A Model-Driven Conformance Testing Method for 3G Network Management North Bound Interface

Liu Yi-chang, Chen Ying-hui, Qi Feng, Qiu Xue-song
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing, China
ycliu@bupt.edu.cn

Abstract—**Based on the Model-Driven Testing (MDT), this paper proposed a model-driven conformance testing method for 3G network management north bound interface in order to minimize the impact from specifications changes and technological innovations. The core thought of the method is to derive interface technology independent test model (PIT) from platform independent model (PIM) defined in analysis specifications and then according to mapping rules defined in design specifications transform PIT to interface technology specific test models (PSTs) which can be used in conformance testing of SUTs by proper test tools. In this method, the conformance testing can be performed with lower cost and in higher efficiency with reuses of the testing models.**

Keywords-MDA; MDT; conformance testing; network management north bound interface; 3G

I. Introduction

The 3G network management north bound interface (Itf-N) defined by 3GPP is the interface between Network Management System (NMS) and Element Management System (EMS) [1]. The standardization of Itf-N is the basis of information exchange and collaboration between NMS and EMS. To meet the demand of centralized management and maintenance, 3GPP applied the methodology of MISM [2] proposed by ITU-T to standardize Itf-N. To ensure that the 3G network management interface specifications are followed up and to guarantee the interoperability of network management interface between NMS and EMS, interface conformance test is an important means [9].

For network management interface conformance testing, the test system acts as a simulative manager, sends management commands to the system under test (SUT) in agent role, receives responses from SUT, and checks whether the information in responses is consistent with the expected one [3]. For this test system is designed and coded according to specific network management specification or interface technology, changes of specifications or introductions of new interface technologies will result in the failures in it and needs of recoding. With continuous changes in 3G network management requirements and diverse development in network management interface technologies, specifications of 3G network management interface have been in frequent evolutions, which bring great challenges to perform the conformance testing.

To minimize the impact of requirement changes and technological innovations, the method of Model-Driven Testing (MDT) is introduced into network management interface testing [5]. Based on the analysis of 3G network management interface specifications, this paper proposes a model-driven conformance testing method for 3G network management Itf-N. This method is to transform test model from interface analysis model defined in 3G network management interface specifications, and use the test model to drive the conformance test.

II. Instruction of Model-Driven Testing

A. MDA

Model Drive Architecture (MDA) is a methodology proposed by OMG [4], giving guide to software development. To take the challenges from requirement changes and technological innovations, it separates service model from system implementation and uses model to direct the course of software development. As shown in Fig.1, MDA prescribes three kinds of models from different viewpoints on a system.

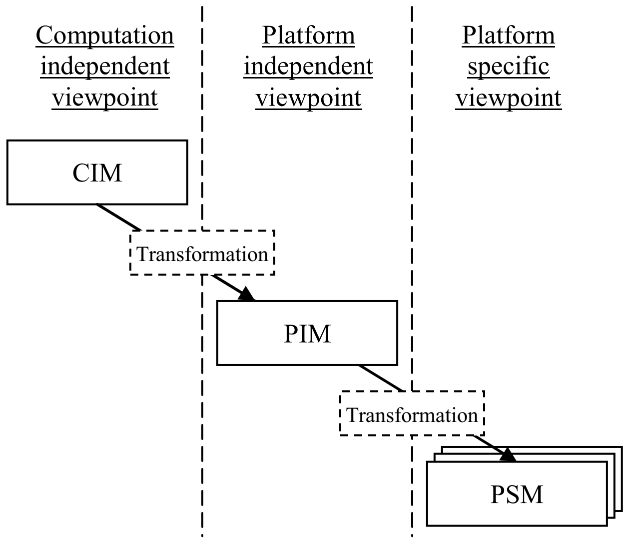

Figure 1. Three viewpoints and models of MDA

Supported by 973 project of China (2007CB310703), NSFC (60802035, 60973108), the Fok Ying Tung Education Foundation (111069), and NCET-06-0092

978-1-4244-6055-7/10/$26.00 ©2010 IEEE

The Computation Independent Model (CIM) describes requirements of the system. The Platform Independent Model (PIM) focuses on describing the pure functions of the system independently from technologies or platforms to realize it. The Platform Specific Model (PSM) describes system functions with a specific platform technique. By the definition of transformation rules and use of proper tools, a PIM can be transformed into several PSMs, which can be transformed into system codes ultimately.

