A Use Case Modeling of Telecommunication Network Fault Correction for the Component-Based Software Development

김효수*, 신정길**, 이용권***, 백청호****,
*한림정보산업대 전산정보처리과, **남서울대학교 정보통신학부,
경동대학교 정보통신공학부,*강원대학교 컴퓨터과학과,
e-mail: khyosoo@mail.hallym-c.ac.kr, pch@cc.kangwon.ac.kr

TMN 컴포넌트 기반 소프트웨어 개발을 위한 망 장애해결 Use Case 모델링

Hyo-Soo Kim *, Shin, jeong gyll**, Yong-kwon Lee ***, Cheong Ho Baek ****,
*Dept. of Computer Aided Information Processing, Hallym College of Information & Industry, **Dept. of Computer Science, Namseoul University,
***School of Information Technology Engineering, Kyung Dong University,
****Dept. of Computer Science, Kangwon National University

요 약

통신산업환경의 변화 즉 통신시장의 개방화, 글로벌화 및 경쟁체제의 돌입으로 통신사업자는 기존의 단위시설의 운용관리 개념에서 네트워크관리, 서비스관리, 사업관리개념으로 통합 발전되고 있는 국제 표준권고의 통신망의 운용 경영 방식인 TMN(Telecommunications Management Network)에 부합한 개방형 운용정보공유 및 일원화된 통신망 운용관리시스템 구축 및 관리로 전환하려는 추세이다. 본 연구에서는 TMN 컴포넌트 기반 소프트웨어 개발을 위한 망 장애 해결 Use Case 모델링을 설계하였다.

1. INTRODUCTION

A component is a software package, which offers service through interfaces [1]. A component is a coherent package of software artifacts that can be independently developed and delivered as a unit and that can be composed, unchanged, with other components to build something larger [2]. Component perspectives are based on the following three areas:

① A package perspective (as a unit of delivery) focusing on identifying a set of elements, which can be reused as a unit.

② A service perspective (as a provider of functionality) emphasizing the notion of a contract between the provider and the consumer of those services.

③ An integrity perspective (as an encapsulation boundary) being independent of the implementation of other components.

These perspectives are not mutually exclusive [3]. Components can be reused from in-house building components, components purchased from outside vendors [4]. Component Based Development is an approach to application development in which ready-made pieces of software are assembled together to enable the rapid construction of applications [2]. General CBD approach is described as follows [5].

① to decide on the scope of the application to be developed

② to specify the components that will form the application

③ to acquire, reuse, or build the component implementations

④ to assemble, test, and field the application

TMN (Telecommunications Management Network) is based on the OSI (open system interconnection) management concept and uses a standard language to communicate. It's particular strength is it's ability to cater to multivendor environments while meeting traditional operation and maintenance needs (provisioning, testing, data collection, analysis, fault locating, network and service restoration, customer reconfiguration & control and bandwidth capacity management)[6]. TMN is an excellent and comprehensive tool for examining the different layers of management that are required by a service provider [7]. The Bellcore TMN document references the ITU-T Recommendations M.3010 that serves as an international standard on TMN [8]. As noted in [9], TMN must start to be implemented in direct response to clear business goals, such as the automation of business processes in order to reduce operating costs while improving customer service, rather than just because it is a 'standard'.

2. OBJECTIVE OF THIS RESEARCH

The objective of this research is to introduce TMN-based architecture, which provides industry-standard foundation for world-class telecommunications system development and acquisition, linkage between telecommunications business strategy and operational execution.

It will concentrate on the network fault correction process pertaining to TMN fault management. And it also develops that use of the TMN-based architecture will lead to the Use Case modeling of network fault correction for the component-based software development.

3. LAYER MODEL OF BELLCORE FAULT CORRECTION

Bellcore fault management consists of the following areas: fault localization, fault correction, trouble administration, alarm surveillance, testing, RAS(Reliability, Availability and Survivability) quality assurance. This research concentrates on fault correction area.

Fault correction is responsible for the repair of a fault and for the control of procedures that use redundant resources to replace equipment that have failed. The layer model of the fault correction shows the functions of the individual layers [Table 3.1].

