Agent composition for 5G management and orchestration

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

***Abstract*—Virtualized execution environments in 5G network call for a linkage between network management and orchestration. Execution of 3GPP functionalities and management applications in the cloud also presents an opportunity to innovate outside of traditional paradigms. We describe a framework focusing on a virtual execution environment and utilizing agent composition to serve as a platform for realizations with particular goals — for example — in terms of coordination. The framework allows for focusing on the opportunities provided with cloud environment and microservices-based agent composition, and describing relevant aspects of orchestration, while avoiding aspects of orchestration which would bring unnecessary complexity to the analysis. The framework is planned to be used as a basis for research demonstrator later on for implementing 5G use cases.**

I. Introduction

Draft goals for improvement of performance in 5G networks over 4G have been described in [1], leading to certain architectural choices in 5G design. These are reflected in network management. For example, the network architecture is cloud-based, with 3GPP applications executed as virtualized network functions (VNFs)[2]. In general, the 5G network deployment is expected to increase technological complexity, highlighting the need for advanced automation. In this article, we focus on network management and orchestration (M&O) in virtualized 5G execution environment for cloud-based 5G network.

Relatively static network management utilizing manual effort and templates has been improved with the advent of Self-Organizing Networks (SON, [3]) in Long-Term Evolution (LTE), typically providing rule-based agent capability for network management automation. Basic LTE SON has limited adaptability due to use of fixed rule bases defining the behaviour of SON functions, the creation and governance of which requires specialized competences. In this article, we shall generalize the concept of 4G agents both in terms of application domain and by considering interactions of agents.

Cloud based execution platforms in 5G necessitate putting network management in context with orchestration, i.e., a mechanism for managing and structuring the execution of virtualized software modules. The ETSI MANO [6] is an initiative for developing management and orchestration for virtualized network functions. In virtual environment, network management deals with configuration, monitoring etc. of VNFs. There are interactions between orchestration and network management, whereby it makes sense to consider them as two aspects of the discipline of M&O. In this article, we present a framework for virtualized 5G M&O. We shall not discuss physical network ”stack” here, even though it is in the scope of the overall 5G system.

Service-oriented architecture [7] superseded tightly integrated Remote Procedure Call (RPC) paradigm by allowing for flexible governance of large-scale systems. The original realization of SoA, WS-*, is somewhat rigid and often substituted with REST [8] in modern realizations for greater flexibility. The Commune project [9] described a system where operability applications can be composed of modules either in a shared run-time execution environment, or interacting with each other via REST calls. A further development of service orientation is the microservices approach [10], [11], where responses to requests are composed out of elementary services which are relatively simple in terms of software implementation. In terms of choosing between orchestration and choreography [11], we use the former due to service composition approach that we are using. From the viewpoint of orchestration, both 3GPP resources for network slices and M&O agents may be targets.

From business viewpoint, the target system for 5G M&O should support minimization of OPEX and human errors resulting from increasing complexity of the networking technology. To this end, system design should support automation in configuring the system so that when additional capabilities are added, new capabilities are automatically detected to the extent possible. Similarly, human users should not be required to immerse themselves in complexities of virtualized application interfaces, but rather communicate with the system by means of high-level objectives.

In what follows, we shall next review some relevant aspects of 5G architecture, and will describe our framework for 5G M&O.

II. 5G architecture

The operator coalition Next Generation Mobile Networks (NGMN) has put forth a set of targets for fifth generation networking [1]. Requirements such as support for sub-10 ms latencies have direct impact on structuring of network capabilities and traffic routing within the network. The foundations of the emerging 5G architecture are described, for example, in [12], [2]. We shall highlight later on some aspects which are relevant to the topic area of this article.

A 5G network leverages cloud-based execution platforms for flexibility and as a basis for economic deployment of mobile network functionality. Cloud platforms will be deployed both in central locations, and near the network edge, both of which have specific roles in 5G system. Telecommunications applications (VNFs) are executed on these virtualized platforms, with access to underlying Software Defined Networks (SDNs) via Application Programming Interfaces (APIs).