B. Model-Driven Testing Approach

With the enlightenment of the idea of MDA, Model-Driven Testing (MDT) is put forward for software testing [6-8]. As shown in Fig.2, just as the processes of model-driven software development includes the construction of PIM and the transformations from PIM to PSMs to system codes, the processes of model-driven testing includes the construction of Platform Independent Test Model (PIT) and the transformations from PIT to Platform Specific Test Models (PSTs) to test codes. By the construction and reuse of test model, the testing design can be more open and shut, systematic, accurate to the customers, and can be protected effectively from requirements changes respectively.

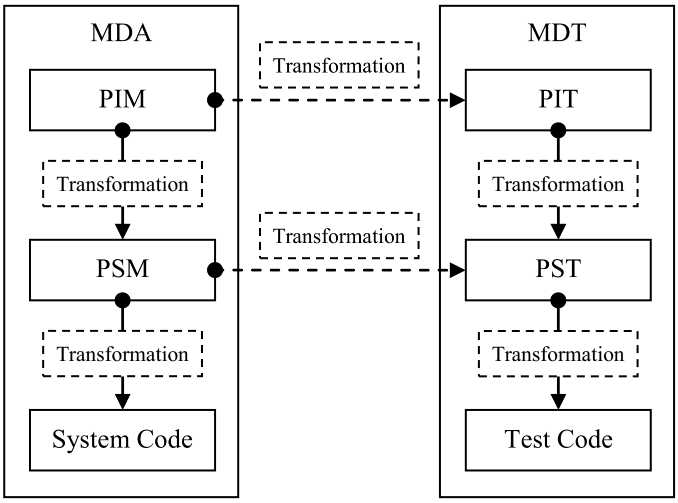

Figure 2. MDA vs. MDT

Furthermore, the test model (PIT or PST) can be transformed from system model (PIM or PSM) directly.

III. ANALYSIS OF 3G NETWORK MANAGEMENT INTERFACE SPECIFICATIONS

To construct the models of 3G network management Itf-N, 3GPP TS32 specifications are derived by using MISM methodology. According to MISM, the process to derive interface specifications is based on three phases, which are user requirements phase, analysis phase and design phase [2].

In user requirements phase business requirements and specification requirements are described in requirement specifications. Since the readability of requirements is critical, 3GPP defines requirement specifications in natural language.

In analysis phase, to identify interacting entities, their properties and the relationships among them, implementation independent specifications, named as Information Services (ISs) by 3GPP, are derived to define the interface analysis model, which includes Information Object Classes (IOCs) and their attributes, operations, notifications. While IOCs are described in UML notation, the following aspects of attributes, operations, and notifications of IOC are described in natural language respectively.

- Attribute: attribute name, definition, read qualifier, write qualifier, and legal values.
- Operation: operation name, definition, qualifier, input parameters, output parameters, pre-condition, post-condition, and exceptions.
- Notification: notification name, definition, qualifier, input parameters, from-state and to-state for triggering notification.

In design phase implementation independent specifications can be mapped into different technology specific specifications, named as Solution Sets (SSs) by 3GPP, dependent on specific management protocol which can be CORBA, SOAP, CMIP or SNMP. So far, two kinds of SSs have been defined by 3GPP, CORBA SS and SOAP SS. Each of them defines the interface design model in specific formalization modeling language e.g., IDL for CORBA, WSDL for SOAP, and mapping rules from analysis model to design one in natural language.

