Challenges Towards Automation of Live Telco Network Management: Closed Control Loops

Ishan Vaishnavi
Lenovo (Deutshcland) GmbH
Munich, Germany
ivaishnavi@lenovo.com

Laurent Ciavaglia
Nokia
Paris, France
laurent.ciavaglia@nokia.com

***Abstract*—Higher automation is seen as a key milestone in the evolution of telecom networks. Maintaining a myriad of different technologies over constantly evolving architectures (from 2G-5G) increases operational expenditures and poses real integration and deployment challenges for operators. In addition, the new network dynamism introduced with virtualization, eventually leading to multi-tenancy, will add further complexities for the operator in managing the network. This paper outlines the essential role of open and closed control loops in achieving higher levels of network automation, identifies the related challenges that operators face in deploying control loops in live environments and describes preliminary, standards-oriented solutions.**

Keywords—network management automation, 5G and beyond, Slicing, Analytics, Closed control loops

I. INTRODUCTION

One of the primary aspects that influenced the design of 5G networks is the interest of operators in virtualization. Virtualization offered three main features: the *first* one is the move to inexpensive COTS hardware reducing the immense CAPEX that previous generations of the telecom infrastructure required. The *second* feature is the ability to adapt the resources to the demand. Virtualized resources can be scaled up and out when the demand increases and scaled down when it subsides. *Lastly*, and probably the most impactful, it introduces the concept of slices (or End-to-End services (E2E services)) proposing different dedicated network architectures wherein the network adapts to the different use cases it supports. All these new features also complexify operation and maintenance. New hardware, new technologies and corresponding new management software are likely incompatible with legacy telecom systems management; The ability to adapt resources on demand adds the need to detect and maybe even predict the resource and E2E service usage. Finally, different network architectures or slices over the same physical network implies the ability to manage those independently while minimizing cross-influence over the shared physical infrastructure.

Automation is therefore seen as a key enabler of beyond-5G networks. Approaches to leverage automation for network management are in development in current standardization organizations. Major changes include I) the incorporation of Artificial Intelligence (AI) into the network control and management planes and II) the understanding of how closed control loops could function in multi-vendor contexts and across technology boundaries. Closed control loop aspects are being standardized in ETSI Zero touch network and Service Management (ZSM) ISG and in 3GPP TSG Service and System Aspects WG5 Telecom Management (SA5). While ETSI ZSM works on a set of capabilities required to automate network management across management domains in the operator network, 3GPP SA5 has a telecom architecture oriented focus. Based on our analysis, this paper lists the ongoing challenges faced with deploying and operating closed control loops and proposes the possible enablers required to address these challenges. Where applicable, the development status of the enablers in standardization groups such as ETSI ZSM or SA5 is also mentioned.

978-3-903176-31-7 © 2020 IFIP

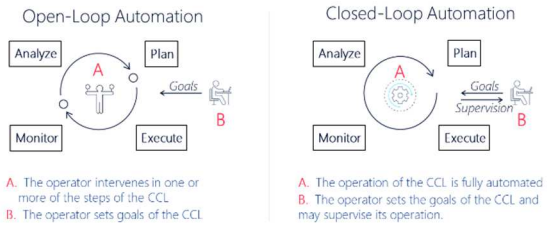

Fig. 1. Open and closed control loops concepts

Section II presents the basics of closed control loops, particularly, works in the design and modeling of closed control loops. Section III looks at the specific architecture of applying automation to a deployed telecom operator infrastructure. Section IV summarizes relevant state of art contributions on closed control loops. Section V presents the main contributions of the paper: the list of challenges identified from the authors' analysis, together with proposed solutions. Finally, conclusions and the future work is presented in Section VI.

