



Towards an Autonomic Network Architecture for Self-healing in Telecommunications Networks

Jingxian Lu, Christophe Dousson, Benoit Radier, Francine Krief

► **To cite this version:**

Jingxian Lu, Christophe Dousson, Benoit Radier, Francine Krief. Towards an Autonomic Network Architecture for Self-healing in Telecommunications Networks. Burkhard Stiller; Filip Turck. 4th International Conference on Autonomous Infrastructure, Management and Security (AIMS), Jun 2010, Zurich, Switzerland. Springer, Lecture Notes in Computer Science, LNCS-6155, pp.110-113, 2010, Mechanisms for Autonomous Management of Networks and Services.

HAL Id: hal-01056640

<https://hal.inria.fr/hal-01056640>

Submitted on 20 Aug 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Towards an autonomic network architecture for self-healing in telecommunications networks

Jingxian Lu, Christophe Dousson, Benoit Radier, and Francine Krief

Orange Labs, 2 Avenue Pierre Marzin,
22300 Lannion Cedex, France
{jingxian.lu, christophe.dousson, benoit.radier}@orange-ftgroup.com
francine.krief@labri.fr

Abstract. We propose a solution to achieve a global self-healing process based on local knowledge of each network element using a model-based diagnosis approach. Then we illustrate this solution in an IMS platform and detail the corresponding causal graph and diagnosis process.

Key words: Self-healing, Model-based diagnosis, Causal graph

1 Introduction

With the increased complexity of system installation, maintenance, configuration and recovering, an increase in intelligence in the system is required and the concept of autonomic system will come in.

An autonomic system manages itself and becomes autonomic without human interventions. It has four distinct objectives: Self-configuring, Self-healing, Self-optimizing and Self-protecting. The closed control loop proposed by IBM is implemented by autonomic managers, which control managed resources [1]. With the development of autonomic system, the identifying, tracing, and determining of the cause of failures and the repairing action became more and more important to the operators in order to satisfy the needs of customers. Thus, the purpose of our studies is to introduce self-healing in operator networks.

2 Model-based diagnosis for self-healing

The causal graph is a general, intuitive and effective representation to model the consistent causalities governing the function of the system to monitor [2]. It is supposed to be an acyclic graph that limits the expressiveness of the representation, however, this restriction is important to reduce the complexity of reasoning [3]. The causal graph structure in this paper is based on five kinds of nodes and two kinds of arcs which are shown in Figure 1.

The diagnosis process consists in collecting all the current observations, and then, determines all the primary causes which could explain them. To achieve this process, each node has a unique state as following: **Guilty** (a test result or an active alarm or a proved cause), **Innocent** (a test result or an absent

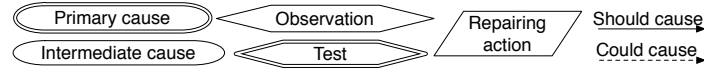


Fig. 1. Nodes and arcs of causal graph

alarm or an exonerated cause), **Suspect** (a test to run or an unknown cause to investigate), **Unknown** (the default state).

Then, applying the causal propagation rules that follows until stability of the node states over the whole graph gives the diagnosis and repairing action:

- R1** : if a node is guilty, then all its “should cause” successors are guilty.
- R1’** : if a node is guilty, at least one of its predecessors is guilty.
- R2** : if a node is innocent, then all its “should cause” predecessors are innocent.
- R2’** : if all the predecessors are innocent, the node is innocent.
- R3** : if a node is guilty, then all its predecessors become suspect.
- R4** : if a node is suspect, then all its “should cause” predecessors are suspect.
- R4’** : if a node is suspect, then all its “should cause” successors are suspect.
- R5** : if a **test** node is suspect, then the test is performed. Depending of the result, the state becomes guilty or innocent.
- R6** : if a repairing action is guilty, then it should be processed.

3 Application to IMS service

The autonomic network architecture used a distributed Knowledge Plane (KP) to interconnect different network elements [4, 5]. We define the **ARM** (Autonomic Resource Manager) to manage network element types and **ANM** (Autonomic Network Manager) to manage the ARM plane and to control the network [6, 7].

We illustrate the previous diagnosis process by applying our approach to the case of a SIP-based VoIP service delivered over an autonomic management architecture (Fig. 2). Let’s suppose that there is the following problem: I-CSCF sends the user’s IP Multimedia Public Identity (IMPU) to SLF which can not find the corresponding user information.

According to the above problem, the global diagnosis relies on the causal graph shown in the Figure 3. We have divided it into four regions. Each region is a subgraph which corresponds the ARM or the ANM. The operators validate each subgraph in each ARM to decide if it could be involved in a global diagnosis when the equipment are integrated in the network. ARM allows the administrator to understand in a smooth way how the equipment could interact with contextual information. If the ARM is compatible with the operators’ expectation, the subgraph is integrated in the KP of ARM and ANM by connecting the different subgraphs together. As a consequence, a primary cause in a subgraph could become an intermediate cause in the implicit global graph. It’s why we need to standardize the interface (showed in the Fig. 3 as arrows between

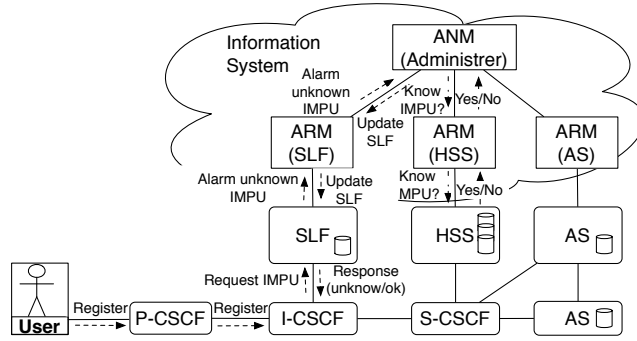


Fig. 2. The case of a SIP-based VoIP service

the black nodes, which cross the domains) between different domains to permit the connection between different subgraphs. After that, the global graph can be used by the diagnosis process in a closed control loop way. Today, there is no autonomous solution which allows to solve this validation step, human expert is still required.

