanced by asynchronous checkpointing. *IEEE Access*, 6:5538–5547, 2018.

[152] W3C. Wsmo-lite: Lightweight semantic descriptions for services on the web, w3c recommendation, world wide web consortium, tech. rep., aug. 2010. *Available: https://www.w3.org/Submission/WSMO-Lite/ (accessed March 2017)*, 2010.

[153] Yi Wei and M Brian Blake. Service-oriented computing and cloud computing: Challenges and opportunities. *IEEE Internet Computing*, 14(6):72–75, 2010.

[154] Chen Wen-Hui, Cho Po-Chuan, and Yong-Lin Jiang. Activity recognition using transfer learning. *Sensors and Materials*, 29(7):897–904, 2017.

[155] Ryen W White and Resa A Roth. Exploratory search: Beyond the query-response paradigm. *Synthesis lectures on information concepts, retrieval, and services*, 1(1):1–98, 2009.

[156] Qihui Wu, Guoru Ding, Yuhua Xu, Shuo Feng, Zhiyong Du, Jinlong Wang, and Keping Long. Cognitive internet of things: A new paradigm beyond connection. *IEEE Internet of Things Journal*, 1(2):129–143, April 2014.

[157] Sheng Wu, Min Zhang, Qiaojia Huang, Yannan Zhang, Cheng Wan, Kaixuan Zhang, Jun Cao, Zhipeng Gui, and Kun Qin. Design a web portal for visualizing and exploring service quality of global ogc web map services. In *2015 23rd International Conference on Geoinformatics*, pages 1–5. IEEE, 2015.

[158] Fengli Xu, Yuyun Lin, Jiaxin Huang, Di Wu, Hongzhi Shi, Jeungeun Song, and Yong Li. Big data driven mobile traffic understanding and forecasting: A time series approach. *IEEE transactions on services computing*, 9(5):796–805, 2016.

[159] Jun Xu, Junmei Yao, Lu Wang, Kaishun Wu, Lei Chen, and Wei Lou. Revolution of self-organizing network for 5g mmwave small cell management: From reactive to proactive. *IEEE Wireless Communications*, 25(4):66–73, 2018.

[160] Barbaros Yet, Zane B. Perkins, Nigel R. M. Tai, and D. William R. Marsh. Clinical evidence framework for bayesian networks. *Knowledge and Information Systems*, 50(1):117–143, Jan 2017.

[161] Zhongliang Zhao, Eryk Schiller, Eirini Kalogeiton, Torsten Braun, Burkhard Stiller, Mevlut Turker Garip, Joshua Joy, Mario Gerla, Nabeel Akhtar, and Ibrahim Matta. Autonomic communications in software-driven networks. *IEEE Journal on Selected Areas in Communications*, 35(11):2431–2445, 2017.

[162] Hai-Tao Zheng, Bo-Yeong Kang, and Hong-Gee Kim. An Ontology-Based Bayesian Network Approach for Representing Uncertainty in Clinical Practice Guidelines. In PauloCesarG. da Costa, Claudia d'Amato, Nicola Fanizzi, KathrynB. Laskey, KennethJ. Laskey, Thomas Lukasiewicz, Matthias Nickles, and Michael Pool, editors, *Uncertainty Reasoning for the Semantic Web I*, volume 5327 of *Lecture Notes in Computer Science*, pages 161–173. Springer Berlin Heidelberg, 2008.

[163] Qianru Zhou, Cheng-Xiang Wang, Stephen McLaughlin, and Xiaotian Zhou. Network virtualization and resource description in software-defined wireless networks. *IEEE Communications Magazine*, 53(11):110–117, 2015.

[164] Miguel Ángel Rodríguez-García, Rafael Valencia-García, Francisco García-Sánchez, and J. Javier Samper-Zapater. Ontology-based annotation and retrieval of services in the cloud. *Knowledge-Based Systems*, 56:15 – 25, 2014.

Publication I

Vilho Räisänen and Kasper Apajalahti. Reasoning in Agent-Based Network Management. In *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, IEEE, Taipei, Taiwan, pages 1–7, DOI https://doi.org/10.1109/NOMS.2018.8406171, April 2018.

© 2018 IEEE

Reprinted with permission.

Reasoning in agent-based network management

Vilho Räisänen
Nokia Bell Labs
Espoo, Finland
Email: first.last@nokia-bell-labs.com

Kasper Apajalahti
Department of Computer Science
Aalto University
Espoo, Finland
Email: first.last@aalto.fi

***Abstract*—An increasing complexity of mobile networks and use cases is reflected in requirements for network management. Advanced automation is required to address this challenge in an economically feasible manner. We describe a system based on agent mapping as a platform for automation for network management in 5G networks and beyond. We use classical and probabilistic reasoning for composing solutions to complex requests by means of relatively simple software agents. We describe different variants of the approach in terms of capabilities, ranging from triple store only to system with semantic and probabilistic reasoning functionalities. This approach provides flexibility for network functionality evolution and facilitates software reuse and is compatible with the use of task-specific machine learning algorithms in network management agents. We describe test system used for evaluating the concept, as well as use case evaluation obtained with it.**

I. INTRODUCTION

Goals for improvement of performance in 5G networks over previous systems have been laid out in [1]. The implementation of the requirements lead to architectural choices in network architecture. On the one hand, the network architecture is cloud-based, with 3GPP applications executed as virtualized applications [2]. On the other hand, the 5G network deployment is expected to increase technological complexity, highlighting the need for advanced automation. Furthermore, simultaneous support for 5G traffic types — massive machine type communications (mMTC), critical machine type communication (cMTC), and extreme broadband (xMBB) — requires new technologies for 5G access, which in most cases needs to co-exist with legacy access technologies. The role of network management needs to be re-assessed in the new architecture.

It is expected that new technologies in 5G such as network slices [3], [4] serve as a platform for new services. Taken together with the increasing complexity of radio access, flexibility is needed for network management since future needs cannot be fully predicted. The paradigms for network management have evolved from manual and template-based management towards agent-based management in Long-Term Evolution (LTE) Self-Organizing Networks (SON) paradigm [5], [6], [8]. Agents are a realization of autonomic computing concept for mobile network management [7]. The limitations of rule-based SON have become apparent in terms of feasibility of tailoring to varying network contexts.

In 5G, cloud-based execution is an integral part of the architecture, and configuration targets for NM are partly virtualized [3]. As a consequence of virtual execution environments in 5G, there is an interplay between orchestration of virtualized resources and NM. For example, orchestration affects the set of virtual functionalities managed by Network Management (NM). In [9], a framework was proposed for virtualized NM which lends itself to an analysis of interactions of agent-based network management for such a case.

In this article, we describe an architecture based on agent composition for NM. The composition in our approach makes use of classical reasoning based on semantic models. We argue that this approach facilitates agile development of network management capabilities while supporting efficiency in terms of software reuse.

In what follows, we discuss the role of agents in operability, followed by an account of the use of reasoning in the same area. We then proceed to describe our approach and a demonstrator environment and use case evaluation. We conclude this article with a summary.

We shall describe relevant prior references within the following technologies Sections in the interests of compactness of presentation and understandability.

II. REASONING IN NETWORK MANAGEMENT

Automated reasoning based on First Order Logic (FOL) can be used to partially replace traditional programming, with the advantage that the consistency of the logical models employed are automatically checked. This lessens the need for testing in validation of proper functioning of a module. The use of logical models such as Description Logic -derived web ontology language (OWL) allows expressing computations with constructs akin to domain models, more understandable for humans than software implementations.

The scaling properties of reasoning algorithms are well understood, and pose limits for application in NM area. It is not feasible to represent the entire state of a mobile network in a single logical model for management purposes. For focused uses such as semantic modelling of configuration management [13], [14], mapping between concepts across domains [12], or analysis of agent coordination [10], classical reasoning is a valid choice. An architecture for knowledge delivery for the purposes of semantic interoperability of autonomic agents has been proposed in [15].

Earlier we noted that rule-based agents are not feasible for tailoring of NM to individual cell contexts. The use of a suitable variant of Case-Based Reasoning [16] allows for interpolation between cases, but is still limited by the ”space” spanned by the case base. In view of the increasing complexity of networking technology, one should allow for the choice of a machine learning (ML) algorithm that is appropriate for the data at hand. Several ML algorithms may thus be used concurrently in agents.

The use of a set of machine learning algorithms in agents gives rise to semantic challenge in integrating the output of the learning agents to NM. Another challenge is related to knowledge of network state. For example, what-if analyses might be useful for human users given a set of information about network which may be incomplete. It can be argued that a combination of semantic reasoning and probabilistic reasoning addresses both of these problems [11]. Semantic modelling lends itself well to mapping between sets of concepts. Probabilistic reasoning complements semantic reasoning with ability to resolve conflicting information. Combining the two technologies, a mapping within an ontology can be achieved for concepts specific to machine learning algorithms.

Probabilistic reasoning is employed to establish the most likely explanation for the set of input information at hand — possibly contradictory – in the context of a domain model [11]. The most likely explanation can then be used in classical reasoning. The consistency of the output of probabilistic reasoning with classical reasoning can be ensured by using a method such a Markov Logic Network (MLN, [17]) which supports both a combination of certain and uncertain rules (latter ones associated with weights).

III. REASONING IN AGENT COMPOSITION

In this section, we describe our reasoning system that dynamically composes agents to perform NM tasks (requests). We use triple store (graph database) with SPARQL query language, Description Logic (DL) reasoner, and probabilistic reasoning providing additional capabilities. This kind of system allows for inference over historical data which is represented as triples, with additional capabilities providing further functionality which is described later on. Such inference can be used in composing a solution to a request by means of agents. The triple store reasoning step itself replaces some traditional programming as we shall see later on.

Processing based on data included in requests is enabled by a reasoner which uses an ontology to infer further triples based on request data. As we shall see later on, probabilistic reasoning can be employed *en route* to accommodate hypothetical information such as output of machine learning algorithms.

The triple store can be viewed as a realization of an ontology, which in the ”triple store only” case does not exist as a separate entity. When reasoning is used, it employs a formal ontology describing the knowledge model used for reasoning and infers implies triples. In our case, the main ingredient of ontology is domain model which ensures consistency of reasoning in NM.

A. Description of the approach

The software architecture is shown in Figure 1. Requests are received by a service interface, processed by a composition entity, and mapped to execution in $[0, N]$ agents. The ”zero agents” case corresponds to performing all related reasoning in a combination composition entity and triple store query. The mapping could be performed by means of a rule base, but we argue that semantic modelling and reasoning provides more flexibility compared to static rule bases.

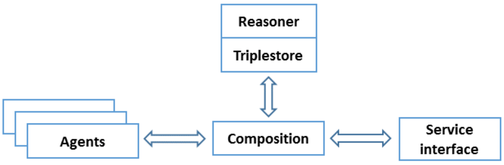


Fig. 1. Mapping architecture overview. Composition receives requests from service interface and uses query to triple store in agent mapping. The contents of triple store may be updated with triples induced by reasoner.

The core concept in our ontological approach is mapping of incoming requests to agents via operations and effects. Figure 2 describes the simplified ontology used for this mapping. The idea is adapted from a simple semantic web service model, WSMO-lite [20]. An agent — analogous to a service in WSMO-lite — has $[1, N]$ operations that monitor or change the status of the target. Operations have effects that represents the desired impact in the target. Furthermore, operations have metadata; for example, operation area (the part of the network where the agent is operating) and temporal range.

Fig. 2. Ontology constructs for agent (top), request (middle), and network measurements (bottom).

Inference over historical data only does not require the use of reasoner, and SPARQL query is sufficient. This correponds to a case where the query does not include new data instances, only parameters via which to query the existing data. Such parameters can be e.g. network scope (set of cells) or temporal range. In this approach, reasoner is not involved in processing the query, but may have been used when relevant data have arrived for inferring new triples.

From the viewpoint of knowledge model, a request including parameters corresponds to one or more instances that activate the OWL reasoner when added to the ontology. The Figure 2 shows a case in which the request is associated with effects in the ontology, reflecting concrete objectives that need to be met with the operations. The request instance is associated with a network scope (relevant cells) and point in time. The request may also have pre- and post conditions that need to be met with, such as a cell loading level. An example of such a request in LTE could be improvement of service quality in a set of cells, which would be associated with the effect of changes in CQI distribution.

In earlier research, Web service execution environment for creation and execution of semantic web services based on ontology [18]. In another article, Distributed Management Task Force (DMTF) Common Information Model (CIM) is "lifted" to OWL and used for service composition with OWL-S [19]. Compared to these approaches, our system is not based on WS-* web services suite and takes a minimalist approach to ontology. As we shall see, the construction of knowledge model is driven by relevant network information and use cases.

The OWL reasoner generates triples by using class model and instances contained within request. By using semantic reasoning together with probabilistic reasoning, we can extend the capabilities of the system to inconsistent and hypothetical information, useful for linking to machine learning algorithms. This requires structured approach to interaction of semantic and probabilistic reasoning.

In our approach, the input model of probabilistic reasoning supports directly statements from the domain model. This is supported by the syntax of MLN, where ontological facts can be represented using logical syntax together with statement the likelihood of which is to be evaluated. The output of probabilistic reasoning is thus guaranteed to be compatible with domain model, and can be added to ontology. A test system combining semantic reasoner and MLN is described in Section IV.

B. Effect modelling and operation-request mapping

We shall next describe an example of how ontology constructs defined above (Figure 2) are used [12]. The approach allows linking effects of operations and requests in the ontology to each other by means of human-defined Key Performance Indicator (KPI) mappings and network measurement effects (Figure 2). The mappings provide dependencies (correlations) between KPI effects, a crucial part of the ontology for mapping. For example, effect dependencies might be used to infer that two functions, that are monitoring thresholds of different KPIs, are both associated with the same goal, since the KPI effects are positively correlated (increasing or decreasing simultaneously). Thus, both of the functions would be valid solutions to a request having similar effect as an objective. More details about the effect modelling is defined in [12].

As the Figure 2 depicts, the reasoner uses effects of agent operations, incoming requests, and network measurements. Using these as input, reasoner first performs inference to find relationships between effects [12]. Next, reasoner maps operations to requested effects with Equation 1. The rule is a Horn clause like rule written in Semantic Web Rule Language (SWRL) and is supported by the OWL reasoner [21]. The rule indicates that if the effect of an operation ($?oe$) is dependent on the effect of a request ($?re$), the operation satisfies (is able to produce) $?re$.

$$\begin{aligned} Operation(?op) \sqcap Request(?req) \sqcap \\ hasEffect(?op, ?oe) \sqcap hasEffect(?req, ?re) \sqcap \\ hasDependency(?re, ?oe) \Rightarrow satisfies(?op, ?re) \end{aligned} \tag{1}$$

Eventually, the rule binds relevant operations into effects of a request. An effect in the request may be bound with multiple operations that produce the effect. Analogously, a particular operation may satisfy several effects in the request. Let us consider an example in which user request is mapped to effects for increasing a KPI value (effect re_1), monitoring the result (re_2), and rollback of the operation (re_3) in case the re_1 is not achieved. The ontology may contain operations that fulfil one of the effects or complex operations that fulfil multiple requested effects, such as a function that executes a configuration and outputs a boolean value whether the objective is achieved. Table II illustrates possible operations mapped to the the described request. As it can be seen, five operations are mapped to one or several effects of the request.

Operation	Satisfies effect
op_1	re_1
op_2	re_2
op_3	re_1, re_2
op_4	re_3
op_5	re_1, re_2, re_3

TABLE I
OPERATIONS MAPPED TO ONE OR SEVERAL EFFECTS OF A REQUEST.

The operation-request mappings may now be processed to obtain combinations of operations that fulfil the whole request. With SPARQL queries, we generate a list of responses that satisfy the request. In the given example, we get three sets of composed operations: $\{\{op_1, op_2, op_4\} \{op_3, op_4\} \{op_5\}\}$.

With multiple responses for a request, a ranking method is needed to select the best response. We address this issue by allowing classification of effects as primary or secondary effects. A primary effect is analysed and scored, whereas a secondary effect only needs to be executable. In the earlier example, an obvious primary effect would be re_1 as it is the actual objective of the request. Operations that are mapped to primary effects may be analysed by using historical data of similar operations in order to define the success ratios of the corresponding operations. The success ratios can then be used to determine the best response for a request.

Based on the operation analysis, the final step is executing the adequate operations of agents with specific sets of parameters.

C. Agents in network management

We shall next discuss our approach from the viewpoint of using agents in network management.

An important advantage of the proposed approach is flexibility. If there is a change either in the NM capabilities or telecommunications functionalities managed by NM, it can be reflected automatically in composition, provided that it has been described in the domain model. Similarly, up-to-date

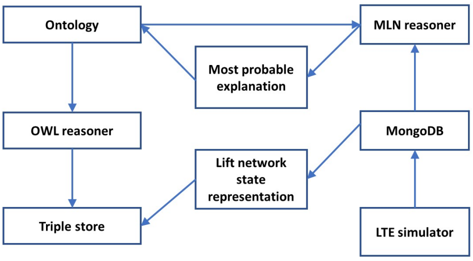


Fig. 3. Information flows in test system. For simplicity SON function and information bus have been omitted from the figure. As discussed in the text, lifting network state directly and importing it into ontology ABox via MLN are mechanisms which may be used depending on the use case.

network status can be used in composition with reasoning and not just in the behaviour of individual agents.

The agents used in composition can be traditional complex SON-style agents, microservices for composing ”virtual” agents, or a mixture of the two types [9]. The ontology model for composition assumes metadata about agents [22]. It is not necessary to semantically model the functioning of the agents, only the operations they provide.

The stand-alone agents or software modules used in composition of virtual agents may employ a variety of machine learning algorithms according to the use case. The output of machine learning needs to be semantically integrable to domain model concepts. Consequently, the domain model is important for ”sanity checks” of output of probabilistic reasoning. As discussed earlier, e.g. Markov Logic Networks allows for inclusion of ”hard facts” in probabilistic reasoning, guaranteeing consistency.

Agent composition can be executed once as a response to incoming request, or run for a period of time (activated) . Requests can be high-level goals from human users, for example. Reasoner and domain models map these to technical parameters using agents as tools. Requests can also be anomalies. In this case, composition identifies anomaly detection and recovery planning functionalities [9] required for processing, composing a virtual agent.

IV. TEST SYSTEM

We are using the test system shown in Figure 3 to validate our approach.

Network is represented in the test system by an internal LTE simulator with configuration and Performance Management (PM) Application Programming Interfaces (APIs). The simulator supports research of agents monitoring LTE KPIs such as Channel Quality Indicator (CQI) and Radio Link Failure (RLF) statistics and performing configurations to transceiver (TRX) power and antenna tilt (RET) parameters, for example [12]. We used Capacity and Coverage Optimization (CCO) SON function based on fixed rule set in the test system. Configuration Management (CM) and PM data from simulator are streamed over publish/subscribe bus [23] to information cross-linking functionality [24] which stores resulting information in MongoDB.

In the first test, metadata representation of cross-linked CM and PM data was lifted directly from MongoDB to triple store (AllegroGraph). Lifting was performed by transforming MongoDB JSON data structure into triples. The source data consisted of 1453 data structures — each corresponding to a CM operation performed by the SON function — which resulted in 193,369 triples. This count is based on preprocessing of PM data associated with operations cases. We also tested a version where CM and PM data were lifted to triple store, which lead to quintupling of triple count in our case. With preprocessing, some of the computation needs in query phase can be eliminated.

In the second test, we used MLN reasoner, which can perform probabilistic reasoning on network data and domain model elements (semantic ontology). In essence, MLN performs analysis of likelihood of hypotheses on data, given a set of ground truth statements [11]. The statements with the highest likelihood are then used in semantic reasoning together with domain model. The results of reasoning on amended ontology are then inserted into the triple store for use with SPARQL queries.

The second test with MLN is a simplified version of learning agent architecture in the sense that machine learning capability was performed by the MLN reasoner rather than a learning agent (CCO function in our case). In a future system, the agents would execute machine learning algorithms, and their outputs would be combined with domain model with probabilistic reasoning. This would change some of the detailed information flows in Figure 3, but the end result — combining machine learning with domain model — would be the same.

A demonstration of substituting traditional programming with reasoning was already possible with our test system. The original goal was to implement the query phase or self-operation system [24] as a warm-up exercise, mapping it to agents. It was found that two SPARQL queries from the coordinator were sufficient to achieve the same end result than the previous Clojure software implementation for similarity search. Our preliminary results also indicate that the similarity search with SPARQL queries have the same level of query processing time as the earlier implementation. Since similarity query of self-operation only retrieves and aggregates historical data and query parameters related to metadata, OWL reasoner was not needed for this demonstration.

The test system described above is flexible, since it allows for routing of information in multiple ways. For the first test, we lifted network data from MongoDB directly to triple store. The second test illustrates more advanced reasoning on network state is facilitated by import into OWL ontology.

LTE simulator was run on a laptop, and the rest of the system functionalities on a Linux server. The system specifications for server used in SPARQL substitute for similarity search: four-core i7 with 32 GB of RAM and RAID SCSI hard discs.

V. CASE STUDY FOR AGENT MAPPING

In this Section, we evaluate simple agent mapping with selected use cases with an LTE simulator. The results of such evaluations could be used in the system described in SectionIV by evaluating the results of respective agent executions with MLN, and transferring the top ranking agents to be used in triple store mappings.

A. *Scenario description*

The simulator environment comprises 20 LTE base stations with 32 LTE macro cells covering an area with a radius of about 5 km. The simulator creates Performance Management (PM) data reports that contain cell level KPIs. The cell level KPIs are aggregations of the measurements made by the user equipments (UEs) that constantly report the experienced signal status to cell they are attached to. The PM data of the cells are reported periodically in 15 minute intervals in simulation time, amounting to 5-6 hours of simulation time per scenario.

We use three network management scenarios for our experiments: coverage problem, local overload, and mobile overload. The scenarios reflect network issues with similar objectives but in different contexts. In all scenarios, users demand higher throughput, but the solutions differ from each other.

In the coverage problem scenario, the UEs are located uniformly in an area where the coverage is insufficient. The objective is to increase the throughput by increasing the TXP of the cells. The second scenario, local overload, has a few hundred UEs located in a small area near one base station hosting three cells. The throughput of these cells should be increased by adjusting the antenna TRX tilt angles (remote electrical tilt, RET) towards the group of UEs. In the third scenario, mobile overload, there are 1000 uniformly located background UEs and two groups of 200 UEs constantly moving in the simulated area causing abrupt load peaks in the cells. This issue should be addressed by balancing the load between the nearby cells.

B. *Analysing the SON agents*

For each scenario, we have deployed an agent that executes a desired action. A naïve TXP agent increases the TXP of all target cells by 5 dB, with the target of improving coverage. A naïve RET agent reduces the angles of target cells by two to three degrees (downtilt) aiming to improve the capacity near the antenna. The TXP and RET adjustments are part of a Cell Coverage Optimization (CCO) agent but we have separated these operations for our analysis. A Mobility Load Balancing (MLB) agent constantly adjusts the handover parameters of the cells while it is active. The MLB redirects UE connections from an overloaded cell to neighbor cells. In order to get diverse and comparable results, each scenario is tested with all of three operations, even though enhancements are not expected in some cases.

Figure 4 shows the relative changes of the throughput with standard error margins before and after actions on every scenario. As assumed, the best solution for the coverage problem is the TXP agent (17 % increase), for the local overload the RET agent (13 % increase), and for the mobile overload the MLB agent (18 % increase).

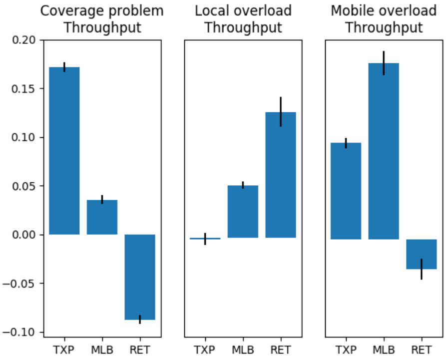


Fig. 4. Scenario-specific relative changes of the throughput values after actions were made.

For further use, we shall use these context-specific results to suggest suitable agents for upcoming network issues in similar contexts.

C. *Evaluation of the mapping task*

To evaluate the context-specific agent mapping, we create new simulations corresponding to the three contexts, but having diverse parameters. For example, we create coverage problem scenario with more users (2500 instead of 1000), but reduce slightly the coverage holes (more transmission power in antennas). In the same manner, the setups of the other two contexts differ from the earlier simulations by the number of the UEs and by the size, shape and movement direction of the UE groups.

All the new scenarios are simulated multiple times for each agent in order to evaluate extensively the agent mapping. To give an overview of the agent performances in the new scenarios, Figure 5 shows the relative changes they produced in the throughput values.

Based on the relative changes in the throughput values, the table II describes the classification and mapping performance of the agents. The case base data (Figure 4) defines the ”predicted” classification of the agents and the new simulations the ”true” classes. The first row depicts the results, when agents are classified with thresholds to four groups. For example, the agent performance is classified as neutral, if the relative change of the throughput is between 0 % and +5 %. The second row depicts an agent mapping task in which +5 % is set as the threshold for mapping results (hits). Last row shows an agent mapping task, when only the best agents are considered (those improving the throughput more than +15 %).

As the classification and retrieval metrics indicate, the accuracy of the classification improves when the threshold for

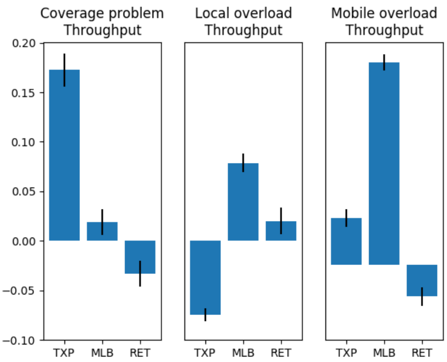


Fig. 5. Scenario-specific relative changes of the throughput for the new simulation scenarios.