It can be found that there are great similarities between MISM and MDA. The corresponding interface models in user requirements phase, analysis phase and design phase of MISM can be regarded as CIM, PIM and PSM in MDA framework respectively. So the method of MDT can be applied to the conformance testing of 3G network management Itf-N to minimize the impacts of frequent evolution of specifications.

As shown in Fig.3, there are two ways to transform interface models into test models.

- **Transformation in PIM level**: The PIM defined in IS is semi-formalization model, described in combinations of UML notation and natural language. Since the computer system cannot auto-recognize the non-formalization languages correctly such as natural language, it is more difficulty to derive test model from interface model automatically by transformation in PIM level.
- **Transformation in PSM level**: PSMs defined in SSs are formalization models in specific management protocol paradigm, such as CORBA/IDL or SOAP/WSDL. By transformation tools, PSMs can be auto-transformed into test models. To be mentioned, since behaviors of IOCs, describing relationships and interaction activities of IOCs, are defined in PSMs in notes or textual description, behavioral test model should be manually added in the test models, auto-transformed from PSMs [5].

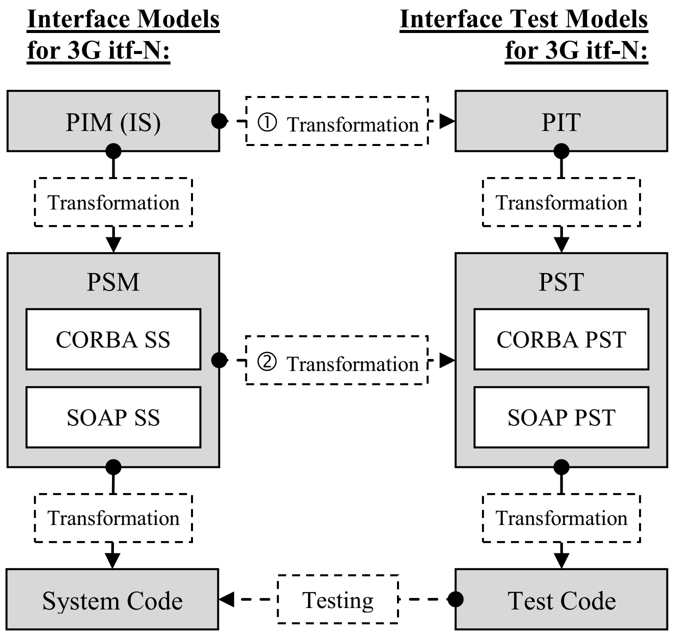

Figure 3. Two ways of transformation from interface models to test models

Transformation in PSM level can improve the degree of auto-transformation, but take sacrifice the portability of test model. Since the specifications of 3GPP have defined two PSMs based on CORBA and SOAP, both CORBA PIT and SOAP PIT should be transformed in PSM level with corresponding tools. It is difficult to assure the same testing criteria among the different transformation tools. Furthermore, once a new technology or management protocol is accepted by 3GPP, a new transformation tool is needed to be found or developed. But for the transformation in PIM level, the above problems can be avoided. Based on the analysis in this section, this paper put forward a model-driven conformance testing method on the transformation in PIM level, which is to derive PIT from PIM manually and transform PIT to PSTs by the utilization of mapping rules from PIM to PSMs.

IV. Model-Driven Conformance Testing Method for 3G Network Management Itf-N

The core thought of the model-driven conformance testing method for 3G network management Itf-N is to derive PIT from PIM defined in 3GPP IS, and then according to mapping rules defined in 3GPP SSs transform PIT to PSTs which can be used in the conformance testing of SUTs. The key steps will be introduced as follows:

A. Deriving PIT from PIM

The process of deriving PIT from PIM is essentially generating test cases, independent of management protocol, on analysis of PIM according to the assigned testing criteria [10]. The formal definition of a test case is as follows:

Definition *TestCase=<id, caseType, name, qualifier, inputParamSeq, outputParamSeq, preCondition, postCondition, result>*