After that, the diagnosis in SLF and AS domains are processed locally. When the diagnosis cannot be completed at the local level, it needs to rise to the global environment (ANM) and collaborates with the other local diagnosis to find or discard or confirm the primary causes. That will ensure an end-to-end diagnosis service. The causal graph modelling can be merged or split according to the needs of the distribution of diagnosis service in a network environment.

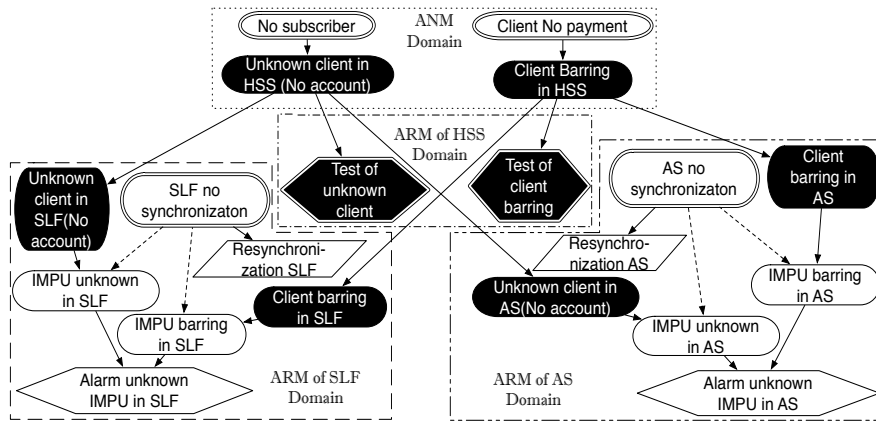


Fig. 3. The global diagnosis

Running the diagnosis process (Section 2) on the example of IMPU alarm is procedure in the following steps by applying the specified rules:

1. (R3): As “Alarm unknown IMPU in SLF” is guilty (alarm present), then “IMPU unknown in SLF” and “IMPU barring in SLF” are suspect.
2. (R4): “IMPU unknown in SLF” is suspect, then its predecessor “Unknown client in SLF (No account)” is set to suspect.
3. (R4): “Unknown client in SLF (No account)” is suspect, then “Unknown client in HSS (No account)” is set to suspect.
4. (R4’): “Unknown client in HSS (No account)” is suspect, then “Unknown client in AS (No account)” and “IMPU unknown in AS” are set to suspect.
5. (R2): “Alarm unknown IMPU in AS” is innocent (no alarm presents), then “IMPU unknown in AS”, “Unknown client in AS (No account)” and “Unknown client in HSS (No account)” are set to innocent.
6. (R2’): Then “Unknown client in SLF (No account)” is set to innocent.
7. (R2,R2’,R4): “IMPU barring in SLF” is suspect, and the same process will be launched as steps 2 to 6. Then “Client barring in SLF” is set to innocent.
8. (R1’): As “Unknown client in SLF (No account)” and “Client barring in SLF” are innocent, then “SLF no synchronization” is guilty.
9. (R6): The repairing action “Resynchronization SLF” is performed.

4 Conclusion

We have proposed a solution to achieve a global network diagnosis with local knowledge of each network element using causal graphs validated by the operators. It permits to connect different necessary processes (subgraphs) to achieve an end-to-end diagnosis and to perform the repairing actions.

The further research should aim to specify the interface between elements or domains which corresponds to the connection between different local diagnosers and consider how the performance of the global diagnosis process could be adapted to our autonomous architecture by defining when and how to send information. Then, we will implement our self-healing approach on an operational IMS platform.

References

1. IBM: An architectural blueprint for autonomic computing. Technical report, IBM White paper (June 2005)
2. Poole, D.: Normality and faults in logic-based diagnosis. *Artificial Intelligence* (1989) 1304–1310
3. Console, L., Dupré, D.T., Torasso, P.: A theory of diagnosis for incomplete causal models. In: *Proc. of IJCAI, Detroit, USA* (1989) 1311–1317
4. ANA: Autonomic Network Architecture. <http://www.ana-project.org>
5. Mbaye, M., Krief, F.: *Autonomic Networking : The Knowledge Plane. System and information sciences notes edn. Volume 1.* (July 2007)
6. Strassner, J.C., Agoulmine, N., Lehtihet, E.: *FOCALE – A novel autonomic networking architecture. Volume 3.* (May 2007) 64–79
7. Strassner, J., Kim, S.S., Hong, J.W.K.: The design of an autonomic communication element to manage future internet services. In: *APNOMS.* (2009) 122–132