II. BASICS: THE CLOSED CONTROL LOOP (CCL)

A. Control Loops – Open and Closed

Previous work [1] defines Open Control Loops (OCL) and Closed Control Loops (CCL) as shown in Fig. 1. With OCLs at least one of the stages in the loop is manually performed. In contrast, with CCLs, the operator only defines a goal and once it is configured, the loop runs automatically. In both OCL and CCL the operator may perform some configurations as part of supervision of the closed loops. Both control loops attempt in controlling the status of a *managed object or managed entity* trying to keep it as close as possible to an operator specified desired goal.

B. Closed Control Loop Models

MAPE-K [4] and OODA [5] are two well-known models of CCL, and have been further adapted in several self-management or autonomic networking architectures such as

Autonomic Computing [4], FOCALE [6], or GANA [7]. Five major logical functions emerge from the design of CCL: 1) Information acquisition, 2) Information analysis, 3) Decision making, 4) Action execution, and Knowledge (K), as illustrated in Fig. 2. The boundaries between these logical functions are not always well defined and may vary from one design to another. Nevertheless, the general functioning of the CCL remains similar. The Knowledge functionality stores and provides access to various types of information (acquired or generated, such as context, historical data, etc.) useful for the operation of the CCL. Different levels of sophistication and cognition levels are also possible depending on the computational techniques employed in each of the CCL steps, such as the approach proposed by the Cognitive-MAPE [8].

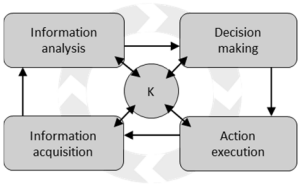

Fig. 2. Generic Closed Control Loop

III. THE TELECOM ARCHITECTURAL VIEW

A. Concept: Management Domains

Management domains are a collection of resources that have their own management system. A management system is for example any set of Management Services (MnS) or their implementations as Management Functions (MnF). Examples of management domains are vendor devices with their management systems, vendor solutions, technical domains such as Radio Access Networks, mobile core, cloud domains (See Fig. 3), datacenters, transport networks with their own controllers, operator administrative domains and so forth. Further details are in ETSI GS ZSM002 [9]. Management domains can be composed recursively of additional management domains.

B. Closed control loops across management domains

To achieve generalized lifecycle automation of the communication services, numerous CCLs will exist at end-to-end and management domain levels, creating a web of interconnected CCLs (Fig. 3). At the top level, the lifecycle management loops (1) are driven by business needs and span multiple levels and domains; the service operation loops (2) encompasses activities such as customer experience management, network slice management and SLA enforcement; finally, the domain loops (3) addresses fault management, performance optimization, security issues, resource scaling. Of upmost importance is coordination between loops to ensure system-wide consistency.

IV. STATE OF ART

The FOCALE architecture [6], loosely based on the OODA model, relies on two interacting sets of loops: the outer control loops react to broader, network-wide changes and influence inner control loops, which are used for fine-grain functional adjustment within a particular context. FOCALE concepts such as the DEN-ng information model or the DENON-ng ontology have been developed in close connection with the standards organizations (TMF and ITU-T) and the Autonomic Communications Forum (ACF); however it has never been widely adopted as a standard solution, although it still constitutes today foundational contributions to autonomic networking.

Closed control loops exist in limited scope as self-organizing networks (SON) since release 9 of 3GPP in [10]. The SON architecture is not a generic CCL architecture but rather a set of specific solutions based on closed control loops for specific use cases broadly classified in three categories:

- Self-configuration: The ability of managed entities to be automatically configured without operator intervention
- Self-optimization: the ability to optimize functionalities of various managed entities and across managed entities
- Self-healing: The ability of detecting possible error or inefficient situations and correct them

SON solutions are still far from a generic CCL-based solution where the operator may assign an appropriate goal or a condition that the CCL must meet, and the CCL thus manages the network. A detailed discussion on SON is found in [11].

