An evaluation of using structural testing information to perform web service monitoring

Marcelo Medeiros Eler and Paulo Cesar Masiero
ICMC – USP
P.O. 668 -- Sao Carlos -- Brazil -- 13560-970
Tel: +55 (16) 3373-9700 / Fax: +55 (16) 3371-2238
e-mail: marceloeler@gmail.com, masiero@icmc.usp.br

Abstract: Service monitoring is useful to detect changes of services. Active monitoring lies on the execution of regression test to detect changes of services' behavior. It is usually based on black-box testing since the source code of monitored services is not available. In such case, modifications that do not affect the behavior tested by the regression test may not be identified. We claim that structural testing information is useful to enhance the change detection mechanism of monitoring approaches and in previous work we have devised an approach to monitor testable services, which are services that provide their clients with structural testing facilities. The approach is called Testable Web Service Monitoring (TWSM) and in this paper we present an experiment designed to investigate the benefits of using TWSM in comparison to a functional approach. Our hypothesis is that structural testing can improve the change detection mechanism of monitoring approaches and increase the monitors' confidence on the results obtained. The results of the experiment showed that TWSM can identify changes of a service with more accuracy than the functional approach. Also, the confidence of the monitors using TWSM is greater than the confidence of the monitors using the functional approach.

Key words: Web services, service monitoring, structural testing, evaluation

1. INTRODUCTION

Service Oriented Architecture (SOA) applications are usually developed by composing autonomous services provided by third parties. Differently from Commercial Off-The-Shelf components, services are not physically deployed at the integrator's environment and they are only under the control of their providers. Service providers can change their services without notification and, in this case, service integrators cannot perform suitable verification and validation activities to detect breaks of SLA (Service Level Agreement) contracts and/or fails to deliver the expected functionality when a third party service is replaced by a new version [1].

Service monitoring is an important activity of a SOA development process that is used to detect changes of services during runtime. Many approaches have been proposed in the literature to solve this issue [1]–[10]. Most of monitoring strategies are based on black- box testing since the source code of services is usually unavailable. The black-box testing is more useful when the behavior of the service changes due to the modifications. However, there can be modifications that do not affect the behavior of the service and yet introduce failures or new pieces of code that are not tested by the regression test set used by the monitoring strategy. White-box testing, on the other hand, can be useful to detect changes of services even when the behavior is the same.

Structural testing is not commonly used in SOA contexts because of the black box nature of services, but in previous work we have developed an approach called BISTWS (Built-in Structural Testing of Web Services) to increase the testability of services by making them more transparent to external users [11]. Such services have been called testable services. They are services which were instrumented to trace information on its own execution (instructions, branches and data exercised). This information is then used to generate a structural coverage analysis regarding criteria such as all-nodes, all-edges and all-uses.

We have developed a generic monitoring strategy called Testable Web Service Monitoring (TWSM) to monitor testable services or services that can provide structural testing information to clients to investigate our assumption that structural testing could be used to improve traditional service monitoring (based on functional testing).

In this paper we briefly review the TWSM approach and present an experiment designed to compare this approach with an approach based only on functional testing. Experimental results have shown that the use of structural testing improves the change detection mechanism and give to the monitor more confidence on the results of the monitoring activities. This paper is organized as follows: Section 2 briefly shows the TWSM approach. Section 3 shows the validation of our approach by means of an example application and an experiment. Section 4 presents the concluding remarks of this paper.

2. TESTABLE WEB SERVICE MONITORING (TWSM)

Testable Web Service Monitoring (TWSM) is a generic approach we devised to use structural testing information on service monitoring. TWSM combines functional and structural testing elements to discover structural modifications, rather than only detecting behavioral changes. TWSM performs two kinds of analysis: static and behavioral. The static analysis is used to detect changes of the structure of the service by analysing the test requirements of the testable service. The behavioral analysis is used to detect changes of the functional behavior by executing a regression test set. Moreover, a structural coverage analysis is performed using the test facilities of the testable services to check whether the coverage of the test criteria has changed.