Network slicing is intended as mechanism for providing logical mobile networks on a single physical infrastructure [13]. In addition to isolation of traffic, 5G network slices enable differentiated end-to-end services of use case types by marshalling related networking and computing resources into virtual networks for service types [14], [1], [2]. Orchestration of related virtual capabilities is a relevant mechanism for instantiating network slices in 5G. In what follows we shall not discuss network slices in detail, but will consider interplay of orchestration with network management.

The addition of orchestration to 5G architecture complicates use case analysis somewhat. Even though it has a specific task in 5G architecture which does not overlap with network management, NM and orchestration have potential for affecting each others' behaviour. At the most basic level, orchestration needs to ensure that relevant network management capabilities are orchestrated along the 3GPP applications for network slices. A more complex scenario, the orchestration of resources may have an effect on network management. For example, the set of objects that a network management function instance is monitoring and configuring may change as a consequence of orchestration.

III. 5G M&O FRAMEWORK

The framework considers VNFs and M&O agents are executed in a cloud environment. The detailed lifecycle mechanisms relating to VNFs in the execution environment are outside of the scope of this article, and we focus on VNF instantiation and orchestration. Thus, a new instance for a particular functionality may be created in a the cloud, an existing instance may be used as a component in orchestration, new instances of a functionality may be created or existing ones deleted for elasticity management purposes.

Agents react to measurements, and can perform configurations autonomously. Sources of measurement data and configuration targets may be either within the virtualized execution environment (VNFs), or external (e.g. virtualized network configured via SDN API). We focus on the first of these options. Agent composition may be used to build solutions to complex goals by using multiple agents. We shall discuss triggers to composition in the next Section.

We shall next discuss aspects of VNFs, agents, and framework functionalities in more detail.

A. Virtual Network Functions (VNFs)

The VNFs in the virtualized execution environment can be viewed from traditional network management perspective or from orchestration perspective. In this article, we consider these as two aspects of the 5G M&O system.

For network management, VNFs are configuration targets and sources of performance management and fault management data. VNFs are assumed to be accompanied by an Element Management (EM) or an equivalent functionality in the virtual EE. VNF instances are managed by lifecycle functionalities outside of the agent framework.

The VNFs are handled according to relevant system rules, e.g. ones relating to slice orchestration. Network and element management need to be part of VNF orchestration so that when a new VNF instance is created, relevant slice instances are created where necessary. Network management needs to be made aware of changes in the composition of a particular class of agents.

We shall not discuss the data exchange to and from the agents here. One approach for this is to use a suitably designed publish/subscribe bus which provides advantages in telecommunications [16].

B. Agents

An agent in the framework has the capability of autonomously performing configurations to a target based on measurements in a closed-loop manner. As discussed earlier, the automation potential of fixed rule bases is limited from economics perspective. The basic problem is that the response of an agent should take into account varying cell contexts (e.g., loading level), and encoding of the range of variability into a single rule base is not a feasible undertaking, especially taking into use evolution of traffic scenarios with changes in applications, terminal capabilities, and temporal variations in endpoint statistics. Managing a large number of context-specific rule bases is not economically feasible either.

A more advanced agent may adjust its behaviour in an automated manner, either by interpolating between cases [4] or by applying more advanced machine learning (ML) methods. We shall not discuss the algorithms here. The adjustment of an agent's behaviour to measurements requires assessment of the effectiveness of past behaviour. This can be performed either by agent itself, an external verification functionality [5], or a human operability user. The use of MLis believed to be necessary for 5G M&O.

In LTE, SON functions are agents which are invoked to handle a standard use case, such as mobility robustness optimization (MRO) or cell coverage optimization (CCO). Generalizing this idea, agents may be viewed as relating to well-defined use cases not limited to network management. A use case does not need to be implemented by a single agent. Instead, solutions can be composed by a "virtual agent" which is composed of multiple software modules. Composition can be applied to agents proper (e.g., traditional SON functions), or microservice-style "atomic" software modules which do not need to be complete agents in their own right.

Taking generic capabilities of a SON agent as an example, anomaly detection functionality can be a separate module from recovery planning. With this approach, anomaly detection and recovery planning could be combined on case-by-case basis, and individual components used in multiple compositions.

