A Source Code and Test Cases Impact Analysis Tool for Database Schema Changes

Chanwit Sriarpanon and Taratip Suwannasart

***Abstract*—Changes are inevitable and can happen throughout any phase of software development cycle. Changes can occur to database schema. Source code which is used database schema must be affected when the database schema is changed. Test cases which are generated from the source code must be changed as well. This research focuses on source code and test cases impact analysis on database schema changes. We have proposed an approach to analyze impact in our previous work [6]. This paper we propose a tool that is developed based on our approach. The tool can compare between the original schema and a log file to analyze changes, then lines of source code that are affected are hilighted, affected test cases are also notified to the users.**

***Index Terms*—database schema, structural testing, software testing, test case**

I. Introduction

DATABASE which is very important to companies, because the data is important for doing their business. Database schema may be changed over the time due to business changes which may cause problems on applications that operate on the database. The changes unavoidably affect to source code that connects to the database and the existing test cases cannot be used. Software developers have to locate the source code where the changes have impacts. In additional test cases that were generated from the source code must have the impacts as well.

The previous studies about impact analysis of database schema changes [1-4] focus only on source code analysis, but they have not mentioned how changes affected test cases, and they do not guide how to generate new test cases when database schema is changed [5]. Therefore, we have considered this problem, and our previous research [6] suggested an approach of impact analysis to source code and test cases for database schema changes.

In this research, we present a tool for analyzing impact to source code and test cases for database schema changes. The tool can compare the original schema with a log file, analyze impact to source code, display the affected positions and line numbers, as well as fix and update the test cases to be ready for use.

We organize the rest of this paper as follows. Section 2 describes a background of this research. Section 3 presents development of our purposed tool. Finally, conclusion and future work are illustrated in section 4.

Manuscript received January 10, 2015; revised January 20, 2015.

C. Sriarpanon and T. Suwannasart are with the Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand e-mail: Chanwit.S@student.chula.ac.th, Taratip.S@chula.ac.th

II. Background

A. Source Code with Embedded SQL

In this research, source code is written by using embedded SQL in java. The symbol starts with "#" shows embedded SQL that connects to database. An example of the source code with embedded SQL in java used in this research is shown in figure 1.

```
private static void memberLogIn()
    throws SQLException, IOException {
  String mmid1,pass1;
  String mmid2 = readEntry("Account#: ");
  String pass2 = readEntry("Password: ");

  try {
    #sql {select mid, password
          into   :mmid1, :pass1
          from   member
          where  mid = :mmid2 and password = :pass2};
```

Fig. 1. Example of a source code with embedded SQL

B. Test Cases

In this research, test cases are generated using structural testing techniques. A test case used in this research is depicted in figure 2.

Test case ID	TC1		
Path	34-35-36-37-38-39-40-41-42-48		
Input			
Name	Type	Size	Value
mmid2	STRING	10	Chanwit
pass2	STRING	10	aaaaaaaaa
Expected output	Member login success		

Fig. 2. Example of a test case

C. Impact Analysis

There were studies [1-4] about impact analysis using several methods such as program slicing, traceability relationships and configuration management to identify factors that make changes, but the researchers do not consider impact analysis on source code with embedded SQL and test cases based on structural view when database schema is changed. In this paper we focus on these main issues as follows.

1. How do changes affect to source code?
2. How do we locate the affected positions on the source code?
3. How do changes affect test cases?

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

4. Can we fix and update the test cases to be ready for used?

III. Development of the Proposed Tool

This research presents a tool for impact analysis to source code and test cases for database schema changes. The structure of our proposed tool is depicted in figure 3.

The structure of this tool is divided into four parts including analysis and comparison of original database schema with a log file to find out how different it is, source code impact analysis, test cases impact analysis, as well as fixing and updating the affected test cases. The tool will then display positions of the source code that are affected and analyze which test cases are needed to be fixed and updated. Our approach are described in the steps as follows.