”good” performances is higher. The recall values for the two agent mapping tasks indicate high sensitivity in identifying suitable agents, as all possible solutions are present in the search results. Precision values show that some false positive cases are also retrieved. With respect to the Figures 4 and 5 the most probable explanation for the false positive hits are the RET agent in the local overload scenario and the TXP agent in the mobile overload scenario. Clearly, the new local overload scenario is not solved with the downtilt, because the user group is located in a wider area around the base station. Finally, the F_1 score, which is the combination of precision and recall, indicates that the overall performance of the classification is good in this demonstrated use case.

Classes	Thresholds	Precision	Recall	Acc.	F_1
{bad,neutral, ok,good}	0 %, +5 %, +15 %	0.74	0.74	0.74	0.74
{reject,hit}	+5 %	0.73	1.0	0.85	0.85
{reject,hit}	+15 %	0.83	1.0	0.96	0.91

TABLE II
STATISTICS OF THE AGENT MAPPING TASK

All in all, we may conclude that the context-specific agent mapping works in this experimental case study. Moreover, the local overload scenarios demonstrated the challenges in defining contexts; scenarios sharing the same context might actually differ from each other. For this purpose, the contexts should be enriched with semantic contextual metadata that explains the scenarios in a more detailed level. This is an important task that will be addressed in the future research.

VI. NETWORK MANAGEMENT PERSPECTIVE

We shall make some notes regarding the use of our proposed system in network management.

If network information is directly lifted to triple store without an ontology, its consistency depends on the information model that was lifted. A domain model encapsulated in OWL ontology brings benefits by providing a central coordinating role for information used in the system. Furthermore, the OWL ontology allows advanced reasoning. The domain model needs to be kept up to date with information models of network state and learning algorithms (where applicable).

Ontology constructs result in induced triples (e.g. ABox instance `isA` relations to classes). In principle, ontology could be generated by lifting a suitably formal information model (e.g. SID [25]). Previous work has shown that not all information models are sufficiently consistent or formal for this. We have approached ontology construction from minimal viewpoint, focusing on its roles: facilitation of agent mapping, ensuring consistency of output of machine learning, and providing use case specific reasoning to replace traditional programming.

Human input to ontology can be validated by MLN in the same way as machine learning to avoid inconsistencies in the ontology.

The architecture for inserting the output of machine learning algorithms to probabilistic reasoning, such as MLN is not within the scope of this article and will be accounted for in other publications. Similarly, we have not considered the interfaces between MLN and semantic reasoner.

VII. SUMMARY

We presented an approach where multiple agents or software modules are composed to provide a solution to a request such as a high-level goal. Composition supports software reuse through participation of agents to multiple solutions. Knowledge-based reasoning in composition can by itself substitute software implementations.

First level of functionality is achieved with SPARQL queries to a triple store, sufficient for reasoning over historical information about the network. With semantic reasoner triggered by the query, advanced inferences —- including parameters of the request — are possible. Combined with suitable method for probabilistic reasoning such as Markov Logic Networks, the ontology used by semantic reasoner can be used for semantic integration of machine learning output from diverse algorithms chosen to match data. Probabilistic reasoning is important since the outputs of different algorithms may be conflicting. Provided that the domain model part of the ontology is used in probabilistic reasoning, its output can be introduced to ontology.

All in all, the combination of technologies provides a flexible platform for future network management use cases. The reduction of traditional programming was demonstrated with an example where Clojure implementation of similarity query [24] was replaced with SPARQL calls. Agent composition, in particular together with microservices paradigm, supports efficient use case driven network management.

An important advantage is provided by a support for centralized processing of domain model, query parameters, and network state, avoiding the need to replicate this in individual agents. This is crucial for using composition to reduce the need for software implementations in agents.

REFERENCES

[1] NGMN Alliance and M Iwamura, *NMGN view on 5G architecture*, in Proc. VTC Spring 2015.

[2] V. Ziegler, *et al.*, *Architecture vision for 5G era*, in Proc. IEEE ICC, 2016.

[3] 5GPPP architecture working group, *View on 5G architecture*, URI: https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP-5G-Architecture-WP-For-public-consultation.pdf (accessed March 2017).

[4] NGMN, *NGMN 5G white paper*, v1.0, February 2015.

[5] S. Hämäläinen *et al.* (eds.), *LTE Self-Organising Networks (SON): Network Management Automation for Operational Efficiency*, John Wiley & Sons, Chichester, England, 2011.

[6] D. Goldszmidt and Y. Yemini, *Delegated Agents for Network Management*, IEEE Communications magazine **36**, p. 66 ff., 1998.

[7] J. O. Kephart and D. M. Chess, *The vision of autonomic computing*, Computer **36**, p. 41 ff., 2003.

[8] NGMN alliance, *NGMN Recommendation on SON and O&M Requirements*, 2008.

[9] V. Räisänen, Agent composition in 5G management and orchestration, in Proc. CNSM 2017.

[10] V. Räisänen and H. Tang, *Knowledge Modeling for Conflict Detection in Self-organized Networks*, in Proc. MONAMI 2011.

[11] K. Apajalahti *et al.*, *Combining ontological modelling and probabilistic reasoning for network management*, J. Ambient Intelligence **9**, p. 63 ff., 2017.

[12] K. Apajalahti *et al.*, *Sharing performance measurement events across domains*, to appear in Proc. IFIP/IEEE IM, 2017.

[13] A. K. Y. Wong, *et al.*, *Ontology Mapping for the Interoperability Problem in Network Management*, J. selected areas in communications **23**, p. 2058 ff., 2005.

[14] J. E. Lopez de Vergara *et al.*, *Ontologies: Giving Semantics to Network Management Models*, IEEE Network, p. 15 ff., 2003.

[15] J. Keeney *et al.*, *Runtime Semantic Interoperability for Gathering Ontology-based Network Context*, in Proc. NOMS, p. 56 ff., 2006.

[16] P. Szilagyi, and S. Novaczki, *An Automatic Detection and Diagnosis Framework for Mobile Communication Systems*, IEEE Transactions on Network and Service Management **9**, p. 184 ff., 2012.

[17] M. Richardson and P. Domingos, *Markov Logic Networks*, Machine learning **62**, p. 107 ff., 2006.

[18] A. Haller, *et al.*, *WSMX — a semantic service-oriented architecture*, in Proc. International conference on Web services (ICWS), 2005.

[19] J. Keeney *et al.*, *Ontology-based Semantics for Composable Autonomic Elements*, in Proc. Workshop of AI in Autonomic Communications at the 19th Intnl Joint Conf. on AI, Edinburgh (2005)

[20] *WSMO-Lite: Lightweight Semantic Descriptions for Services on the Web*, W3C Recommendation, World Wide Web Consortium, Tech. Rep., Aug. 2010, Available: https://www.w3.org/Submission/WSMO-Lite/ (accessed March 2017).

[21] *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*, W3C Recommendation, World Wide Web Consortium, Tech. Rep., May 2004. URI: https://www.w3.org/Submission/SWRL/ (accessed March 2017).

[22] K. Apajalahti, *Multivariate Time Series-Based Annotation Process for Semantic IoT Services*, in preparation.

[23] V. Kojola *et al.*, *Distributed Computing of Management Data in a Telecommunications Network*, in Proc. MONAMI 2016.

[24] H. Tang, *et al.*, *Automatic Definition and Application of Similarity Measures for Self-Operation of Network*, in Proc. MONAMI 2016.

[25] TMForum, *GB922, Information framework (SID) R16.5.1*, URI: http://https://www.tmforum.org/information-framework-sid/ (accessed March 2017).

Publication II

Kasper Apajalahti, Juha Niiranen, Shubham Kapoor, and Vilho Räisänen. Sharing Performance Measurement Events Across Domains. In *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, IEEE, Lisbon, Portugal, pages 463–469, DOI https://doi.org/10.23919/INM.2017.7987313, May 2017.

© 2017 IEEE

Reprinted with permission.

Sharing Performance Measurement Events Across Domains

Kasper Apajalahti
Aalto University
Semantic Computing Research Group (SeCo)
Espoo, Finland
Email: first.last@aalto.fi

Juha Niiranen
and Shubham Kapoor
University of Helsinki
Helsinki, Finland
Email: first.last@helsinki.fi

Vilho Räisänen
Nokia Bell Labs
Espoo, Finland
E-mail: first.last@nokia-bell-labs.com

***Abstract*—Network management activities, such as fault analysis and configuration management, are eventually related to changes in network measurements. Some measurement event might be either a trigger or objective of a management activity. We argue that sharing the semantics of performance data across networks provides a basis for more advanced automation. This paper presents an ontology-based system for sharing information about network measurements across network domains. The represented information contains correlations and human-defined mappings between network measurements and the system is based on semantic reasoning that identifies dependencies which arise by combining local and shared information. We demonstrate the usage of the system in a Long Term Evolution (LTE) network domain. Our experiments from an LTE simulator and LTE test network show that a combination of correlations, human-defined mappings, and ontological reasoning produces useful cross-domain information that can be accessed with ontology queries.**

I. INTRODUCTION

Making the best use of cellular network infrastructure investment requires the optimization of network parameters in view of changing usage patterns and traffic mix. The target is to achieve the best utilization of capacity while providing the right level of service quality for different applications [1].

The growth in the complexity of mobile networks has necessitated the adoption of automation. The situation has been recognized by Next Generation Mobile Networks (NGMN), and an automation framework has been provided for Long Term Evolution (LTE) networks in the form of Self-Organizing Networks (SON) concept [2]. It provides a closed-loop automation for the classes of operability use cases implemented as SON functions. In a typical realization, fixed rule bases are used for defining the behaviour of SON functions. The definition and governance of rule bases are expensive [3] which puts a price tag on optimizing closed-loop automation on per-cell level. Such needs arise from geographic and temporal variety on a cell level of radio access networks.

In addition to the limitations associated with SON function management, more automation is expected to be needed in many network management tasks such as the fault identification, troubleshooting and continuous optimization of networks [4]. It is expected that adaptive automation is needed in 5G to manage the more complex network.

The use of adaptive automation is not limited to 5G wireless systems. Below, we use the wireless network as a concrete example, but the core idea can be used in any other network management domains.

A. Information exchange

Traditional information sharing needs to take care of network-specific parameters and mappings between networks by means of software. Semantic mappings can be performed using Common Information Models (CIMs) such as TMForum Shared Information/Data (SID) model, but these typically need to be extended for use in mobile domains, not to mention service provider or vendor specific parameters.

We argue that semantic representations of network status together with domain ontologies and reasoning simplifies implementation by facilitating query-based access to knowledge representations [5]. Reasoning can involve both classical and probabilistic aspects [6].

B. Our approach

In this paper, we present Effect Sharing Service (ESS), a framework for sharing performance measurements globally. The motivation behind our approach is enabling effective information exchange across networks which lends itself to automation. Every network operation from anomaly detection to network configuration is eventually related to changes in network measurement values, and thus, these relations could be utilized to connect information about network problems and solutions across domains. For example, a local network might have deployed a SON function with some settings to address an anomaly. Another local network can find this solution via the ESS by querying a solution for its own anomaly. Even though these anomalies have occurred in different networks and might address different metrics, the ESS finds a relation between the metrics in question (e.g. via a statistical correlation and semantic mapping).

The rest of the paper is organized as follows: the next Section II briefly presents an overview of the framework and related use cases. Section III describes technical details and logical axioms for the components of the ESS. Section IV provides statistical experiments from two measurement scenarios (LTE simulator [7] and test LTE network), semantic reasoning results, human-defined mappings and a query example. Finally, Section V discusses related work and Section VI concludes the paper and clarifies our future work.

II. OVERVIEW OF THE ESS

A. Architecture

The core concept of the ESS framework is to express context-specific dependencies between metric effects (value changes in metrics) in network management systems (NMS). A network metric can be any measurement that has a scalar value and characterizes some aspect of the network status, such as a low-level counter, a key performance indicator (KPI), or a high-level business objective, such as customer satisfaction. Vector-valued measurements, such as the Channel Quality Indicator (CQI), are converted to a set of scalar measurements. Currently, our model covers only per-cell measurements, but it is expandable to cell-pair measurements or base station-specific metrics.

The ESS contains a knowledge base and a reasoning capability that are used for linking and finding relations between metrics across domains. Figure 1 presents the ESS architecture in the context of virtualized networks and global network applications. The ESS can be utilized directly from a virtualized NMS that provides performance data and context-specific metadata about a network domain. An NMS can query the ESS, obtaining cross-domain relations from the performance measurements of another network domain. Cross-domain relations between metrics can be utilized to find solutions from one network to a problem in another. For example, one network might have an algorithm addressing changes in a particular metric. Due to a cross-domain relation between metrics in the two networks, the algorithm can be found and re-used also in the other network.

The ESS can also be used by global network applications that aggregate heterogeneous data from network domains and provide global NMS-relevant information such as anomalies, configurations, or network planning. These applications share cross-domain information from particular network management activities, such as fault analysis or configuration management. The ESS may serve these applications with a global representation of network status changes (the metadata of activities), along with an automated tool analysing relations between network states.

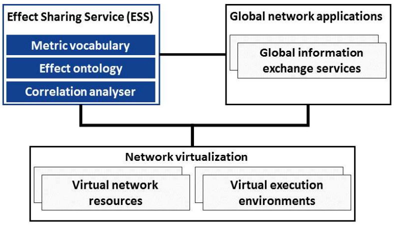


Fig. 1. The ESS architecture and its relation to network virtualization and global network applications.

The ESS consists of a *correlation analyser* that calculates correlations from network domain Performance Management (PM) data, a *metric vocabulary* that maintains a list of metrics used in the ESS, and *effect ontology* that stores and infers semantic metric effects and information about them (Figure 1).

B. The interaction between a local network and the ESS

An NMS have three use cases towards the ESS: mapping of common metrics, sending PM time series, and querying inter-domain effect relations.

1) Mapping of common metrics: Local operators can map some metrics used in their systems to commonly used metrics (described in the ESS). Standard metrics such as those described in 3GPP specifications[1], may be used here. For example, the CQI is specified in the 3GPP specifications and could be defined as common metric in the ESS. A local network operator may measure CQI in its own network and due to the equivalence of the two metrics, the operator can map the local CQI to the CQI instance in the ESS.

2) Sending PM time series: A human operator may send PM time series to the ESS. From this data, correlations are analysed between metrics and the effect ontology is populated with new instances (metrics appeared in the measurement data) and updated with respect to the correlation data. A semantic reasoner can infer new logical implications from the updated ontology. The outcome of the reasoner includes new correlations and contradictions (effects which cannot be achieved simultaneously) between effects. A detailed explanation of the reasoner can be found in the Section III. Information about correlations and contradictions can be used, for example, for automating the governance of closed-loop automation such as SON: correlations and contradictions between metrics clarify which SON functions can operate parallel in the same network without having conflicts.

3) Querying inter-domain effect relations: As a result of the preceding steps, the ESS contains a graph of metric effects and their dependencies. Subsequently, the NMS may now query the ESS to find context-specific effects that are related to queried effects. When the ESS receives a context-specific effect query, the reasoning of effects with similar context is triggered, returning all effects that are satisfying the query via dependency mappings (human-defined mappings or correlations). An example query is demonstrated in the Section IV-E.

III. DESCRIPTION OF THE COMPONENTS IN THE ESS

A. Statistical correlation analysis

A fundamental part of the ESS framework is analysing statistical correlations between metrics. From a statistical point of view a metric is considered as a random variable, whose observed values are values of the metric of a certain cell at a certain point of time. Correlations between metrics are computed within the ESS in order to make results comparable

---

[1] http://www.3gpp.org/specifications

across domains. We want to detect pairs of metrics with a linear relation, e.g., pairs where a positive change in one metric usually occurs simultaneously with a negative change in the other metric. We use Pearson's product-moment correlation coefficient r_{xy}, which measures the linear correlation between two metrics x and y. The correlation is calculated over all cells included in the context.

The correlation coefficient always lies between -1 and 1. We classify the correlation between two metrics as significant, if $|r_{xy}| > 0.5$. With respect to a transitive correlation between x and z (as a consequence of their correlations to y) the following inequality holds for coefficients r_{xy}, r_{yz}, and r_{xz} [8]:

$$\begin{aligned} r_{xy}r_{yz} - \sqrt{(1-r_{xy}^2)(1-r_{yz}^2)} &\leq r_{xz} \\ &\leq r_{xy}r_{yz} + \sqrt{(1-r_{xy}^2)(1-r_{yz}^2)} \end{aligned} \tag{1}$$

From Equation (1) we can see that if both r_{xy} and r_{yz} are greater than $\frac{\sqrt{3}}{2}$ or less than $-\frac{\sqrt{3}}{2}$, then the correlation between x and z is significant as $r_{xz} > 0.5$. We can also see that if one of r_{xy} and r_{yz} is less than $-\frac{\sqrt{3}}{2}$ and the other is greater than $\frac{\sqrt{3}}{2}$ then there is significant negative correlation between x and z.

To make use of this information, we further classify the correlation between x and z as strong correlation if $|r_{xy}| > \frac{\sqrt{3}}{2} \approx 0.866$. The transitivity of individual correlations through a pair of strong correlations is an important feature in our semantic model as it is described in following subsections.

B. *Effect ontology*

We define an effect as an event where a metric value increases or decreases significantly. The limit for "significant" varies across metrics. For this article, we use 5 % change as an arbitrary uniform criterion for significant change by way of demonstration.

The ontology uses the resource description framework[2] (RDF) and web ontology language[3] (OWL 2) to model effects and facts about them. The OWL 2 is compatible with description logic (DL) and semantic web rule language[4] (SWRL). Thus, in addition to the semantic representation of the effects, the ontology facilitates reasoning which we implement with a semantic reasoner, Pellet [9], to infer dependencies between effects.

An effect in the ontology is defined with the following data elements:

- *Metric*: Universal Resource Identifier (URI) as an identifier.
- *Impact direction*: increase or decrease.
- *Context attributes*: describing the environment where the effect has occurred.
- *Dependencies*: Links between effects that may occur at the same time.
- *Contradictions*: Links between effects that are not achievable at the same time.

Every effect metadata contains a metric URI and direction of the value change. The environment where the effect has occurred is expressed with context attributes, such as spatial, temporal or network-related attributes. An effect might have dependencies, such as statistical, subsuming, and human-defined mappings, to other effects.

1) Context attributes: Context attributes describe the measurement scenario the effect is related to. Some context classes and their attributes are described below. These attributes are presented for demonstration purposes and the list of attributes is assumed to expand over time, when measurement scenarios need more sophisticated definitions.

- *Location type:* urban, suburban, rural, highway
- *PM status:* classified attributes of the network performance, such as low/medium/high load and throughput
- *Network Technology:* LTE, UMTS, GSM, Hetnet
- *Overall attribute:* general

Due to the open world assumption of the semantic representation (the absence of an attribute is considered as $unknown$ rather than $false$), new attributes will not violate existing effects. A special context attribute is $general$ that covers all attributes and contexts.

2) Statistical dependencies: Statistical correlations higher than 0.5 or less than -0.5 are represented in the ontology with a general *hasDependency* property. If two metrics have a positive (negative) correlation, then their effects with the same (opposite) impact direction are depended. Effects might also have a *hasStrongDependency* property, which is defined for correlations higher than 0.866 or less than -0.866. A strong dependency has transitive characteristics in a sense that two strong dependencies between effects XY and YZ will produce a general dependency between XZ.

3) Subsumption: A subsumption property, *subsumptionOf*, can be defined in two cases. First, it describes a link from an effect which is part of an aggregated effect to its aggregation, for example from a low-level counter effect to an effect of an aggregated KPI value of several counters, including the linked one. Second, it describes a link from an effect with a limited context (few attributes only) to an effect with a more general context (including all attributes in the other effect). Subsumption is a non-symmetric property; the subsumption relation holds only from the sub-effect to its parent effect.

4) Human-defined mappings: Human experts can share their information about effect dependencies with a *hasLogicalDependency* property. For example, the property can be used to map semantically equal effects between two networks.

5) Transitive dependencies: In order to utilize the transitivity of statistical and human-defined dependencies described above, we define *hasStrongDependency*, *subsumptionOf*, and *hasLogicalDependency* also as subproperties of *hasTransitiveDependency*. Due to this definition, we can utilize an SWRL rule (Equation (2)) for generating dependencies between effects that are transitively connected with two *hasTransitiveDependency* properties. Using the general superproperty,

---

[2] https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/

[3] https://www.w3.org/TR/owl2-overview/

[4] https://www.w3.org/Submission/SWRL/

we can identify transitive links whether they are statistical, human-defined, or a mix of them. The resulting property is a general *hasDependency*. The notation of the equation describes an SWRL rule in the ontology. The rule indicates that if the ontology contains a transitive dependency between effects $?x$, $?y$, and between $?y$, $?z$, then a dependency link is generated between $?x$ and $?z$.

$$\begin{aligned} &hasTransitiveDependency(?x, ?y) \sqcap \\ &hasTransitiveDependency(?y, ?z) \\ &\Rightarrow hasDependency(?x, ?z) \end{aligned} \tag{2}$$

6) Contradictions: A contradiction is defined as a symmetric and transitive property between effects that cannot occur at the same time. Equation 3 shows a rule that generates a contradiction between effects $?x$ and $?y$, because they have the same metric $?metric$, but their impact directions $?impX$ and $?impY$ have different types ($Decrease$ and $Increase$).

$$\begin{aligned} &hasMetric(?x, ?metric) \sqcap hasMetric(?y, ?metric) \sqcap \\ &hasImpact(?x, ?impX) \sqcap hasImpact(?y, ?impY) \sqcap \\ &Decrease(?impX) \sqcap Increase(?impY) \\ &\Rightarrow hasContradiction(?x, ?y) \end{aligned} \tag{3}$$

With respect to the SWRL rule above and to *hasStrongDependency* property between effect instances, the semantic reasoner can infer new contradictions with a rule defined in Equation 4. The rule defines that if effects $?x$, $?y$ contradict and $?y$, $?z$ have a transitive dependency, then a contradiction link is generated between $?x$ and $?z$.

$$\begin{aligned} &hasContradiction(?x, ?y) \sqcap \\ &hasTransitiveDependency(?y, ?z) \\ &\Rightarrow hasContradiction(?x, ?z) \end{aligned} \tag{4}$$

C. Common metrics

We define common metrics as generally used metrics among stakeholders. For demonstration purposes, Received Signal Strength Indicator (*RSSI*) is defined as common metric in the vocabulary. The list of common metrics will be complemented as new use cases and new data sources are added to the ESS. Generally, these could be extracted from 3GPP specifications, such as [10], or otherwise commonly used metrics in SON function and autonomous network management.

Common metrics provide global links between separate network domains. Once network operators have uploaded network-specific metrics and effects to the ESS, they can map some of the metrics to common metrics. Then, the ESS can be queried for inter-domain effect relations. As an example, if metrics in network domains D_1, D_2, and D_3 have been mapped to a same common metric, then their metric effects are also related (increases in the two metrics correlate). Now, if network D_1 has an algorithm addressing issues related to its metric, network D_2 and D_3 will find information about this algorithm by querying the ESS and finding the link to the network D_1 via the common metric mappings.

IV. EXPERIMENTS

We analysed the usage of the ESS in two environments: LTE simulator [7] and LTE test network. In the simulator, the measurement scenario consists of a 2 GHz LTE network with 32 macro cells covering an urban area with a diameter of 5 km and 2000 terminals. The test network is a live LTE network for research purposes. The network operates at 2.6 GHz and comprises 20 LTE base stations with 36 LTE cells. The test network can host up to 200 real and simulated LTE users.

A. Context for scenarios

The information about measurement scenarios described above can be expressed with context attributes as shown in the table I. Given the context attributes above, all cells in the simulator are included in correlation analysis. From the test network, we include four cells fulfilling the context criteria.

	Location type	Network tech.	PM status
Simulator	Urban	LTE, 2,0GHz	high thrp, high load
Test network	Urban	LTE, 2,6GHz	medium thrp, medium load

TABLE I
CONTEXT ATTRIBUTES FOR SIMULATION AND TEST NETWORK SCENARIOS.

B. Simulator correlations

We analysed the following metrics in the simulator: individual CQI classes, average CQI (CQI_Avg), Radio Link Failures (RLF), terminals per cell (CUEs), and average RSRP. The table II shows correlation coefficients between the metrics. Strong correlations higher than $|0.87|$ are highlighted in the table. As the table shows, CQI class 1 (CQI_1) has a strong correlation with RLF and the CQI_Avg has a strong negative correlation with the RSRP.

Measurements also revealed that other low CQI classes (1 to 3) correlated with the RLF and that the CQI_Avg correlated with several CQI classes; to classes from 3 to 11 CQI_Avg had coefficients higher than 0.5. For simplicity, the table presents only CQI_1 from the CQI classes.

	CQI_1	CQI_Avg	RLF	CUEs	RSRP
CQI_1	1	0.19	**0.87**	0.69	-0.44
CQI_Avg	0.19	1	0.07	0.46	**-0.93**
RLF	**0.87**	0.07	1	0.78	-0.27
CUEs	0.69	0.46	0.78	1	-0.58
RSRP	-0.44	**-0.93**	-0.27	-0.58	1

TABLE II
CORRELATIONS FOUND IN THE SIMULATOR. SEE TEXT FOR EXPLANATION OF METRICS.

C. Test network correlations

The following per-cell metrics were analyzed in the test network: a signal-to-noise ratio (SINR), RSSI, uplink throughput (U-THR) and downlink throughput (D-THR). All the cells had similar behaviour and correlations for given metrics. Correlations are reported in the table III, which presents coefficients between selected metrics. From these metrics, the RSSI correlated strongly with the SINR and the D-THR with the U-THR.

	RSSI	SINR	D-THR	U-THR
RSSI	1	**0.88**	0.49	0.31
SINR	**0.88**	1	0.66	0.44
D-THR	0.49	0.66	1	**0.90**
U-THR	0.31	0.44	**0.90**	1

TABLE III
CORRELATIONS FOUND IN THE TEST NETWORK. SEE TEXT FOR EXPLANATION OF METRICS.