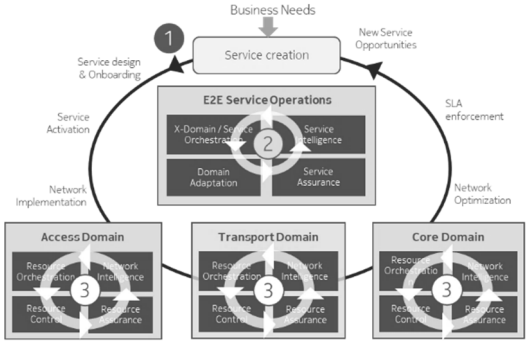

Fig. 3. Automation involves complex interconnected CCLs

The Generic Autonomic Networking Architecture (GANA) [14] , released by ETSI AFI ISG in April 2013, represents the first standardized reference model describing a holistic design of CCLs applied both vertically: from network-level to protocol-level, and horizontally: in intra- and inter-domain, and in multi-technology contexts. A central piece of GANA is the Decision Element that drives a CCL over the "management interfaces" of its assigned Managed Entities, implementing the logic of self-* functionalities. GANA also emphasized the concepts of knowledge and knowledge plane as means to correlate contextualized information to the abstract concepts defined by the information model and the ontology. Although GANA has been regularly extended to emerging technologies and application domains, its inherent complexity resulted in lack of widespread adoption in operational networks.

More recently, the Unified Management Framework (UMF) [15] focused on defining common functionalities and artefacts essential for the management of multi-vendor CCLs, namely Governance, Coordination and Knowledge. In 2015, the UMF contributed to the Autonomic Networking initiative of the IRTF Network Management Research Group (NMRG) with publication of RFC 7575 [16] and RFC 7576 [17], and later with the creation of the IETF ANIMA WG. ANIMA is currently developing standards for different facets of

978-3-903176-31-7 © 2020 IFIP

autonomic networks such as A Generic Autonomic Signaling Protocol (GRASP) [18] or Guidelines for Autonomic Service Agents (ASA)[19] employing CCLs as core components.
Authors in [3] present complimentary challenges towards the limitations in AI/ML models in deployment of automation solutions. The challenges herein are more related to the practical deployment aspects of closed control loops.

V. CHALLENGES IN DEPLOYING CLOSED CONTROL LOOPS

This section investigates the key challenges in deploying multi-vendor CCLs in operational networks.

A. Types of Closed Control Loops

When specifying interoperable solutions, there are essentially two types of CCL that are relevant: 1) Ready-made, pre-integrated CCL and 2) Made-to-order, dynamically composable CCL. Legacy CCLs, with no or limited compliance with standard specifications and requiring specific adaptors, represents a third type of CCLs but are not considered further here.

- Ready-Made (RM) CCLs are pre-assembled CCL, which the operator can configure, deploy and operate. They are provided "as is", i.e. with a pre-defined set of capabilities
- Made-to-Order (M2O) CCLs are assembled on demand. M2O-CCL can be also be configured, deployed and operated; however, M2O-CCLs first need to be built based on standard-compliant building blocks to complete a successful composition. M2O-CCLs provide more flexibility and choice in the CCL capabilities and their (re-)composition, at the cost of additional complexity in managing their assembly at the deployment time

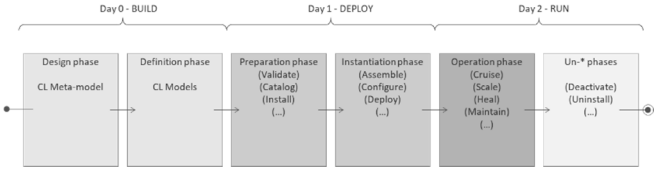

Fig. 4. Main phases of the CCL lifecycle

To enable operators to manage uniformly the different types of CCLs across multiple management domains (see Section III), both CCL types must share common lifecycle operations as shown in Fig. 4 (the M2O-CCL having the mandatory extra-phases of composition and assembly) and common external interfaces. The M2O-CCL requires additional standard specifications for the interactions between its internal components, as illustrated in Fig. 5

B. Control and supervision

Entities external to a managed domain need to understand and possibly influence the behavior of the CCLs within management domains that may in return influence their E2E services. This goes beyond just knowing the status of a CCL. It may require further abilities such as:

- Designing and choosing the right model for the CCL for the operator and the E2E services being managed
- Managing the lifecycle of a CCL
- Providing performance information beyond the simple status such as the average time for the CCL to execute.