A monitor using the TWSM approach must execute the following steps. The monitor firstly collects the structural testing requirements of the testable service. It then sets the testable service to a test session mode and then executes the regression testing activity. Next, the monitor gets the results of the test set execution, stops the test session and gets the coverage analysis report. After these steps, the monitor checks for functional and structural changes to decide whether the monitored service has changed.

The data collected (test requirements, test case results and coverage analysis) the first time in which the monitoring activities are performed are used to create a baseline. From this moment on each time the monitoring activities are performed the results are compared with the values of the baseline. The monitor can also update the baseline at any time.

2-1 Static Analysis

The main activity of this phase is to compare the test requirements received from the testable service with the test requirements of the baseline. If any difference is noticed then there is strong evidence that the implementation of the web service has changed. It is easy to define which operation has changed because the test requirements are generally provided by each operation.

The most challenging task in regression testing is to select a suitable test set. In many situations it is recommended to execute only a subset of all test cases. It is a more challenging task when the tester does not know which parts of the code was modified. Using the test requirements analysis, however, the monitor has a clue about which operations of the service has changed.

The test requirements reflect directly the inner structure of a program in many aspects and they change as the structure of the program changes. They are sensitive enough to help detecting minimal changes. Table 1 shows an example of the comparison of test requirements performed during the static analysis. This comparison is related to the test requirements of the two operations provided by the BankWS services. We show here only the requirements of the criteria all-nodes and all-edges because the test requirements of the all-uses criterion are too long. Note that the test requirements of loanMoney have changed, while the requirements of getMonthlyTax have not changed. This is strong evidence that the provider of BankWS has changed only the implementation of the operation loanMoney.

Table 1 – Test requirements of BankWS's operations

BankWS	Baseline	1st execution	2nd execution
loanMoney/all-nodes	36,28,26,50,18,…	36,28,26,50,18,…	**33,97,64,94,134,…**
loanMoney/all-edges	(28,35), (28,33),…	(28,35), (28,33),…	**(59,97), (28,35),…**
getMonthlyTax/all-nodes	26,21,18,15,…	26,21,18,15,…	26,21,18,15,…
getMonthlyTax/all-edges	(26,39), (5,15), …	(26,39), (5,15), …	(26,39), (5,15), …

2-2 Behavioral Analysis

The behavioral analysis consists of verifying if the behavior of the service has changed. The first analysis is related to the functional results of the test set executed during the regression testing activity. The result of each test case is compared with the result obtained from the previous execution. If any test case has a different result from the previous execution and is no longer satisfying the functional requirements, then the behavior of the web service has changed. Table 2 shows an example of the functional results of the regression test set executed against a monitored service. Notice that the results have changed only at the second execution.

The functional result of each test case is not the only evidence to define if the behavior of the web service has changed. The TWSM approach also uses the coverage analysis report that shows how much of the instructions,

branches and data of the service the test cases executed have exercised. The monitor compares the coverage analysis achieved with the baseline. Suppose, for example, that the coverage value of the criterion all-nodes was 85% and has now dropped to 70%. If the test requirements of the service and the results of the test cases have not changed and the coverage analysis have changed, it means that the instructions, branches and data exercised by the test cases are different from those executed by the test cases executed to create the baseline. Such changes are also evidences of changes of the monitored service code. Table 3 shows the coverage analysis of a service called BankWS obtained after two executions of the behavioral analysis activity. Note that the coverage information has changed in the second execution. This can indicate that the service has new pieces of code that should be tested to avoid unexpected failures.