D. Semantic contradiction analysis

The semantic reasoner infers contradictions between metric effects. The obvious contradictions can be found with respect to the axiomatic rule in Equation 3 (opposite impacts of the same metric cannot occur at the same time). In addition to these, table IV shows which contradictions the semantic reasoner has inferred from the simulator and test network with respect to the ontology rule 4 and earlier presented correlation data (tables II and III). The first column depicts the environment, second contradicting metrics and third the direction of a contradiction (+/–). The direction of the contradiction is opposite to the sign of the correlation coefficient. For example, the first row of the table defines that CQI_1 and RLF contradict negatively meaning that CQI_1 cannot increase when RLF decreases and vice versa.

Environment	Metric pair	Direction of contradiction (+/–)
Simulator	CQI_1, RLF	–
	CQI_Avg, RSRP	+
Test network	SINR, RSSI	–
	U-THR, D-THR	–

TABLE IV
INFERRED CONTRADICTIONS FROM THE SIMULATOR AND TEST NETWORK.

Information about contradictions can be used to validate an action in an NMS. For example, an action that increases CQI_1 in the simulator in the given context is undesirable if RLF should not increase at the same time. Similarly, it is not possible to execute an action that has an objective to increase downlink and decrease uplink throughput in the test network (in the given context), as these effects contradict.

E. Common metric mappings and inter-domain query example

Let us consider a scenario where one network expert has access to the simulator data and another to the test network data. They examine common metrics and look for correspondencies in their networks. If we have an RSSI as a common metric, one obvious mapping in the test network is between the common and local RSSI metrics as they are most likely equal KPIs. With respect to the RSRP metric in the simulator, let us make a hypothesis based on the theory of RSSI and RSRP formulas [11][12], that effects in RSRP are part of effects in RSSI (e.g. when RSRP increase, also RSSI increases). Thus, a one-way *subsumptionOf* link from the simulator RSRP to the common RSSI is created. After the mappings are done, we may assume that if these metrics would be available in the same network environment, there would be correlations between them and for this reason, we may find inter-domain effect relations.

As an example of related inter-domain effects, Figure 2 illustrates them with a graph visualisation. The figure depicts effect relations in the simulator, test network, and global effect vocabulary. Every square node represents an effect with information about its metric, impact direction, and context attributes included. The results of the abovementioned RSSI mappings have produced effect relations to the global vocabulary from the simulator (between nodes 3 and 4) and the test network (4 and 5).

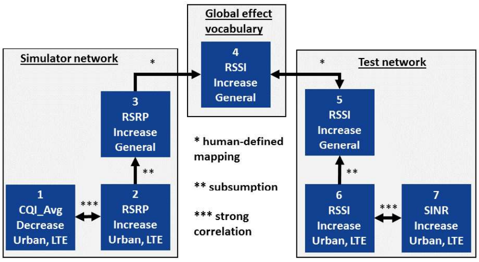


Fig. 2. Query results for decreasing CQI_Avg in simulator with attributes: urban, and LTE.

Now, assuming that during an LTE network simulation, a significant decrease in the average CQI has been identified in cells in an urban area. The operator sends a query to the ESS to find effects related to this anomalous effect (with context attributes urban and LTE). Reasoning over results is initiated with a SPARQL interface and the results for the query are shown in the Figure 2, as a graph. The table shows matched effects: the first match (the node 1) is to an effect having a match in impact, metric, and context attributes (may be general or specific attributes). After that, RSRP_Inc is matched via a strong correlation (2), which is turn has a subsumption relation to its general effect (3). The general RSRP_Inc is matched to RSSI_Inc (4) in the global vocabulary due to the human-defined subsumption mapping. The global RSSI_Inc is in turn matched to the corresponding test network metric effect (5). After the context-specific RSSI_Inc is matched (6), we finally get a match to a context-specific SINR_Inc (7) that is found via strong correlation. From the results, we can conclude that in a context, in which simulator metric CQI_Avg decreases, the test network metric SINR would increase. Further assuming that there would be an algorithm addressing anomalies in SINR in the test network (some link between the algorithm instance and the SINR effect), the example query would provide a path to this algorithm, even though there was no predefined semantics between the CQI in the simulator and SINR in the test network.

The aforementioned query demonstrated the use of the ESS; one can use this information to find related measurements across networks.

V. RELATED WORK

A study in [13] presents a conceptual architecture for cross-domain 5G network management system between cellular and industrial networks. An architecture for shared infrastructure in future cellular networks has also been addressed in [14] which proposes a network configuration platform that would aggregate multi-domain resources and translate requests between several control planes and NMSs.

Context-specific measurements have also been presented in [15] by means of an adaptive SON function management mechanism. Here, the goal is to bind context-specific metric measurements together with SON configurations to enable the dynamic adoption of suitable SON configurations with respect to current network status. [15]

A service management-related research paper has proposed an ontology-based global service management framework that links local systems together [16]. The paper describes a common taxonomy for service management concepts, also for KPIs which are linked to service level agreements SLAs. [16]

Cited studies relate to inter-domain platforms and information exchange mechanisms that react or analyse changes in performance metrics. The ESS instead provides a unified framework for representing these metric value changes in the network status. Thus, this work can be seen as a common ontology and reasoning capability from which network-related global platforms can benefit, for example for linking performance events to the platform and information (in a view of metric effects) across platforms.

Semantic and logical models for performance metrics are researched earlier. In [17] has been presented an ontology for representing the units of measurements. In [18] and [19] ontologies have been defined for the semantic representation of performance metrics and their metadata, such as formulas (including the units of measures), targets, and dimensions (temporal, organizational, etc.). [18][19] These works focus more on describing metric itself, whereas the ESS describes a measurement event (metric value change) and context metadata about it.

In [20] predicate logic has been used to define similar logical relations between performance metrics as in our system. For example, the authors define causality, aggregation (similar to subsumption in our work) and correlation for metric relations. However, the axioms are not directly adaptable, as we define our axioms specifically for context-specific metric effects. In addition to logical differences, our framework provides a semantic representation of metric effects which enables better interoperability between heterogeneous data sources and global usage of effects.

VI. CONCLUSIONS AND FUTURE DIRECTIONS

This paper presented a framework, Effect Sharing Service (ESS), for distributing information about context-specific metric effects across virtual networks. We have described the semantic and logical foundation of our system and described its functionality with an architecture, examples, and experiments from two data sources: LTE simulator and test network. The results of our first experiments with the ESS validate the interaction between local networks and the ESS. Both human-defined mappings and experiments from correlation analysis from the data sources have been translated into effects which are linked globally via the effect ontology. In particular, the use case example in the Section IV-E demonstrates the benefit of the ESS, as the operator utilizes new effect information extracted from the measurement scenarios and points out linked effects between the simulator and test network.

Although the first results look promising, some issues still needs to be further analysed. One issue is the level of statistical significance in the correlation analysis. As defined in III-A, the threshold for a normal metric correlation is set to 0.5. This value should be further investigated to find out a probability for metric effects to co-occur, given a certain metric correlation coefficient. Another issue to be considered in statistical analysis is a confidence value indicating the quality of the analysed dataset. The confidence value would express the reliability of analysis. For example, what is the variance in the metric values, what is the time range considered, and how many cells are included.

Instead of analysing metric value correlations, we could directly examine event correlations. For example, classifying value changes into effects, such as "no change", "small change", and "significant change", and comparing the occurrences of these metric effects. This way, we might get deeper insight to effect relations and possibly the statistical analyser would reveal various types of relations, such as subsumptions (e.g. subeffect correlates with the parent effect but not the opposite) and other one-way metric correlations (e.g. the increases of two effects co-occur while decreases do not).

A potentially important aspect for the future work is the exploration of context attributes which produce semantically valuable relations between effects. Current scenarios serve as practical examples of how to use the ESS, but might be too general and different from each other to make further conclusions about the similar behaviour of cells in these contexts. Thus, the ESS would need the analysis of semantically similar metrics in various contexts and with many data sources in order to extract the relevant sets of context attributes.

Altogether, the goal for building the ESS was to enable semantic connections between metric-related data across networks. This paper gives promising results from experiments indicating the need for the semantic representation of metric data in the view of global information exchange.

REFERENCES

[1] V. Räisänen, "Service quality support — an overview," *Computer communications*, vol. 27, p. 1539 ff., 2004.

[2] NGMN, "NGMN recommendation on SON and O&M requirements," *Next Generation Mobile Networks, White paper*, 2008.

[3] S. Hämäläinen, H. Sanneck, and C. Sartori, *LTE Self-Organising Networks (SON): Network Management Automation for Operational Efficiency*, 1st ed. Wiley Online Library, 2012.

[4] NGMN, "5G white paper," *Next Generation Mobile Networks, White paper*, 2015.

[5] "SPARQL query language for RDF," W3C Recommendation, World Wide Web Consortium, Tech. Rep., Jan. 2008.

[6] K. Apajalahti, E. Hyvönen, J. Niiranen, and V. Räisänen, "StaRe: Statistical reasoning tool for 5G network management," in *The Semantic Web: ESWC 2016 Satellite Events*, H. Sack, G. Rizzo, N. Steinmetz, D. Mladenić, S. Auer, and C. Lange, Eds. Springer-Verlag, May 2016.

[7] I. Viering, M. Dottling, and A. Lobinger, "A mathematical perspective of self-optimizing wireless networks," in *2009 IEEE International Conference on Communications*. IEEE, 2009, pp. 1–6.

[8] E. Langford, N. Schwertman, and M. Owens, "Is the property of being positively correlated transitive?" *The American Statistician*, vol. 55, no. 4, pp. 322–325, 2001.

[9] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, "Pellet: A practical owl-dl reasoner," *Web Semantics: science, services and agents on the World Wide Web*, vol. 5, no. 2, pp. 51–53, 2007.

[10] 3GPP, "Evolved Universal Terrestrial Radio Access Network (E-UTRAN);Self-configuring and self-optimizing network (SON) use cases and solutions. Technical report 36.902 version 9.3.1," http://www.etsi.org/deliver/etsi_tr/136900_136999/136902/09.03.01_60/.

[11] H. Xian, W. Muqing, M. Jiansong, and Z. Cunyi, "The impact of channel environment on the RSRP and RSRQ measurement of handover performance," in *Electronics, Communications and Control (ICECC), 2011 International Conference on*, Sept 2011, pp. 540–543.

[12] L. Luan, M. Wu, Y. Chen, X. He, and C. Zhang, "Handover parameter optimization of LTE system in variational velocity environment," in *Communication Technology and Application (ICCTA 2011), IET International Conference on*, Oct 2011, pp. 395–399.

[13] C. Mannweiler, L. C. Schmelz, S. Lohmüller, and B. Bauer, "Cross-domain 5G network management for seamless industrial communications," in *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, April 2016, pp. 868–872.

[14] A. Khan, W. Kellerer, K. Kozu, and M. Yabusaki, "Network sharing in the next mobile network: TCO reduction, management flexibility, and operational independence," *IEEE Communications Magazine*, vol. 49, no. 10, pp. 134–142, 2011.

[15] S. Lohmüller, L. C. Schmelz, and S. Hahn, "Adaptive SON management using KPI measurements," in *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, April 2016, pp. 625–631.

[16] A. Castro, V. A. Villagrá, B. Fuentes, and B. Costales, "A flexible architecture for service management in the cloud," *IEEE Transactions on Network and Service Management*, vol. 11, no. 1, pp. 116–125, 2014.

[17] H. Rijgersberg, M. van Assem, and J. Top, "Ontology of units of measure and related concepts," *Semantic Web*, vol. 4, no. 1, pp. 3–13, 2013.

[18] C. Diamantini, L. Genga, D. Potena, and E. Storti, "Collaborative building of an ontology of key performance indicators," in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, 2014, pp. 148–165.

[19] A. del Río-Ortega, M. Resinas, and A. Ruiz-Cortés, *Defining Process Performance Indicators: An Ontological Approach*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 555–572. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-16934-2_41

[20] V. Popova and A. Sharpanskykh, "Modeling organizational performance indicators," *Information systems*, vol. 35, no. 4, pp. 505–527, 2010.

Publication III

Haitao Tang, Kaj Stenberg, Kasper Apajalahti, Juha Niiranen, and Vilho Räisänen. Automatic Definition and Application of Similarity Measures for Self-Operation of Network. In *Agüero Ramón, Zaki Yasir, Wenning Bernd-Ludwig, Förster Anna, Timm-Giel Andreas (editors) Mobile Networks and Management. MONAMI 2016. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, Springer, Cham, Abu Dhabi, volume 191, pages 206–219,
DOI https://doi.org/10.1007/978-3-319-52712-3_15, October 2016.

© 2016 Springer, Cham

Reprinted with permission.

Automatic Definition and Application of Similarity Measures for Self-Operation of Network

Haitao Tang[1], Kaj Stenberg[2], Kasper Apajalahti[3], Juha Niiranen[4], and Vilho Räisänen[1]

[1]Bell Labs
Nokia
Finland

[2]Mobile Network Products
Nokia
Finland

[3]Semantic Computing Research Group
Aalto University
Finland

[4]Dept. of Mathematics and Statistics
University of Helsinki, Finland

`haitao.tang@nokia.com`

Keywords: similarity measure, context aware, network operations, self-operation, operation experience, OSS, cellular network, case based reasoning.

Abstract. Self-operation concept is proposed to learn the past experiences of network operations and apply the learned operation experiences to solve new but similar problems. It works based upon the observation that actions appropriate for achieving an objective resemble each other in similar network contexts. Plenty of such similarities exist at the level of network elements, functions, and their relations. Similarity measure definition and application are essential components for the self-operation to apply the learned operation experiences. This paper provides a solution for self-operation to define and apply two types of similarity measures for two self-operation use cases. The first use case answers how to select a best suitable function to achieve any given objective. The second use case tells how the selected function should be configured with the most optimal parameter values so that the given objective could be achieved. This solution is realized on a demonstrator implementing the self-operation concept. Corresponding experiments are made with the demonstrator. The experimental results show that the self-operation solution works well.

1 Introduction

The network environments of multi-RAT, multi-access, and multi-vendor have added significant complexity to the network operations. Self-x functions (e.g., SON and traffic steering functions [1-3]) have become an essential part of the 3~4G networks and their management. These self-x functions have reduced a clear part of manual work related to operations that would be needed otherwise for the 3~4G networks. This effectively reduces the operational complexity perceived by human operators as well. The coming 5G systems (i.e., their networks and management systems [4]) are expected to have a much wider scope and, a larger number and variety, of self-x functions and multi-x network environments. In addition, one of the 5G goals is to minimize the need of human involvement in their operations.

adfa, p. 1, 2016.
© Springer-Verlag Berlin Heidelberg 2016

These fundamental developments have created the industry-wide determination to gradually evolve towards cognitive network management systems. In such systems, relevant past experiences could be used to predict the future status of a network. The corresponding decisions could thus be made to improve either network performance or subscriber perceived experience. In such systems, the minimal but still critically needed role of human operator can be seamlessly integrated to observe the systems and instruct them when the non-human parts of systems have no knowledge to deal with certain situations or are otherwise incapable of drawing conclusions or making decisions by their own based on their predefined logics. Such systems can prevent the functions (self-x or not) of the systems from executing the operations that may cause (and are known to have caused already earlier) degradation in network performance metrics or unfavorable user experience. Such systems can also make an operation of a function favorable if its execution is expected to induce improvement in network performance or customer experience.

Self-operation [5] proposes a solution to realize such an aforementioned cognitive network management system. It creates a self-operation case for each relevant event, learns every corresponding operation and outcome of the system, and stores the learned experience to the self-operation case. The outcome consists of the performance metrics and customer experience, etc. The self-operation case also stores the learned context data such as system conditions and other relevant circumstances (e.g. cell configuration, location, and traffic profile) that may have impacted the triggering of the event. All the data relevant to a corresponding operation execution are learned and collected in the data elements of the self-operation case, and thus inherently linked into a piece of useful corresponding experience, which can thus be applied on the fly. The availability of such experience is very important. As of today, these data (if any) are quite scattered and distributed in a system. Some data elements are stored in different locations. Other necessary data elements may not yet be stored at all. In such a situation, data mining cannot help to find the experience. In addition, data mining is usually time / resource consuming, and cannot therefore meet a request for such experience in timely fashion.

The self-operation solution in [5] does not answer how to define a similarity measure to find the corresponding operation experiences from the potentially large number of learned but different self-operation cases. The similarity measure definition is an essential and critical component of the self-operability and, it is specific to the given use case of self-operation. It determines if the operation experience cases (i.e., knowledge) learned from the earlier executed operations can be applied to future operations of a given use case.

As a major use case of self-operation, an operator may want to request the self-operation for guidance on how to achieve given objective(s) for network performance or service in a certain area (i.e. scope of the network). The objectives are usually related to improvement of certain Key Performance Indicators (KPI) regarding for example coverage, traffic, mobility, or quality. Different functions may however cause impacts on many of those KPIs at the same time. It is thus difficult for the operator to select the best function (out of several candidate functions) to achieve the objective. This is where the definition and application of the corresponding similarity measure can help. The relevant self-operation cases can be matched from the knowledge database, with the corresponding objective-specific similarity measure. The information about the best suitable function can then be extracted from the relevant self-operation cases.

As another major use case of self-operation, the operator can have difficulty to determine the (best) suitable configuration (e.g., SCV - SON function Configuration parameter Value) for the selected function to achieve the result expected by the objective. The suitable configuration of the function depends on the corresponding conditions (network configurations, status, traffic, etc.) of the managed objects (MOs) where the function is planned to be executed. This is again where the definition and application of the corresponding similarity measure can help. The relevant self-operation cases could be matched from the knowledge database, with the corresponding function-specific similarity measure. The information of the (best) suitable function configuration can then be extracted from the relevant self-operation cases.

The function-specific similarity measure could also be used to find the corresponding operation experience case(s) and extract the knowledge concerning another related major use case of self-operation that answers the question: “Can an action request for the function be executed or not?” Thus, the function-specific similarity measure enables both the operation to select the corresponding configuration for the chosen function, and the decision on an action of the function.

The motivation of this paper is therefore to design a complete solution by solving the following problems: (1) how to define an objective-specific similarity measure to match an objective to its corresponding function; (2) how to automatically match the objective and corresponding rule given by an operator to the best suitable function for achieving the given objective; (3) how to automatically define the corresponding function-specific similarity measure; (4) how to automatically apply the function-specific similarity measure for a function-specific operation. For example, the corresponding operation information and the configuration(s) can be found from the matching operation experience cases.

The sections of this paper are organized as the follows. In Section 2, the self-operation architecture to define and apply a similarity measure is described. In Section 3, the approach to define an objective-specific similarity measure is introduced. In Section 4, it is explained how an objective-specific similarity measure is used to find the relevant function. In Section 5, the approach to define a function-specific similarity measure is presented. Section 6 depicts how a function-specific similarity measure is used to find the proper configuration value for the selected function. Section 7 introduces the example implementation of a self-operation system and its experiment results. Section 8 summarizes and discusses the major finding of the current work.

2 Self-Operation Architecture for Similarity Definition and Application

Figure 1 shows the architecture of defining and applying similarity measures based on stored operation experience cases. The arrows are logical and can be implemented by direct or indirect connections between the entities in the real implementations.

The definition of a similarity measure is started when a request message for action recommendation (i.e., Message 1, A, or I) is received. These messages serve as the triggers to define relevant similarity measures and use them to find the matching self-operation cases with the principle of case based reasoning [6]. The corresponding experiences in the matching self-operation cases are replied back to the requesting entities.

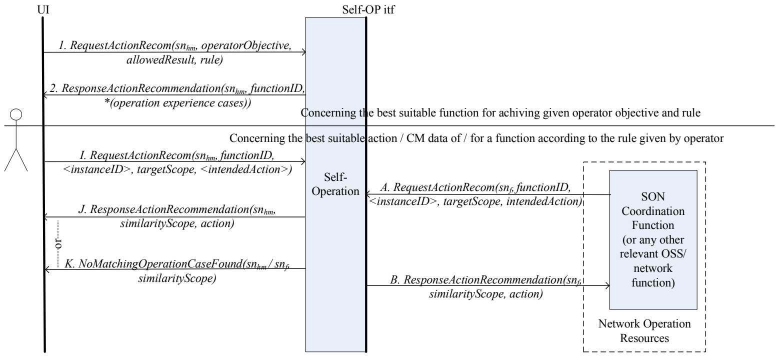


Fig. 1. The architecture of defining and applying similarity measures based on stored operation experience cases.

There are different types of similarity measures, which are usually specific to their actual applications (i.e., use cases) [7-9]. The similarity measures of this paper belongs to the family of semantic similarity measures. The similarity measures of this paper are used to find the exactly matching self-operation cases (if any) stored in the database of the self-operation entity. Their specific definitions and applications of the similarity measures are given in the following sections.

3 Definition of Similarity Measure for a Given Operator Objective and Rule

The operational objectives and rules of a network (e.g., [10]) are usually defined with a set of high level KPIs (e.g., [11, 12]) for the network operations. When an operator needs to achieve a specific objective for network performance under certain rules (i.e., constrains and options), the operator sends a request (Message 1 in the Figure 1) to self-operation function. This message carries the information of the operation objective including the target scope (i.e., the targeted MOs), allowed result, and rules. The rules can be created either by the system vendor or by the operator via means of Rule Editor, which is a specialized tool for the creation and maintenance of the rules.

The self-operation function uses the received information to define the corresponding objective-specific similarity measure, which can be simply in the form of a text string carrying the provided information elements.

4 Selection of a Suitable Function for the Given Operator Objective and Rule

The self-operation function uses the objective-specific measure (defined in Sec. 3) to find all the matching operation experience cases and their functions. For demonstration purposes, we present two Capacity and Coverage Optimization (CCO) functions, CCO-SURROUNDED (optimizing a cell surrounded by its first-tier neighbor cells) and CCO-HOTSPOT (optimizing a hotspot source cell). For example, these functions (CCO-SURROUNDED and CCO-HOTSPOT, shown in Figure 6) have caused similar operation experiences (in areas containing both surrounded and hotspot cells) in the past.

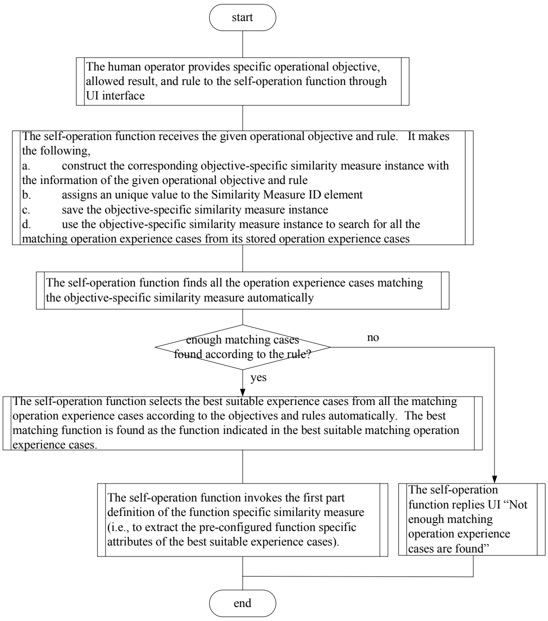
According to the rule, the self-operation function selects the best suitable experience cases from all the matching operation experience cases. For example, *CCO-SURROUNDED* is the function that has achieved the optimization objective in most of the matching cases (96 % of all the matching cases). *CCO-SURROUNDED* function is thus selected automatically as the best suitable function to achieve the intended operation. The decision for the selection can be made based on several different criteria such as the highest probability to achieve successful results, operations' priorities or operator's preferences and policies. The criteria is actually defined by the rules provided by the operator. The general procedure of objective specific similarity definition and function selection is described with the diagram shown in Figure 2.

5 Definition of Similarity Measure for the Selected Function

The common information elements needed by a function-specific similarity measure instance are defined as a set of general similarity attributes and function-specific attributes, as shown in Table 1. The function-specific attributes are always explicitly defined for the specific function selected. A function-specific similarity measure instance is always operation specific.

After the best suitable function is found by self-operation, the self-operation is invoked to define its function specific similarity measure, which consists of two parts. With the information of the function (e.g., *CCO-SURROUNDED*) and the objective-specific similarity measure (e.g., coverage-related optimization of the cells with ID 1-5), the self-operation function defines the first part of the corresponding function specific similarity measure. Here, the information of any function in the network is predefined and made available in the form of function metadata [13] by the operator or its vendor. The function information also defines the impacting scopes [14] of the function.

The first part of the function-specific similarity measure is the static information of the function and the MOs that are either pre-defined or available beforehand. This "static" part is defined by extracting the information of the function, the corresponding cells, and the relevant rule. For example, the first part of *CCO-SURROUNDED* -specific similarity measure can consist of the information elements (and their values) of such as CCO ID, cell technology, cell type, and antenna mode. For simplicity, we assume the target scope consists of only one similarity scope in this example. In reality, if multiple similarity scopes exist in a given target scope (as often the case), their corresponding function-specific similarity measures are defined one by one with the same approach shown in this example.



The human operator provides specific operational objective, allowed result, and rule to the self-operation function through UI interface

The self-operation function receives the given operational objective and rule. It makes the following,
a. construct the corresponding objective-specific similarity measure instance with the information of the given operational objective and rule
b. assigns an unique value to the Similarity Measure ID element
c. save the objective-specific similarity measure instance
d. use the objective-specific similarity measure instance to search for all the matching operation experience cases from its stored operation experience cases

The self-operation function finds all the operation experience cases matching the objective-specific similarity measure automatically

enough matching cases found according to the rule?

The self-operation function selects the best suitable experience cases from all the matching operation experience cases according to the objectives and rules automatically. The best matching function is found as the function indicated in the best suitable matching operation experience cases.

The self-operation function invokes the first part definition of the function specific similarity measure (i.e., to extract the pre-configured function specific attributes of the best suitable experience cases).

The self-operation function replies UI "Not enough matching operation experience cases are found"

Fig. 2. Procedure to find the best matching function and its corresponding operation experience cases that can achieve a given operator objective.

The self-operation function then uses the defined first part of the function-specific similarity measure to further select the matching operation experience cases from all the cases still fulfilling the search criteria. For example, there are 51 self-operation cases found under the selected *CCO-SURROUNDED* function. 25 self-operation cases match the defined first part of the CCO-specific similarity measure.