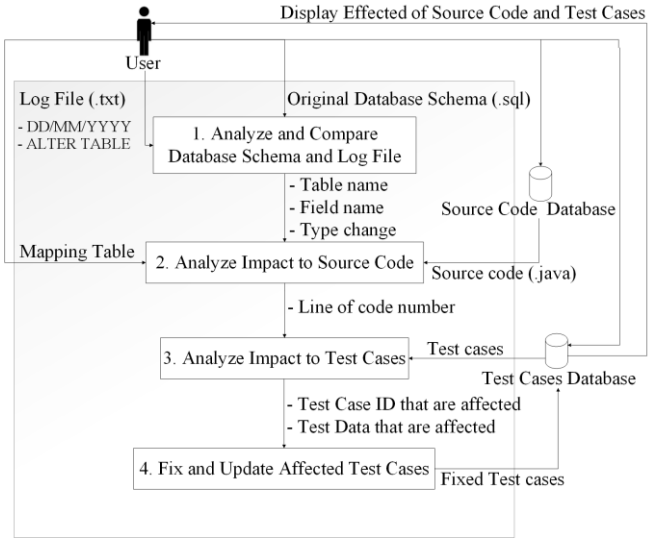

Fig. 3. Structure of the tool

A. Analyze and Compare Database Schema and Log File

A user can import an original database schema file to compare with a log file. Figure 4 shows SQL statement for creating a table in a database. This statement can be used to create a table while figure 5 is commands that are used to change the structure of the table.

```
CREATE TABLE IF NOT EXISTS `member` (
`mid` varchar(10) NOT NULL,
`password` varchar(10) NOT NULL,
`firstname` varchar(50) NOT NULL,
`lastname` varchar(50) NOT NULL,
`address` varchar(100) NOT NULL,
`email` varchar(100) NOT NULL,
`cash_balance` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Fig. 4. SQL statement for creating a table

```
ALTER TABLE `member` CHANGE `password` `password` VARCHAR(5) NOT NULL
ALTER TABLE `member` CHANGE `firstname` `firstname` CHAR(50) NOT NULL
ALTER TABLE `member` CHANGE `cash_balance` `cash_bal` INT(11) NOT NULL
ALTER TABLE `member` ADD `description` VARCHAR(50) NOT NULL
ALTER TABLE `member` DROP `lastname`
```

Fig. 5. Example of a log file for altering the table structure

The example in this research we assume that we use a change in figure 5. The table ‘member’ is altered with dropping ‘lastname’.

The user interface of this step is shown in figure 6. The tool can analyze changes that can be happened to database schema. In our approach, allow 5 types of changes as follows.

1. Add new field
2. Drop field
3. Change name
4. Change type
5. Change size

The tool analyze and compare between the original schema (figure 4) and the log file (figure 5) to find changes. From this example the change is drop field (lastname).

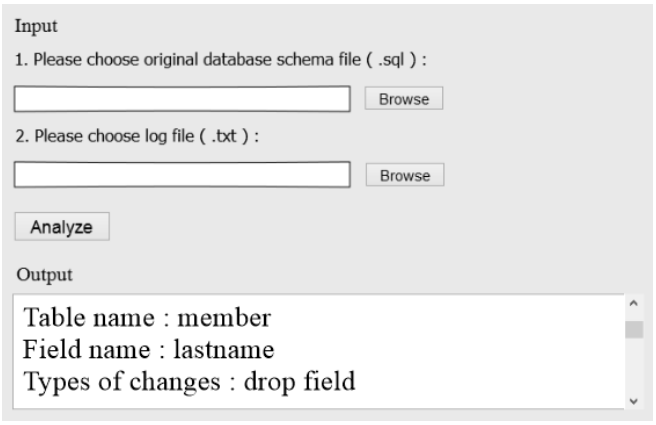

Fig. 6. User interface for analyzing and comparing the original schema and the log file

The output of this step is shown in figure 6, the information includes table name and field name that are changed, as well as types of changes.

B. Analyze Impact to Source Code