The advantages of reuse are avoidance of replication of functionality between agents, and subsequent easier requirement management and software revision processes. In this example, machine learning can be applied separately in anomaly detection and recovery planning, and separate learning cycles can be applied to them. More generally, one can envision having enablers for agent tasks in the same way as in the Commune architecture [9], but instead of running in a JADE environment, the participating software modules would be orchestrated.

An alternative approach uses simple, complete agents which solve a limited problem autonomously. Solutions to complex tasks could then be composed with a combination of focused agents. While it is difficult to rule out this strategy without a comprehensive use case analysis, the previous approach would seem to facilitate better software reuse. In general, the two approaches do not need to be mutually exclusive. Indeed, one can even mix complex and simple agents in a single mapping environment.

C. Framework services

Software modules — agents or not — may make use of framework services, depending on the realization of the framework. Some of these reside within the same execution environment, whereas some are conceptually external to it.

Configurations to VNFs may require element management (EM) functionality for isolating vendor-specific details from configuration interfaces. Such a module translates configuration intents such as "tilt TRX up" to vendor specific data structures and parameter values. This approach allows creation of software modules which are not tied to vendor software in VNFs.

Lifecycle management of VNFs is another type of framework service which may be used by some software modules. For example, a software module could detect that its processing capacity needs to be boosted in view of the volume of input events encountered. The module itself or a supervising entity could then send a request to lifecycle management capability to create further instances of the module in question within the execution environment. Composition a solution to a complex might involve a software module which is presently not active. In such a case, a request for instantiation would be made to the lifecycle management functionality.

Software modules used in composition need a way to obtain measurements from VNFs. Depending on the case, this can be a peer-to-peer connection, access to measurement database, or subscription to information bus [16]. The interface used for configuration depends on the arrangement, but may be assumed to be a peer-to-peer connection between the origin and the target.

Software module repository describes metadata pertaining to agents. Mapping capability needs this in order to perform composition of agents to carry out a complex task. This topic is beyond current article and will be reported elsewhere.

The configuration targets may reside outside of the execution environment and are also classified as framework services here.

IV. Service composition

Service composition starts with a trigger. The trigger for 5G M&O may be a high-level goal such as "optimize coverage within network scope X", referring to previously defined goal for coverage optimization. The trigger may also be an anomaly observed in the network. The source of the trigger may be a software module or a human operability user. The response to trigger is the outcome of the composite computation if it is successful, an indication of a module computation failing, or "solution does not exist" if no composition exists.

Service composition requires an identification of software modules required to achieve the result – if any – as well as computation required by the composition. Identification can be based on automated processing of metadata describing the modules, or fixed mapping to agents. Composition may take into account previously active compositions or stand-alone software modules to avoid clashes between modules.

Matching of the output of a previous module to the input of the next one is an example of the latter. Information model for inputs and outputs of modules is advantageously part of the metadata of the module, and facilitates automated conversion of information from one module to the next one. More details of the agent composition as well as advanced examples of computation performed as part of the composition will be reported elsewhere.

The participating modules may be one of two types: ones called explicitly from the service composition capability and ones used indirectly in the composition. The former type is called "explictly participating" here, whereas the latter ones may be in communication with the explicitly participating modules. If necessary, service composition entity needs to create instances of explicitly participating modules prior to execution. The presence of the indirectly participating modules requires a communication means between the modules during execution.

The execution phase of service composition consists of calling each explicitly participating module in turn (or possibly also in parallel where feasible) and performing computations required for utilizing the output of the previous module in the input of the next one. If the execution of an explicitly participating module has not been successful, there are two basic possibilities: attempt a recomposition to circumvent the problem, or terminate the composition. In the latter case, an error is returned to the originator of the trigger. The failure of the composition may be stored and used in future compositions to avoid unnecessarily ending in the same situation again. Human user may be alerted to the situation to see whether there is something that can be learned from the situation.

V. Perspective

The framework we have discussed has been designed to allow for analysing interactions between network management and orchestration, and for developing new approaches utilizing the capabilities provided by virtualized environment. Focusing on agents (real or virtual) within the execution environment supports the latter goal. From the viewpoint of network management, lifecycle management of VNFs affects the set of targets for network management agents.

