Anti Pattern Detection over Services through Optimization techniques

[1]Kakarla sai Krishna,
[1]M.Tech,
[1]Computer Science and Engineering,
[1]JNTUACEA, Pulivendula, India

Abstract : **Service Oriented Architecture mostly resulted via integrated services rather than multiple individual services. As a result Service Based Systems are considered popularly in past. The problem with design over architectures related to SBS is considered to be poor designs which resulted as anti patterns. Those poor structured designs lead to low Quality of Service. Customer will be dissatisfied by suing those services. Therefore importance is given to design where service anti patterns and also services related to SBS are detected. So, our proposed work concentrates on detection of s anti patterned services and business related anti-patterns. The framework exploit co-operative parallel evolutionary algorithm (P-EA) to provide anti-patterns. Measures like cost and time are assessed in the framework. An algorithm called Particle Swarm optimization is built to demonstrate the proof of concept. The empirical results revealed the utility and performance improved over a state-of-the-art approach. It will be useful in validating SOA-based applications in software industry to improve QoS and leverage customer satisfaction.**

IndexTerms: Anti Patterns, ***P-EA, PSO, SBS, Web services, Web service antipatterns.***

I. INTRODUCTION

Web services technology and distributed computing has revolutionized the way applications work. The applications now can have heterogeneous environment. It does mean that there is integration achieved between heterogeneous applications for chain of businesses. The usage of web services makes applications inter-operable. The reason behind that is SBS can help users to have services with a single integrated application rather than multiple individual applications. Business process refers to a collection of web services that are combined with definite workflow. The problem with such applications is the design flaws or anti-patterns that may lead to failure of systems or any such serious issues.

To overcome anti-patterns issue, many researchers contributed. Web service anti-pattern detection is explored in [1], [9], [10] while business process anti-pattern detection is studied in [11], [14], [19] and [20]. In the literature it is found that there are many kinds of methods to detect anti-patterns. Methods are found for detection of both service anti-pattern and business process anti-pattern. The problem with existing approaches is that they focus in either of the applications. There is no framework that can cater to the needs of both with unified solution. In this paper we proposed a unified framework for detection of various kinds of anti-patterns of both the applications that form SBSs. Our contributions in this paper are as follows.

- We proposed a framework known as Automated Anti-Pattern Detection Framework which guides the users to have detection of web service anti-patterns

- Anti-patterns. We employed two benchmark datasets to evaluate the proposed methodology.
- We built a prototype application to demonstrate proof of the concept. The experimental solutions of the proposed technique show the importance of predicting anti-patters accurately with the help of proposed application.

The closing of the paper is arranged as follows. Segment II features related work. Segment III informs the proposed work. Segment IV contains implementation details. Segment V shows experimental effects. Segment VI concludes the paper following with future arrangements.

II RELATED WORK

This section provides review of literature on the web service and business process anti-pattern detection. Search based web service anti-patterns are explored in [1] and [8] while the best practices discovering and consuming services is studied in [2]. In the process of service creation finding bad smells in the source code is investigated in [3]. Similarly quality assurance in detecting such code smells is explored in [4]. Software testing to improve QoS is the focus in [5] while the trends and technologies in

search based software engineering is the main work in [6]. With respect to SOA quality measure are discussed in [7]. Important research on SOA anti-patterns is found in [10], [11], [16], [17] [22].
Very popular SOA anti-patterns are studied in [9]. Service anti-patterns and their taxonomy are described in [12]. Service patterns in cloud and their QoS are the main focus in the study made in [13]. Service patterns, Service anti-patterns in terms of change proneness are studied in [14]. A heuristic approach is found in [15] for anti-pattern detection over web services. Anti-patterns over web services with respect to SOA are studied in [16].Recognition for detection of SOA antipatterns is studied in [17]. Service oriented computing [18], SOA based anti-pattern detection [19], improving the methodology to detect SOA anti-patterns [20], a tool to improve detection of anti-patterns [21] are the important studies found in the literature. From the literature, it is found that there are many studies on web services and business process anti-patterns. However, they are made individually and a unified framework that can detect both of them is not found in the literature. Therefore, in this paper, we proposed a unified framework that can discover both web service anti-patterns and business process anti-patterns.