Detailed information and control are vital for the deployment of CCLs that manage a myriad of different E2E services that themselves require different sorts of control behaviors.
ETSI GS ZSM009-1 [12] has currently defined CCL Governance Management Services (MnS) for controlling and supervising CCL . Solutions that use these MnS are specified in ETSI GS ZSM009-2 [13].

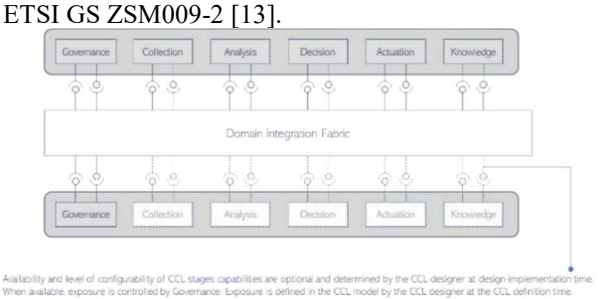

Fig. 5. Service-based representation of RM- and M2O-CCLs in ZSM

C. Trust in CCLs' operation

The operator's trust in the execution of the CCLs is a crucial aspect of automation. Without such a trust an operator would be reluctant to enable the CCLs in the deployed system. Consequently, mechanisms to enhance operator's confidence in CCLs operation need to be introduced. Trust can be addressed in three major ways:

- Ability to quickly update/replace software components that form the CCL. This includes updating the decision logic (e.g. AI-based) software, changing sets of policies and so forth
- Ability to control the execution of a CCL on operator request or set conditions, such as the ability to pause the execution of the CCL
- Logging the actions executed by the CCL for retracing what happened later in the network

While some of these features already exist (e.g. action logging), some other features such as enabling the operator to manage the execution state of a CCL are still under development in ETSI GS ZSM009-1 [12] and ETSI GS ZSM009-2 [13]. Thus far, 3GPP SA5 has specified only CCL goal setting as a supervision feature for Rel 16 in [2].

D. Operational policies

The CCLs autonomy is defined by so-called operational policies following imperative or declarative patterns. Policies allow determining conditions under which autonomous operation is allowed i.e. levels of human oversight, of reporting, and conditions for escalation and delegation; and include specification of objectives (i.e. the concrete goal and scope that the CCL must manage within a certain qualitative and quantitative envelope), and other governance information, needed for proper service operation.
Normally, the CCLs operate within the boundaries defined by the operational policies. Exceptionally, the CCLs try to remediate an abnormality with all possible means under their control, and within the limits set by the operational policies. If having tried all possible and relevant remediation actions, the CCL fails in solving the problem(s), it then generates an "escalation" as defined per the operational policies (i.e. who is the target of the escalation, relevant information, on

remediation actions tried, augmenting the identified situation (exception) with contextual and historical information to further help the analysis by the receiving entity. This escalation forms part of the coordination amongst CCLs.
Further work is required into policy management as applicable to CCLs. Thus far, ETSI GS ZSM009-2 [13] has specified solutions relating to escalation of events that cannot be managed at the domain level.

E. Planned Interactions between cooperating of CCLs

The architecture in Section III.B depicts multiple levels of CCLs across the different management domains. This implies some forms of interactions management to achieve a consistent, collective outcome. Two ends to this interaction have been proposed in ETSI GS ZSM009-1 [12] – peer and hierarchical. With hierarchical interaction the parent loop can completely determine the behavior of the child loop, while in peer relationship this is not true. However, in practice, interactions between CCLs will rarely fall into extremes and are expected to be based on a more authorization-based model where authenticated users are authorized to perform a subset of actions on the CCLs. The questions then remain: which of a CCL's behavior can other CCLs influence? Under what conditions? How is the functionality separated between CCLs? Is there a central closed control loop entity that controls all other closed control loops, or do they function is a distributed consensus driven approach? How is the influence transmitted between closed control loops?