Table 2 – Test results/behavioral analysis

TC-ID	Baseline	1st exec	2nd exec
01	Passed	Passed	Passed
02	Passed	Passed	**Failed**
03	Passed	Passed	**Failed**
…	…	…	…
10	Passed	Passed	Passed

Table 3 – Coverage analysis/behavioral analysis

Operation/criterion	Baseline	1st exec	2nd exec
loanMoney/all-nodes	24/26 (92%)	24/26 (92%)	**33/42 (78%)**
loanMoney/all-edges	10/10 (100%)	10/10 (100%)	**18/22 (81%)**
loanMoney/all-uses	10/10 (100%)	10/10 (100%)	**19/26 (73%)**
getMonthlyTax/all-nodes	7/8 (87%)	7/8 (87%)	7/8 (87%)
getMonthlyTax/all-edges	8/9 (88%)	8/9 (88%)	8/9 (88%)
getMonthlyTax/all-uses	7/8 (87%)	7/8 (87%)	7/8 (87%)

3. VALIDATION OF TWSM

We performed an experiment to evaluate the TWSM approach by comparing it with a monitoring approach that uses only functional information. Two experimental objects and 11 subjects were used. The details of the experiment are presented as follows.

3-1 Experimental setup

The following null and alternative hypotheses were defined. $H0_1$ states that the amount of changes correctly identified by TWSM is less or equal to the amount of changes correctly identified by the functional approach, while $H1_1$ states that the amount of changes correctly identified by TWSM is greater than the amount of changes correctly identified by the functional approach. $H0_2$ states that the number of correct identifications made by TWSM of which operations of the web service have changed is less or equal to the number of correct identifications made by the functional approach, while $H1_2$ states that the number of correct identifications made by TWSM of which operations of the web service have changed is greater than the number of correct identifications made by the functional approach. $H0_3$ states that the confidence of the monitors using the TWSM approach is less or equal to the confidence of the monitors using the functional approach, while $H1_3$ states that the confidence of the monitors using the TWSM approach is greater than the confidence of the monitors using the functional approach.

The independent variables of the experiment are the following: the TWSM and the functional approach; the experimental objects BankWS and IRWS; and the experience of the subjects. The dependent variables are the following: number of changes identified; number of correct identifications of which operations have changed; and the confidence of the monitor on the results obtained by the approach used.

3-2 Experimental subjects and objects

The experiment was performed from the perspective of the web services monitors. The subjects of the experiment were eleven grad students of the Software Engineering Lab of the ICMC-USP. Two of them played the role of service providers, who make changes in the services. The nine others played the role of monitors, who try to identify any change on the monitored services. Five monitors used the TWSM approach and four used the functional approach. The choice of which approach should be used by each subject was randomized.

The experimental objects of the experiment are two web services: BankWS and IRWS. BankWS has three public operations that provide bank loan functionalities. IRWS also has three public operations that provide functionalities for the Brazilian annual income tax declaration. These two services were developed by the author of the TWSM approach for other case studies and they were reused in this experiment.

3-3 Threats to validity

The threats to the internal validity of this experiment are the experience of the subjects and the productivity under evaluation. The first factor does not impact the experiment because the subjects have almost the same experience. The second factor cannot be considered because, despite being conducted in the context of a grad course, the experiment was not a component of the student's grade.

One of the threats to the construct validity of this experiment is the experimental objects. Both of them were developed by the author of the TWSM approach, which could mean that the objects were built to be suitable to the TWSM approach. To mitigate this threat the experimental objects used in the experiment were reused from other case studies non related to monitoring activities, which means that they were not built specifically for this experiment.

There are some threats to the external validity of the experiment. The population is not representative as they are grad students and not professionals. Moreover, the instrumentation may not reflect the real state of practice. The providers have to change the web service but they are not motivated by a real need of change. The amount of times the service change in two weeks may not be realistic. Thus, the results of the experiment cannot be generalized for every context and domain.

3-4 Operation of the experiment

The experiment was conducted during two weeks. In the first week, the providers had to change BankWS and they were free to make any kind of modification, but only once a day. The monitors had to use the appropriate approach to detect whether the monitored service had changed that day; which operations had changed and indicate which their confidence on the results given by the approach used. In the second week the providers and monitors performed the same activities, but this time for IRWS.