The self-operation function extracts the information of the 25 cases. What to extract depends on the given rule or otherwise a default configuration. For example, an extraction can be done from all those performance metrics information elements and their value ranges shared by some or all of the 25 cases. These performance metrics are, for example, the impacting and impacted metrics of *RLF INPUT* and *RLF OUTPUT*. The extracted result serves and becomes the remaining part of the similarity measure definition. Now, the complete function-specific similarity measure has been defined.

Table 1. The common information elements of a function-specific similarity measure instance.

	Element Name	**Definition**
General Attributes	Function-specific Similarity Measure ID (F_SM_ID)	A character string that uniquely identifies this similarity measure instance. It helps the further process and application of this similarity measure instance.
	Similarity Scope	The type of the managed objects (MOs) relate to this similarity measure instance. For example, A similarity scope can be one type of *{individual cell, cell pair, first-tier neighbor cells, second-tier neighbor cells, subnetwork, network, etc.}*. A similarity scope is usually specific to a function. For example, a *CCO-SURROUNDED* function is optimizing the coverage performance of the given cells. Thus, the similarity scope for this function is the given cell and its 1st-tier neighbors of the same type.
Function-Specific Attributes	Function ID	The unique ID of a function (e.g., *CCO-SURROUNDED*) that is selected as the most relevant function to pursue the requested operation.
	Function Specific Attribute1	The first feature specific attribute and value that is automatically extracted from the selected experience cases. Note: an attribute and its value are extracted only when this attribute is impacting the function or is impacted by the output of the function. The attribute is identified according to the impacting scopes [14] of the function.
	…	
	Function Specific Attribute$_n$	The n^{th} feature specific attribute and value that is automatically extracted from the selected experience cases, where $n \geq 0$ and, $n=0$ means there is no feature specific attribute for the specific similarity measure instance.

6 Selection of Suitable Configuration for the Selected Function

With the defined function-specific similarity measure, self-operation function finds, e.g., 9 self-operation cases (out of the 25 cases) matching the similarity measure exactly. The configuration values (SCVs) of the 9 self-operation cases are collected into a configuration set called "*CCO-SURROUNDED Config Set*$_1$". The self-operation function then uses the extracted configuration value to configure the *CCO-SURROUNDED* function and activate it to achieve the given objective.

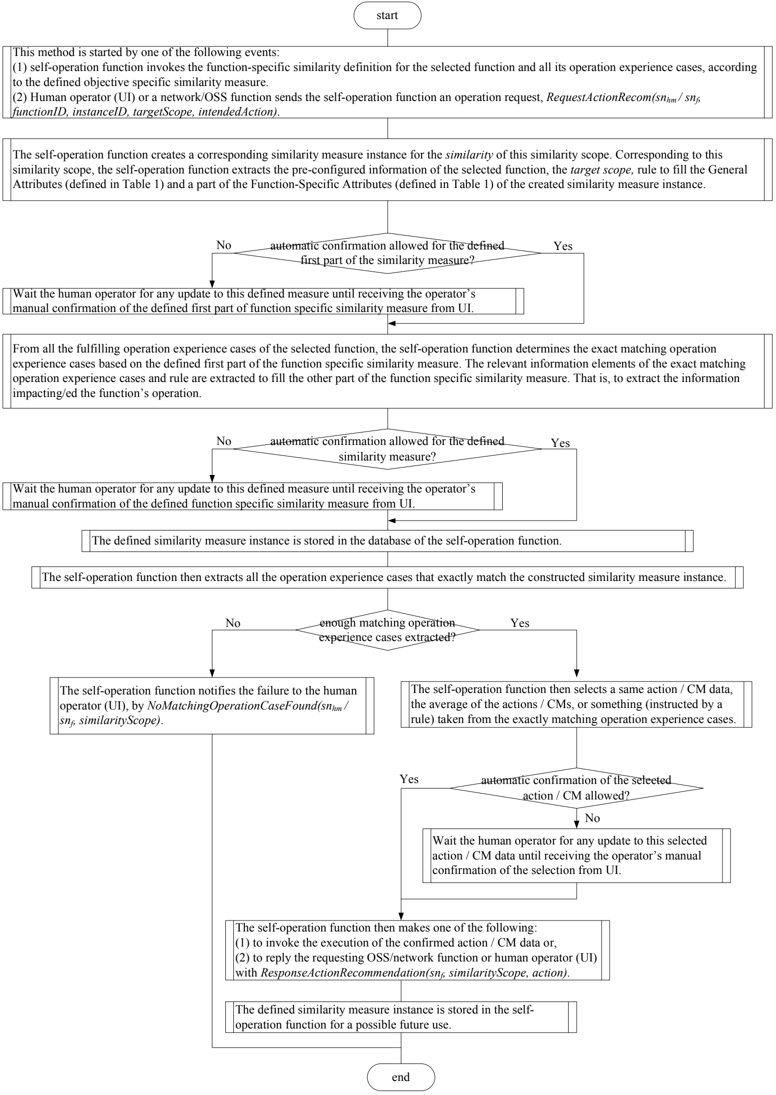


Fig. 3. Procedure for the automatic function-specific similarity measure construction and application.

The general procedure of the definition and application of function specific similarity measure is described with the diagram shown in Figure 3. In this procedure, the intervention of human operator is supported in the otherwise automatic definition and application of a function-specific similarity measure. For example, the operator may need to update or confirm a selection.

7 Experiments on Determining Suitable Function and Configuration Automatically

In this section, we describe a demonstrator for self-operation and show its experimental results. It currently realizes two use cases: 1) finding the suitable function to achieve a given (high level) objective and 2) finding the corresponding configuration for the selected function so that the objective can be achieved. The details concerning these two use cases have been presented in Sec. 3-6.

7.1 Demonstrator Description

The demonstrator set-up for learning operation experiences is shown in Figure 4. It also supports the applications of operation experiences for the self-operation use cases that receive their configuration or instruction from self-operation with means not shown in Figure 4.

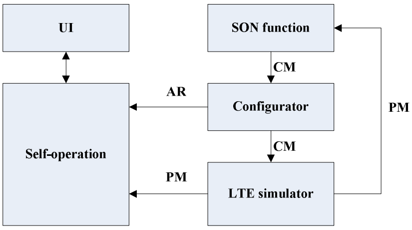


Fig. 4. Architecture of self-operation demonstrator for learning operation experiences, where AR = action request, CM = configuration management data, PM = performance management data, and UI = user interface.

An LTE simulator (Nokia internal tool and its main principle introduced in [15]) acts as a source of PM data, which are sent via REST (Representational State Transfer) interfaces to a SON function and the self-operation control logic. Two groups of SON functions are used in the demonstrator, i.e., RET (Remote Electrical Tilt)-based CCO and energy saving (ES). Configurator adjusts cell and other (e.g., function) parameters on the one hand, and amends CM data with metadata to create Action Requests (ARs) for self-operation on the other. The UI can be used for similarity definitions and related operation case searches. Direct configuration from the UI is not implemented at the moment.

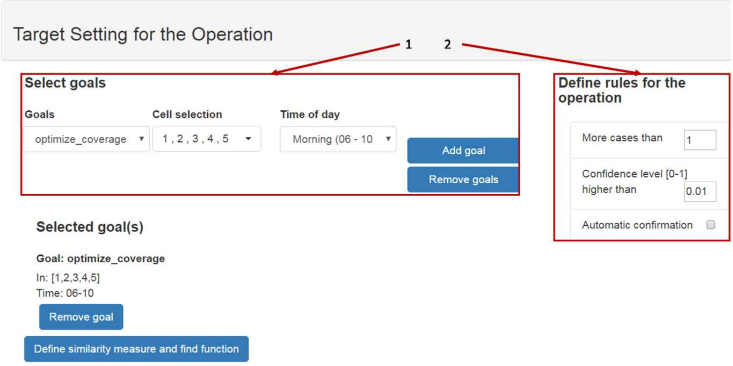
PM data are stored into self-operation internal database as they arrive. ARs are received by the self-operation system in such a way that each received AR triggers the creation of a corresponding operation case. A MongoDB NoSQL database is used for storing PM data, operation cases, and function profiles. The REST interface is implemented with Java, the self-operation control logic with Clojure, and the user interface with HTML 5 / JavaScript. SON functions, self-operation, and UI JavaScript back-end are run on Ubuntu desktop machine (Intel Core2 2.5 GHz, 2GB memory, 64-bit Ubuntu). UI front-end is run on browser (Chrome) over a Windows 7 laptop (8 GB).

7.2 Experiment and Result

This section shows an example experiment and results in which a human operator provides high-level operation objective and then finds a best matching function and its configuration set in a desired context. The experiment is done in three phases, i.e., defining goal and preconditions, retrieving relevant search results based on the automatically defined similarity measure, and, if needed, sharpening the results by adjusting similarity measure.

7.2.1 Defining Goal and Precondition for the Objective-Specific Similarity Measure

Figure 5 depicts the first phase in the view. Through UI, the user can define a goal and its context by Wizard 1 of Figure 5 and set numerical boundaries (or other rules) for the search results by Wizard 2 of Figure 5.



Goals: optimize_coverage
Cell selection: 1 , 2 , 3 , 4 , 5
Time of day: Morning (06 - 10

More cases than 1
Confidence level [0-1] higher than 0.01
Automatic confirmation ☐

Goal: optimize_coverage
In: [1,2,3,4,5]
Time: 06-10

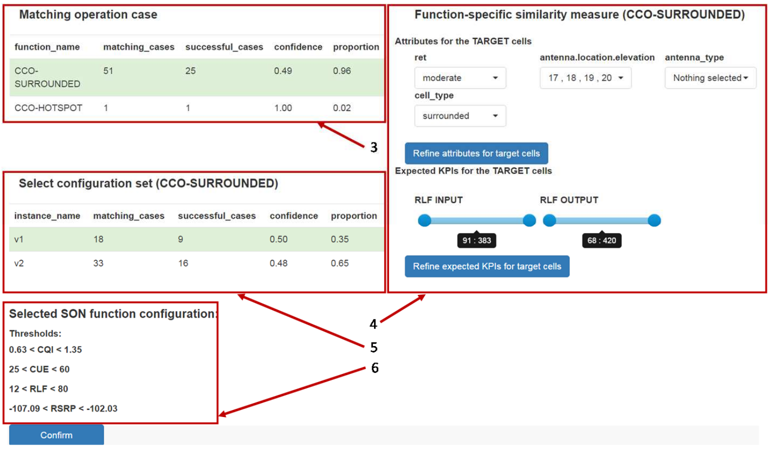
Fig. 5. A snapshot of an example to define objective specific similarity measure via specifying corresponding goal, network context, and numerical boundaries.

In this use case, the user wants to find suitable operations for optimizing coverage for an area including, e.g., five cells (1, 2, 3, 4, and 5), and with a time range from, e.g., 6AM to 10AM. In the rule definition form (Wizard 2), numerical boundaries can be selected to further exclude irrelevant search results. For example, the minimum amount of cases per retrieved SON function is set to 1. The minimum confidence level (success ratio to achieve a goal) is set to 0.01. Here, the boundaries have low values in order to maximize the amount of operation cases in the search results.

The above information is used to define the objective-specific similarity measure. With this similarity measure, Wizard 3 of Figure 6 searches and shows the result of the self-operation cases matching the objective-specific similarity measure. The self-operation analysed approximately 500 cases in several seconds. Two SON functions are then identified in these self-operation cases. *CCO-SURROUNDED* is a CCO function instance that optimizes a source cell surrounded completely by its 1st-tier neighbor cells. *CCO-HOTSPOT* is a CCO function instance that optimizes a hotspot source cell. The columns in Wizard 3 describe the name of the function (*function_name*), the total amount of self-operation cases matching the target that the SON function has been involved with (*matching_cases*), the amount of successful cases achieved the target (*successful_cases*), the success ratio of the matching cases (*confidence*), and the proportion of the number of matching cases of the function to the total number of the matching cases of all functions (*proportion*). The function *(CCO-SURROUNDED)* and its 51 self-operation cases in the first row are selected as this function has the highest proportion value.

7.2.2 Function-Specific Similarity Definition and Search for Configuration

For the selected function *CCO-SURROUNDED*, the available function-specific similarity attributes are antenna elevation, antenna type, and cell type of the source cells, as well as their value ranges for RLF. The value ranges indicate the values before (*RLF INPUT*) and after (*RLF OUTPUT*) the operation case has been executed.



Matching operation case

function_name	matching_cases	successful_cases	confidence	proportion
CCO-SURROUNDED	51	25	0.49	0.96
CCO-HOTSPOT	1	1	1.00	0.02

Function-specific similarity measure (CCO-SURROUNDED)

Attributes for the TARGET cells

ret: moderate

antenna.location.elevation: 17 , 18 , 19 , 20

antenna_type: Nothing selected

cell_type: surrounded

Expected KPIs for the TARGET cells

RLF INPUT: 91 : 383

RLF OUTPUT: 68 : 420

Select configuration set (CCO-SURROUNDED)

instance_name	matching_cases	successful_cases	confidence	proportion
v1	18	9	0.50	0.35
v2	33	16	0.48	0.65

Selected SON function configuration:

Thresholds:

$0.63 < CQI < 1.35$

$25 < CUE < 60$

$12 < RLF < 80$

$-107.09 < RSRP < -102.03$

Fig. 6. A snapshot of the objective-based function selection, the function-specific similarity definition, and the function configuration selections.

Wizard 4 of Figure 6 collects the relevant attributes of the source cells to define the first part of the function-specific similarity measure for the selected function. With this first part of the similarity measure, the 25 matching self-operation cases (found by Wizards 3) are further filtered. The remaining part of the function-specific similarity measure is then defined by extracting the relevant KPI information of the further matched self-operation cases. With this fully defined function-specific similarity measure, the first configuration set (*v1*) of the 9 self-operation cases is selected, as shown in Wizard 5. The actual configuration values of the selected configuration set are presented in the configuration list shown in Wizard 6.

The selected configuration can now be confirmed (automatically or manually) and configured to the selected function so that the selected function can make its decision (e.g., CM output) accordingly. If not, the function-specific similarity measure can be updated to find another configuration set, or another function can be re-selected and the process is repeated from Wizard 3.

7.2.3 Refining Function Specific Similarity Definition and its Match Results

In addition to the automatic confirmation of a matching result, a user can also take the manual control of the confirmation when needed. In this mode, the user can explore the current search results to see if more accurate results for the context are needed. Figure 7 demonstrates such a situation, in which the user has decided to refine the results by reducing the value range of the *RLF OUTPUT* shown in Entry A and the elevation value shown in Entry B.

Matching operation case

function_name	matching_cases	successful_cases	confidence	proportion
CCO-SURROUNDED	15	9	0.60	0.88
CCO-HOTSPOT	1	1	1.00	0.06

Select configuration set (CCO-SURROUNDED)

instance_name	matching_cases	successful_cases	confidence	proportion
v2	4	3	0.75	0.27
v1	11	6	0.55	0.73

Selected SON function configuration:

Thresholds:

0.63 < CQI < 1.35

15 < CUE < 40

20 < RLF < 80

-107.09 < RSRP < -102.03

Function-specific similarity measure (CCO-SURROUNDED)

Attributes for the TARGET cells

ret: moderate

cell_type: surrounded

antenna.location.elevation: 17

antenna_type: Nothing selected

Expected KPIs for the TARGET cells

RLF INPUT: 91 : 383

RLF OUTPUT: 68 : 185

Fig. 7. A snapshot of the match result and configuration via refining the values of the function specific similarity measure.

Figure 7 shows how the amount of cases has reduced (with respect to Figure 6.) for the *CCO-SURROUNDED* and for its configurations *v1* and *v2*, shown in Entry C and

Entry D. The confidence levels of these elements have increased. Configuration set *v2* is now automatically selected as the preferred configuration for the user to confirm. The user could even pick and confirm the configuration set *v1*, if the user would prefer so instead.

8 Conclusion and Discussion

This work provides a solution to define and apply two types of similarity measures for two self-operation use cases. The use cases are as follows: Self-operation uses its learned operation experiences to answer the question "Given any objective and its corresponding network context, what function should be used to achieve it?" Self-operation uses its learned operation experiences to answer the question "Given any objective and network context, what should be the suitable configuration for that function so that it could achieve the objective?"

The solution consists of the self-operation architecture for similarity measure definition and application, the data elements needed by the two types of similarity measures, and their definition and application procedures.

This work also describes a demonstrator implementation of self-operation, which learns operation experiences into self-operation cases and applies operation experiences for certain self-operation use cases. These use cases receive their configuration or instruction from self-operation with the means not shown in the demo architecture. The demonstrator uses an LTE simulator and SON function instances as a source of data, where the LTE simulator simulates a whole LTE network. In the experiments, the demonstrator defines an objective-specific similarity measure based on the given network context, objective, and rule. It then matches the corresponding self-operation cases with the defined objective-specific similarity measure. The best suitable function is extracted from the matching self-operation cases by the demonstrator. The demonstrator then automatically defines a function-specific similarity measure based on the selected function and the given network context, objective, and rule. The more relevant self-operation cases are further matched with the defined function-specific similarity measure by the demonstrator. The (best) suitable function configuration is extracted from the further matched self-operation cases by the demonstrator. The function can then be configured with the suitable configuration and activated to achieve the given objective.

The experimental results of the demonstrator (including the implemented solution, proposed by this paper) show the concept of self-operation (including the solution) work well as expected. This self-operation scales well (with respect to use of a distributed database, MongoDB) and works automatically while being able to interact with human operator through UI during the network operations.

The network operations of 5G networks are expected to have much more automation capabilities when compared with the current network operations. The proposed solution for self-operation by this paper serves naturally as an important part of the 5G network operations.

As the future work, the current demonstrator is expected to be enhanced to support the direct configuration of the functions and the network from UI. In addition, a machine to machine interface is expected to be added to the demonstrator so that self-operation can directly control and configure the functions and the network.

9 References

1. 3GPP, "Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Self-configuration and self-optimizing network use cases and solutions (Release 8)," 3GPP TR 36.902 V1.0.1, Sept. 2008.
2. NGMN, "NGMN Use Cases related to Self Organising Network, Overall Description," Deliverable (https://www.ngmn.org/uploads/media/), NGMN Alliance, Dec. 2008.
3. Nokia, "Business aware traffic steering," White Paper of Nokia Networks, http://networks.nokia.com/sites/default/files/document/nokia_traffic_steering_white_paper.pdf, 2013.
4. 5G-PPP, "5G Empowering Vertical Industries," Brochure (https://5g-ppp.eu/wp-content/uploads/2016/02/BROCHURE_5PPP_BAT2_PL.pdf), February 2016.
5. Haitao Tang and Kaj Stenberg, "Self-Operation of a Network," Proceedings of IEEE DataCom 2016, pp. 647-653, Aug. 2016.
6. Agnar Aamodt and Enric Plaza, "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches," Artificial Intelligence Communications 7: 1, pp. 39-52, 1994.
7. Ahmad Rawashdeh and Anca L. Ralescu, "Similarity Measure for Social Networks– A Brief Survey," Proceedings of Modern AI and Cognitive Science Conference (MAICS), pp. 153-159, April 2015.
8. Djamel Guessoum, Moeiz Miraoui, and Chakib Tadj, "Survey of Semantic Similarity Measures in Pervasive Computing," International Journal of Smart Sensing and Intelligent Systems, VOL. 8, NO. 1, pp.125-158, March 2015.
9. Wael H Gomaa and Aly A Fahmy, "A Survey of Text Similarity Approaches," International Journal of Computer Applications, vol. 68, no. 13, pp. 13-18, April 2013.
10. TRAI, "The Standards of Quality of Service for Wireless Data Services (Amendment) Regulations," Regulation (http://www.trai.gov.in/Content/Regulation/1_0_RegulationUser.aspx), Telecom Regulatory Authority of India, New Delhi, India, July 2014.
11. 3GPP, "Key Performance Indicators (KPI) for Evolved Universal Terrestrial Radio Access Network (E-UTRAN): Definitions, v13.0.0," 3GPP TS32.450, Jan 2016.
12. TRAI, "The Indian Telecom Services Performance Indicators, July - September, 2015," Indicator Report (http://www.trai.gov.in/WriteReadData/PIRReport/Documents/Indicator_Reports.pdf) Telecom Regulatory Authority of India, New Delhi, India, pp.61-84, Feb. 2016.
13. NGMN, "NGMN Informative List of SON Use Cases," An Annex Deliverable (https://www.ngmn.org/uploads/media/NGMN_Informative_List_of_SON_Use_Cases.pdf) , NGMN Alliance, pp.6-47, April 2007.
14. Tobias Bandh, Haitao Tang, Henning Sanneck, and Chistoph Schmelz, "SON Operation," Chapter 9 (pp.322-356), LTE Self-Organizing Networks (SON), WILEY, ISBN 978-1-119-97067-5, 2012.
15. Ingo Viering, Martin Döttling, and Andreas Lobinger, "A mathematical perspective of self-optimizing wireless networks," Proc. ICC'09, p.1 ff., Dresden, Germany, June 2009.

Publication IV

Kasper Apajalahti, Eero Hyvönen, Juha Niiranen, and Vilho Räisänen. Combining Ontological Modelling and Probabilistic Reasoning for Network Management. *Journal of Ambient Intelligence and Smart Environments*, IOS Press, volume 9, number 1, pages 63–76, DOI https://doi.org/10.3233/AIS-160419, January 2017.

© 2017 IOS Press

Reprinted with permission.

Journal of Ambient Intelligence and Smart Environments 1 (2017) 1
IOS Press

Combining ontological modelling and probabilistic reasoning for network management

Kasper Apajalahti [a], Eero Hyvönen [a], Juha Niiranen [b], and Vilho Räisänen [c]
[a] *Aalto University, Dept. of Computer Science, first.last@aalto.fi*
[b] *University of Helsinki, Dept. of Mathematics and Statistics, first.last@helsinki.fi*
[c] *Nokia Bell Labs, first.last@nokia-bell-labs.com*

Abstract. Advanced automation is needed in future mobile networks to provide adequate service quality economically and with high reliability. In this paper, a system is presented that takes into account the network context, analyses uncertain information, and infers network configurations by means of probabilistic reasoning. The system introduced in this paper is an experimental platform integrating a mobile network simulator, a Markov Logic Network (MLN) model, and an OWL 2 ontology into a runtime environment that can be monitored via a Resource Description Framework (RDF) -based user interface. In this approach, the OWL ontology contains a semantic representation of the relevant concepts, and the MLN model evaluates elements of uncertain information. Experiments based on a prototype implementation demonstrate the value of semantic modelling and probabilistic reasoning in network status characterization, optimization, and visualization.

Keywords: Network management, 5G, semantic modelling, probabilistic reasoning, ontologies, Markov Logic Networks

1. Introduction

Mobile networks are a part of a critical infrastructure, facilitating wireless access to Internet with all its services. The number of users, devices, and applications is expected to continue to grow [9]. A dramatic increase in the number of Internet of Things (IoT) endpoints is expected [24]. The advent of IoT is one of the most important drivers for fifth-generation (5G) mobile networks. Consequently, 5G networks need to cater for an increase in data volume and massive growth in the number of terminals served. Furthermore, 5G networks also need to support services with requirements of a new kind, such as ultra-low latency and high reliability [24].

Networks need to be configured optimally to provide customers with high service quality at a reasonable price. Manual network configurations employed in the past make per-cell optimization unfeasible in practice. Given the growth in the complexity of networks, more automation is needed from Operations and Support Systems (OSS) [19], a collective term for capabilities used for managing mobile networks.

State-of-the art operability in fourth-generation networks (4G) is based on the concept of Self-Organizing Networks (SON), which amounts to a set of closed-loop agent systems reacting to measurements, typically by means of a fixed model [19]. Each agent is an implementation of an operability use case. The challenge with this approach is creating and maintaining up-to-date models in view of geographic and temporal variety across cells in radio access networks.

The application of autonomic computing to various areas has been envisioned to the mobile networks with new architectures and software components [14,15,19]. These capabilities are based on architectures which support learning and adaptation to context-specific situations by means of Machine Learning (ML) algorithms, knowledge representation, and reasoning with knowledge bases [14,15,19].

1876-1364/17/$35.00 © 2017 – IOS Press and the authors. All rights reserved

Machine Learning is expected to increase the level of automation in the OSS area, for example, by improving the analysis of traffic patterns and cell-related data to learn statistical correlations. Another application of ML is SON verification [8], where learning capability is used for identifying effective solutions. The output of a stand-alone ML system can be characterized as hypothetical in contrast to the deterministic results of a traditional rule-based system. This paper will argue that an effective approach to utilizing ML in the complex environment of future mobile networks involves both classical and probabilistic reasoning.

In this paper, a new approach is proposed to automate mobile network management by using statistical relational learning with a Markov Logic Network model (MLN) [32] for handling uncertainty in mobile network analysis. An OWL 2 ontology is used to complement the MLN model by providing global meaning and a semantic description of the system. The ontology is currently used as a semantic storage with a SPARQL interface for the MLN model and for an RDF-based faceted graphical user interface (GUI). In summary, the practical reason for combining semantic technologies with probabilistic reasoning is the decrease the complexity in monitoring of the MLN model and to making it dynamically modifiable.

To enable a human user to monitor and understand complex network management operations, intelligent storage and presentation of data is needed [22]. In this work, the OWL 2 model targets this issue with a semantic representation of the network state and the related automated configurations. The use of a formal knowledge model which supports automated reasoning reduces the need for case specific software design and implementation, and provides a mechanism for assessing the degree of consistency of the relevant models. Suitably chosen probabilistic reasoning accommodates partial and conflicting data, which would be challenging to address using subsets of First Order Logic (FOL) alone. Additionally, it has been argued that knowledge models bring benefits as a basis of future telecommunication systems both in view of systems design and from the perspective of value networks [30].

The paper is structured as follows: First, the approach is presented in Section 2 and an overview of the system is given in Section 3. After that, Section 4 presents the concept of MLNs and how they are applied to our implementation. Section 5 describes the OWL 2 model and Section 6 presents the RDF-based GUI used to monitor the MLN reasoning. Section 7 presents experiments with the system, such as the evaluation of the MLN reasoning, the affection of the MLN model size to execution times (MLN reasoning and SPARQL queries), and use case examples of the GUI. Finally, Section 8 discusses related work and Section 9 concludes the paper and presents ideas for future work.