A solution to such questions is the ability to create a possible list of foreseeable CCL interactions based on influencing the various stages in the CCL. External entities including other CCLs may be then authorized to configure or modify varying aspects of the list. Solution examples of such interactions are specified in ETSI GS ZSM009-2 [13]

F. Unplanned CCL interactions - Conflict management

Unforeseen undesired interactions between active CCLs could also occur. Such interactions are typically referred to as conflicts. Managing conflicting situations within and across domains' CCL is required to detect design errors that may have crept into CCL operation. Such a feature should:

- Be applicable for the whole network lifecycle (i.e. build, deploy, and operation), as detailed in TABLE I
- Offer multiple strategies (algorithms) to solve different coordination problems
- Operate on low knowledge and control of CCL internals
- Use common CCL descriptors, lifecycle and information/knowledge models.

TABLE I. CCL COORDINATION LIFECYCLE AND ARTEFACTS

Build time	Static map, a priori knowledge based on CCL descriptor (metrics, parameters, actions…)
Deploy time	Deployed interaction map based on: per instance i) inventory of metrics monitored, of actions performed and computation paths; ii) connected control loops graphs, iii) conflicting control loops identification
Run time	Dynamically updated interaction groups used to i) arbitrate conflict based on coordination strategies and available mechanism, and ii) infer new dependencies

978-3-903176-31-7 © 2020 IFIP

Managing the planned and unplanned interactions between CCLs need further investigation. Management abilities such as performing CCL Coordination considering the different, complementary approaches (e.g. pre- and post-action coordination, policy-based coordination, etc.). Using such enablers, solutions to detect CCL conflicts can be conceived.

G. Integration of AI/ML in CCLs

According to the Cognitive-MAPE principle [7], a cognitive CCL includes ML into its stages making the CCL adaptive and self-driven as opposed to CCLs that are self-regulating but not self-learning (i.e., they do not change the logic based on the outcomes). Applying ML within a CCL requires the integration of one or more trained ML models; each operating on different sets of input data and producing different outputs. The reasons for multiple models may include:

- Using them in combination to implement a more complex analytics and decision process. In this case, the involved models are simultaneously active and should be supplied with their respective input data
- Some models are alternatives to others with different level of functionality or capability. E.g., different models may be created to analyse time series data coming in at different measurement intervals. or if the availability of the input data changes, the CCL may automatically switch between models to select the best fitting one
- Other models may be used only under specific circumstances. E.g., special unexpected cases (e.g., suspicion of an anomaly) trigger further analysis using new models. New models may require different input data, the CCL may need to pro-actively initiate specific data sources to get the data for a newly activated model

These considerations forecast a dynamic interplay between stages of a CCL in contrast to a linear collection-analytics-decision-actuation cycle. With enhanced analytics, CCLs may implement self-learning capability by analyzing the outcome of the actions and reasoning on how well the actions have reached the goal for which they have been initiated.

VI. CONCLUSION AND FUTURE WORK

This paper presented the analysis of the authors on challenges and possible enablers in adopting CCLs in an operational environment. Where applicable, the paper presented the status of the main standards specification bodies with regards to network management automation – particularly ETSI ZSM. Work done in ETSI ZSM on CCLs is already being adopted in 3GPP SA5 and in other groups (e.g. LF ONAP) and is expected to be finalized towards the end of 2020.
Future work will be to further develop CCL enablers at stage-2 specification levels in ETSI GS ZSM009-1[12]. Cooperation between the standards, industry and academic bodies would be required to demonstrate efficient implementation of the aforementioned enablers and the automations solutions being specified in ETSI GS ZSM009-2 [13] in multi-vendor management environments.. This will allow to validate the design and specifications of CCLs in practice and to deliver them as essential building blocks towards network automation in late 5G and beyond-5G networks.

