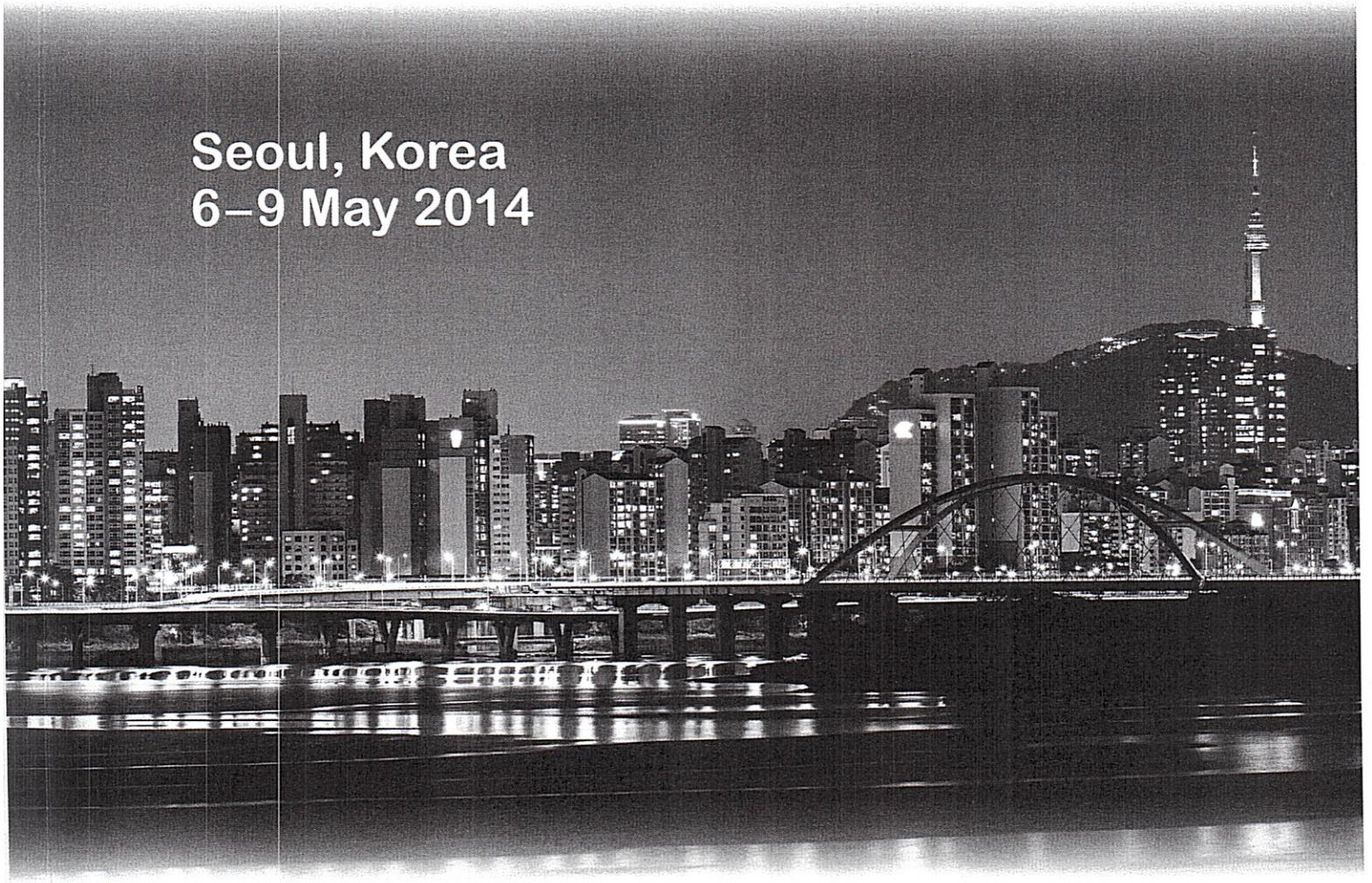
2014 Fifth International Conference on Information Science and Applications

