Analytics-based Operation for Implementing Service Co-creation Networks

Keisuke Ishibashi

Abstract

At NTT Network Technology Laboratories, we are working on ways to accelerate the speed of responding to network faults, as well as to increase accuracy and save labor when doing so. This includes analyzing data obtainable within and outside the network such as equipment logs, traffic, and trouble tickets, in order to achieve service co-creation networks. A key method to achieve this is analytics-based operation, which we introduce in this article.

Keywords: fault detection, cause identification, recovery automation

1. Introduction

At NTT Network Technology Laboratories, one of our objectives is to accelerate the speed of responding to network faults, and to do so with greater accuracy and with less labor. The ultimate goal is to realize service co-creation networks, which will enable us to achieve safe, secure, and easy-to-handle end-to-end service management [1]. As networks have become larger and more complex, it has become more difficult to detect network faults, isolate the causes, and understand their effects on services. As a result, the longer times needed for recovery work and the increasing amounts of recovery work are becoming problems. Although progress is being made in formulating responses to frequent, stereotypical faults and automating them to increase speed and save labor, there are also cases for which recovery is taking longer, particularly silent faults, which are faults detected through user reports, and unique faults that occur infrequently.

However, the developments in big data and machine learning technologies in recent years may be useful in detecting faults, identifying their causes, and estimating their effects on services by inferring the underlying network state from data obtained within and outside the network. NTT Network Technology Laboratories is conducting research and development on network operation methods, which we call analytics-based operation, that can be used to respond to faults more quickly and accurately and save labor by inferring the network state from network data using machine learning technology.

2. Elemental methods of analytics-based operation

We are studying recovery of network service faults in the three stages of detection, cause analysis, and recovery. For the detection stage, we are developing methods such as service state visualization, support for visual monitoring work, silent fault detection, and predictor detection. For the cause-analysis stage, we are developing methods to identify the causes of faults, identify the locations, and understand their effects on services. For the recovery stage, we are developing methods to formulate recovery tasks and automate recovery work (**Fig. 1**).

For detection, we infer the network state from data within and outside the network and detect whether it is a fault or a predictor of a fault by determining whether that state is associated with a past fault and whether it is an unusual or abnormal state. For cause analysis, particularly for silent faults not detected by

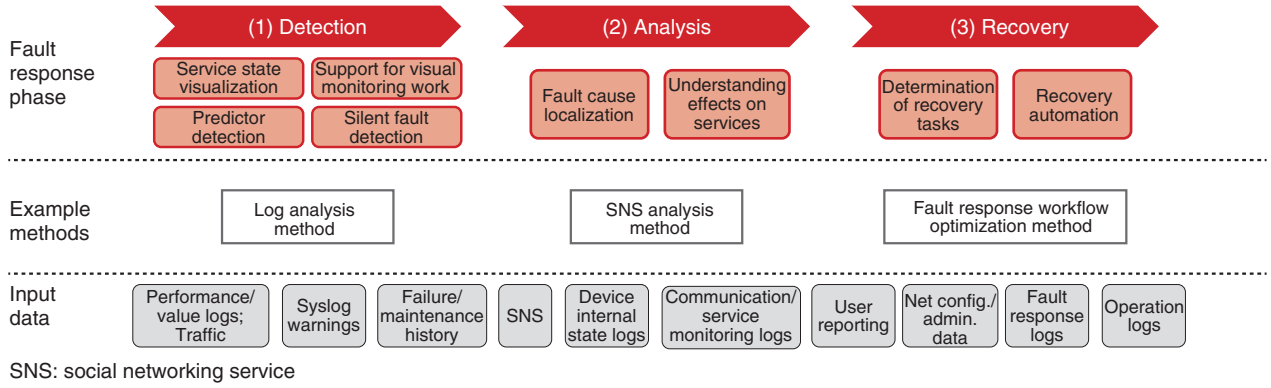


Fig. 1. Elemental methods of analytics-based operation.

device alerts, we identify the location and cause. At the same time, we evaluate the level of urgency in responding to the fault by estimating any effects on services. Finally, for recovery, we take recovery measures such as replacing or restarting components based on the causes determined by the analysis. We give an overview of these elemental methods used for implementing analytics-based operation below.

2.1 Detection

Data sources that can be used to infer the network state include alerts issued by equipment, syslogs indicating state changes, and performance logs indicating usage of resources such as links and the CPU (central processing unit). Data from outside the network such as service monitoring data from test calls, customer reports, and data from Twitter* can also be used to infer the network state. We combine these data sources to infer the network state.

However, syslogs are text logs with vendor-specific formats, so they are difficult to process statistically. They also generally have one message per line, while state changes result in multiple messages. Thus, detecting state changes is a matter of grouping multiple messages. We are studying machine learning approaches to handling this issue. Specifically, we have established a vendor-independent method for creating templates that do not require prior knowledge by estimating parameter settings from multiple syslogs and by grouping logs of the same type by ignoring parts that are the same. We have also established a method for extracting events, which groups messages likely to occur at the same time [2] (**Fig. 2**).