2. Approach

The radio access network of a mobile network domain is composed of cells. In terms of hardware, cells have base stations having dedicated transmitter/receiver (TRX) units, each serving a sector of the cell. Each base station serves a number of terminal devices, known as User Equipment (UE) in industry terminology. Terminal devices may roam across cells and their traffic characteristics (uplink/downlink) traffic may vary depending on the services used.

Mobility and traffic characteristics exhibit diurnal and weekday variations. Additionally, the patterns vary according to the location of the cell in question; a suburban cell may be expected to have the highest loading outside of business hours, whereas the opposite is true for a downtown cell in a business district. There are longer term trends relating to varying usage patterns of services and residential/business user densities. There are also short-term variations due to special events like concerts or sports events.

Given the complexity and variability of network loading, it is difficult to try to optimize network status as one entity. In 4G networks, SON agents execute specific use cases in a limited scope (one or a few cells), performing automated configurations based on measurements. A run-of-the-mill implementation of SON agent utilizes a fixed model. This works well when the model reflects the situation of the cell, but becomes a bottleneck in tailoring the behaviour of the agent on cell level when cell characteristics vary.

Automated adaptation is the next step beyond the fixed model. An approach utilizing Case-Based Reasoning (CBR) has been utilized [33,7]. Interpolation across cases is an enhancement, but is nevertheless limited by the case base.

We study the use of probabilistic reasoning in coping with complexity and adaptability. Instead of using a rule or case base, this system is driven by formulae for reasoning performed over facts about system status. In addition to raising the abstraction level of implementations, a suitably chosen method also supports incomplete and conflicting data, which classical reasoning cannot address.

3. System Overview

3.1. Components

The architecture view of the system used in this article is depicted in Figure 1. The bottom component is the data source, e.g. cellular network or a network simulator. In this paper, we demonstrate the system with a Long Term Evolution (LTE) simulator.

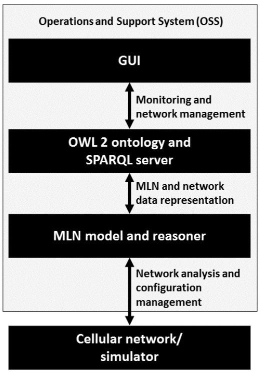


Fig. 1. System architecture.

The simulator interacts with the MLN model which analyses the network data and provides data for decision support in the configuration management. The OWL 2 ontology and reasoner are used to create a semantic representation of the network and MLN data and to describe the data as a graph. The ontology is stored in a SPARQL server providing access to the ontology. The GUI is used to monitor and manage the system. In a real network, the three topmost components would belong to an OSS system which manages the network.

3.2. Scenario

The simulation scenario consists of a small urban area (a diameter of 5 km) with 2000 terminals and 32 sectors. In the context of LTE, term "cell" is also used as a synonym for a sector of a base station. The simulator sends performance data that contains measurements from key performance indicators (KPIs) for various cases. KPIs utilized in the system are the number of connected terminals (CUE) per cell, the channel quality indicator (CQI) distribution vector for measuring the signal quality of a cell, and the radio link failures (RLF) for measuring the amount of connection failures per cell. The simulator receives configuration data that contain possible changes in the transmission power (TXP) and angle (remote electrical tilt, RET) of a cell antenna. The data are sent periodically in 15 minute intervals in simulation time.

The MLN model analyses the CUE, CQI, and RLF measurements and infers posterior marginal probabilities for potential network configuration changes in order to optimize the CQI and RLF metrics. A fit of model parameters of the MLN reasoner is performed to historical performance data and past executed configuration actions. The OWL 2 ontology is constructed by transforming the MLN reasoner model into a semantic representation that can be utilized as an RDF graph in a SPARQL endpoint. An operator interface for managing the system and the underlying mobile network is implemented on top of the SPARQL endpoint with HTML5 based GUI.

3.3. Data sequence

Figure 2 depicts the data sequence between system components starting with a measurement report from the LTE simulator on the left. The simulator sends the MLN reasoner performance management (PM) data (1) periodically. The reasoner processes data into its evidence (classified KPI values, such as low, moderate, and high CQI) and infers the probabilities for network actions (2) from them. Then, the reasoner sends a set of action proposals (configurations with high probabilities) to the simulator (3). In addition, the MLN reasoner sends further data, including evidence, formulae, and action proposals to the ontology processor (4), which processes the data into RDF and populates the ontology with new network- and MLN-related instances (5). The RDF graph is then uploaded into a SPARQL server based on Fuseki[1] (6). The server dynamically updates facets from the ontology with SPARQL update scripts (7). The user can monitor the system via the GUI that interacts with the SPARQL endpoint in order to retrieve network- and MLN-related data from the ontology (8) and to update MLN formulae (9). Similarly, the MLN model queries the SPARQL endpoint in order to retrieve an updated model (10).

---

[1] https://jena.apache.org/documentation/serving_data/

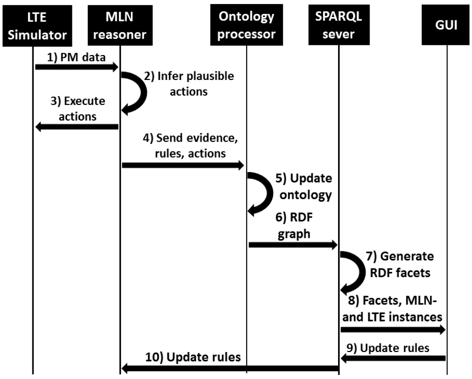


Fig. 2. Data sequence diagram for managing a mobile network (simulator) with system components.

4. Markov Logic Network model

4.1. Definition

MLNs allow uncertain and contradictory knowledge in a first-order logic (FOL) subset model by introducing a weight parameter for each formula in the FOL subset knowledge base. The weighted set of formulae defines a template for a Markov network, where the features and feature weights are determined by the formulae and formula weights.

Definition 4.1 *A Markov logic network L is a set of pairs (F_i, w_i), where F_i is a first-order formula and w_i is a real-valued weight parameter. Together with a set C of constant terms, over which the formulae in L are applied, it defines a Markov network $M_{L,C}$ with a binary variable for each possible grounding of each predicate appearing in L and a feature for each possible grounding of each formula in L. The value of the feature corresponding to a grounding of formula F_i is 1 if the ground formula is true, and 0 otherwise. The weight of the feature is w_i, the weight associated with F_i in L.*

Each state of the variables in a Markov network $M_{L,C}$ represents a possible world, for example a truth assignment for each of the ground atoms for (L, C). The probability distribution over possible worlds $x \in \mathcal{X}$ specified by $M_{L,C}$ is defined in Eq. (1).

$$P_{M_{L,C}}(X = x) = \frac{1}{Z} \exp\left(\sum_i w_i n_i(x)\right), \qquad (1)$$

where $n_i(x)$ is the number of true groundings of F_i in x and Z is a partition function given by $Z = \sum_{x\in\mathcal{X}} \exp\left(\sum_i w_i n_i(x)\right)$. Intuitively this means that the weights of the true ground formulae give the logarithmized factors of the distribution function. If two worlds differ only on a single ground formula, then the weight of the formula gives the logarithmic odds of choosing one world over the other.

4.2. Inference

Given an MLN L and a set of constants C, the most likely state of the world can be inferred, $\hat{x} \in \mathcal{X}$, according to the joint probability given in Eq. (1), such as the most likely truth assignment for the variables in $M_{L,C}$. Also, the marginal probability for each of the variables can be inferred.

However, a typical inference task is to deduce the most likely state or marginal distributions for a subset of the variables, called query variables, using the values of (some of) the rest of variables as evidence. Given the values x_E of a set of evidence variables $X_E \subset X$, the most likely state $\hat{x}_Q$ for query variables $X_Q \subset X$ is inferred according to Eq. (2).

$$\begin{aligned} &P(X_Q = x_Q | X_E = x_E) \\ &\quad = \frac{P(X_Q = x_Q, X_E = x_E)}{P(X_E = x_E)} \\ &\quad = \frac{\sum_{x\in\mathcal{X}_Q\cap\mathcal{X}_E} P_{M_{L,C}(X=x)}}{\sum_{x\in\mathcal{X}_E} P_{M_{L,C}}(X = x)}, \end{aligned} \qquad (2)$$

where $\mathcal{X}_\varphi$ is the set of worlds where $X_\varphi = x_\varphi$ holds for $\varphi = E$ and $\varphi = Q$.

As Richardson and Domingos show [32], exact inference over an MLN model is infeasible in practice. Instead, they introduce an efficient approximation algorithm for this problem using stochastic simulation.

4.3. Weight learning

The weights of an MLN model can be learned from one or more databases. A database is effectively a Boolean vector stating the observed truth values for each ground predicate. Given a database (or databases), the most likely weight values are those that maximize the probability of the database given in Eq. (1). The standard method for maximum likelihood estimation (MLE) is the gradient descent method. However, computing the gradient of the probability in Eq. (1) requires computing the estimated number of true groundings for each formula, which is infeasible, as shown by Richardson and Domingos [32]. They propose a solution using gradient descent for optimizing the pseudo-likelihood [4] of the ground Markov network.

Typically, in optimizing the weights of an MLN, it is known in advance which of the predicates in the model will be used for querying and which will be used as evidence. In that situation, Lowd and Domingos [23] propose a discriminative learning solution where they use a gradient descent algorithm to optimize the likelihood of the conditional probability, as given in Eq. (2), instead of the full joint probability, given in Eq. (1). They show that this solution outperforms the solution based on pseudo-likelihood.

4.4. Application in OSS setting

4.4.1. Structure of the model

The MLN model of the system is defined in terms of three types of predicates:

- *Context predicates* reflect the current status of the relevant network scope and its environment. A context predicate can indicate, for example, that a KPI value for a cell is currently below the acceptable level, or that two cells are neighbors in the network topology.
- *Objective predicates* indicate required changes to KPI values to achieve performance targets defined by the operator. For example, an objective predicate can indicate that a particular KPI value for some cell is too low and needs to be increased.
- *Action predicates* indicate changes to network configuration parameter values.

Each predicate represents an attribute of a cell in the network or a relation among the cells. The domain of a predicate can be either the set of cells or an n-ary Cartesian product of the set of cells.

The MLN model is composed of formulae with these predicates. The formulae are intended to describe a correlation between a set of $Objectives$ and a set of $Actions$ in a certain $Context$. A typical inference task is to query for appropriate actions using the current context data and objective requirements as evidence. Therefore, the formula format is defined as

$$\mathcal{C} \Rightarrow (\mathcal{O} \Leftrightarrow \mathcal{A}). \tag{3}$$

Above, $\mathcal{C}$, $\mathcal{O}$ and $\mathcal{A}$ are sets of one or more context, objective and action predicates, respectively.

The values of the context predicates are computed from the PM data measured from the LTE network. For numerical data, the measurements need to be first classified, for example to low, moderate, and high classes. When values for the action predicates are inferred, the values of the objective predicates are derived from the classified PM data according to performance requirements set by the network operator. For example, if the CQI value measured for a cell is classified as "low", the derived objective could be to increase the CQI for that cell. On the other hand, when the weights of the MLN model are learned from a database of historical PM and configuration data, the objective predicate values are computed by analyzing realized changes in PM values. In a similar way, the values of the action predicates are computed from changes in configuration data.

After the weights are defined for the MLN model (either learned iteratively or set by an expert), the most likely state of the action predicates can be inferred. Furthermore, the network can be configured according to the inferred action predicates values, because they give the best possible solution with respect to the model. The model weights can still be adjusted by configuring the network with respect to the marginal probabilities of actions and monitoring performance changes. In this manner, the model is dynamically adapted to the network situation.

4.4.2. Inference example

To illustrate the usage of the MLN model, the inference is demonstrated with an example.

Example Let L be a simple MLN model consisting of the weighted formulae defined in Table 1.

w_i	F_i
2.4	$I(c, Rlf, High) \Rightarrow (O(c, Rlf, Dec) \Leftrightarrow A(c, Txp, Inc))$
0.5	$I(c, Rlf, High) \Rightarrow (O(c, Rlf, Dec) \Leftrightarrow A(c, Ret, Dec))$
0.9	$(N(c,d) \wedge I(c, Cqi, Low)) \Rightarrow (O(c, Cqi, Inc)$ $\Leftrightarrow (A(c, Ret, Dec) \wedge A(d, Txp, Inc)))$

Table 1

Examples of weighted formulae in the MLN model.

Here the variables c and d denote cells in the mobile network. I is a context predicate indicating KPI category, N is a context predicate indicating the neighborship of two cells, O is an objective predicate indicating a change in a KPI value and A is an action predicate indicating change in a parameter value. Suppose that there is a mobile network with two neighbor cells named $C1$ and $C2$ and that low CQI value for cell $C1$

and a high RLF value for $C2$ is measured. This information is used to infer proper configuration actions to get the CQI and RLF values to a normal level. The MLN reasoner is used to query the MLN model L for marginal probability distributions for action proposals $A(c, Txp, Inc)$, $A(c, Txp, Dec)$, $A(c, Ret, Inc)$, and $A(c, Ret, Dec)$ for each cell c given the evidence:

$$
\begin{aligned}
E = \{&N(C1, C2), I(C1, Cqi, Low),\\
&I(C2, Rlf, High), O(C1, Cqi, Inc),\\
&\qquad\qquad O(C2, Rlf, Dec)\}
\end{aligned}
$$

An example of the reasoning output is shown in Table 2, which shows inferred marginal probabilities for cell configurations given the model L and the evidence E. The output indicates that decreasing RET for cell $C1$ and increasing TXP for cell $C2$ are the most likely actions to achieve the objectives according to the model.

Action	$P(Action\|L, E)$
$A(C1, Txp, Inc)$	0.35
$A(C1, Txp, Dec)$	0.30
$A(C2, Txp, Inc)$	0.70
$A(C2, Txp, Dec)$	0.14
$A(C1, Ret, Inc)$	0.28
$A(C1, Ret, Dec)$	0.41
$A(C2, Ret, Inc)$	0.29
$A(C2, Ret, Dec)$	0.41

Table 2

Marginal probabilities for cell configurations.

4.4.3. *Generating formulae for the model*

The CQI, CUE, and RLF KPIs are measured for each cell. RLF and CQI characterize the performance and CUE is an indicator of the cell load. To facilitate usage across varying load levels, RLF value is normalized by dividing it with the CUE value. Furthermore, to get a scalar value from the CQI distribution vector, the average channel efficiency is computed using the CQI vector values as weights for the channel efficiency values of each CQI class as defined in the technical specification [13]. For simplicity, these two aggregate metrics are referred to as RLF and CQI, respectively.

For the MLN model these three metrics, CUE, RLF and CQI, were classified as low, moderate, or high. The context predicates of the model describe classified values for these metrics for a cell variable. Furthermore, another context predicate was introduced describing the neighborship status of pairs of cells.

Objective predicates were introduced for the RLF and CQI metrics. For inference, the values of the predicates were computed according to operator goals from the context predicate values so that if the RLF value was high, the objective was to decrease the value, and if the CQI value was low, the objective was to increase the value. For weight learning the objective predicate values were computed from the realized changes in the numeric RLF and CQI values.

For the network configuration parameters, RET and TXP, action predicates are introduced to indicate a fixed-size decrease or increase of the value.

Algorithm 1 shows how weighted formulae are generated. The pseudocode illustrates updating the set of formulae; after a combinatorial generation of formulae, all unachievable formulae are removed from the model. A formula is unachievable, if a KPI is included in the objective (e.g. decrease RLF) but not in the context (e.g. high RLF). A formula is also unachievable, if it has a KPI value in the context that violates an operator goal (e.g. high RLF), but has not a corresponding objective to change that value (e.g. decrease RLF).

Algorithm 1. Generating weighted formulae for the model

```
Create all combinations of formulae with given
KPIs, operator goals, and configuration parameters
for formula in model do
    If KPI in objective not contained in context
        Then Remove formula from the model
    If KPI in context does not achieve operator goal
    and corresponding objective does not exist
        Then Remove formula from the model
end for
```

In addition to creating weighted formulae, some constraints are added to the model, such as: 1) each KPI must have exactly one value, 2) a KPI value can not be both increased and decreased, 3) a parameter value can not be further decreased (increased) from its minimum (maximum) level.

5. OWL 2 Model

The OWL2 ontology[2] has been designed to support GUI based on reasoning. Because of this, the ontology is minimal and use case driven rather than an all-encompassing ontology of the entire mobile network

---

[2] https://www.w3.org/TR/owl2-overview/

domain. The ontology contains mobile network concepts such as KPIs and cells on the one hand, and MLN model concepts like rules, actions, and parameters on the other. The MLN model supports learning of the most effective action to achieve a particular goal in a specific network context. The results of learning and as well as evidence (input data used by MLN) are reflected in the ABox of the OWL2 ontology.

The MLN model concepts are linked to mobile network concepts, which helps exploration. Furthermore, this link clarifies the interrelations between the MLN model on the one hand and network state and MLN action inference on the other. An added benefit is that the governance of the MLN model is easier as the network-related metadata is defined for it.

5.1. Evidence and action proposals

In Figure 3, mobile network concepts are described together with MLN evidence and action proposals. The *Cell* is the most fundamental class in the model and has properties *hasKPI* to its performance metrics (instances of the class *CellKPI*) and *hasParameter* to its configuration parameters (instances of the class *CellParameter*). *CellKPI* has a crisp description for its value (*KPIValue*), such as low, moderate, or high, which is defined in the MLN model. Also, according to the MLN model, a KPI might have an objective defined using the class *KPIObjective*. *KPIObjective* in turn has some *EventImpact* defining the direction of the change of an impact (increase or decrease). The *CellParameter* can have an *ActionProposal*, if the parameter needs to be adjusted with respect to the MLN inference. The *ActionProposal* also has a relation to an instance of the class *EventImpact* to describe its impact direction.

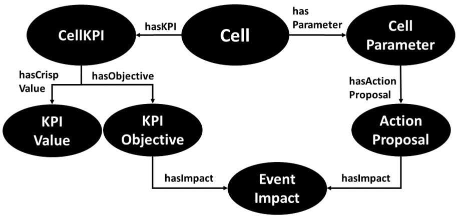


Fig. 3. MLN evidence, action proposals, and their cell-related concepts in the ontology TBox.

5.2. Formulae

In addition to the evidence and action proposals, the weighted formulae of the MLN model are represented with concepts and mapped to mobile network concepts in the OWL 2 TBox, as shown in Figure 4. In the OWL 2 model, the "rule" term is used as a synonym for a formula. The *MLNRule* class defines a formula, which has a numerical value *hasRuleWeight* defining its weight and relations to formula classes *RuleContext*, *RuleObjective*, and *RuleAction*. The figure also depicts that the formula classes are bound to network classes *CellParameter* and *CellKPI*. A *RuleAction* has a relation to an instance of a *CellParameter* (such as *Txp*) and that is bound to an instance of *EventImpact* class (such as *Increase*). Similarly, *RuleObjective* and *RuleContext* have relations to *CellKPI* instances, which in turn contain instances of an *EventImpact* (in case of *RuleObjective*) and *KPIValue* (in case of *RuleContext*).

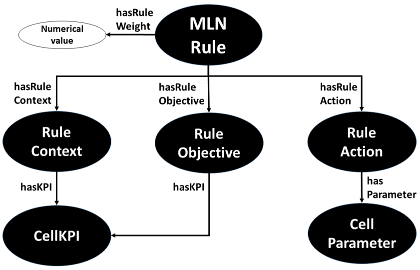


Fig. 4. Rules (same as formulae in the MLN model) and their cell-related concepts in the ontology TBox.

5.3. Ontology query

To demonstrate the ontology, Listing 1 shows RDF triples representing data regarding the cell with ID 1. The cell has information about the neighboring cells, amount of neighbors, KPI instances (:hasCqi, :hasCue, and :hasRLF are subproperties of :hasKPI), parameter instances (:hasRet and :hasTxp are subproperties of :hasParameter) and facet values that are generated on-the-fly in the SPARQL server. The facets designed for the cell and MLN formula specific data are explained in more detail in the next section.

```
:Cell1 a :Cell,
  owl:NamedIndividual ;
:hasCellId 1 ;
:hasAmountOfNeighbors 3 ;
```

```
:hasNeighbor :Cell25,
  :Cell2, :Cell3 ;
:hasCqi :Cqi1 ;
:hasCue :Cue1 ;
:hasRlf :Rlf1 ;
:hasRet  :Ret1 ;
:hasTxp :Txp1 ;
:hasFacetCqi :LowCqi ;
:hasFacetCue :LowCue ;
:hasFacetNeighbors :Few ;
:hasFacetRlf :HighRlf .
```

Listing 1: RDF representation of Cell 1 and its related data.

The ontology can be queried with a SPARQL query pattern shown in Listing 2. The query returns cell-specific data for cells fulfilling the given facet selections (high RLF is chosen in this example). The return variables of the query are cell URI (?*c*), link property (?*link*) to a related node (link to a KPI or parameter instance of a cell), column name in the GUI (?*col*), or column value (?*val*).

The query has three union parts, which find column properties (:hasColumnProperty and :hasColumnDataProperty) that specify instances represented in the tabular view of the GUI which is explained in more detail in the next section. The first and second union parts find column properties directly related to the cell, such as :hasFacetRlf and :hasAmountOfNeighbors. The last union part finds instances from the related nodes, such as an objective to increase a KPI value of a cell. The third part also contains a filtering condition that prevents search results from neighboring cell instances.

```
SELECT DISTINCT ?c ?link ?col ?val
{
  ?c a :Cell .
  ?c :hasFacetRlf :HighRlf .
{
  ?col rdfs:subPropertyOf
    :hasColumnProperty .
    ?c ?col ?val .
} UNION {
  ?col rdfs:subPropertyOf
    :hasColumnDataProperty .
  ?c ?col ?val .
} UNION
{
  ?c ?link ?k_or_p .
  FILTER NOT EXISTS
        {?k_or_p a :Cell}
  ?k_or_p ?col ?obj_or_act .
  ?obj_or_act ?has_impact ?val .
  ?has_impact rdfs:subPropertyOf
    :hasColumnProperty .
} }
ORDER BY ?c
```

Listing 2: SPARQL query for cell-specific data with facet selection high RLF.

The SPARQL query returns all cells having a high RLF value. Cell 1 is fulfilling this condition and it is shown as an example in the table 3. Each column depicts a SPARQL return variable (cell URI (?*c*), link property (?*link*), column name (?*col*), or column value (?*val*)).

Cell (?*c*)	Link (?*link*)	Column (?*col*)	Value (?*val*)
:Cell1		:hasAmountOf Neighbors	3
:Cell 1		:hasCellId	1
:Cell1		:hasFacetCqi	:LowCqi
:Cell1		:hasFacetCue	:LowCue
:Cell1		:hasFacetRlf	:HighRlf
:Cell1	:hasCqi	:hasObjective	:Increase
:Cell1	:hasRlf	:hasObjective	:Decrease

Table 3

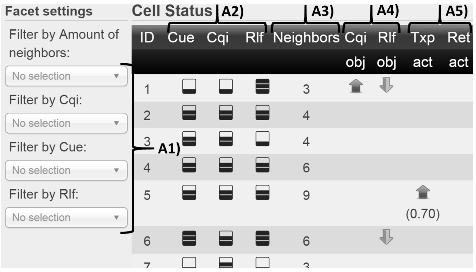
A snippet of SPARQL query results for the example query. Cell 1 fulfils the facet condition high RLF.

The SPARQL query pattern shown in Listing 2 can be used with slight modifications to retrieve MLN formula-specific data. MLN formulae have own column properties defining retrieved values in a different tabular view and facets to filter search results.

6. Graphical User Interface

The system presented in this paper is monitored via a faceted RDF-based GUI [1] that visualizes the ontology instances processed using the MLN model and supports exploration. The purpose of the GUI is to provide the end-user with informative and interactive tools for evaluating the MLN functionality. Thus, views are implemented to present MLN reasoner-related cell states and MLN formulae.

Figures 5 and 6 show cell states in two alternative visualizations: tabular and graph-oriented. With these views, the user examines how MLN evidence (KPI values and KPI objectives) affects the MLN reasoning outcome (action proposals). In the tabular visualization (Figure 5), the rows depict cell instances and the columns their attributes, such as classified KPI values (A2), KPI objectives (A3), amount of neighbors (A4), and action proposals (A5). The data describe the current states of cells and thus are based on the latest PM report from the simulator. In the graph visualization (Figure 6), positioning of cells corresponds to their Cartesian coordinates in the simulator. The arcs in the graph depict neighborship relations between cells and the size of the node depicts the classified value for some KPI in the evidence (CUE, CQI, or RLF). In the figure, RLF value has been selected as the node size from the settings (B2). The color of a node indicates the desired impact of a KPI objective or an action proposal (in the regarding grayscale figure, light gray indicates no action and dark gray an increase). In the figure, TXP action has been selected for the node color and it can be dynamically changed between objectives and action proposals from the settings (B3). Both in the tabular and graph visualization user can interactively browse cells with similar states by selecting facet values (A1 in Figure 5 and B1 in Figure 6), such as the amount of neighbors and classified KPI values.