ICISA 2014

Technically Co-Sponsored by IEEE Computer Society



Technically Co-Sponsored by

IEEE

IEEE computer society

Test case Generation from Cause-Effect Graph based on Model Transformation

Hyun Seung Son
Dept. of Computer & Information Communication, Hongik University, Sejong Campus, 339-701
Korea
son@selab.hongik.ac.kr

R. Young Chul Kim
Dept. of Computer & Information Communication, Hongik University, Sejong Campus, 339-701
Korea
bob@selab.hongik.ac.kr

Young B. PARK
Dept. of Computer Science, Dankook University, 330-714
Korea
ybpark@dankook.ac.kr

Abstract— **In software testing, cause-effect graph assures coverage criteria of 100% functional requirements with minimum test case. The existing test case generation from cause-effect graph implements the algorithmic approach. It has disadvantages to modify the entire program if the input model is different. In contrast, model transformation approach can flexibly implement with even a different input models. In the future, we need to study the method of automatic generation of test cases from UML Diagram. It is possible to generate the test case when mapping between cause-effect graph and UML diagram. In this paper, as a first research step, we propose the method to generate test cases from cause-effect graph based on model transformation. To implement the proposed method, we write the rules of model transformation with ATLAS Transformation Language (ATL), and execute the rules in development environment of Eclipse. The implemented tool of the proposed method can be easily extended by rewriting with the mapping rule between cause-effect graph and UML diagram. We just define the relationship between each models to generate the test case.**

Keywords— Model Transformation; Cause-Effect Graph; Test case Generation; Testing; Metamodel

I. INTRODUCTION

Automatic testing is appearing as an important factor because it is able to fast design and execute test case than manual testing. In a part of test automation, the automatic generation of test case can simplify the test during the repetitive development of the complex model, and can reuse test case and execute test even who is not an expert. Therefore, test automation and automatic generation of test cases has a characteristic to save time and money of developers [1]. Because of these features, automatic generation of test cases in software development was carried out many studies [2-5].

To automatically generate the test cases, the existing methods develop test tool using program language. But, the tools written by a programming language is difficult to change it if the input model once determined. When it needs to change the model to use testing tools, developers rewrite the program code to need modification, and recompile it. Because of this point, the traditional programming methods have limitations when the number of models consists of a variety relationship and the frequent changes.

The model transformation [6-7] is able to flexibly implement when connecting to a different input model than the existing method of program because the relationship between the input model and the output of the model is a loose and open to modification. But, to apply the techniques of model transformation, we must have the element such as metamodel, model transformation language (rule), and model transformation engine essentially.

In software testing, cause-effect graph that maps a set of causes to a set of effects assures coverage criteria of functional requirements of 100% with minimum test case. Our final goal is test case generation from UML diagram. But, the existing tools for cause-effect are difficult to map UML diagram. As a first research step, this paper proposes the method based on model transformation for test case generation of cause-effect graph. To implement the proposed method, we write the rules of model transformation using ATLAS Transformation Language (ATL) and execute the rules in development environment of Eclipse.

The proposed method consists of three parts of cause-effect graph, decision table, and test case. Also, the method uses two rules of model transformation. First step translated by model transformation engine to decision table from cause-effect, Second step translated by the engine to test case from decision table. And, we demonstrate the process of test case generation of this method through a case study that is simple calculate program for international post fee.

Our proposed method based on model transformation can be easily extended by when write the mapping rule between cause-effect graph and UML diagram because it uses the model

978-1-4799-4441-5/14/$31.00 ©2014 IEEE

of each step and defines the relationship between the models to generate the test case.

The paper is organized as follows. Chapter 2 presents the model transformation language such as ATL. Chapter 3 addresses the proposed method for test case generation based on model transformation. Chapter 4 describes a case study about automatic test case generation. Chapter 5 gives conclusion and future works.