We can also detect faults by inferring the state from these data sources. Conventionally, faults have been detected using rules such as those indicating when particular strings appear in the syslog or when threshold values are exceeded in performance logs. However, in some conditions it is difficult to determine whether or not a fault exists, and false positives and false negatives can be a problem in these cases. Such problems can potentially be resolved using machine learning.

Fault detection with machine learning can be broadly divided into supervised approaches that learn states related to faults that have occurred in the past and that use those states to detect faults or predictors of faults, and nonsupervised approaches that detect abnormalities by finding statistical deviations from the normal state. The latter is advantageous in that it can apply to previously unknown faults, although it can be difficult to identify specifically what happened. For the latter approach, we have established a way of detecting abnormalities by using a dual approach of extracting occurrences in syslogs that are unique to faults, and also extracting those faults that occur periodically or infrequently [3].

2.2 Identifying fault causes and understanding effects on services

To recover or otherwise handle faults that are detected, the causes must be isolated and identified, and the effects on services must be understood. This consists of estimating both the causes and results of the phenomenon. For faults that occur frequently in a set manner, procedures can often be defined for isolating and identifying them, but unique and unknown

* Twitter is a registered trademark of Twitter, Inc.

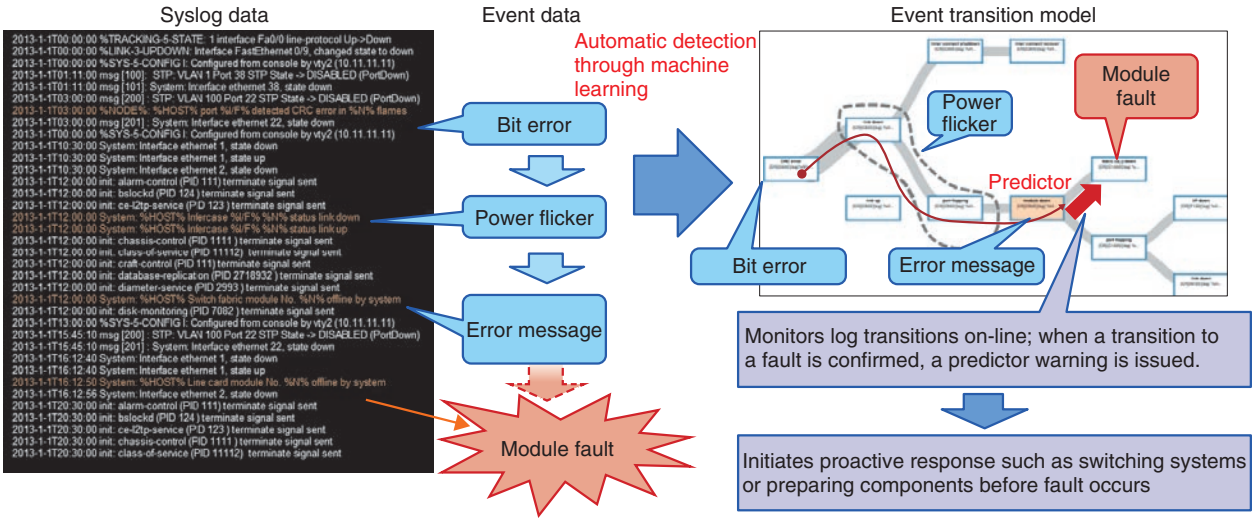


Fig. 2. Method for extracting events.

faults can take longer to isolate and identify. One potentially effective way to handle this is to use the machine learning results described earlier in the opposite direction to infer the fault that is the cause of the observed state. Also, to estimate the effects on services, it can be useful to obtain information outside the network such as service monitoring data using test calls, customer-reported information, or Twitter data [4], although the coverage of test calls and the relation between faults and customer reports is not always reliable. As such, it is possible that effects on services may be understood more accurately by using methods for estimating the state from sampled, incomplete, or noisy data.

2.3 Fault recovery

To reduce the operations workload and the amount of down time, procedures for handling faults must be clarified. If such procedures are not clear, operators can decide how to respond to the fault from the response history, which describes how abnormalities have been handled in the past (trouble-ticket data). However, reading and understanding the response history, which contains large amounts of mixed information, and deciding how to deal with the fault requires a great deal of work by the operator and prolongs the time that the fault is occurring. Therefore, to formulate and automate the work of fault recovery, we examined the free-form descriptions in the fault-response records (trouble-ticket logs) and developed techniques to extract task items from these records, generate work flows based on tasks identified in multiple records, and extract branch points of workflows using clustering, in order to establish an overall fault response workflow visualization method [5] (**Fig. 3**).

3. Future development

In this article, we described technical problems and potential ways to solve them using data analysis in fault detection, analysis, and recovery procedures to achieve more sophisticated fault handling. We also introduced related technologies that NTT Network Technology Laboratories is working on. We will continue our technical development in the future using a diversity of data to improve quality for customers and reduce operational workloads.