III PROPOSED METHODOLOGY

We proposed a methodology is for predicting anti-patterns in service based systems. The web service anti-patterns and business process anti-patterns are collected and used for experiments. The framework has mechanism to make use of metrics that can help in determining the presence of absence of anti-patterns. Different metrics considered for the web service anti-patterns are NPT, NOD, NCT, NOM, COH, COUP, NOC, etc. In the same fashion we used metrics for detection of business process anti-patterns. They are known as NICF, NIDF, NII, NIO, NOF, NOM, NUI, NUO, NIU and NIP. If one or more metrics show the value 1, then such web service or business process is considered to be an anti-pattern. This could help in having training and testing data for detecting anti-patterns in both kinds of applications.
The benchmark datasets are used to extract metrics as mentioned before. The metrics are used to create training and testing datasets. In our framework detection rules are generated by first population using genetic programming while simultaneously using the same algorithm second population tries to find anti-patterns. These algorithms are used to label unlabelled testing datasets in order to achieve proposed detection of web service and business process anti-patterns.
A prototype application is implemented using the proposed framework. The flow of the proposed application is as in Figure 1.

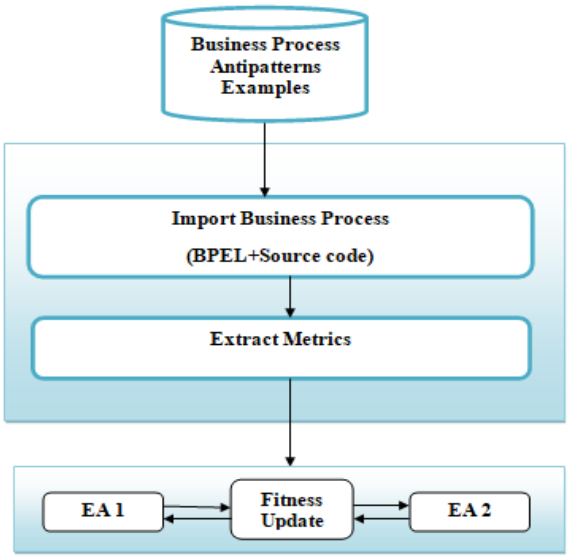

Figure1: Flow of the Proposed application

As shown in Figure 1, it is evident that the application is built to extract metrics from benchmark datasets and analyze them. The metrics are associated with different types of anti-patterns. The web service and business process anti-patters are detected using the prototype application.

IV IMPLEMENTATION DETAILS

The main application Interface is as displayed in Figure 2.

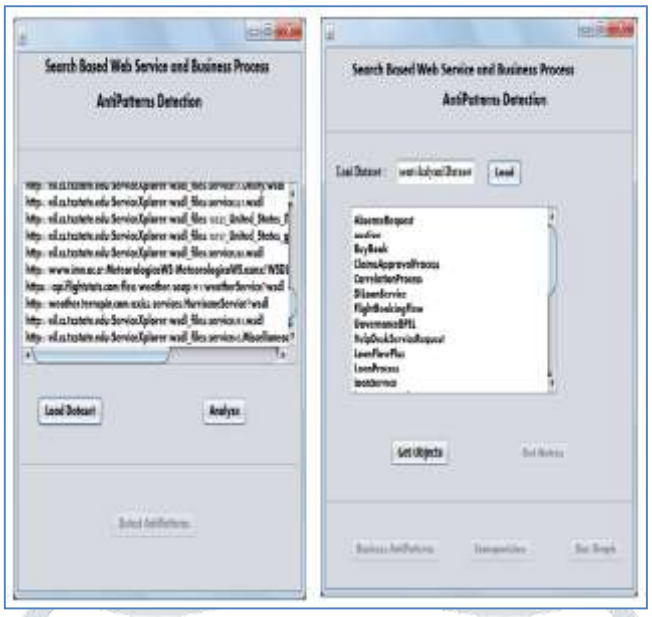

Figure 2: Main UI of the application for detecting web service and business process anti-patterns

As shown in Figure 2, it is evident that the application provides intuitive interface to proceed with the functionality described by the proposed framework. The training and testing are performed as per the P-EA proposed. The results are assessed based on precision and recall measures.