- *id* – Unique identifier of the test case.
- *caseType*=(*op*|*notif*|*get*|*set*) – Type of the test case. *op, notif, get* and *set* denote test case of operation, notification, attribute-reading and attribute-writing respectively.
- *name* – Name of attribute, operation or notification to be tested by this case.
- *qualifier*=(*m*|*o*) – Qualifier of the test case. *m* is a mandatory test case and *o* is a optional one；
- *inputParamSeq*={(*name*,*value*)} – A sequence of input parameters. Each input parameter is expressed in *name/value* pair. The *value* may be a constant or macro defined before. *inputParamSeq* is invalid for test cases of attribute reading and writing.
- *outputParamSeq*={(*name*,*expectValue*)} – A sequence of output parameters. Each output parameter is expressed in *name/expectValue* pair. The *expectValue* may be a constant or value range. *outputParamSeq* is invalid for test cases of notification, attribute-reading and attribute-writing.
- *preCondition*={(*object,attribute*,*expectValue*)|(*id*,*result*)} – A set of pre-conditions of performing the test case. Each pre-condition may be *object/attribute*/*expectValue* triple or *id/result* pair. The *object/attribute*/*expectValue* triple is the expression of an attribute of some object and its expected value, and the *id/result* pair is another test case and its result.
- *postCondition*={(*object,attribute*,*expectValue*)} – A set of post-conditions expected after the performance of the test case. Each post-condition is an attribute of some object and its expected value expressed in *object/attribute*/*expectValue* triple.
- *result*=(*p*|*f*) – The result of this test case. *p* and *f* denote pass and failure respectively.

There are the three following steps of deriving PIT from PIM:

STEP 1 – Construct the transformation model from PIM to PIT shown in Fig.4, including definitions of *Attribute*, *Operation*, *Exception*, *Notification* classes to describe attributes, operations, operation exceptions and notifications of IOCs in formal representation, and the definition of *PITestCase* class to describe the test case for PIT.

STEP 2 – Create instances of the corresponding classes defined in step 1, according to definitions of attributes, operations, operation exceptions and notifications of IOCs defined in PIM.

STEP 3 – According to the assigned testing criteria, design the function *generateTestCase()*of *Attribute*, *Operation*, and *Notification* classes respectively to generate instances of *PITestCase* class.

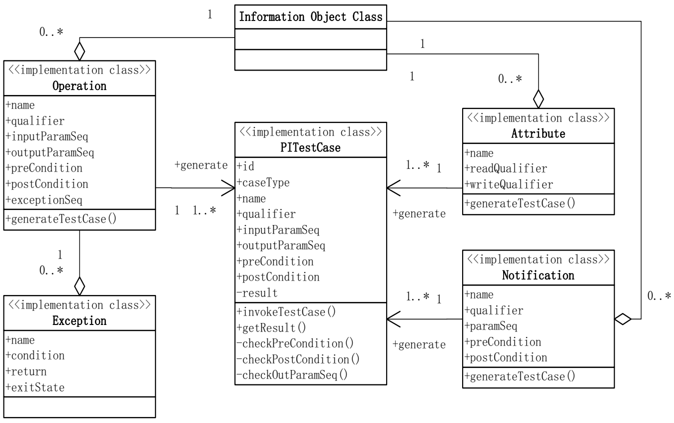

Figure 4. The transformation model from PIM to PIT

The generated instances of *PITestCase* class are the test cases independent of management protocol. Only those mapped into the test cases dependent on specific management protocol can be used in conformance testing.

B. Transformation from PIT to PSTs

This transformation is that from *PITestCase* independent of management protocol to *PSTestCase* dependent on specific management protocol according to mapping rules from PIM to PSMs defined in 3GPP SSs. e.g. transform *PITestCase* to *CORBATestCase* according to mapping rules defined in CORBA SS or transform *PITestCase* to *SOAPTestCase* according to mapping rules defined in SOAP SS.