Table 3.1 Layer Model of Fault Correction

Layer	Function
BML	- Management of repair process
SML	-Arrangement of repair with customer
NML	-Scheduling and dispatch admin of repair force
EML	- NE(s) fault correction
NE	- Automatic restoration

4. A TMN-BASED USECASE MODELLING

The TMN-based fault management use case diagram is presented in the first section. The second section is allocated to an OMT (Object Modeling Technique) structural diagram for network fault correction process. The last section is allocated to a sequence diagram for network fault correction.

4.1 A TMN-BASED FAULT MANAGEMENT USE CASE DIAGRAM

During a development, we can models of the system we are to design from the requirement specification, a requirements model is created in which we specify all the functionality of the system. This is mainly done by use cases in the use case model, which is a part of the requirement model. They represent everything that needs to exchange information with the system [10][11][12]. The Bellcore fault management use case diagram is presented in Figure 4.1[13].

4.2 AN OMT STRUCTURAL DIAGRAM FOR NETWORK FAULT CORRECTION

Fault correction process model is invoked when a trouble ticket has been issued with the root cause of the fault located. Further test may be done to confirm the root cause. The trouble is repaired by scheduling and dispatching of the repair work force. A TMN-based OMT structural diagram for fault correction is presented in Figure 4.2.

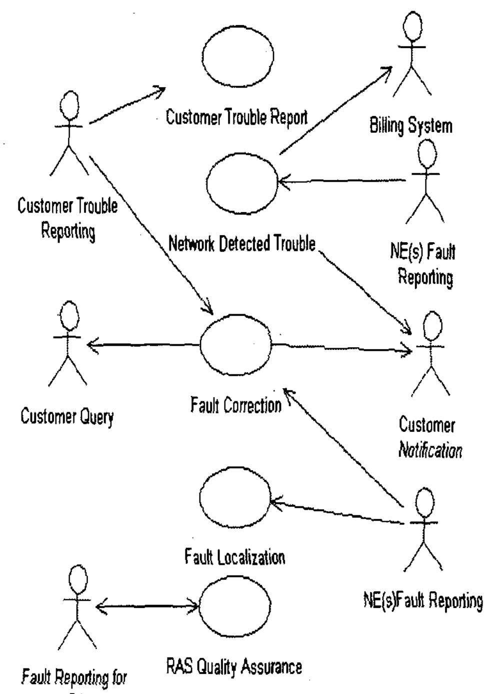

Figure 4.1 A Fault Management Use Case Diagram

4.3 A SEQUENCE DIAGRAM FOR NETWORK FAULT CORRECTION

A sequence diagram is useful for capturing interactions. To describe a sequence of stimuli (that is the event of one object communicating with another)[10], we use Rational Rose 2000 and describe a sequence diagram as a sequences of stimuli sent between the blocks within the system and the actors. A sequence diagram for TMN-based network fault correction is presented in Figure 4.3.

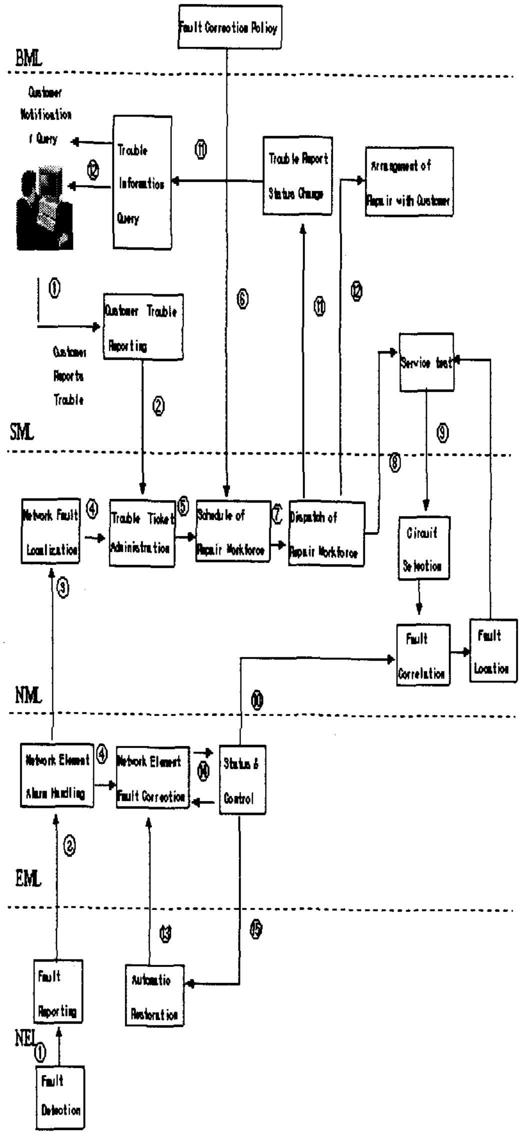

Figure 4.2 An OMT Structural Diagram For Fault Correction

The TMN-based network fault correction starts with either the customer reports trouble or the network element detects a fault. The activities in the SML primarily deal with customer notification, customer query, arrangement fault correction at the EML will attempt to correct the fault automatically. The TMN-based network fault correction model has formulated scenario as follows:

① *SML Customer Trouble Reporting* receives customer trouble report. Failure event detection in *NEL Fault Detection* is sent to N*EL Fault reporting.*

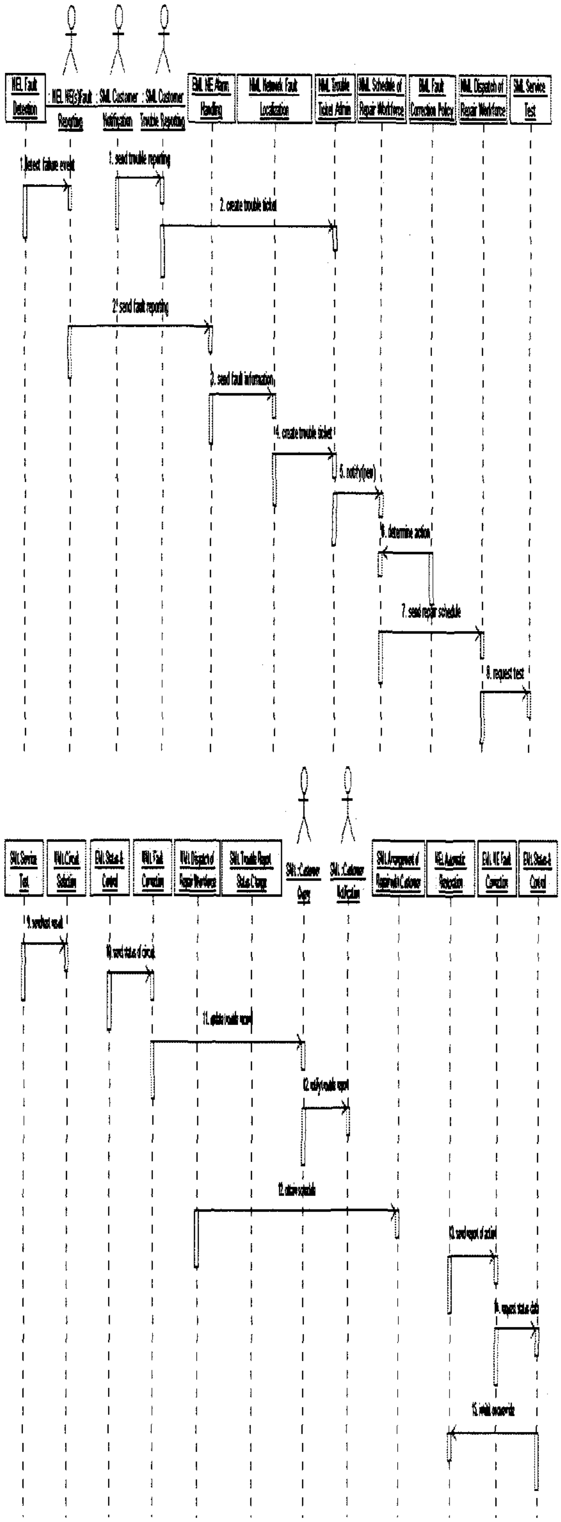

Figure 4.3 A Sequence Diagram For TMN-based Network Fault Correction

② *SML Customer Trouble Reporting* creates a trouble ticket on a customer's circuit. The trouble ticket is stored in *NML Trouble Ticket Administration*'s database. A fault reporting in *NEL Fault Reporting* is sent to *EML NE(s) Alarm Handling* for alarm correlation and filtering of a new alarm event.

③ *EML NE(s) Alarm Handling* sends the fault information and the root cause of a trouble to *NML Network Fault Localization.*

④ *NML Network Fault Localization* creates a trouble ticket, which is stored in *NML Trouble Ticket Administration*'s database.