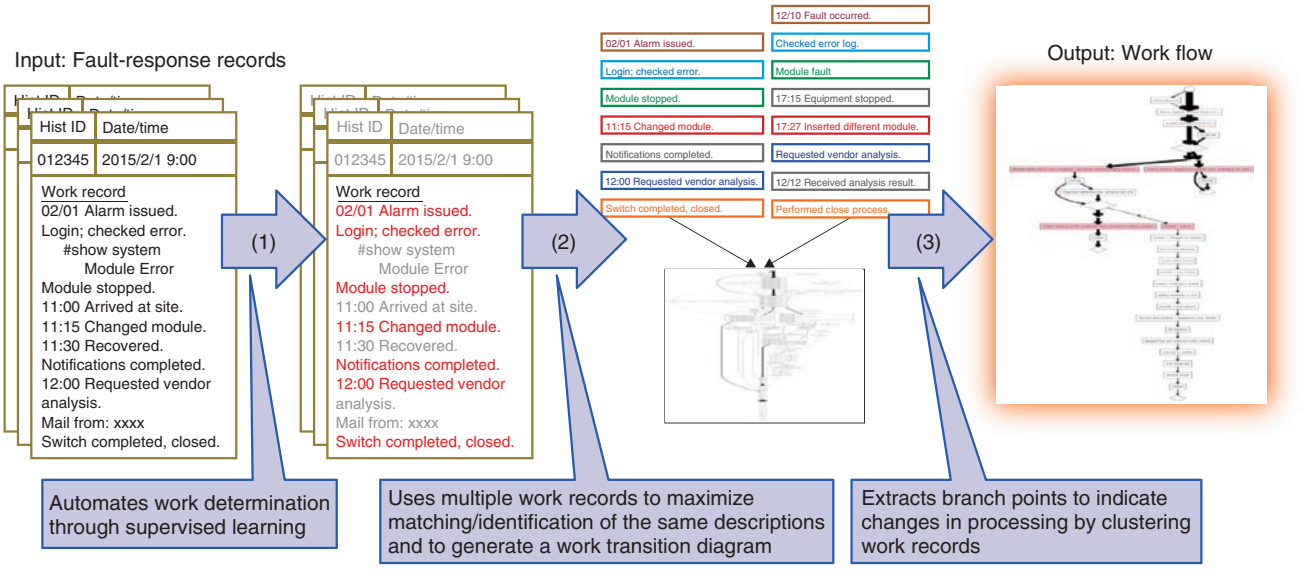


Fig. 3. Workflow visualization method.

References

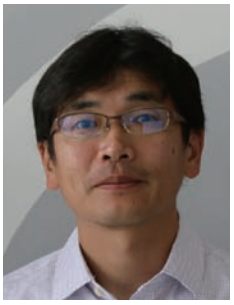
[1] K. Shiomoto, "Approach to Network Science—Solving Complex Network Problems through an Interdisciplinary Approach," NTT Technical Review, Vol. 13, No. 9, 2015. https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201509fa1.html

[2] T. Kimura, K. Ishibashi, T. Mori, H. Sawada, T. Toyono, K. Nishimatsu, A. Watanabe, A. Shimoda, and K. Shiomoto, "Spatio-temporal Factorization of Log Data for Understanding Network Events," Proc. of INFOCOM 2014 (33rd IEEE International Conference on Computer Communications), pp. 610–618, Toronto, ON, Canada, Apr./May 2014.

[3] T. Kimura, T. Mori, T. Toyono, K. Ishibashi, and K. Shiomoto, "Detecting Anomalous Network Events Based on the Log Data Generation Patterns," Proc. of IEICE General Conference, B-6-80, Gifu, Japan, Mar. 2013.

[4] K. Takeshita, M. Yokota, and K. Nishimatsu, "Early Network Failure Detection System by Analyzing Twitter Data," Proc. of IM 2015 (14th IFIP/IEEE Symposium on Integrated Network and Service Management), Ottawa, ON, Canada, May 2015.

[5] A. Watanabe, T. Kimura, T. Toyono, and K. Ishibashi, "Branch Point Extraction from Process Event Logs for Operational Workflow Mining," IEICE Tech. Rep., Vol. 114, No. 523, ICM2014-63, pp. 55–60, 2015.



Keisuke Ishibashi

Senior Research Engineer, Supervisor, NTT Network Technology Laboratories.

He received a B.S. and M.S. in mathematics from Tohoku University, Miyagi, in 1993 and 1995, respectively, and a Ph.D. in information science and technology from the University of Tokyo in 2005. Since joining NTT in 1995, he has been researching traffic issues in computer communication networks. He received the Young Researcher's Award from the Institute of Electronics, Information and Communication Engineers (IEICE) in 2002, the Information Network Research Award in 2002 and 2010, and the Internet Architecture Research Award in 2009. He is a member of the Institute of Electrical and Electronics Engineers, IEICE, and the Operations Research Society of Japan.