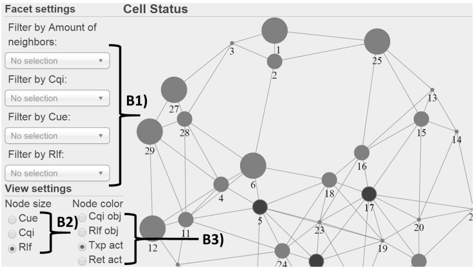


Filter by Amount of neighbors: No selection
Filter by Cqi: No selection
Filter by Cue: No selection
Filter by Rlf: No selection

ID	Cue	Cqi	Rlf	Neighbors	Cqi obj	Rlf obj	Txp act	Ret act
1	▁	▁	█	3	⇧	⇩		
2	▄	▄	▄	4				
3	▄	▄	▁	4				
4	▄	▄	▄	6				
5	▄	▄	▄	9			⇧ (0.70)	
6	█	▄	█	6		⇩		
7	☐	▄	☐	3				

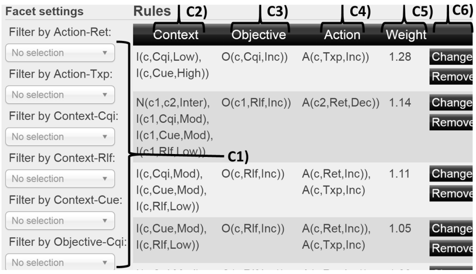
Fig. 5. A faceted view for a cell-specific tabular visualization

Figure 7 depicts the weighted formulae in a tabular view by dividing each formula into a formula weight (C5) and the formula classes defined earlier: context (C2), objective (C3), and action (C4). The user examines this view to learn the contents of the formulae and may modify or create formulae in order to change the behaviour of the MLN reasoner. For example, modification can be done by removing a formula or by changing its weight (C6). Facets (C1) in this view are generated as a combination of formula classes (contexts, objectives, and actions) and their objects (CQI, RLF, CUE, TXP, and RET).



View settings
Node size: ○ Cue ○ Cqi ● Rlf
Node color: ○ Cqi obj ○ Rlf obj ● Txp act ○ Ret act

Fig. 6. A faceted view for a cell-specific graph visualization.



Filter by Action-Ret: No selection
Filter by Action-Txp: No selection
Filter by Context-Cqi: No selection
Filter by Context-Rlf: No selection
Filter by Context-Cue: No selection
Filter by Objective-Cqi: No selection

Context	Objective	Action	Weight	
I(c,Cqi,Low), I(c,Cue,High))	O(c,Cqi,Inc))	A(c,Txp,Inc))	1.28	Change Remove
N(c1,c2,Inter), I(c1,Cqi,Mod), I(c1,Cue,Mod), I(c1,Rlf,Low))	O(c1,Rlf,Inc))	A(c2,Ret,Dec))	1.14	Change Remove
I(c,Cqi,Mod), I(c,Cue,Mod), I(c,Rlf,Low))	O(c,Rlf,Inc))	A(c,Ret,Inc)), A(c,Txp,Inc)	1.11	Change Remove
I(c,Cue,Mod), I(c,Rlf,Low))	O(c,Rlf,Inc))	A(c,Ret,Inc)), A(c,Txp,Inc)	1.05	Change Remove

Fig. 7. Faceted view for MLN formulae.

7. Experiments

7.1. Statistical evaluation of the MLN reasoner

The MLN model is used to optimize the performance of a simulated LTE network by adjusting the parameters of each cell simultaneously with respect to PM data retrieved from the simulator in 15 minute intervals (simulation time). Initially, all formula weights of the MLN model were set to zero. The weights were updated after every 48 measurement rounds using the measurements as training data. The Alchemy 2.0 software package [21] is used for the MLN inference and weight learning. The Alchemy implements the marginal inference and discriminative weight learning algorithms described in Sections 4.2 and 4.3.

Figure 8 shows the changes in the total number of RLFs in the LTE network as the model is updated. From the figure can be seen that in the beginning the number increases, as the model weights have not yet been learned. After the weights are first updated, the number begins to rapidly decline and stabilizes after a number of iterations.

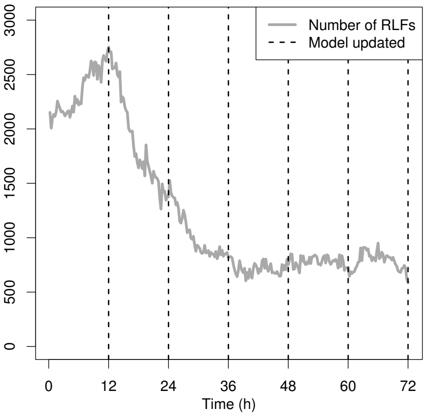


Fig. 8. Total number of radio link failures in a 15 minute interval. The vertical dotted lines indicate an update of the model weights.

7.2. Rule model statistics

Table 4 describes the results of some experiments on the impact of model sizes on MLN reasoning and SPARQL query times. In the table, *model 1* is constructed with a combinatorial generation of formulae from which unachievable formulae are removed, as defined in Algorithm 1. *Model 2* presents the model after formulae with zero weights are removed from the *model 1*. *Model 3* is created by removing formulae having weights less than |0.1| from *model 2*. Apparent from the results, the number of formulae decreases drastically when the model is refined, and finally the model size is only 32 % compared with the original *m*odel 1. Also, a decrease in the model size has an impact on the reasoning and SPARQL query times. The reasoning time has decreased from 12.1s to 8.9s (from *model 1* to *model 3*. This is a slight improvement to the system, although 12.1s is already feasible as the reasoning is run only once every 15 minutes in simulation time (and would also be run every 15 minutes in a real network management system). The SPARQL query time has dropped from 1.6s to 0.8s which enhances the usability of the GUI as the page contents need to be frequently updated in a faceted GUI.

	Number of formulae	Reasoning time	Query time
model 1	1920	12,1s	1,6s
model 2	1063	11,3s	1,0s
model 3	614	8,9s	0,8s

Table 4

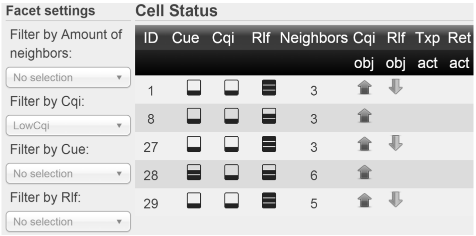
Statistics of model sizes and their impact on MLN reasoning and SPARQL query times.

7.3. Browsing activities in the GUI

To demonstrate the usage of the GUI, a random simulation round is taken from the system and the MLN reasoner and its model is explored via the GUI. First, two use cases are shown for 1) browsing cell states in a table and 2) graph visualization. Then, an example is shown on how modifying the model affects the reasoning outcome.

7.3.1. Observing cell states in the table view

The cell-specific MLN reasoning data of the given simulation round are presented in Figure 5. This view can be further examined interactively by selecting facet values. Figure 9 shows that cells having low CQI values (selected as a facet value) also have objectives to increase these values. In the same manner, from Figure 10 it can be concluded that high RLF values imply objectives to decrease the RLF values. These discoveries relate to the data preprocessing logic of the MLN model (operator goals that need to be achieved), as some classified KPI values (such as low CQI and high RLF in our scenario) trigger KPI objectives into the evidence.



Facet settings

Filter by Amount of neighbors: No selection

Filter by Cqi: LowCqi

Filter by Cue: No selection

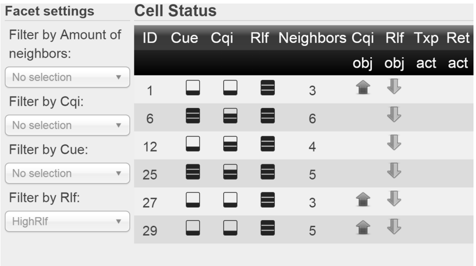
Filter by Rlf: No selection

Cell Status

ID	Cue	Cqi	Rlf	Neighbors	Cqi obj	Rlf obj	Txp act	Ret act
1	▂	▂	█	3	⬆	⬇		
8	▂	▂	▅	3	⬆			
27	▂	▂	█	3	⬆	⬇		
28	▅	▂	▅	6	⬆			
29	▂	▂	█	5	⬆	⬇		

Fig. 9. Cells having low CQI.

Another finding can be done from the table by selecting many neighbors (9 or more) as a facet value. Figure 11 shows the results for this selection and as it can be seen, all the three cells having many neighbors also have high probabilities (more than 0.7) for increasing their TXPs. As explained in Section 4, the



Facet settings

Filter by Amount of neighbors: No selection

Filter by Cqi: No selection

Filter by Cue: No selection

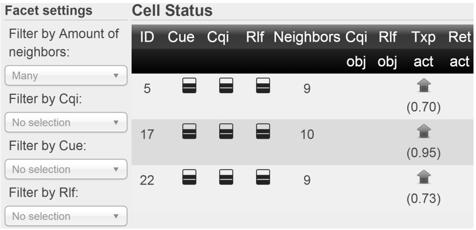
Filter by Rlf: HighRlf

Cell Status

ID	Cue	Cqi	Rlf	Neighbors	Cqi obj	Rlf obj	Txp act	Ret act
1	▁	▁	█	3	⬆	⬇		
6	█	▄	█	6		⬇		
12	▁	▄	█	4		⬇		
25	█	▄	█	5		⬇		
27	▁	▁	█	3	⬆	⬇		
29	▁	▁	█	5	⬆	⬇		

Fig. 10. Cells having high RLF.

MLN model and its formulae consider the states of neighboring cells as well, which might imply that the number of neighbors affects the probabilities of the action proposals in this scenario. Another aspect to observe from this figure is that these cells have neither an objective to increase CQI nor decrease RLF although actions are proposed to them. This also indicates how dependent cells are on their neighbors.



Facet settings

Filter by Amount of neighbors: Many

Filter by Cqi: No selection

Filter by Cue: No selection

Filter by Rlf: No selection

Cell Status

ID	Cue	Cqi	Rlf	Neighbors	Cqi obj	Rlf obj	Txp act	Ret act
5	▄	▄	▄	9			⬆ (0.70)	
17	▄	▄	▄	10			⬆ (0.95)	
22	▄	▄	▄	9			⬆ (0.73)	

Fig. 11. Cells having many neighbors.

7.3.2. *Observing cell states in the graph view*

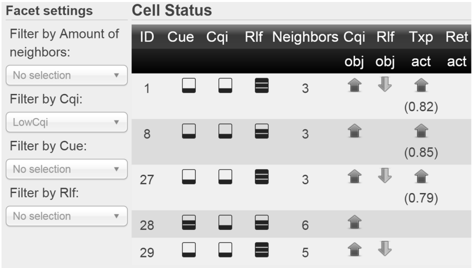
Figure 6 showed cell states in a graph visualization, in which node sizes indicate RLF values and node colors action proposals for TXP parameter. Observing this graph more closely, it can be seen that four of the six cells having high RLF values (cells 1, 25, 27, and 29) are located at the edge of the network and have only a few neighbors (five or less). High RLF values of these cells are most likely caused by terminals located at the edge of the network coverage area. Although the whole simulation area is populated by terminals, cells cannot cover the whole area with high signal strength from current cell locations. Thus, these cells most probably have high RLF values regardless of their configuration (even if coverage areas are expanded, new terminals will occur at the new edge of the coverage area). Moreover, the figure shows that MLN reasoner neither proposes configuration changes to these cells nor to their neighbors. This might indicate that the model has learned that the performance is not improved with any action for cells with given characteristics.

7.3.3. *Examining and modifying the MLN model*

For demonstration purposes, a weight of the following formula in the MLN model is modified:

$$I(c, Cqi, Low) \Rightarrow (O(c, Cqi, Inc) \iff A(c, Txp, Inc) \tag{4}$$

The weight of this formula is changed from 0.41 to 4.0. The Figure 12 shows cell states for cells having low CQI values after MLN reasoner has recalculated the action probabilities. Compared with Figure 9, new action proposals have been generated for cells 1, 8, and 27, which implies that the weight update had an effect on the functionality of the reasoner.



Facet settings

Filter by Amount of neighbors: No selection

Filter by Cqi: LowCqi

Filter by Cue: No selection

Filter by Rlf: No selection

Cell Status

ID	Cue	Cqi	Rlf	Neighbors	Cqi obj	Rlf obj	Txp act	Ret act
1	▁	▁	█	3	⬆	⬇	⬆ (0.82)	
8	▁	▁	▄	3	⬆		⬆ (0.85)	
27	▁	▁	█	3	⬆	⬇	⬆ (0.79)	
28	▄	▁	▄	6	⬆			
29	▁	▁	█	5	⬆	⬇		

Fig. 12. Updated states (actions re-inferred after model update) for cells having low CQI.

8. Related Work

The use of probabilistic reasoning in telecommunications has been studied previously. For example, Bayesian networks (BN) have been investigated in automatic network fault management [3,18] and in configuration evaluation [11]. MLNs have been used earlier to diagnose anomalous cells [8] in the network. Ontological modelling has been used together with BNs to evaluate network management activities in [10], which proposes using an ontology to describe domain-specific knowledge which is then utilized to dynamically generate a BN for a context-specific probabilistic evaluation task. To the best of our knowledge, statistical relational models (such as MLNs or a combination of an ontology and BNs) have not been applied earlier for cell configuration tasks specifically. Also, MLNs provide a template from which multiple Markov networks can be analysed. This approach provides more flexibility than a basic Bayesian or Markovian network model.

In the fields of pervasive computing and ambient intelligence, probabilistic reasoning and ontologies have been studied in several works. For example, MLNs are applied to context-aware decision processes in smart home environments [6]. The ontology is used to interpret and recognize situations from incoming data streams and an MLN model is dynamically constructed with respect to ontological knowledge. The MLN model is then used for decision making (smart home activities) from incomplete information [6]. Another project ([16,17]) examined human activity recognition from sensor data; first with MLN [16] and later with a combination of an ontology and a log-linear DL model (a model integrating description logics with probabilistic log-linear models [26]) [17]. Human activity recognition is also considered in another project [31] that uses an ontology to recognize potential human activities and statistical analysis to examine their confidence level [31].

In other problem domains, experiments in combining an ontological approach with probabilistic methods have been investigated. For example, BN-specific projects have use cases for medical decision support [34], financial fraud detection [5], and instance matching in a geological domain [27]. MLNs have been applied with semantic technologies in problem domains for ontology matching [25] and for natural language processing, where ontological concepts are extracted from text [12,28]. Aforementioned studies are not addressing cellular networks or network management tasks and thus they are not fully comparable. Yet, they provide similar technologies and address tasks, such as decision making support and graph analysis, that could be adapted and examined also in our work.

Another statistical relational model used for similar problems as MLN is the Probabilistic Soft Logic (PSL) [20]. It is based on First Order Logic, and provides weighted formulae and probabilistic inference. One application for a combination of ontologies and PSL is in analysing semantic similarities between natural language sentences. [2] Another project utilizes these techniques to extract a knowledge graph from text by using ontologies for representing domain-specific constraints and PSL to infer the most probable meaning for the text as a graph. [29] It has been argued that PSL is more efficient than MLN [29]. Our experiments indicate that MLN analysis can be applied to mobile networks by analysing a limited scope. The analysis of an entire network would in any case be futile for a system having the complexity and dynamicity of a mobile network domain. The comparison of PSL and MLN should be examined more thoroughly in order to conclude their difference in our case.

9. Conclusion and Future Work

This paper presented an experimental network management platform that uses MLN to analyse uncertain information from the LTE simulator and to infer suitable actions for the simulator. Along with the MLN model, OWL 2 ontology is used to semantically represent relevant network and MLN concepts. The ontology is utilised with SPARQL queries by the operator (via the faceted GUI) and by the MLN model.

Experiments of the platform in the Section 7 show that the MLN model works well in practice after it is trained to the current simulation context. The average number of RLFs reduced significantly during the weight learning phase. Also, experiments show the importance of monitoring the model; the combinatorial generation of formulae leads to a high amount of insignificant formulae. Removing these formulae with pruning makes both the MLN inference (the calculation of action probabilities) and the SPARQL query executions more robust. A potential future direction is investigating the scalability of the MLN model in larger networks and performing pruning of low-weight formulae in run time.

Some use cases of the GUI were shown to clarify how the MLN reasoning and network status can be monitored in view of this platform. The semantic representation of the underlying data and SPARQL queries provide a versatile access to the data and enable flexible information exploration with faceted browsing activities. Moreover, the SPARQL interface and the GUI provide an easy modification in order to investigate alternative reasoning results.

A further potential future research topic is the enhancement of the OWL-MLN interaction so that the MLN model settings can be dynamically modified by a human or by a description logic (DL) reasoner. Model settings include the selection of measurement variables, their threshold values, and formulae to be generated from the set of variables. Model settings could even include some initial formula weights with respect to prior knowledge. Moreover, the system will be enhanced by creating high-level goals which the user can use to modify the behaviour of the reasoning. For example, high-level goals could be mapped to corresponding MLN model settings.

The reason for combining semantic technologies with probabilistic reasoning was the decrease in the complexity in monitoring of the MLN model and to making the model dynamically modifiable. Our current implementation gives promising results to continue this work in order to enhance the system towards autonomic computing and towards adapting these technologies in more complex scenarios in the field of network management.

References

[1] Kasper Apajalahti, Eero Hyvönen, Juha Niiranen, and Vilho Räisänen. StaRe: Statistical reasoning tool for 5g network management. In Harald Sack, Giuseppe Rizzo, Nadine Steinmetz, Dunja Mladeniċ, Sören Auer, and Christoph Lange, editors, *The Semantic Web: ESWC 2016 Satellite Events*. Springer-Verlag, May 2016.

[2] Islam Beltagy, Katrin Erk, and Raymond Mooney. Probabilistic soft logic for semantic textual similarity. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1210–1219, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[3] Leila Bennacer, Laurent Ciavaglia, Abdelghani Chibani, Yacine Amirat, and Abdelhamid Mellouk. Optimization of fault diagnosis based on the combination of Bayesian Networks and Case-Based Reasoning. In *Network Operations and Management Symposium (NOMS), 2012 IEEE*, pages 619–622. IEEE, 2012.

[4] Julian Besag. Statistical analysis of non-lattice data. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 24(3):179–195, 1975.

[5] Rommel Carvalho, Shou Matsumoto, Kathryn Laskey, Paulo Costa, Marcelo Ladeira, and Laécio Santos. Probabilistic Ontology and Knowledge Fusion for Procurement Fraud Detection in Brazil. In Fernando Bobillo, PauloC.G. Costa, Claudia d'Amato, Nicola Fanizzi, KathrynB. Laskey, KennethJ. Laskey, Thomas Lukasiewicz, Matthias Nickles, and Michael Pool, editors, *Uncertainty Reasoning for the Semantic Web II*, volume 7123 of *Lecture Notes in Computer Science*, pages 19–40. Springer Berlin Heidelberg, 2013.

[6] Pedro Chahuara, François Portet, and Michel Vacher. *Making Context Aware Decision from Uncertain Information in a Smart Home: A Markov Logic Network Approach*, pages 78–93. Springer International Publishing, Cham, 2013.

[7] Fedor Chernogorov, Ilmari Repo, Vilho Räisänen, Timo Nihtilä, and Janne Kurjenniemi. Cognitive self-healing system for future mobile networks. In *IWCMC*, 2015.

[8] Gabriela Ciocarlie, Christopher Connolly, Chih-Chieh Cheng, Ulf Lindqvist, Szabolcs Nováczki, Henning Sanneck, and Muhammad Naseer-ul Islam. Anomaly Detection and Diagnosis for Automatic Radio Network Verification. In Ramón Agüero, Thomas Zinner, Rossitza Goleva, Andreas Timm-Giel, and Phuoc Tran-Gia, editors, *Mobile Networks and Management*, volume 141 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 163–176. Springer International Publishing, 2015.

[9] Cisco. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update 2014–2019 White Paper. Technical report, Cisco, 02 2015. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html.

[10] Ann Devitt, Boris Danev, and Katarina Matusikova. Ontology-driven automatic construction of bayesian networks for telecommunication network management. In *2nd Int. Workshop: Formal Ontologies Meet Industry (FOMI 2006)*, 2006.

[11] George Dimitrakopoulos, Kostas Tsagkaris, K Demestichas, E Adamopoulou, and Panagiotis Demestichas. A management scheme for distributed cross-layer reconfigurations in the context of cognitive {B3G} infrastructures. *Computer Communications*, 30(18):3807 – 3822, 2007. Optical Networking: Systems and Protocols.

[12] Lucas Drumond and Rosario Girardi. Extracting Ontology Concept Hierarchies from Text Using Markov Logic. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, SAC '10, pages 1354–1358, New York, NY, USA, 2010. ACM.

[13] European Telecommunications Standards Institute (ETSI). LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures, 8 2016. (3GPP TS 36.213 version 13.2.0 Release 13).

[14] Carolina Fortuna and Mihael Mohorcic. Trends in the development of communication networks: Cognitive networks. *Computer Networks*, 53(9):1354 – 1376, 2009.

[15] Liljana Gavrilovska, Vladimir Atanasovski, Irene Macaluso, and Luiz A DaSilva. Learning and reasoning in cognitive radio networks. *IEEE Communications Surveys Tutorials*, 15(4):1761–1777, Fourth 2013.

[16] Rim Helaoui, Mathias Niepert, and Heiner Stuckenschmidt. Recognizing interleaved and concurrent activities: A statistical-relational approach. In *Pervasive Computing and Communications (PerCom), 2011 IEEE International Conference on*, pages 1–9, March 2011.

[17] Rim Helaoui, Daniele Riboni, and Heiner Stuckenschmidt. A probabilistic ontological framework for the recognition of multilevel human activities. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 345–354. ACM, 2013.

[18] Carole Hounkonnou and Eric Fabre. Empowering Self-diagnosis with Self-modeling. In *Proceedings of the 8th International Conference on Network and Service Management*, pages 364–370. International Federation for Information Processing, 2012.

[19] Seppo Hämäläinen, Henning Sanneck, and Cinzia Sartori. *LTE Self-Organising Networks (SON): Network Management Automation for Operational Efficiency*. Wiley Online Library, 1st edition, 2012.

[20] Angelika Kimmig, Stephen Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. A short introduction to probabilistic soft logic. In *Proceedings of the NIPS Workshop on Probabilistic Programming: Foundations and Applications*, pages

1–4, 2012.

[21] Stanley Kok, Parag Singla, Matthew Richarson, and Pedro Domingos. The Alchemy System for Statistical Relational AI. Technical report, Department of Computer Science and Engineering, University of Washington, Seattle, WA, 2007.

[22] Sihyung Lee, Kyriaki Levanti, and Hyong S. Kim. Network monitoring: Present and future. *Computer Networks*, 65:84 – 98, 2014.

[23] Daniel Lowd and Pedro Domingos. Efficient Weight Learning for Markov Logic Networks. In *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases*, PKDD 2007, pages 200–211, Berlin, Heidelberg, 2007. Springer-Verlag.

[24] NGMN Alliance. 5G White Paper. Technical report, Feb 2015. https://www.ngmn.org/5g-white-paper.

[25] Mathias Niepert, Christian Meilicke, and Heiner Stuckenschmidt. A probabilistic-logical framework for ontology matching. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI'10, pages 1413–1418. AAAI Press, 2010.

[26] Mathias Niepert, Jan Noessner, and Heiner Stuckenschmidt. Log-linear description logics. In *IJCAI*, pages 2153–2158, 2011.

[27] David Poole, Clinton Smyth, and Rita Sharma. Semantic Science: Ontologies, Data and Probabilistic Theories. In PauloCesarG. da Costa, Claudia d'Amato, Nicola Fanizzi, KathrynB. Laskey, KennethJ. Laskey, Thomas Lukasiewicz, Matthias Nickles, and Michael Pool, editors, *Uncertainty Reasoning for the Semantic Web I*, volume 5327 of *Lecture Notes in Computer Science*, pages 26–40. Springer Berlin Heidelberg, 2008.

[28] Hoifung Poon and Pedro Domingos. Unsupervised Ontology Induction from Text. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 296–305, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[29] Jay Pujara, Hui Miao, Lise Getoor, and William W Cohen. Using semantics and statistics to turn data into knowledge. *AI Magazine*, 36(1):65–74, 2015.

[30] Vilho Räisänen. Information and knowledge in telecommunications: a service-dominant view. In *SRII Global Conference (SRII), 2012 Annual*, pages 211–218. IEEE, 2012.

[31] Daniele Riboni and Claudio Bettini. COSAR: hybrid reasoning for context-aware activity recognition. *Personal and Ubiquitous Computing*, 15(3):271–289, 2011.

[32] Matthew Richardson and Pedro Domingos. Markov Logic Networks. *Machine Learning*, 62(1-2):107–136, February 2006.

[33] Péter Szilágyi and Szabolcs Nováczki. An automatic detection and diagnosis framework for mobile communication systems. *An automatic detection and diagnosis framework for mobile communication systems*, 9:184 ff., 2012.

[34] Hai-Tao Zheng, Bo-Yeong Kang, and Hong-Gee Kim. An Ontology-Based Bayesian Network Approach for Representing Uncertainty in Clinical Practice Guidelines. In PauloCesarG. da Costa, Claudia d'Amato, Nicola Fanizzi, KathrynB. Laskey, KennethJ. Laskey, Thomas Lukasiewicz, Matthias Nickles, and Michael Pool, editors, *Uncertainty Reasoning for the Semantic Web I*, volume 5327 of *Lecture Notes in Computer Science*, pages 161–173. Springer Berlin Heidelberg, 2008.

Publication V

Kasper Apajalahti. Creating Time Series-Based Metadata for Semantic IoT Web Services. In *Database and Expert Systems Applications. DEXA 2018.*, Springer, Cham, Regensburg, Germany, pages 417–427
DOI https://doi.org/10.1007/978-3-319-98812-2_38, September 2018.

© 2018 Springer, Cham

Reprinted with permission.

Creating Time Series-Based Metadata for Semantic IoT Web Services

Kasper Apajalahti

Aalto University School of Science, Finland,
`kasper.apajalaht@aalto.fi`
https://seco.cs.aalto.fi/projects/mobile5g/

Abstract. In the near future, the Internet of things (IoT) will rapidly change and automate tasks in our everyday life. IoT networks have sensors measuring the environment and automated agents changing it with respect to predefined objectives. Modeling agents as web services requires lots of metadata from the environment in order to define the desired performance in a specific context. For this purpose, we propose an automatic measurement-based metadata creation method that analyses multivariate time series gathered from the sensors during agents change the environment. The time series analysis uses a cumulative sum algorithm (CuSum) to detect events and association rule learning to find temporal patterns. We evaluate our system with a Long-Term Evolution (LTE) simulator having mobile phones corresponding to IoT devices, LTE macro cells as the data source, and the Self-Organised Network (SON) functions as the automated agents in the network. Our experiments give promising results and show that the metadata creation process can be utilised to characterise IoT agents.