⑤ *NML Schedule of Repair Workforce* is notified of the new trouble ticket.

⑥ *NML Schedule of Repair Workforce* reads data tables in *BML Fault Correction Policy* to determine the next action for the task.

⑦ *NML Schedule of Repair Workforce* sends the repair schedule of the new trouble ticket to *NML Dispatch of Repair Workforce.*

⑧ *NML Schedule of Repair Workforce* may request *SML Service Test* to perform a test.

⑨ *NML Circuit Selection, NML Fault Correlation* and *NML Fault Location* may request *SML service test* to perform a test and request the result.

⑩ *NML Circuit Selection, NML Fault Correlation* and *NML Fault Location* may read information on the status of the circuit components from *EML NE(s) Status & Control.*

⑪ As work on the trouble progresses, *NML Dispatch of Repair Workforce* may update a trouble record in *SML Trouble Report Status Change.*

⑫ The *SML Customer Query* may access *SML Trouble Information Query* to read trouble records. *SML Trouble Information Query* notifies *SML Customer Notification* for the trouble report status change or trouble ticket creation. If it is necessary to visit the customer premises, *NML Dispatch of Repair Workforce* may obtain a schedule from *SML Arrangement of Repair with Customer.*

⑬ *NEL Automatic Restoration* isolates the failed resource and activates a spare resource. A report of the action is sent to *EML NE(s) Fault Correction.*

⑭ In the course of repair, *EML NE(s) Fault Correction* may request status data from the *EML Status & Control. EML NE(s) Fault Correction* sends a change in restoration status to a database in NE(s) functionality of the *EML Status and Control.*

⑮ In the course of repair, *NEL Automatic Restoration* may be inhibited or overridden by *EML Status & Control.*

5. CONCLUSION

The objective of this research is to concentrate on the network fault correction process pertaining to TMN fault management area. And it will concentrate on the network fault correction process pertaining to TMN fault management. And it also develops that use of TMN-based architecture will lead to the use case modeling of network fault correction for the component-based software development. In chapter 3, we reviewed the layer model of Bellcore fault correction process.

In chapter 4, we have developed a TMN-based use case modeling for network fault correction. The fault management use case diagram is presented in the first section. The second section is allocated to an OMT structural diagram for network fault correction. The last section is allocated to describe a sequence diagram for network fault correction. A TMN-based fault management use case diagram has been presented in Figure 4.1. A TMN-based OMT structural diagram for network fault correction has been presented in Figure 4.2. A sequence diagram for TMN- based network fault correction is presented in Figure 4.3.

The results indicate that this TMN-based use case modeling for network fault correction can be used to help CIO and process managers effectively analyze and establish TMN-based fault correction for the component-based software development. Also this research can be used to build the fault management system, which is one of the network management application building blocks. The future work of this research is to develop use case and sequence diagram of network fault localization process.

REFERENCES

[1] Peter Herzum and Oliver Sims Business Components Factory: A Comprehensive Overview of Component-Based Development for the Enterprise, John Wiley & Sons, Incorporated, 1999.

[2] D.F. D Souza and A.C. Wills Objects, Components, And Frameworks with UML The Catalysis Approach, Addison-Wesley, 1998

[3] Alan Brown, Sterling Software, White Paper, Building Systems from Pieces with Component-Based Software Engineering, 1999

[4] http://nkcho.isp.st

[5] Alan Brown, Moving form Components to CBD: Supporting distributed computing paradigms, Component Strategies, April 1999

[6] http://www.telecoms-mag.com/moong/articles/feb97 /vuilli, Feb.1997.

[7] NMF, "Implementing Omnipoint ", part 2: The NMF's PROGRAMS

[8] M.3010 ITU-T Services M recommendation; TMN, ITU-T, Geneva, 1992.

[9] Keith J. Willetts and Elizabeth K. Adams, "TMN2000: Evolving TMN as Global Telecommunications prepares for the Next Millennium", IEICE Trans. Commun, Vol.E80-B, No.6 June 1997.

[10] Ivar Jacobson et al., Object-Oriented Software Engineering: A Use Case Driven Approach. acm press, Addison-Wesley, 1995.

[11] http://www.kisco.co.kr11-body.html.

[12] http://www.toa.com/pub/html/use_case.html

[13] UML Rational Rose 2000,Rational Software.

[14] http://www.rational.com