The mapping rules from PIM to PSMs defined in 3GPP SSs can be expressed in a set of two-tuples as *MappingRuleSeq*={*MappingRule*}={<*ISElement*, *SSElement*>}. And where *ISElement* can be attribute, operation, notification, operation parameter or notification parameter defined in IS, *SSElement* can be method or method parameter defined in SS.

There are the three following steps of the transformation from PIT to PSTs:

STEP 1 – Construct the transformation model from PIT to PST shown in Fig.5 and define PIT2PSTTranslator class to map from instances of *PITestCase* class to those of *PSTestCase* class.

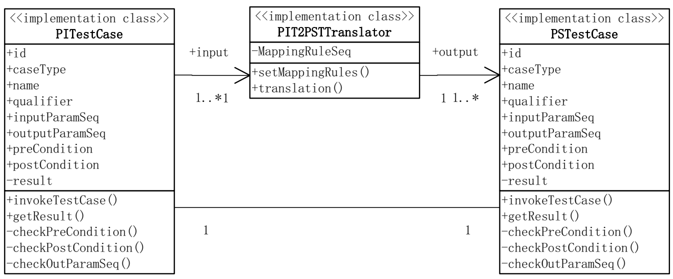

Figure 5. The transformation model from PIT to PST

STEP 2 – Create an instance of *PIT2PSTTranslator* class and add the mapping rules defined in 3GPP SS to its attribute *MappingRuleSeq* by using its function *setMappingRules()*.

STEP 3 – Create a corresponding instance of *PSTestCase* for each instance of *PITestCase* and copy attributes of the latter to those of the former. Then substitute *SSElement* for *ISElement* in attributes *name*, *inputParamSeq*, *outputParamSeq*, *precondition* and *postCondition* of instances of *PSTestCase* .

The instances of *PSTestCase* are test cases dependent on specific management protocol such as CORBA and SOAP. On condition that pre-conditions of a test case are satisfied, with proper test tools management command messages can be derived from the test case and sent to SUT for conformance testing.

V. CONCLUSION

A conformance testing method by reference to the idea of MDT is proposed in this paper to take the challenge from frequent evolution of 3G network management interface specifications and the diversification of implementation technologies. This method is to derive PIT from PIM defined in 3GPP IS, and then according to mapping rules defined in 3GPP SSs transform PIT to PSTs which can be used in the conformance testing of SUTs. With reuses of the testing models, the conformance testing can be performed with lower cost and in higher efficiency.

REFERENCES

[1] 3GPP TS 32.101. "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Telecommunication management; Principles and high level requirements". 2007

[2] ITU-T Rec. M.3020. "Management interface specification methodology". 2008

[3] ITU-T Rec.X.290. "OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications - General concepts".1995

[4] OMG. Model Driven Architecture. http://www.omg.org/mda

[5] Chen Ying-hui, Hou Lin-min, Liu Yi-chang, Zhang Yong-ping, Qi Feng. "Study and Implementation of Model-Driven Testing Method for Network Management Interface". International Conference on Communication Technology and Application 2009 (ICCTA 2009).

[6] Wang Lin-zhang, Li Xuan-dong, Zheng Guo-liang. "Research on Model-Driven Software Testing". Computer Science. 2005

[7] Zhen Ru Dai. "Model-Driven Testing with UML2.0". http://www.cs.kent.ac.uk/projects/kmf/mdaworkshop/submissions/Dai.pdf

[8] Marc Born, Ina Schieferdecker, Hans-Gerhard Gross, Pedro Santos. "Model-Driven Development and Testing – A Case Study". European Workshop on Model Driven Architecture with Emphasis on Industrial Application (MDA) <1, 2004, Enschede, Netherlands>

[9] Chen Ying-hui, Meng Luo-ming. "The study of network management interface standardization techniques". Beijing: Beijing University of Posts and Telecommunications, 2005.

[10] YE Junmin, ZHAO Liang, LUO Jing, WANG Zhenyu,ZHAO Heng. "Research on Testing Criteria of Software Architecture". Computer Engineering. 2005.