1 Introduction

Internet of Things (IoT) services combine sensors and IoT devices into applications that solve predefined use cases. Some services provide access to data gathered from sensors and others analyse the data and perform actions on the environment with respect to some objectives [8], for example, to increase the temperature of a room or signal quality of a mobile device. A key challenge in IoT is the interoperability between services and applications [5]. The services may be deployed to a variety of environments. Yet, the semantics of services should be defined with relevant metadata in order to find similarities in services across network domains. Analogously to a cellular network, one domain might want to prevent an anomalous behaviour by understanding how other domains have experienced and solved an issue in a similar context. For example, a sudden bad weather may be the root cause for terminals (IoT devices) to increase the load of nearby indoor cells (data sources) as people escape rain indoors. In this case, web services accessing local weather data combined with network-specific data and agent services would be necessary to act proactively to handle the upcoming load peak indoors.

A general challenge with web services is the need to involve domain experts in developing the services [14]. Particularly, semantic IoT services have development costs in modeling the services with relevant metadata [12] and linking it to other ontologies in order to make services discovered and invoked [9]. The potential development costs also lead to a cold-start problem among the web service modelers: the SWS system cannot recommend suitable and interoperable services before developers in several domains have used their resources to manually model them. Hence, there is a need to automate the service development process.

In this paper, we propose a time series-based metadata creation process for semantic IoT services. The process is evaluated in a Long Term Evolution (LTE) network simulator environment having mobile devices that measure the quality of service and LTE macro cells that periodically report aggregated measurements. With respect to the LTE networks, the Self-Organised Networks (SON) has brought automation to the management tasks [11]. The SON functions can be viewed as specialised agents controlling the LTE cell configurations with respect to predefined objectives. The simulator contains simple SON functions which optimise the network performance and our goal is to characterise their behaviour with the metadata creation process.

The paper consists of the following parts. Related work is discussed in Section 2. After that, Section 3 gives an overview of the SWS methodology in the scope of IoT and cellular networks. Section 4 explains the theory and statistical methods used to create time series-based metadata for the services. Section 5 presents the experiments for the metadata creation process in an LTE simulator. Finally, Section 6 concludes the paper.

2 Related Work

Bytyçi et. al [4] propose a method that combines association rule learning and ontologies to mine patterns from water quality measurement data. They managed to enrich the mining results by first populating the context ontology with sensor data and then using the ontology as an input to the association rule learning.

Fan et. al [6] use association rule learning for sensor-based constructions to find contextual patterns from sensor measurements [6]. The experiments show that temporal patterns can be identified with respect to time metadata, such as a public holiday, weekday, or weekend.

Galushka et. al [7] examine data mining techniques for smart home environment. The authors present techniques both to transform time series into labeled segments and to use association rule learning to find temporal patterns from them.

Labbaci et. al [13] analyse web service logs and interactions between web services to learn frequent compositions and substitutions of services in a web service system. Their method has a similar focus on analysing the past data related to the web service invocations in order to learn characteristics of their behaviour. Such as in our work, they use association rule learning to find the most frequent item sets from the web service-related data.

The related work focuses on association rule learning of measurement-based sensor data and web service logs. However, none of the earlier works learn association rules from sensor measurements in order to characterise IoT services, which is the focus of this paper.

3 Semantic web service model

3.1 Methodology

The core idea in the Semantic Web Service (SWS) is to discover, compose, and invoke services with respect to user requests [2]. This idea applies both to IoT and cellular networks: requests based on the multivariate measurements of IoT devices or mobile phones need to be handled with an action that causes a desired impact to the measured values. Figure 1 describes a simplified web service ontology, where arrows depict *"hasElement"* relations. It also explains the analogy to IoT and cellular networks, such as the service corresponds to an agent, operation to an action, and effects to changes in the metric values. The architecture of the model is adapted from a simple SWS model, WSMO-lite [15]. A service has operations that aim to change the status of its environment.

Effects play a central role in using the ontology; the service model is used by linking the effects of requests and operations. For example, the goal of a network operation could be to enhance customer satisfaction for mobile users during a rush hour. This context-specific intention is mapped to some metric effects, such as an increase in the throughput and balancing the usage of physical resource blocks (PRBs). Based on the network measurements, some service operations are known to produce the requested context-specific effects and thus, they are mapped as responses to the request.

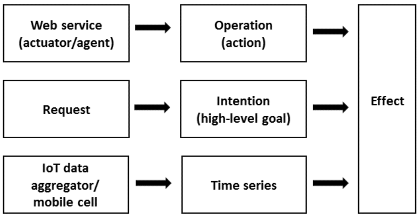


Fig. 1. Service ontology constructs for a service (top), request (middle), and environment/object status (bottom).

As domains may have dedicated data and means of management, there is a need to also understand the semantics of cross-domain effects. For example, domains might have services that monitor threshold values for different key performance indicators (KPIs). Yet, some KPIs are associated (correlate) in the given context and are part of the same intention. Thus, both of the services would be valid solutions to a request having similar effect as an objective. More details about the dependency modeling of effects are defined in earlier research [3].

3.2 Cross-domain request

An example of a cross-domain user request from mobile network management is illustrated in Figure 2. An operator from the network X wants to request better Quality of Service (QoS) in some context, such as during a public event. The requested intention can be achieved by increasing the throughput. Another network domain Y has deployed a SON function as a web service with an operation that is known to increase Reference Signal Received Power (RSRP) during a public event. Knowing the semantic similarity between the same cross-domain KPIs and that the throughput and RSRP in the domain Y are statistically associated, the given service operation can be discovered and mapped as a response to the user request. Altogether, even though a problem and solution would be in different networks and might address different parameters, the SWS system finds a request-operation mapping relation with respect to semantic modeling and statistical dependencies.

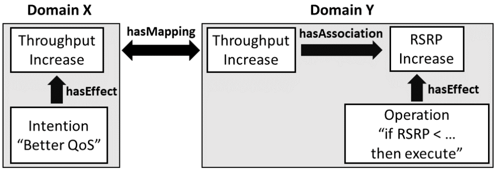


Fig. 2. An example of a cross-domain request-operation mapping.

In order to have a functional SWS system in a distributed and multi-domain sensor network, contextual metadata is needed, for example, the relevant operation-specific effects that enable the discovery of the web services. Creating and maintaining these manually is resource-intensive. Moreover, metadata mappings (such as corresponding KPIs) between different environments need to be resolved before the full benefit and utilization of the services can be realised. Automation for the service modeling and metadata addition can reduce the cost of deployment significantly by utilising SWS systems. In the next subsection, we introduce a process to address this problem with automatic metadata creation methods.

4 Methods to create service metadata from sensor measurements

4.1 The data sequence of the process

The measurement-based metadata is added as effects to the service operations and it is used to bind requests to operations. Our process analyses statistically the behaviour of a service operation in a given context while it is executed. Figure 3 illustrates the data sequence of processing metadata for a service operation based on realised actions and measured metric values both before and after the actions have been taken. In the beginning, the user decides how an operation is defined. For example, a service could be an algorithm and an operation a set of parameter values. The actions fulfilling this criteria are retrieved from the database (step 1) and based on their timestamps, a time series for every available metric of the operated object (such as the mobile cell) is retrieved from the measurement database (step 2). Time series are analysed with an event detection method (step 3) and the detected events are sent to the association learning component (step 4). This component detects whether one or more metrics have temporal correlation. Finally, the associated events are sent to the ontology and the service operation is populated with these events (step 5). The events are later used as service operation effects, as described in Figure 1.

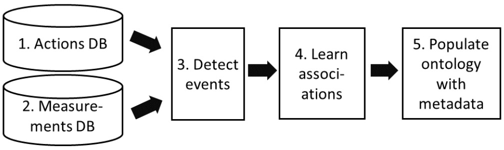


Fig. 3. Process flow for identifying associated events and adding them as operation metadata.

4.2 Event detection with CuSum

The CuSum algorithm is a statistical quality control method that can be used to detect value changes in a time series. The basic concept is to cumulatively sum up changes between data points and a comparison value and flag a change if the sum exceeds a predefined threshold value. The Equation 1 describes how to detect increasing event in our system. The equation contains a max of zero and the cumulative sum of value s_h, the data point x_t, and the combined comparison value of mean and standard deviation, μ and σ, calculated from the time series. σ is used as a threshold sum value for increasing trends.

$$s_h = max(0, s_h + x_t - \mu - \sigma) \tag{1}$$

For analysing decreasing trends in a time series, the Equation 2 is used instead. Compared with the earlier equation, now a min operator is used and CuSum contains a positive sign for the σ. The threshold sum for detecting a decreasing trend is $-\sigma$.

$$s_l = min(0, s_l + x_t - \mu + \sigma) \tag{2}$$

When CuSum is executed for all operation-specific actions, the outcome is a dataset where each row depicts a single action having a list of measurement events it produced. From this dataset, we may further learn operation-specific patterns between measurement events.

4.3 Temporal pattern mining with Apriori

Association rule learning is a data mining method that learns rules between the sets of items in a database. The idea is to analyse the co-occurrence of items in a database row and to use some measure and threshold to find out relevant rules. The simplest measure is the support, which is calculated as a proportion of the database rows containing the given set of items. [10] In our use case, the support is the proportion of detection timestamps containing a set of metric events. Thus, it indicates the frequency of the events occurring simultaneously in the given context.

In addition to support, confidence is another measure to determine associations between items. The Equation 3 shows the definition of the confidence. It can be interpreted as an if/then pattern: if set of events X occurs, then set of events Y also occurs. As it can be seen, the measure indicates the proportion of X (the support of X) that also contains Y (the support of X and Y). [10]

$$conf(X \rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)} \tag{3}$$

In this system, the objective is to learn support and confidence values for measurement events occurred during a set of actions made by an agent. For this purpose, we use an open-source implementation[1] that of a well-known Apriori algorithm (see [1] for further details).

5 Evaluation

5.1 Case study: LTE network simulation with SON functions

The applicability of our metadata creation process is evaluated with an LTE network simulator. The simulator environment comprises 20 LTE base stations with 32 LTE cells covering an area with a radius of 5 km. The simulator creates Performance Management (PM) data reports that contain cell level KPIs such as the average cell throughput, radio link failures (RLFs), average Reference Signal Received Power (RSRP), the overall usage of the Per Resource Blocks (PRB), and average channel quality indicator (CQI) level. The cell level KPIs are aggregations of the measurements made by the user equipments (UEs) that constantly report the experienced signal status to a cell they are attached to. The PM data of the cells are reported periodically in 15 minute intervals in simulation time. The time series gathered from the PM data contains 10 time steps before and after the actions have been executed or activated.

Table 1 describes the scenarios created for our experiments. The idea was to create network issues with similar objectives but in different contexts; in all scenarios, users have issues getting the required throughput level, but the required actions differ significantly from each other.

In the coverage problem, the UEs are located uniformly in an area where the coverage is insufficient and the solution is to increase TXP to enhance the coverage and therefore the overall throughput level. The second scenario, local overload, has a few hundred UEs located in a small area near one base station hosting three cells. Now the throughput should be increased by adjusting the antenna tilt angles (remote electrical tilt, RET) towards the group of UEs. The third scenario, mobile overload, has 500 uniformly located background UEs and a group of 500 UEs constantly moving in the

---

[1] https://github.com/asaini/Apriori

Table 1. Simulation scenarios

Scenario	UEs	Objective	Solution
Coverage problem	1000	Inc. Thr.	Increase power
Local overload	350	Inc. Thr.	Downtilt
Mobile overload	500	Inc. Thr.	Balance load

simulated area causing abrupt load peaks in the cells. An increase in the throughput in this case should be achieved by balancing the load between the nearby cells.

5.2 Context-specific support values

Figure 4 presents the action-specific support values in different scenarios and KPIs. Figure presents one subplot for each scenario and each subplot presents KPI-specific support values for each action. Positive support value indicates a support measurement for an increase and negative a decrease. For example, the first five bars show support values for the increasing and decreasing events for the KPIs when no action has been taken in the coverage problem scenario. The first bar shows that increasing events for CQI has been measured with a support value of 0.12 and decreasing events with a value of 0.09. With respect to these experiments, a threshold level of ± 0.15 (marked with two dashed lines) is suitable for labelling action-specific KPI effects.

In general, we may conclude that the distribution of the scenario- and action-specific support values show that the detection of single KPI events works well as the values are plausible with respect to the actions. Especially, the throughput values show that the best agents in every scenario also enhance the throughput in the network, which is the desired outcome. Also, the fact that the number of false positive support values (values when no action is taken) is low, indicates an adequate performance of the CuSum method.

5.3 Context-specific associations and their applicability as metadata

The association rules for every scenario-specific action were generated with a minimum support level of 0.15 and confidence level of 0.70. Figure 5 shows the quantities of associations learned among the recorded events. With the given parameters, the best agents also generate the most associations between the KPI effects, whereas few associations are produced from other agents in the coverage problem. This is a desired outcome as our goal is to highlight the best matching agents in different contexts.

The final step in the metadata creation process is to populate the ontology instances with relevant events and associations. As defined in the Section 4.1, the populated ontology instances are web service operations: three scenario-specific operations for each web service (network agent). Finally, we may examine the request (ontology queries) that naps the queried effects with the relevant service operation metadata. Table 2 illustrates the examples of combining association rules that we tested and verified to retrieve correct mappings to the operations. For example, if a request contains associations from $IncTHR$ (throughput) to $DecRLF$ and from $DecRLF$ to $DecPRB$, it gives a unique

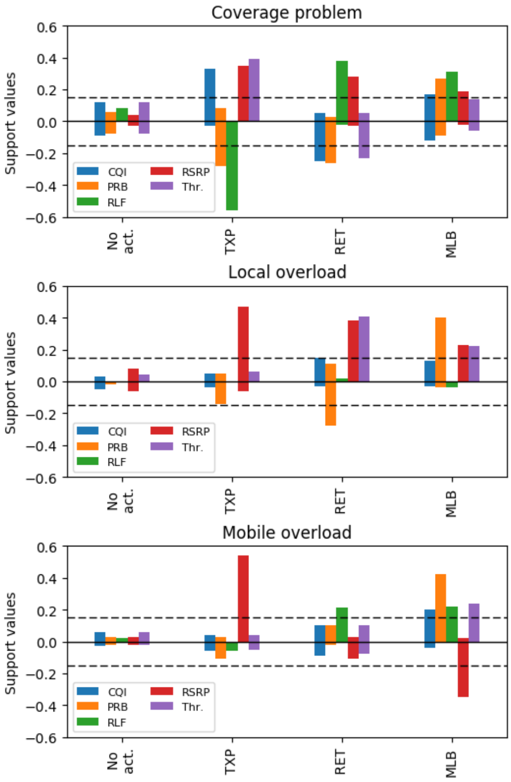


Fig. 4. Support values for action- and KPI-specific events in three scenarios.

mapping to the TXP agent operation on a coverage problem scenario. Similarly, the two other rows show rules that are also unique among all rule sets and that corresponds to the best solution in the scenario. In addition to the association rules shown in the table, the request query may include effects that pass the minimum support level (e.g. ”increase the throughput”) or negations of undesired effects (e.g. ”do not decrease the throughput”).

Altogether, the demonstrated associations indicate that we managed to distinguish the important agent operations and scenarios from each other with our metadata creation process.

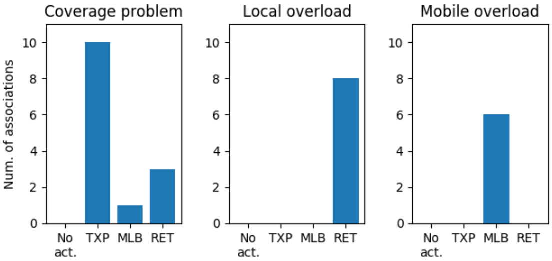


Fig. 5. Scenario-specific quantities of associations for service operations. Threshold for support is 0.15 and for confidence 0.70.

Table 2. Unique set of rules that characterise suitable agents in every scenario.

Scenario	Action	Matching rules
Coverage problem	TXP	$IncTHR \rightarrow DecRLF$, $DecRLF \rightarrow DecPRB$
Local overload	RET	$IncTHR \rightarrow DecPRB$, $IncRSRP \rightarrow DecPRB$
Mobile overload	MLB	$IncTHR \rightarrow IncPRB$, $DecRSRP \rightarrow IncPRB$

6 Conclusions and future work

We proposed a method of creating time series-based metadata for services that operate in IoT networks. The metadata creation process is a combination of statistical methods, event detection and association rule learning, and it is based on analysing multivariate time series gathered from the network elements while some actions (service operations) are executed. The process was evaluated with a Long Term Evolution (LTE) simulator where automated agents (web services) configure the antenna parameters of LTE macro cells in order to enhance the network quality. We created three simulation scenarios and evaluated the results of three agents in those. Our experiments show that the presented metadata creation process works on these scenarios; all suitable service operations can be characterised with the generated metadata. For future work, we examine and compare different event detection and pattern mining methods and evaluate them in more complex IoT environments.

Acknowledgment

We thank Vilho Räisänen and Eero Hyvönen for comments and support on early versions of this paper. This work is partially funded by Nokia.

References

1. Agrawal, R., Srikant, R., et al.: Fast algorithms for mining association rules. In: Proceedings of the 20th VLDB Conference. pp. 487–499 (1994)

2. Ankolekar, A., Burstein, M., Hobbs, J.R., Lassila, O., Martin, D., McDermott, D., McIlraith, S.A., Narayanan, S., Paolucci, M., Payne, T., Sycara, K.: DAML-S: Web Service Description for the Semantic Web, pp. 348–363. Springer Berlin Heidelberg, Berlin, Heidelberg (2002). https://doi.org/10.1007/3-540-48005-6_27
3. Apajalahti, K., Niiranen, J., Kapoor, S., Räisänen, V.: Sharing performance measurement events across domains. In: 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM). pp. 463–469 (May 2017). https://doi.org/10.23919/INM.2017.7987313
4. Bytyçi, E., Ahmedi, L., Kurti, A.: Association Rule Mining with Context Ontologies: An Application to Mobile Sensing of Water Quality, pp. 67–78. Springer International Publishing, Cham (2016). https://doi.org/10.1007/978-3-319-49157-8_6
5. Da Xu, L., He, W., Li, S.: Internet of things in industries: A survey. IEEE Transactions on Industrial Informatics **10**(4), 2233–2243 (Nov 2014). https://doi.org/10.1109/TII.2014.2300753
6. Fan, C., Xiao, F., Yan, C.: A framework for knowledge discovery in massive building automation data and its application in building diagnostics. Automation in Construction **50**, 81 – 90 (2015). https://doi.org/http://dx.doi.org/10.1016/j.autcon.2014.12.006
7. Galushka, M., Patterson, D., Rooney, N.: Temporal Data Mining for Smart Homes, pp. 85–108. Springer Berlin Heidelberg, Berlin, Heidelberg (2006). https://doi.org/10.1007/11788485_6
8. Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M.: Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems **29**(7), 1645 – 1660 (2013). https://doi.org/http://dx.doi.org/10.1016/j.future.2013.01.010
9. Gyrard, A., Serrano, M., Atemezing, G.A.: Semantic web methodologies, best practices and ontology engineering applied to internet of things. In: 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT). pp. 412–417 (Dec 2015). https://doi.org/10.1109/WF-IoT.2015.7389090
10. Hahsler, M., Grün, B., Hornik, K.: arules - a computational environment for mining association rules and frequent item sets. Journal of Statistical Software, Articles **14**(15), 1–25 (2005). https://doi.org/10.18637/jss.v014.i15, https://www.jstatsoft.org/v014/i15
11. Hämäläinen, S., *et al.*: LTE Self-Organising Networks (SON): Network Management Automation for Operational Efficiency. John Wiley & Sons (2011)
12. Heß, A., Kushmerick, N.: Learning to Attach Semantic Metadata to Web Services, pp. 258–273. Springer Berlin Heidelberg, Berlin, Heidelberg (2003). https://doi.org/10.1007/978-3-540-39718-2_17
13. Labbaci, H., Medjahed, B., Aklouf, Y.: Learning interactions from web service logs. In: Benslimane, D., Damiani, E., Grosky, W.I., Hameurlain, A., Sheth, A., Wagner, R.R. (eds.) Database and Expert Systems Applications. pp. 275–289. Springer International Publishing, Cham (2017)
14. Pautasso, C., Zimmermann, O., Leymann, F.: Restful web services vs. "big"' web services: Making the right architectural decision. In: Proceedings of the 17th International Conference on World Wide Web. pp. 805–814. WWW '08, ACM, New York, NY, USA (2008). https://doi.org/10.1145/1367497.1367606
15. W3C: WSMO-Lite: Lightweight Semantic Descriptions for Services on the Web, W3C Recommendation, World Wide Web Consortium (2010). Available: https://www.w3.org/Submission/WSMO-Lite/ (accessed May 2018)

Publication VI

Kasper Apajalahti, Ermias Walelgne, Jukka Manner, and Eero Hyvönen. Correlation-Based Feature Mapping of Crowdsourced LTE Data. In *2018 IEEE 29th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, IEEE, Bologna, Italy, September 2018.

© 2018 IEEE

Reprinted with permission.

Correlation-Based Feature Mapping of Crowdsourced LTE Data

Kasper Apajalahti
Aalto University
http://seco.cs.aalto.fi
Email: first.last@aalto.fi

Ermias Andargie Walelgne
Aalto University
http://comnet.aalto.fi
Email: first.last@aalto.fi

Jukka Manner
Aalto University
http://comnet.aalto.fi
Email: first.last@aalto.fi

Eero Hyvönen
Aalto University
http://seco.cs.aalto.fi
Email: first.last@aalto.fi

***Abstract*—There have been efforts taken by different research projects to understand the complexity and the performance of a mobile broadband network. Various mobile network measurement platforms are proposed to collect performance metrics for analysis. Data integration would provide more thorough data analyses as well as prediction and decision models from one dataset to another. The crucial part of the data integration is to find out, whether two datasets have corresponding features (performance metrics). However, finding common features across datasets is a challenging task. For example, features might: 1) have similar names but be different metrics, 2) have different names but be similar metrics, or 3) be same metrics but have differences in the underlying methodology.**

We designed a feature mapping methodology between two crowdsourced LTE measurement-based datasets. Our method is based on correlations between the features and the mapping algorithm is solving a maximum constraint satisfaction problem (CSP). We define our constraints as inequality patterns between the correlation coefficients of the measured features. Our results show that the method maps measurement features based on their correlation coefficients with high confidence scores (between 0.78 to 1.0 depending on the amount of features). We observe that mapping score increases as a function of the amount of features. Altogether, our results show that this methodology can be used as an automated tool in the measurement data integration.

I. INTRODUCTION

There have been efforts taken by different research projects to understand the complexity and the performance of a mobile broadband network. Various crowsourced-based platforms such as Netradar [1], RTR Nettest [2], Mobiperf [3], OpenSignal [4], and Speed Test [5] have been developed to collect network related metrics from different vantage points. Also, controlled cross-operator measurement test platforms, such as the MONROE [6] has been built for the same purpose. These platforms are collecting measurement metrics independently. It is possible to use each of this measurement dataset separately for analyzing the behaviors of mobile broadband networks, as it is recently done by several research groups [7], [6], [8].

However, to enable the richer use of the collected data, the data sources should be integrated. The data integration would provide more thorough data analyses, for example with a wider range of Mobile Network Operators (MNO)s included in the data. Moreover, the data integration would provide the dynamic adaptation of prediction and decision models from one dataset to another. The integration of the dataset needs to find the features (performance metrics) that are common to these separate datasets, such as throughput, latency, and network-level metrics. For instance, there is an EU project about mapping of broadband services in Europe [9] and the main challenge in the project is to present the variety of data in one mapping application. Currently, the application shows separate datasets by country level. One reason of displaying such different datasets separately is that the data differs in terms of methodology approaches and that there is no easy solution to find similar features and to merge them. Therefore, such projects would also benefit by applying a feature mapping method that enables integrating datasets into a single country-level view.

The crucial part of the data integration is to find out, whether two datasets have corresponding features (performance metrics). The challenge in finding similar features rises in comparing whether they have: 1) similar names but different metric (such as "download speed" depicting either the throughput or average bit rate), 2) different names but similar metric (such as "latency" and "ping duration"), names, and 3) same metrics in general but measurements have differences in the underlying methodology (such as latencies measured with different protocols). Our objective is to automatically analyse and map similar features across platforms, without a need for manually analyse their similarities and solve the abovementioned vagueness.

Our approach addresses these issue by mapping features between crowdsourced datasets. We use platform-specific correlation coefficients between features and try to find the same correlation patterns from another dataset. In the mapping, we assume that ranking between coefficients is domain invariant. In other words, the ranking order of the coefficients is more or less the same in both datasets. We present a methodology for measurement-based feature mapping of different data sources only using the correlations between the features. Thus, our method is independent of the actual feature values which might be biased between the datasets. We find mappings between performance metrics computed under different conditions, such as different protocol in latencies (TCP and UDP) and between biased metrics that at first seems different, such as Reference Signal Received Power (RSRP) and Arbitrary Strength Unit (ASU). The results show that the method maps measurement features based on their correlation coefficients with high confidence scores (between 0.78 to 1.0 depending on the amount of features). The applicability of our methodology is that it can be used as an automated tool in the measurement data integration.

This paper is structured as follows: Section II presents the related work in feature mapping, Section III describes two datasets, Section IV explains our methodology, Section V presents the results of our approach, and Section VI concludes the paper.

II. RELATED WORKS

Our work is mainly related to studies which have the research objective of integrating measurement data from different sources and need to solve the problem of mapping the features across datasets.

Mapping of performance metrics and QoS features in the LTE networks is addressed in earlier mobile network research. Malandrino et. al [10] have the similar objective of merging two crowdsourced LTE measurement datasets. Their focus is however on using human expertise in order to map the metrics from the datasets, whereas we present a method that analyses the data and proposes the mappable features without manual analysis.