Dataset Details

Benchmark datasets are collected from [23] and [24] on web services anti-patterns and business process anti-patterns respectively. The datasets are studied and altered to have training and testing datasets. These datasets are used for empirical study while learning a classifier and then detect anti-patterns in given web services and business processes.

- Web service training dataset contains webs services wsdl file, the service name and values for metrics like NPT, NOD, NCT, NOM, COH, COUP, NOC, etc. followed by a label telling whether that service is anti-pattern or not antipattern.
- Business process training dataset contains bpel file, the process name and values for metrics like NICF, NIDF, NII, NIO, NOF, NOM, NUI, NUO, NIU and NIP followed by a label telling whether that process is anti-pattern or not anti-pattern.

V EXPERIMENTAL RESULTS

Table 1: Time and cost performance

Solution	cost	Time
PEA	89	93
PSO	90	95

As shown in Table 1, the cost and time measures for solutions such as web service anti-pattern detection and business process anti-pattern detection are presented. It is evident that the proposed solutions for service anti-pattern and business process anti-pattern detections are evaluated. Results showed that the cost and time terms differ for both the solutions. It is based on the number of samples in the dataset and prediction performance of the classifier.

VI CONCLUSION AND FUTURE WORK

In this paper, we target to detect web service anti-patterns and business process anti-patterns in the applications based on Service Oriented Architecture (SOA). The framework guides the users to have detection of web service anti-patterns and business process anti-patterns. The existing approaches in detection of anti-patterns in service based systems did not consider for detection of both web service and business process anti-patterns. In this paper we considered a PSO framework that covers different categories of anti-patterns for both kinds of SOA applications. We used two benchmark datasets covering web service anti-patterns and business process anti-patterns. We employed P-EA to learn from training set for both kinds of applications. Then P-EA builds a model that is used to predict labels for testing set. We built PSO application to demonstrate proof of the concept. The experimental results revealed that the proposed methodology is better than existing one in terms of precision and recall. In future we intend to explore hybrid approaches to have ensemble of methods for more accurate prediction of anti-patterns.

REFERENCES

[1] Ali Ouni, Marouane Kessentini, Katsuro Inoue and Mel O Cinn ´ eide. (2015). Search-based Web Service Antipatterns Detection. *IEEE TRANSACTIONS ON SERVICES COMPUTING*, p1-14.

[2] Rodriguez, Juan, ISISTAN - UNCPBA, Computación y Sistemas Crasso, Marco; ISISTAN - UNCPBA, Computación y Sistemas Mateos Diaz, Cristian; ISISTAN - UNCPBA, Computación y Sistemas Zunino and Alejandro. (2012). Best Practices for Describing, Consuming, and Discovering Web Services A Comprehensive Toolset. *Practice and Experience*, p129.

[3] Fabio Palomba, Gabriele Bavota, Massimiliano Di Penta, Rocco Oliveto, Andrea De Lucia and Denys Poshyvanyk. (2013). Detecting Bad Smells in Source Code Using Change History Information, p1-11.

[4] Eva van Emden. (2002). Java Quality Assurance by Detecting Code Smells. *IEEE*, p1-10.

[5] A. Arcuri and L. Briand, "A practical guide for using statistical tests to assess randomized algorithms in software engineering," in Proc. 33rd Int. Conf. Softw. Eng., 2011, p 1–10.

[6] M. Harman, S. A. Mansouri, and Y. Zhang, "Search-based software engineering: Trends, techniques and applications," ACM Comput. Surveys, 45, 1, p. 11, 2012

[7] R. Sindhgatta, B. Sengupta, and K. Ponnalagu, "Measuring the quality of service oriented design," in Proc. 7th Int. Joint Conf. Service-Oriented Comput., 2009, vol. 5900, pp. 485–499.

[8] W. Kessentini, M. Kessentini, H. Sahraoui, S. Bechikh, and A. Ouni, "A cooperative parallel search-based software engineering approach for code-smells detection," IEEE Trans. Softw. Eng., 40,9, pp. 841–861, 2014.