REFERENCES

[1] 3GPP TS28.535 v16.0.0 Management services for communication service assurance; Requirements; 2020.

[2] 3GPP TS28.536 v16.0.0 Management services for communication service assurance; Stage 2 and stage 3; 2020.

[3] Benzaid C, Taleb T. AI-driven Zero Touch Network and Service Management in 5G and Beyond: Challenges and Research Directions. IEEE Network. 2020 Feb 12;34(2):186-94.

[4] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," in *Computer*, vol. 36, no. 1, pp. 41-50, Jan. 2003, doi: 10.1109/MC.2003.1160055.

[5] J. Boyd, "The Essence of Winning and Losing", 28 June 1995

[6] J. Strassner, N. Agoulmine, E. Lehtihet, "FOCALE – A Novel Autonomic Networking Architecture", ITSSA Journal, Vol. 3, No. 1, May 2007, pages 64-79, ISSN 1751-1461

[7] C. Simon, R. Chaparadza, P. Benkö, D. Asztalos and V. Kaldanis, "Enabling autonomicity in the future networks," 2010 IEEE Globecom Workshops, Miami, FL, 2010, pp. 637-641, doi: 10.1109/GLOCOMW.2010.5700398

[8] S. Ayoubi, N. Limam, M.A. Salahuddin, N. Shahriar, R. Boutaba. Machine Learning for Cognitive Network Management. IEEE Communications Magazine. Vol. 56(1), pp. 158-165, Jan 2018

[9] ETSI GS ZSM002 V1.1.1 Zero-touch network and Service Management (ZSM); Reference Architecture. August 2019.

[10] 3GPP TS32.500 Technical Specification Group Services and System Aspects; Telecommunication Management; Self-Organizing Networks (SON); Concepts and requirements

[11] Moysen J, Giupponi L. From 4G to 5G: Self-organized network management meets machine learning. Computer Communications. 2018 Sep 1;129:248-68.

[12] ETSI GS ZSM009-1 0.8.3 Zero-Touch Network and Service Management (ZSM); Closed-loop automation; Enablers. July 2020.

[13] ETSI GS ZSM009-2 0.3.1 Zero-Touch Network and Service Management (ZSM); Closed-loop automation; Solutions July 2020

[14] ETSI GS AFI002 1.1.1 Autonomic network engineering for the self-managingFuture Internet (AFI); Generic Autonomic Network Architecture, April 2013

[15] Tsagkaris, K., Nguengang, G., Galani, A., Yahia, I.G., Ghader, M., Kaloxylos, A., Gruber, M., Kousaridas, A., Bouet, M., Georgoulas, S., Bantouna, A., Alonistioti, N., & Demestichas, P. (2013). A survey of autonomic networking architectures: towards a Unified Management Framework. Int. J. Netw. Manag., 23, 402-423.

[16] Behringer, M.H., Pritikin, M., Bjarnason, S., Clemm, A., Carpenter, B.E., Jiang, S., & Ciavaglia, L. (2015). Autonomic Networking: Definitions and Design Goals. RFC, 7575, 1-16.

[17] Jiang, S., Carpenter, B.E., Behringer, M.H., (2015). General Gap Analysis for Autonomic Networking. RFC, 7576, 1-17.

[18] A Generic Autonomic Signaling Protocol (GRASP), Carsten Bormann, Brian E. Carpenter, Bing Liu, Internet Engineering Task Force, Workin Progress, https://datatracker.ietf.org/doc/html/draft-ietf-anima-grasp, July 2017

[19] Guidelines for Autonomic Service Agents (ASA), Brian E. Carpenter, Laurent Ciavaglia, Sheng Jiang, Peloso Pierre, Internet Engineering Task Force, Work in Progress, https://datatracker.ietf.org/doc/html/draft-carpenter-anima-asa-guidelines, July 2020

978-3-903176-31-7 © 2020 IFIP