```
private static void memberLogIn()
throws SQLException, IOException {
String mmid1,pass1,fname1,lname1;
String lname2 = readEntry("Account#: ");
String pass2  = readEntry("Password: ");
try {
#sql {select mid, password, firstname, lastname
into :mmid1, :pass1, :fname1, :lname1
from member
where lastname = :lname2 and password = :pass2};
}
```

Fig. 7. Analyze table name and variables related to the changes

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

This step uses the output from the previous step which includes table name, field name, and types of changes.

The tool can analyze impact to all source code files in the system. First, the tool finds the table name (member) in the source code related to changes. Next, the field name (lastname) is analyzed, because the field name is removed from the table, The variables (lname1 and lname2) related to the field name (lastname) will be affected, because this query selects values and assigns to those variables. The tool hilights variables and positions in the source code that are affected by changes.

In our approach we assume that users have a table that defines line numbers of the source code as shown in figure 8, then we know the line numbers that contain parameters/variables names that are affected by database schema changes. From the example in figure 8, line 36, 37 and 41 includes the affected variables : lname1 and lname2. Figure 9 illustrates the user interface of the impact analysis of schema changes to source code.

Line of code number	Statement
34	private static void memberLogIn() throws SQLException, IOException
35	{
36	String mmid1, pass1, fname1, lname1;
37	String lname2 = readEntry("Account#: ");
38	String pass2 = readEntry("Password: ");
39	Try
40	{
41	#sql {select mid, password, firstname, lastname into :mmid1, :pass1, :fname1, lname1 from member where lastname = :lname2 and password = :pass2};
42	}

Fig. 8. An example of mapping table

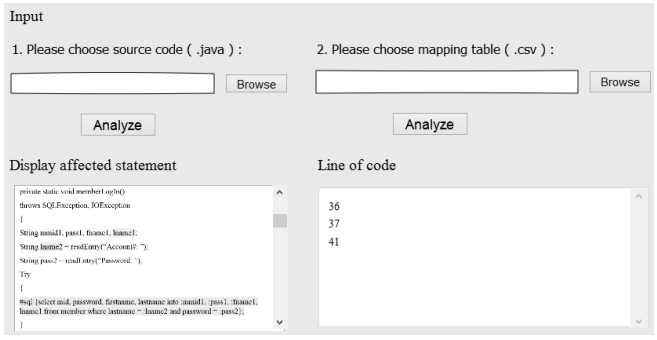

Fig. 9. User interface for analyzing impact to source code

C. Analyze Impact to Test Cases

A test case example used in this step is shown in figure 10. In this research, test cases are created from structural testing technique and have a path that contains line numbers of the source code. Users can import test cases to analyze and find out which test cases are affected. The output from previous step is line numbers of the source code (36, 37 and 41). Our approach finds test cases that have line numbers related to the output from the previous step. Those test cases are need to be changed.

Test case ID	TC1		
Path	34-35-36-37-38-39-40-41-42		
Input			
Name	Type	Size	Value
lname2	STRING	10	Sriarpanon
pass2	STRING	10	aaaaaaaaaa
Expected output	Member login success		

Fig. 10. A test case example

In our approach we analyze in some changes we need to generate new test cases such as, adding new fields to the original schema and changing type or size of fields.

In case that fields are dropped, some test cases can be used, but they must be changed, like in the previous example shown in figure 10.

In the other case, some test cases can be used and no need to be changed. For example when changing name (cash_balance to cash_bal) to the original schema this case affects only on the source code, but does not affect on the test cases.

Fig. 11. User interface for analyzing impact to test cases

Figure 11 shows the user interface for analyzing impact to test cases. The user can import the test cases and the tool analyzes, and then displays which test cases are affected on the screen. This step displays the test cases that are affected by the changes.

D. Fix and Update Affected Test Cases

This step fixs and updates only the values of test cases that are affected. From the previous step, affected test cases must be fixed and updated. A test case example that is fixed and updated is shown in figure 12. Because the change is dropping field name (lastname), so this test case must be updated by deleting the variable (lname2).

Test case ID	TC1		
Path	34-35-36-37-38-39-40-41-42		
Input			
Name	Type	Size	Value
pass2	STRING	10	aaaaaaaaaa
Expected output	Member login success		

Fig. 12. A fixed and updated test case example

The user interface for fixing and updating affected test cases is shown in figure 13.

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Fig. 13. User interface for fixing and updating affected test cases

IV. Conclusion

The purpose of this research is to present a tool for analyzing impact to source code and test cases for database schema changes. This tool analyzes changes by comparing between a log file and the database schema. Then source code and test cases which are created by structural testing technique are analyzed. The tool can display lines in source code that are affected by the changes, and affected test cases are updated or needed to be newly generated. The benefit of this tool is the reduction of cost, time and effort to find impacts on the source code and test cases.

References

[1] N.M. Spyridon K. Gardikiotis, "A Two-folded Impact Analysis of Schema Changes on Database Applications," *International Journal of Automation and Computing*, vol.6, pp. 109-123, 2009-04-15 2009.

[2] A. Karahasanovic. "Identify impact of database schema on application," *Industrial Systems Development Group*, Department of Informatics, University of Oslo, 2001.

[3] A. Maule, W. Emmerich, and D. S. Rosenblum, "Impact analysis of database schema changes," in *Software Engineering, 2008. ICSE '08. ACM/IEEE 30th International Conference on*, 2008, pp. 451-460.

[4] K. Adump, "A Tool for Displaying Effect on C# Program from Database Schema Changes," Master's Project, Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, 2014.

[5] P. Tongrak and T. Suwannasart, "A Tool for generating test case from relational database constraints testing," in *Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on, 2009,* p. 435-439.

[6] C. Sriarpanon and T. Suwannasart, "Impact Analysis of Source Code and Test Cases for Database Schema Changes," *International Computer Science and Engineering Conference, 2014. ICSEC 2014.* p. 104-108.

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)