The providers of the experiment had no problem to change, instrument and publish the testable version of the web service. The monitors had no problem to perform the monitoring activities too, with the exception of the subject number 4. This subject made a mistake and did not use the structural testing information to perform the monitoring activities and the results obtained were excluded from the experiment. With this modification, the experiment has become balanced.

3-5 Data analysis

An ordinal scale was defined to score the experiment according to the results obtained by each approach. For the first hypothesis: the monitor wins one (1) point when correctly identifies that a web service has or has not changed; no point (0) when does not identify a change of the web service when it has changed or when identifies that a web service has changed when it has not. For the second hypothesis: the monitor wins two (2) points when correctly identifies that an operation has changed; one (1) point when correctly identifies that an operation has not changed; no point (0) when incorrectly identifies that an operation has changed; miss one point (-1) when incorrectly identifies that an operation has not changed. For the third hypothesis: the monitor wins three (3) points when the confidence is high; two (2) points when the confidence is medium; and one (1) point when the confidence is low.

The forms filled in by the monitors were compared with the forms filled in by the providers to score the results of the experiments according to the scale defined for each hypothesis. Figure 1 shows the box plots comparing the results of each hypothesis. Concerning the first hypothesis, the TWSM approach seems to be better than the functional approach to identify that a service is changed. The TWSM approach also seems to be more accurate to detect which operations have changed and which have not than the functional approach. Apparently, the monitors using the TWSM approach are more confident with the results obtained than the monitors using the functional approach.

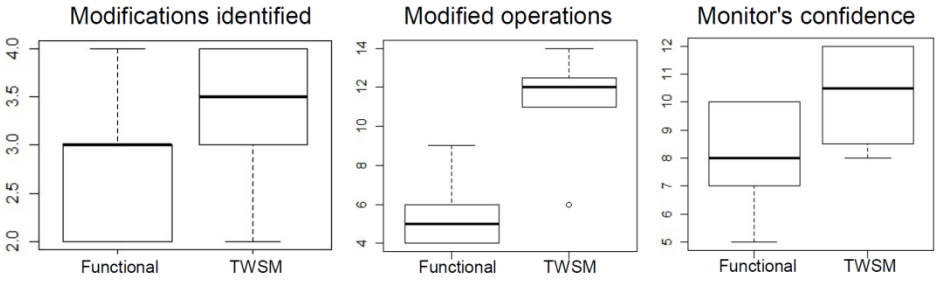

Figure 1 – Boxplots of the results of the experiment

3-6 Hypothesis testing

We used the T-test for testing the hypothesis of this experiment according to the recommendation of [12] given the design of the experiment. When testing the first hypothesis, the p-value is 0.05355 and t0 is 1.7222. According to the table of the t-value distribution, the t14 (16 samples) is 1.761 considering a confidence interval of 95%. These results allowed us to refute the null hypothesis $H0_1$ and accept the alternative hypothesis $H1_1$ because the p-value is similar to 0.05 and t0 >t14.

The p-value of the second hypothesis testing is 0.00003667 and t0 is 5.7138. Using a confidence interval of 99.95% the t14 is 4.14, which can refute the null hypothesis $H0_2$ and accept the alternative hypothesis $H1_2$ because t0 >t14 and the p-value is less than 0.01.

For the third hypothesis testing, the p-value was 0.01751 and t0 was 2.3351. The t14 is 2.145 using a confidence interval of 97.5%. In such case, we can refute the null hypothesis $H0_3$ and accept the alternative hypothesis $H1_3$, since the p-value is less than 0.025 and t0 >t14.

4. CONCLUDING REMARKS

We briefly presented a monitoring approach called TWSM that combines both functional and structural testing techniques to detect changes of services. We also presented an experiment to evaluate the TWSM approach by comparing it to another approach that uses only functional testing. The data analysis and the hypothesis testing of the experiment support some conclusions. Monitors using structural testing information combined with functional testing can identify changes of service with more accuracy than monitors using only functional testing. Also, monitors using structural testing have more confidence on the results of the monitoring activity since they have a better observation of the monitored service than the monitors using only functional testing.