II. RELATED WORK

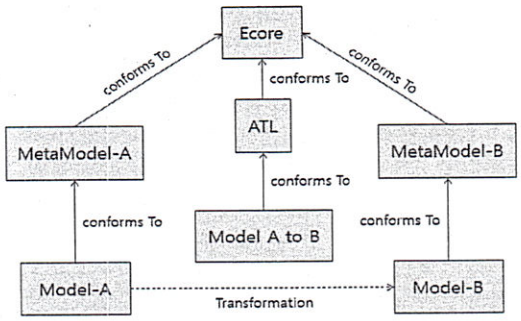


Fig. 1. The overview of ATLAS Transformation Language (ATL)

ATLAS Transformation Language (ATL) developed by ATLAS INRIA & LINA is tools for model transformation between models [8-9]. The ATL currently managed by the Eclipse Foundation, which is open source [10]. ATL consist of rules. This rule defines how to get the model elements and convert the target model. Figure 1 that is step of model transformation through ATL show to convert the model to Model-B from Model-A. ATL is based on the Ecore of Eclipse. Ecore has been trying to convert a model and a model for the conversion information includes a metamodel, ATL model according to the rules defined in the translation.

III. THE METHOD FOR AUTOMATIC TEST CASE GENERATION BASED ON MODEL TRANSFORMATION

The proposed method for test case generation based on model transformation consists of two steps showing figure 2. First step (MT1) convert to decision table from cause-effect graph, and second step (MT2) convert to test case from decision table. In this process of the proposed method, the data of input/output models represent XML Metadata Interchange (XMI) [11].

Test case generation of our suggested model transformation consists of two steps such that 1) transforms from cause-effect graph to decision table, 2) also from decision table to test case. Input and output data of model transformation are executed with all XML metadata Interchange [11].

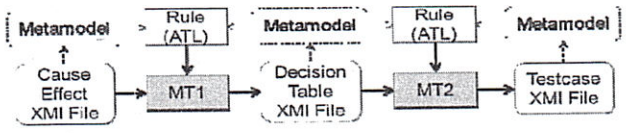


Fig. 2. The steps of Model transformation

At rule definition step of Model transformation, it defines all transformable elements of each model and relationship, and represents transformation rules with model transformation language (ATL). ATL provides engine, language and test editor for executing model transformation.

IV. CASE STUDY

We show to generate test cases from requirements of international post mail fee calculating program at figure 3. This requirements represent cause-effect graph. Through easily ATL plug-in[10], it executes with our defined metamodel and model transformation on Eclipse, and then generates test cases.

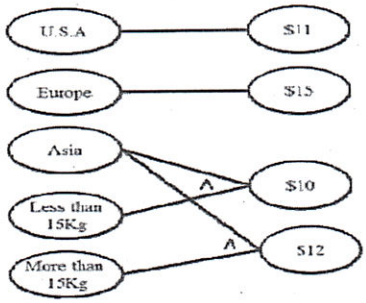
The calculate program for international post fee:

If you enter the country and the weight tells you the price.

Country	Weight	Price
U.S.A	all weight	$11 US
Asia	weight < 15kg	$10 US
Asia	weight ≧ 15kg	$12 US
Europe	all weight	$15 US

Fig. 3. The requirements specification of calculate program for international post fee

To execute model transformation, we represent XMI from modeling cause-effect graph based on requirements. Figure 4 shows cause-effect graph and XMI based on requirements.