[9] J. Kral and M. Zemlicka, "Popular SOA antipatterns," in Proc. Computation World: Future Comput., Service Comput., Cognitive, Adaptive, Content, Patterns, 2009, pp. 271–276.

[10] N. Moha, F. Palma, M. Nayrolles, B. J. Conseil, Y.-G. Gueheneuc, B. Baudry, and J.-M. Jezequel, "Specification and detection of SOA antipatterns," in Proc. 10th Int. Conf. Service-Oriented Comput., 2012, pp. 1–16.

[11] Francis Palma, Naouel Moha, Yann-Gaël Guéhéneuc, Specification and Detection of Business Process Antipatterns. In Proceedings of the 6th International MCETECH Conference, May 12-15th, 2015, Montreal, Canada. (May 2015).

[12] Francis Palma and Naouel Moha, A Study on the Taxonomy of Service Antipatterns, in the proceedings of 2nd on Patterns Promotion and Anti-patterns Prevention (PPAP 2015) co-located with 22nd IEEE International Conference on Software Analysis, Evolution, and Reengineering (SANER 2015), Montréal, Canada.

[13] Geoffrey Hecht, Benjamin Jose-Scheidt, Clément De Figueiredo, Naouel Moha, and Foutse Khomh, An Empirical Study of the Impact of Cloud Patterns on Quality of Service (QoS). In IEEE CloudCom 2014, Singapore.

[14] Francis Palma, Le An, Foutse Khomh, Naouel Moha, and Yann-Gaël Guéhéneuc: Investigating the Change-proneness of Service Patterns and Antipatterns. In Proceedings of the 7th IEEE International Conference on Service Oriented Computing & Applications (SOCA), Matsue, Japan. (November 2014).

[15] Francis Palma, Johann Dubois, Naouel Moha, and Yann-Gaël Guéhéneuc: Detection of REST Patterns and Antipatterns: A Heuristics-based Approach. In Proceedings of the 12th International Conference on Service Oriented Computing (ICSOC), Paris, France. Springer (November 2014).

[16] Francis Palma, Naouel Moha, Guy Tremblay, and Yann-Gaël Guéhéneuc: Specification and Detection of SOA Antipatterns in Web Services. In Proceedings of the 8th European Conference on Software Architecture (ECSA), Vienna, Austria. Springer (August 2014).

[17] Francis Palma, Specification and Detection of SOA Antipatterns, PhD Symposium, in conjunction, with 30th International Conference on Software Maintenance and Evolution (Victoria, Canada), ICSME 2014, September 28 - October 3, Victoria, Canada 2014.

[18] Anthony Demange, Naouel Moha, Guy Tremblay: *Detection of SOA Patterns* Service-Oriented Computing. Lecture Notes in Computer Science, Volume 8274, Springer Berlin Heidelberg (2013) 114--130.

[19] Francis Palma, Mathieu Nayrolles, Naouel Moha, Yann-Gaël Guéhéneuc, Benoit Baudry and Jean-Marc Jézéquel, *SOA Antipatterns: An Approach for their Specification and Detection.* International Journal of Cooperative Information Systems 22 (04) (2013).

[20] Mathieu Nayrolles, Naouel Moha, Petko Valtchev, *Improving SOA Antipatterns Detection in Service Based Systems by Mining Execution Traces*. In Reverse Engineering (WCRE), 2013 20th Working Conference on. (Oct 2013) 321--330

[21] Mathieu Nayrolles, Naouel Moha, Petko Valtchev, *A Tool to Improve the Detection of Antipatterns in SOA Systems by Mining Execution Traces*. Tool Demo Track in Reverse Engineering (WCRE), 2013 20th Working Conference on. (Oct 2013)

[22] Francis Palma, *Detection of SOA Antipatterns*, 8th PhD Symposium (Shanghai, China), in conjunction with ICSOC 2012, (10th International Conference on Service Oriented Computing), November 12-16, Shanghai, China, 2012.

[23] Web Service Anti Patterns. Retreieved from https://github.com/ouniali/WSantipatterns. Accessed on 20 March 2018.

[24] Business Process Anti Patters. Retreived from http://sofa.uqam.ca/soda-bp/. Accessed on 10 April 2018.