Our framework is based on composition of agents, which can be ”legacy” complex agents (e.g., LTE SON), or virtual agents composed of microservices style elementary capabilities. The framework thus allows for seamless evolution towards microservices approach, while also supporting — for example — composition of legacy SON agents into workflows which may consist of both complex and simple agents. The framework thus allows for multiple paradigms in terms of coordination. In one approach, the execution status of agents may be maintained within the composing unit and instantiation of agents which would conflict with already executing ones can be prevented. This approach requires that the composing unit is able to detect conflicts between agents. Another approach is that composing unit simply ”fires off” agents and they coordinate ”out-of-band” from the viewpoint of composing unit. In the interest of generality of the framework, both modes are possible, but a specific realization may choose one of the modes.

An important benefit of the framework is support for software reuse. Instead of developing new stand-alone software implementations for each use case, the previous modules can be reused. At the simplest level, complex agents can be executed in succession to achieve a desired outcome, e.g. energy saving and coverage optimization for optimizing a RAT layer energy consumption during night time. In view of software process, complex agents most likely contain replicated functionality which could be isolated into microservices and used in developing implementations for future use cases as virtual agents composed of multiple modules. The virtual agents can operate autonomously by means of ”out-of-band” communications not involving the composing unit (after instantiation of the virtual agent), or as a composed workflow where the composing agent mediates the communication between constituent parts of the virtual agent. Using the anomaly as a trigger also supports the mode in which the composing unit would be used for anomaly detection and subsequent execution of recovery planning part of an agent. This is however not expected to be an efficient way of utilizing the framework.

The transition towards microservices also has downsides. There are more service interfaces to be created and managed, whereby an automation of related processes becomes important. This topic will be addressed in future publications.

Finally, the framework allows for use case centric composition of agents so that implementation do not need to be pure network management applications, for example. If the introduction of a new end user service or one-time event requires it, a dedicated agent can be deployed which has network management capabilities alongside service specific ones.

VI. Summary

New requirements for 5G networks lead to architecture design choices which have an impact on network management. In particular, the cloud-based execution and orchestration of resources requires and understanding of the interplay of management and orchestration. Cloud-based execution environment also allows for novel possibilities, but analysis of theses is challenging if the complete complex network slice management machinery is brought into the discussion. In this article, we propose a framework focusing on a virtual execution environment with virtual network function lifecycle management capabilities or orchestration as framework functionalities.

An approach using agent composition allows for supporting legacy network management agents while also enabling virtual agents composed out of microservices. The framework can be used as a basis for different realizations in terms of coordination between agents. The presented framework is used as a basis for demonstrator, the results for which will be reported later on. A detailed analysis of 5G management and orchestration use cases is beyond the scope of this article, but will be utilized in forthcoming publications.

References

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

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

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

[4] 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.

[5] T. Tsvetkov *et al.*, A configuration management assessment method for SON verification, in Proc. ISWCS, Barcelona, Spain, 2014.

[6] ETSI, Network Functions Virtualisation (NFV) — Management and Orchestration, GS NFV-MAN 001, v1.1.1, 2014.

[7] T. Erl, Service-Oriented Architecture: Concepts, Technology, and Design, Prentice Hall, 2005.

[8] R. T. Fielding, Architectural Styles and the Design of Network-based Software Architectures, Doctoral dissertation, University of California, Irvine, 2000.

[9] Deliverable D3.1, Specificaton of cognitive network management framework and functions, Commune Celtic project, 2013.

[10] J. Lewis and M. Fowler, Microservices — a definition of this new architectural term, URI: https://martinfowler.com/articles/microservices.html (visited March, 2017).

[11] N. Dragoni, *et al.*, Microservices: yesterday, today, and tomorrow, URI: https://arxiv.org/pdf/1606.04036v2.pdf (visited March, 2017).

[12] 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 (visited March, 2017).

[13] P. Rost, *et al.*, Mobile network architecture evolution towards 5G, IEEE Communications magazine, May issue, p. 84 ff., 2016.

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

[15] R. Mijumbi *et al.*, Management and orchestration challenges in network functions virtualization, IEEE Communications magazine, January issue, p. 98 ff., 2016.

[16] V. Kojola *et al.*, Distributed computing of management data in a telecommunications network, in Proc. MONAMI, Abu Dhabi, UAE, 2016.