(a) Cause-effect graph

```
1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <cedm:CauseEffectModel xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
3    xmlns:cedm="http://cedm/1.0">
4  <cause name="U.S.A" ownedConnector="//@connector.0"/>
5  <cause name="Europe" ownedConnector="//@connector.1"/>
6  <cause name="Asia" ownedConnector="//@connector.2 //@connector.4"/>
7  <cause name="Less than 15Kg" ownedConnector="//@connector.3"/>
8  <cause name="More than 15Kg" ownedConnector="//@connector.5"/>
9  <connector start="//@cause.0" end="//@effect.0" name="U.S.A->$11"/>
10 <connector start="//@cause.1" end="//@effect.1" name="Europe->$15"/>
11 <connector start="//@cause.2" end="//@effect.2" name="Asia->$10"/>
12 <connector start="//@cause.3" end="//@effect.2" name="Less than 15kg->$10"/>
13 <connector start="//@cause.2" end="//@effect.3" name="Asia->$12"/>
14 <connector start="//@cause.4" end="//@effect.3" name="More than 15kg->$12"/>
15 <effect name="$11" ownedConnector="//@connector.0" />
16 <effect name="$15" ownedConnector="//@connector.1" />
17 <effect name="$10" ownedConnector="//@connector.2 //@connector.3" eType="AND"/>
18 <effect name="$12" ownedConnector="//@connector.4 //@connector.5" eType="AND"/>
19 </cedm:CauseEffectModel>
```

(b) Cause-effect's XMI file

Fig. 4. The input data of cause-effect for model transformation

978-1-4799-4441-5/14/$31.00 ©2014 IEEE

We input XML file of the modeled cause-effect into model transformation tool, then generate decision table like Figure 5. After executing Model transformation, it generates XMI file of figure 5(b). Figure 5(a) represents decision table to easy understand it.

	Condition											
	1	2	3	4	5	6	7	8	9	10	11	12
U.S.A	F	T										
Europe			F	T								
Asia					F	F	T	T	F	F	T	T
Less than 15Kg					F	T	F	T				
More than 15Kg									F	T	F	T
Effect	1	2										
S11	F	T										
S15			F	T								
S10					F	F	F	T				
S12									F	F	F	T

(a) Decision table

```
1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <decision:DecisionTableModel xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
3     xmlns:decision="http://decision/1.0">
4   <testcaseSet id="0">
5     <cause>
6       <input name="U.S.A">
7         <in item_id="F"/>
8         <in item_id="T"/>
9       </input>
10    </cause>
11    <effect>
12      <output name="$11" eType="NORMAL">
13        <out item_id="F"/>
14        <out item_id="T"/>
15      </output>
16    </effect>
17  </testcaseSet>
18  <testcaseSet id="1">
19    <cause>
20      <input name="Europe">
21        <in item_id="F"/>
22        <in item_id="T"/>
23      </input>
24    </cause>
25    <effect>
26      <output name="$15" eType="NORMAL">
```

...

(b) Decision table's XMI file

Fig. 5. The execution result of model transformation to decision table from cause-effect graph

No	Pre-Condition	Test Condition	Expectation Result
TC1	U.S.A=F	NORMAL	$11=F
TC2	U.S.A=T	NORMAL	$11=T
TC3	Europe=F	NORMAL	$15=F
TC4	Europe=T	NORMAL	$15=T
TC5	Asia=F, Less than 15Kg=F	AND	$10=F
TC6	Asia=F, Less than 15Kg=T	AND	$10=F
TC7	Asia=T, Less than 15Kg=F	AND	$10=F
TC8	Asia=T, Less than 15Kg=T	AND	$10=T
TC9	Asia=F, More than 15Kg=F	AND	$12=F
TC10	Asia=F, More than 15Kg=T	AND	$12=F
TC11	Asia=T, More than 15Kg=F	AND	$12=F
TC12	Asia=T, More than 15Kg=T	AND	$12=T