Lipenbergs et. al [11] address the European-wide broadband mapping task [9] and analyse the data representation of broadband mapping. Apajalahti et. al combine statistical correlations and human-defined semantic dependencies to enable cross-domain mappings between LTE performance metrics of different network providers. Li et. al [12] map QoS parameters across LTE network components, such as the Evolved Universal Terrestrial Radio Access Network (E-UTRAN), Backhaul transport network, and Evolved Packet Core (EPC) network. All of these works propose models where the actual cross-domain mapping is defined by human, whereas our approach aims to find the mapping automatically.

More generally in the field of wireless networks research, there have been approaches to map features across data sources with statistical and machine learning methods. For example, Manco-Vásquez et. al [13] uses a Kernel Canonical Correlation Analysis (KCCA) method for spectrum sensing in the cognitive radio environment. Although the method is also correlation-based, it requires the actual data sources to be in the same environment (the same time periods and/or location), whereas our method is developed to handle heterogeneous data where time periods and locations of the measurements might be unknown or scattered.

Pan et. al [14] present a transfer component analysis method that learns a cross-domain feature space for indoor WiFi localization. Their method differs from ours as it addresses a supervised learning task where the feature mapping is trained with respect to labels (locations) in the training set, which we do not consider.

The concept of feature mapping has also been addressed in sensor networks research, for example, in the human activity recognition task. Van Kasteren et. al [15] map features with manually define mapping functions by first classifying the features by their type. Chiang et. al [16] also define manually the sensor metadata which is then used to calculate the feature similarities across domains. Wen-Hui et. al [17] propose an algorithm based on Kullback–Leibler divergence to map cross-domain features with respect to the probability distributions of the classification labels. In this case, one needs to map the classification labels in order to learn mappings between the features.

Altogether, the related work shows that cross-dataset feature analysis has gained interest in the related research fields, but most of the work relies either on manually defined mappings or on classified data where labels describe the measurements and feature values. The need for automatic mapping of QoS and other LTE-related measurement parameters between data sources has been recognized, but to our knowledge no earlier work for this exists yet.

III. DATASETS

For this work, we have used two measurement datasets collected from the first of June to the end of Nov. 2017. The first dataset is `Netradar` [18]. It is a crowdsourced mobile measurement platform that measures and collects metrics related to cellular network performance collected from mobile user devices. It has been running actively worldwide since March 2013. The measurement mainly focuses on the data services and analysis of bit-rates (over TCP), UDP based latencies and the context information related to each measurement including, device model, battery level, location, radio signal strength, date and time, the mobile operator.

We processed the dataset by radio technology type and location. For this, we select measurements under LTE network, which has been collected from Helsinki area, Finland. `Netradar` has a number of measurement metrics related to cellular network performance. In this paper, we use the following metrics: TCP-based downlink and uplink throughput, UDP-based latency, signal strength, LTE ASU, RSRP, Reference Signal Received Quality (RSRQ), Reference Signal to Noise Ratio (RSSNR), battery level, and movement speed.

The second dataset we have used in this paper is `RTR Nettest` [2]. It is a mobile application that collects information from the end user with an open dataset access. It records features including the downlink and uplink throughput, signal strength, network metrics such as RSRP and RSRQ for LTE, connection error and IP packet loss, ping based latency, testing time, IP address and host name of the computer. It also collects other quality parameters such as Domain Name System (DNS), ports, transparent connection, downloading speed test website and traceroute. `RTR Nettest` provides more than 60 network-related features. For this work, we only focus on metrics collected under LTE network. These are the TCP-based downlink and uplink throughputs, TCP-based ping latency test, LTE RSRQ, and LTE RSRP.

IV. METHODOLOGY

This section presents the method that maps the measurement-based features between two crowdsourced LTE data platforms, RTR Nettest and Netradar. Thus, the features are performance metrics and QoS parameters collected via end-user measurements. The main problem is to find and map corresponding features between two platforms by only analysing their correlation coefficients with other features. The hypothesis is that we can find feature pair-specific patterns from the correlation coefficients which occur in both platforms. The objective of the method is to rank the coefficient values and represent every coefficient pair with inequalities, such as $r(f_x, f_y) < r(f_y, f_z)$ stating that the correlation between f_x and f_y is lower than f_y and f_z. For example, our analysis shows that we can make a general rule $r(latency, downlink) < r(downlink, uplink)$ (see Section V for more information).

A. Preparing measurements into correlation coefficient rankings

In order to find regular patterns regarding the coefficient rankings of a large dataset (data available from RTR Nettest or Netradar), we need to preprocess the data. Figure 1 shows the preprocessing phase. First, the dataset is split into smaller monthly subsets (step 1). Next, a correlation matrix R_i is calculated for each subdataset (step 2). For evaluation purposes we use both Pearson's linear and Spearman's non-linear correlations. Finally, for each subdataset we calculate coefficient rankings as a set of inequality clauses (step 3).

B. Correlation-based feature mapping

The output of the preprocessing task (Figure 1) is used for the actual mapping. Figure 2 shows an overall picture how the features are mapped across the platforms. The earlier described data preparation are made separately for both source and target platforms. From the source platform, we also need to learn which of the coefficient inequalities are more regular than others regarding the N subdatasets (step 1). We add an inequality $r(f_x, f_y) < r(f_y, f_z)$ between correlation coefficients of features f_x, f_y, and f_y,f_z into the constraint base, if the inequality occurs in majority (more than 0.5 times) of the subdatasets. After learning the constraint base we use it to find similar patterns from the target platform (step 2).

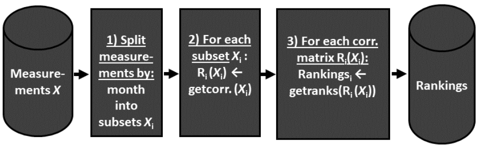


Fig. 1: Preprocessing of the data: 1) splitting into monthly subsets 2) calculating correlation matrices 3) listing the rankings between the correlation coefficients.

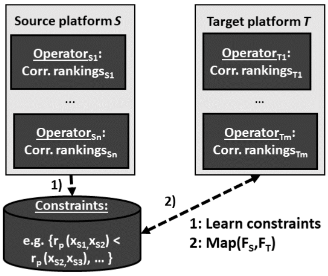


Fig. 2: Steps in the mapping procedure: 1) learning the constraints from the source platform and 2) mapping those to correlation coefficients from the target platform.

Technically, our feature mapping method is solving an approximation of a well-known constraint satisfaction problem (CSP). A CSP is a problem in which values need to be assigned to variables so that given constraints are satisfied [19]. In our case, the constraints are the inequalities that we learn between the correlation coefficients of the source platform and variables the feature pairs of the source features F_S. The problem is then to assign feature pairs from the target platform (features F_T) as variable values to the constraint base (replacing F_S with F_T) so that the assigned feature pairs maximize the number of truth statements when comparing the constraint base to the target datasets. Assuming that a set of features would have a similar ranking of the coefficient values across platforms, the solution of the maximum CSP problem would then also be a mapping of features between F_T and F_S.

Algorithm 1 demonstrates at a high level how the maximum CSP is adapted to the feature mapping. Basically, we try every possible mapping combination between a set of target and source features, and try to maximize the truth statements that the assignments (mapping) in the constraint base will produce while comparing the assigned constraints to the correlation data from the target platform.

Algorithm 1 Pseudoalgorithm demonstrating the functionality of the feature mapping.

```
for each possible mapping $Map_i(F_T, F_S)$ do
    Assign features $F_T$ to the constraint base wrt. $Map_i$
    for each monthly-based correlation matrix in the target platform do
        Count, how many times constraints are satisfied in the target platform with the current assignment.
    end for
end for
return Mappings having the highest count of truth statements
```

The algorithm returns a list of possible mappings, that have the highest satisfiability count. For every possible mapping between features f_Ti and f_Sj, we define a mapping score which is a portion of their occurrence in the returned list. For example, let us consider a mapping case where the problem is to map three features between platforms S and T: $F_s = \{x, y, z\}$ and $F_T = \{a, b, c\}$. The algorithm returns two lists of mappings: $Map_1\{(x, a), (y, b), (z, c)\}$ and $Map_2\{(x, a), (y, c), (z, b)\}$. For this example case, the mapping scores would be: $(x, a) = 1.0$ and 0.5 for $(y, b), (y, c), (z, b)$ and (z, c). The scores indicate that x and a could be mapped with each other while other mappings can not be deduced from these results. Generally, as the method requires inequalities between coefficients, at least three features from both platforms are required at minimum and a higher number of features would provide a richer set of constraints for the analysis.

V. RESULTS

This section evaluates the feature mapping method. The datasets, Netradar and RTR Nettest, are separated into six monthly sub-datasets in order to analyse the variation of the correlation coefficients. With respect to the documentations of the two platforms (Netradar, RTR Nettest), we define the common features as: (dowlink, download_kbit), (uplink, upload_kbit), (latency, ping_ms), (RSRP, RSRP), (RSRQ, RSRQ). These pairs are assumed to be matched with our feature mapping methodology.

A. Correlations

First, we report and analyse the correlation matrices for the features from both platforms. For evaluation purposes we have applied Pearson's correlation method for analysing possible linear relations and Spearman's for non-linear relations between the features.

1) Pearson: Figures 3 and 4 show Pearson's correlation matrices of monthly based data for the Netradar and RTR Nettest platforms. Figures show heat maps with blue indicating positive correlation (1.0 as a maximum value), white no correlation (0.0) and red negative correlation (-1.0 as a minimum value). In Figure 3, the correlations between the common five features are presented in the first five rows and columns.

From the figures can be seen that all five common features clearly have regularities among each other; inside every correlation matrix, the relative positions of the coefficients stay mostly the same. For example, latency (corresponding to ping_ms) and downlink (download_kbit) have mainly lower negative correlation than latency and RSRP in both platforms. Moreover, RSRP has mainly a stronger positive correlation with uplink (upload_kbit) than with downlink in both platforms.

From the Netradar correlation matrix (Figure 3) can be seen that RSRP, signal strength and LTE ASU have 1.0 correlations between each other. This indicates that the features present redundant information about the signal strength and it refers to the issue of having different names but being similar metrics. This is important to consider, because some platform might not have RSRP present in the measurement data, but only LTE ASU or signal strength.

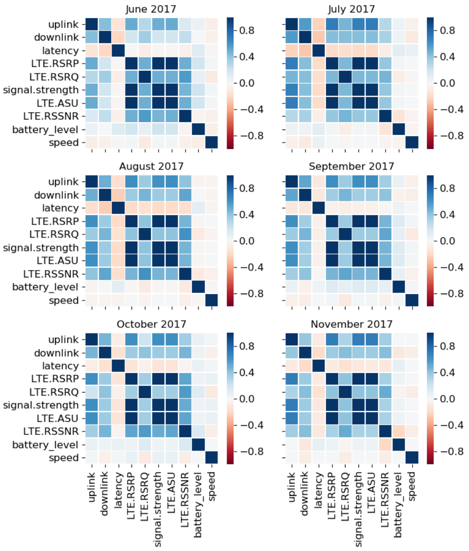


Fig. 3: Pearson's correlations in the Netradar platform. First five rows and columns show the correlations between the common features: uplink, downlink, latency, RSRP, and RSRQ.

2) Spearman: Figures 5 and 6 show the Spearman's correlation matrices for the Netradar and RTR Nettest

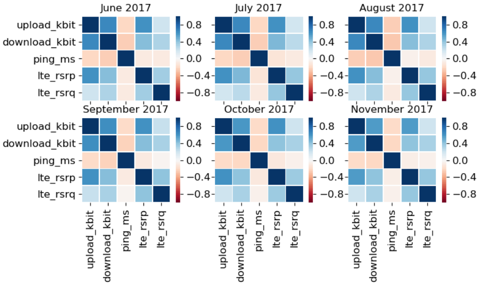


Fig. 4: Pearson's correlations in the RTR Nettest platform.

platforms. Although the actual coefficient values vary between Pearson and Spearman, the relations between the features stay the same. Again, from figures can be noticed that latency correlates stronger with downlink than with RSRP. RSRP in turn correlates stronger with uplink than downlink. Also, Figure 5 shows that the Spearman correlation between RSRP, LTE ASU, and signal strength is 1.0.

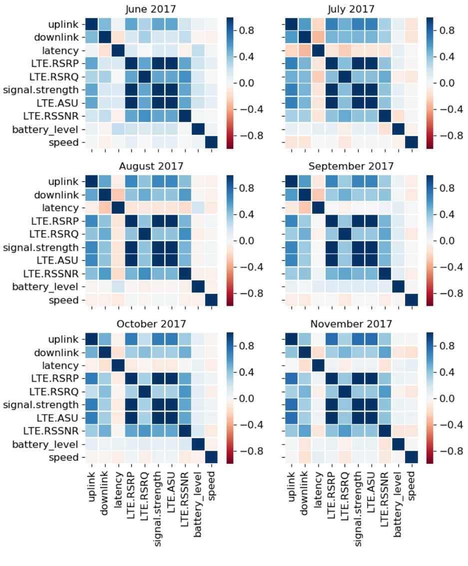


Fig. 5: Spearman's correlations in the Netradar platform. First five rows and columns show the correlations between the common features: uplink, downlink, latency, RSRP, and RSRQ.

Altogether the correlation matrices in Figures 3–6 support our hypothesis that regular patterns between coefficients can be found. Next, we evaluate our methodology with mapping scores for all mapping combinations between the platforms.

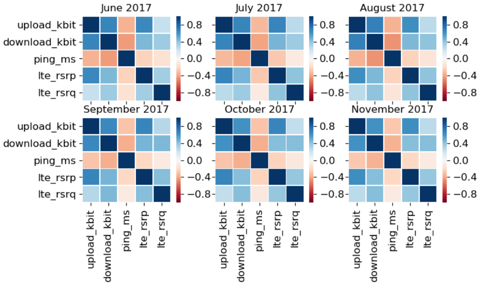


Fig. 6: Spearman's correlations in the RTR Nettest platform.

B. *Average scores of the feature mapping*

To evaluate the feature mapping method, we include all the five features from the RTR Nettest (upload_kbit, download_kbit, ping_ms, lte_rsrp and lte_rsrq) and following eight from the Netradar: uplink, downlink, latency, RSRP, RSRQ, RSSNR, battery_level, and speed. We leave LTE ASU and signal strength out from the evaluation because the correlation matrices clearly indicate that they would give exactly the same results as the RSRP (which is also the desired outcome of the method). The higher number of features in the Netradar dataset allows us to evaluate the mapping in situations when there is not a simple one-to-one mapping between the features, but also some "noisy" features that should be left without a mapping.

For evaluation purposes, we define a mapping score between F_T and F_S for a single mapping case as follows: $\frac{1}{M}\sum_{i=0}^{M} score(f_{Ti}, f_{Si})$, where M is the number of common features between the sets ($M = |F_T \cap F_S|$), f_{Ti} and f_{Si} are the ith common features from the F_T and F_S. As defined in Section III, the common feature-pairs are defined as: (uplink,upload_kbit), (downlink, download_kbit), (latency, ping_ms), (RSRP, lte_rsrp), and (RSRQ, lte_rsrq).

We evaluate the overall performance of the mapping by generating all possible mapping combinations between the source and target features in order to examine how our method catches the different levels of similarities between the feature sets. We present the overall results in plots that show average mapping scores as a function of the ratio of common features (ratio of true positive mappings). This means that we group all mapping cases that have the same ratio of common features and calculate the average mapping score over those. For example, the mapping of three features means that there are 560 mapping combinations ($\binom{5}{3} \times \binom{8}{3}$) in the plot and the point where the common ratio is 3/3, the value is averaged over 10 ($\binom{5}{3}$) different mapping combinations.

Figure 7 shows the average scores of the feature mapping while RTR Nettest is the source and Netradar the target platform. Figure has three subplots separating the mapping results between the mapping of three, four, and five features. All the subplots show that the average scores of perfect mappings (common ratio is 1.0) can clearly be distinguished from mappings having more false positives (common ratio is lower than 1.0). False positives include all imperfect mappings that do not refer to the same metric. Moreover, the increasing trend of the scores as a function of the common ratio can be noticed. This shows the desired outcome that the feature mapping method gives better scores when higher portion of common features are mapped. The different correlation methods, Spearman, Pearson, and their combination (both of them are used in the constraint base), give rather similar results, meaning that linear relations can be used as well as non-linear.

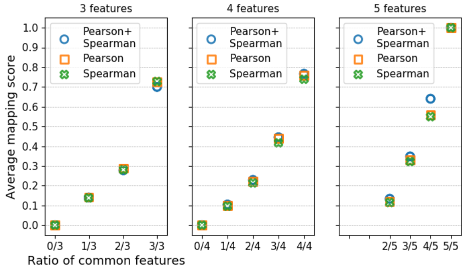


Fig. 7: Average mapping scores when mapping from RTR Nettest to Netradar. The score increases as a function of the common ratio. The perfect mappings (ratio of 1.0) outperforms the incomplete mappings.

Figure 8 shows the mapping results when Netradar is the source, and RTR the target platform and the same features are included in the evaluation. Apart from a small variation, the scores are rather equal to the earlier mapping case shown in Figure 7.

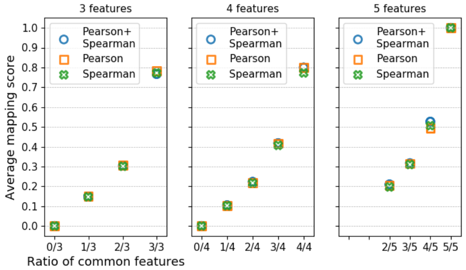


Fig. 8: Average mapping scores when mapping from Netradar to RTR Nettest. The score increases as a function of the common ratio. The perfect mappings (ratio of 1.0) outperforms the incomplete mappings.

Altogether, the average scores show that the feature mapping method is able to distinguish incomplete mappings from the perfect mappings. Regardless of the amount of mapped features, there is a remarkable gap in the scores between the incorrect mappings and correct mappings (between the scores having a common ratio of 1.0 compared with lower ratios). Moreover, we can see that also the scores of the incorrect mappings give an insight of how many correct features there might be between the platforms, as the average score clearly correlates with the common ratio.

C. Feature-specific scores

Next, we evaluate our method from the perspective of the common features in order to report the differences between the features. We define a feature-specific mapping score as an average over all mapping cases, in which the common ratio is 1.0. As we have 16 such cases ($\binom{5}{3} + \binom{5}{4} + \binom{5}{5}$), the feature-specific score of the ith common feature (f_{Ti} and f_{Si} respectively) is $\frac{1}{16}\sum_{j=0}^{16} score(f_{Tij}, f_{Sij})$.

Feature-specific scores of the mappings can be seen in Figure 9. This plot shows that there are more variations between the correlation methods and between the feature scores than that could be seen from the earlier overall results. Because of the differences between Pearson and Spearman score values, we would consider the combined method. The combined method gives rather stable scores regardless of the source and target platform. The highest difference in the performance is in the uplink scores; uplink gets a score of 0.8 while RTR Nettest is the source platform, but 0.96 while Netradar is the source platform.

The latency outperforms all other features having the highest possible mapping score of 1.0. This result is expected as the earlier correlation matrices (Figures 3 – 6) show that latency clearly has the lowest correlations with all the other features, which makes it easy distinguish it with respect to our method. Another finding of these plots is that RSRP and RSRQ are more difficult to map than the other features, as their scores are lower. A closer look to the individual mapping cases shows that RSRP and RSRQ are sometimes mixed up together when only three features were mapped.

Altogether, we may conclude that the feature-specific figure scores are high enough to make correct mappings between all features, but there are some variations between the feature scores. For example, RSRP and RSRQ have lower scores, whereas latency clearly has the highest scores of 1.0. All scores are acceptable as the scores do not present the accuracy, but rather a "voting" score, as explained in Section IV. Thus, any score higher than 0.5 for a common feature implies that on average we would select a correct mapping. Moreover, it should be noted that a random guess would have a mapping score of 0.33, 0.25, or 0.2, depending of the amount of features ($1/N$, in general).

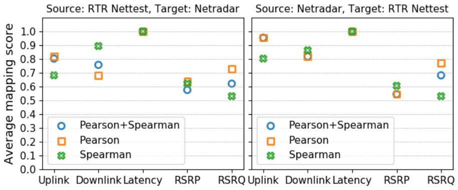


Fig. 9: Feature-specific mapping scores. The scores are high enough to make correct mappings for all features but there is variation between the features.

VI. CONCLUSION

In this work, we designed a feature mapping methodology between two crowdsourced LTE measurement-based platforms. Our objective is to automatically analyse and map similar features across platforms, without a need for manually analyse their similarities. Our method is based on correlations between the features and the mapping algorithm is solving a maximum constraint satisfaction problem (maximum CSP). We defined our constraints as inequality patterns between the correlation coefficients of the measured features.

Our results show that the method maps the common features with high confidence scores (between 0.78 to 1.0 depending on the amount of features). As a desired outcome of the method, the average mapping score increases when more similar features are involved. The results also indicate that there is no significant difference in the average results between using Pearson, Spearman, or their combination. However, some individual features perform slightly better using Pearson (uplink and RSRQ) and some other using Spearman (downlink). Some issues were noticed between features having similar patterns, for example RSRP and RSRQ. However, even then the scores are promising and our results show that this methodology can be used as an automated tool in the measurement data integration.

In the future work, we will include more measurement datasets in the feature mapping and compare to other methods that could be adapted to this task. Moreover, with the merged datasets we may do cross-dataset analysis, for example by using the transfer learning paradigm [20] from the machine learning.

ACKNOWLEDGEMENTS

This work is partially funded by Nokia and the FP7 Marie Curie Initial Training Network (ITN) METRICS project (grant agreement No. 607728)

REFERENCES

[1] "Netradar measurement platform." [Online]. Available: http://www.netradar.org/

[2] "RTR Nettest measurement platform." [Online]. Available: https://www.netztest.at/en/

[3] J. Huang, C. Chen, Y. Pei, Z. Wang, Z. Qian, F. Qian, B. Tiwana, Q. Xu, Z. Mao, M. Zhang *et al.*, "Mobiperf: Mobile network measurement system," *Technical Report. University of Michigan and Microsoft Research*, 2011. [Online]. Available: https://sites.google.com/site/mobiperfdev/

[4] "OpenSignal, Inc. opensignal." [Online]. Available: http://opensignal.com/

[5] "Speedtest.net the global broadband speed test." [Online]. Available: http://speedtest.net/

[6] O. Alay, A. Lutu, M. Peón-Quirós, V. Mancuso, T. Hirsch, K. Evensen, A. Hansen, S. Alfredsson, J. Karlsson, A. Brunstrom, A. Safari Khatouni, M. Mellia, and M. A. Marsan, "Experience: An open platform for experimentation with commercial mobile broadband networks," in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '17. New York, NY, USA: ACM, 2017, pp. 70–78. [Online]. Available: http://doi.acm.org/10.1145/3117811.3117812

[7] K. Kousias, C. Midoglu, O. Alay, A. Lutu, A. Argyriou, and M. Riegler, "The same, only different: Contrasting mobile operator behavior from crowdsourced dataset," in *Personal, Indoor, and Mobile Radio Communications (PIMRC), 2017 IEEE 28th Annual International Symposium on*. IEEE, 2017, pp. 1–6.

[8] E. Walelgne, J. Manner, V. Bajpai, and J. Ott, "Analyzing throughput and stability in cellular networks," ser. IEEE/IFIP Network Operations and Management Symposium, NOMS, Taipei, Taiwan, 2018. [Online]. Available: http://vaibhavbajpai.com/documents/papers/proceedings/netradar-throughput-stability-noms-2018.pdf

[9] "broadbandmapping mapping of broadband services in europe." [Online]. Available: https://www.broadbandmapping.eu/

[10] F. Malandrino, C. F. Chiasserini, and S. Kirkpatrick, "Understanding the present and future of cellular networks through crowdsourced traces," in *2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, June 2017, pp. 1–9.

[11] E. Lipenbergs, A. Stafecka, G. Ivanovs, and I. Smirnova, "Quality of service measurements and service mapping for the mobile internet access," in *2017 Progress In Electromagnetics Research Symposium - Spring (PIERS)*, May 2017, pp. 2526–2532.

[12] L. Li and S. Shen, "End-to-end qos performance management across lte networks," in *2011 13th Asia-Pacific Network Operations and Management Symposium*, Sept 2011, pp. 1–4.

[13] J. Manco-Vásquez, S. V. Vaerenbergh, J. Vía, and I. Santamaría, "Kernel canonical correlation analysis for robust cooperative spectrum sensing in cognitive radio networks," *Transactions on Emerging Telecommunications Technologies*, vol. 28, no. 1, 2017.

[14] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, Feb 2011.

[15] T. van Kasteren, G. Englebienne, B. J. Kröse *et al.*, "Recognizing activities in multiple contexts using transfer learning." in *AAAI Fall Symposium: AI in Eldercare: New Solutions to Old Problems*, 2008.

[16] Y. t. Chiang and J. Y. j. Hsu, "Knowledge transfer in activity recognition using sensor profile," in *2012 9th International Conference on Ubiquitous Intelligence and Computing and 9th International Conference on Autonomic and Trusted Computing*, Sept 2012, pp. 180–187.

[17] C. Wen-Hui, C. Po-Chuan, and Y.-L. Jiang, "Activity recognition using transfer learning," *Sensors and Materials*, vol. 29, no. 7, pp. 897–904, 2017. [Online]. Available: http://dx.doi.org/10.18494/SAM.2017.1546

[18] S. Sonntag, J. Manner, and L. Schulte, "Netradar - Measuring the wireless world," ser. 11th International Symposium and Workshops on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, WiOpt 2013, 2013, pp. 29–34. [Online]. Available: http://ieeexplore.ieee.org/document/6576402/

[19] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Prentice Hall, 1995.

[20] R. Caruana, "Multitask learning," in *Learning to learn*. Springer, 1998, pp. 95–133.

Aalto-DD 193/2019

9 789526 087764

ISBN 978-952-60-8776-4 (printed)
ISBN 978-952-60-8777-1 (pdf)
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)

Aalto University
School of Science
Department of Computer Science
www.aalto.fi

BUSINESS +
ECONOMY

ART +
DESIGN +
ARCHITECTURE

SCIENCE +
TECHNOLOGY

CROSSOVER

DOCTORAL
DISSERTATIONS