We also performed an analysis of the results considering the experimental object separately. The analysis showed that the effectiveness of the functional approach depends on the nature of the modifications. The functional approach is suitable to identify service's changes when the modification affects the behavior and the results of the regression test set. Otherwise, the modifications may pass unnoticed. In such case, information provided by the structural testing technique can be useful to detect the unnoticed changes by analyzing the structure and the coverage of the service's code.

The results of the experiment, however, do not support the generalization of the conclusions for every situations and contexts due to some reasons: the subjects of the experiment are students, and not professionals; and the experiment was performed in a controlled environment with experimental objects that are not real world services developed by third parties. Despite this, the results of the experiment shows evidences that the use of structural testing information on monitoring activities can help monitoring strategies to improve their change detection mechanism. As stated by Myers [13] many years ago, functional testing and structural testing are complementary testing techniques and this is also true regarding service monitoring. Then, structural testing information is very useful to service monitoring activities.

ACKNOWLEDGMENTS

The authors would like to thank the Brazilian funding agencies: FAPESP (process 2008/03252-2), CAPES and CNPq for their financial support.

REFERENCES

[1] M. Bruno, G. Canfora, M. D. Penta, G. Esposito, and V. Mazza, "Using test cases as contract to ensure service compliance across releases," in Proceedings of the 3rd International Conference on Service-Oriented Computing, 2005, pp. 87–100.

[2] F. Barbon, P. Traverso, M. Pistore, and M. Trainotti, "Run-time monitoring of instances and classes of web service compositions," in Proceedings of the IEEE International Conference on Web Services, 2006, pp. 63–71.

[3] L. Baresi, C. Ghezzi, and S. Guinea, "Smart monitors for composed services," in Proceedings of the 2nd International Conference on Service Oriented Computing, 2004, pp. 193–202.

[4] Y. Gan, M. Chechik, S. Nejati, J. Bennett, B. O'Farrell, and J. Waterhouse, "Runtime monitoring of web service conversations," in Proceedings of the 2007 Conference of the Center for Advanced Studies on Collaborative Research, 2007, pp. 42–57.

[5] S. Ho, W. M. Loucks, and A. Singh, "Monitoring the performance of a web service," in Proceedings of the EEE Canadian Conference on Electrical and Computer Engineering, 1998, pp. 109–112.

[6] S. Lamparter, S. Luckner, and S. Mutschler, "Formal specification of web service contracts for automated contracting and monitoring," in Proceedings of the 40th Annual Hawaii International Conference on System Sciences, 2007, p. 63.

[7] H. Liu, Z. Li, J. Zhu, and H. Tan, "Business process regression testing," in Proceedings of the 5th International Conference on Service-Oriented Computing, 2007, pp. 157–168.

[8] S. Rosario, A. Benveniste, and C. Jard, "Monitoring probabilistic slas in web service orchestrations," in Proceedings of the 11th IFIP/IEEE International Conference on Integrated Network Management, 2009, pp. 474–481.

[9] Q. Wang, Y. Liu, M. Li, and H. Mei, "An online monitoring approach for web services," Annual International Conference on Computer Software and Applications, vol. 1, pp. 335–342, 2007.

[10] Q. Wang, J. Shao, F. Deng, Y. Liu, M. Li, J. Han, and H. Mei, "An online monitoring approach for web service requirements," IEEE Transactions on Services Computing, vol. 2, pp. 338–351, 2009.

[11] M. M. Eler, M. E. Delamaro, J. C. Maldonado, and P. C. Masiero, "Built-in structural testing of web services," in Proceedings of the 2010 Brazilian Symposium on Softtware Engineering.

[12] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, Experimentation in software engineering: an introduction. Norwell, MA, USA: Kluwer Academic Publishers, 2000.

[13] G. J. Myers, The Art of Software Testing. New York: Wiley, 1979.