(a) Test case

```
1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <testcase:TestCaseModel xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
3     xmlns:testcase="http://testcase/1.0">
4   <testcaseline no="TC1">
5     <testcase_pre testpre="U.S.A=F"/>
6     <testcase_con testcon="NORMAL"/>
7     <testcase_res testresult="$11=F"/>
8   </testcaseline>
9   <testcaseline no="TC2">
10    <testcase_pre testpre="U.S.A=T"/>
11    <testcase_con testcon="NORMAL"/>
12    <testcase_res testresult="$11=T"/>
13  </testcaseline>
14  <testcaseline no="TC3">
15    <testcase_pre testpre="Europe=F"/>
16    <testcase_con testcon="NORMAL"/>
17    <testcase_res testresult="$15=F"/>
18  </testcaseline>
19  <testcaseline no="TC4">
20    <testcase_pre testpre="Europe=T"/>
21    <testcase_con testcon="NORMAL"/>
22    <testcase_res testresult="$15=T"/>
23  </testcaseline>
24  <testcaseline no="TC5">
25    <testcase_pre testpre="Asia=F,Less than 15Kg=F"/>
26    <testcase_con testcon="AND"/>
27    <testcase_res testresult="$10=F"/>
```

...

(b) Test case's XMI file

Fig. 6. The execution result of model transformation to test case from decision table

V. CONCLUSION

In software testing, cause-effect graph assures coverage criteria of 100% functional requirements with minimum test case. In this paper, as a first research step, we propose the method to generate test cases from cause-effect graph based on model transformation. To implement the proposed method, we write the rules of model transformation with ATLAS Transformation Language (ATL), and execute the rules in development environment of Eclipse. The implemented tool of the proposed method can be easily extended by rewriting with the mapping rule between cause-effect graph and UML diagram. We just define the relationship between each models to generate the test case.

ACKNOWLEDGMENT

This work was supported by the IT R&D Program of MKE/KEIT [10035708, "The Development of CPS (Cyber-Physical Systems) Core Technologies for High Confidential Autonomic Control Software"]

REFERENCES

[1] D.M. Cohen, S.R. Dalal, J. Parelius, G.C. Patton. The combinatorial design approach to automatic test generation. Software, IEEE, Vol. 13, Issue 5, pp. 83-88, 1996.

[2] Woo Yeol Kim, Hyun Seung Son, Jae Seung Kim, R. Young Chul Kim. A Study on Improvement of Test Process for Heterogeneous Embedded Software. Proceedings of 2010 Korea Conference on Software Engineering, Vol. 12, No. 1, pp. 513-515, 2010.

[3] Hyun Seung Son, Woo Yeol Kim, Jae Seung Kim, R. Young Chul Kim. A Study on Test Process of Multiple-jointed robot in virtual environment. Proceedings of 2011 Korea Conference on Software Engineering, Vol. 13, No. 1, pp. 532-535, 2011.

[4] Su Jeong Woo, Hyun Seung Son, Woo Yeol Kim, Jae Seung Kim, R. Young Chul Kim. A Study on test case generation based on M&S for Pre-Testing. Proceedings of 2012 Korea Conference on Software Engineering, Vol. 14, No. 1, pp. 181-183, 2012.

[5] [5] Hyun Seung Son, Woo Yeol Kim, Jae Seung Kim, R. Young Chul Kim. Automatic Test Case Generation using Multiple Condition Control Flow Graph. Information Technology and Computer Science 2012, ASTL 13, pp. 105-109, 2012.

ICITCS2014

http://icitcs2014.org/

ICITCS 5th will be held in Beijing, China from October 27th-29th, 2014. This will also include joint conferences from ICISST, ICMWT and ICIEEM. Beijing is a city of great majestic history and has endless activities for anyone to enjoy. We are excited to invite you and your fellow scholars to come join us for a new experience.

ICISA2015

http://icisa2015.org/

Come join us for the 6th ICISA from January 5th-7th, 2015. ICISA2015 will be held in Las Vegas, USA the City of Lights. Las Vegas has everything from amazing views, performances, food and everything for the family. ICISA will have scholars from all over the world and will be the perfect time to come and participate in the conference.

ICISA 2014

Technically Co-Sponsored by IEEE Computer Society

Technically Co-Sponsored by

IEEE

IEEE computer society