



Monitoring SOA Based Applications According to Business Level Agreement

Sami Ali AlYazidi, Ahmed Zayed Emam
College of Computer and Information Sciences
King Saud University
Kingdom of Saudi Arabia

Abstract: This paper discusses a new type of agreement that might be established in SOA environment; business level agreement -BLA. Compared to SLA; BLA is providing a higher level of monitoring on system's activities. Thus, monitoring BLA being required in order measure where a system/application is performing well according to its business goals.

Key words: Business Level Agreement (BLA), Service Level Agreement (SLA), SOA monitoring.

I. Introduction

Nowadays Service Oriented Architecture SOA has been widely used and many applications are built upon it. Many organizations felt the benefits of that architecture and establish to align their business according to. SOA based applications provide breaks down the business logics and goals of an organization into atomic services that can be easily managed, customized and monitored [1]. Those services are communicated between service provider (the system which provide the service and business logic) and service consumer (the system which consume the service for the requester) through broker. Broker or integration (technically named as Enterprise Service Bus) is the key player which facilitate the communication between service provider and consumer event if they are totally different systems. Monitoring such applications is not an easy thing and needs more research to focus on the key elements which can provide a representative data [2].

SLA can be defined as a contractual obligation between a service consumer and a service provider, which can represent guarantees of quality of service of a service consumer and promises of a service provider [11].

Many researches had been conducted to monitor the SLA (Service Level Agreement) between service consumer and provider to monitor the functional and non-functional aspects from service point of view [3]. On the other hand, the need of monitoring a higher level of focus is becoming noticed for business advantages [4].

II. SLA vs BLA

Service-Level Agreements (SLA) is an approach to establish guarantees for the low-level aspects of the interaction between a service provider and a service consumer [16]. Whereas BLA is looking for higher level agreement of how a business transaction is conducted with all related aspects such as low-level services involved, total execution time, service consistency and data transparency [4][7]. In other words, A BLA concerns the agreement of higher business goals, thus it is created by business analysts, whereas an SLA concerns technical characteristics of a service.

A service level agreement is an agreement regarding the guarantees of a web service. It defines mutual understandings and expectations of a service between the service provider and service consumers. The service guarantees are about what transactions need to be executed and how well they should be executed.

There is a clear difference between SLA and BLA; SLA concerns agreements on the availability degree of a Web service BLA concerns agreements on what a Web service does and how well it does it [4][7].

III. Problem statement

No much researches (as best to my knowledge) focus on the business level monitoring on such architecture (SOA). Here is this paper we are establishing the state of the art of monitoring Business Level Agreement (BLA) which focus on higher level of SOA based applications. BLA is looking into business aspects provided by SOA-based applications that focus on monitoring functions executed by the applications from business point of view.

IV. Purpose

This research aims to introduce a new type of agreement for software services, named Business-Level Agreement (BLA), which is complementary to the technical SLAs and focus on the business arguments and aspects. Moreover, we proposed a framework that can monitor the BLA defined on SOA based applications.

V. Significance of BLA Monitoring

The need for monitoring SOAs environments at run time (based on SLA) has been discussed widely in research and industry recently [1]. Service monitoring can play an important role in cutting testing costs. Monitoring is also useful for such tasks as described in [6]:

- Avoiding failures - e.g. by replacing an unavailable service with another service of the same functionality;
- Enforcing the SLA compliance between the provider and the consumer;
- Ensuring that a service satisfies the pre- and post-conditions; and
- Enabling recovery activities if required.

However, Business monitoring is a crucial aspect that leads any organization to its success. It give insight to the decision makers within the organization to understand the current situation, what happen in the past, why it went like this and what will happen in future. Thus, developing a solution that monitor business aspect will make the system/application aligned with organization business goals since they are not only monitoring functional attributes (SLA) but also business attributes (BLA).

VI. Related Work.

There have been a lot of efforts by the research community towards the development of deferent methodologies for the discovery, negotiation, monitoring, and adaptation of software services using SLAs. However, much less focus has been placed in higher-level aspects of this interaction, i.e. on the actual business activities realized through a software service [7][4]. In a situation where an SBA (Service Based Application) uses a software service that has high availability and high responsiveness, but the service delivers a business activity of low quality, the business value of the SBA will suffer. Therefore, the business-level becomes a principal concern for an organization, in order to maintain the long-term business value of its SBAs.

Generally, there are three different approaches for web service monitoring: consumer-side monitoring, provider-side monitoring, and third-party (broker) monitoring [8]. The drawback of provider-side and consumer-side monitoring is that in the case of problems neither side will trust the others. Third party monitoring is an approach to solve this issue [8]. Main disadvantage of this approach is the resulting bottleneck when all calls between the consumer and provider sides have to be intercepted by this third party. In order to avoid this, one could monitor only a limited number of service calls, leading to a reduced accuracy of the results [9].

Authors in [12] proposed a monitoring framework for WSBPEL (Web Services-Business Process Execution Language) processes for both functional and non-functional requirements. The approach presented in this framework concentrates on client-side monitoring and relies on WS-Policy to express the monitoring policies associated with WS-BPEL processes, that is, the user requirements (constraints) on running Web services compositions. All constraints are written in WS-CoL (Web Service Constraint Language), a domain-independent language for monitoring assertions. All of these mentioned frameworks only monitor the functional characteristics of the services.

VII. Proposed Framework Architecture

The suggested framework is composed of two containers namely BLA interceptor and Run Time environment Interceptor. Hereunder description of those containers and it belonging components. Below figure shows the framework containers and components.

BLA interceptor:

This container mainly managing the BLA, deciding breaches of BLA and reporting monitoring results. It has four component described below.

Business interface: this component is the interface between business analyst and framework. It is the entry point to take the data from business analyst and communicate it to business rule.

Business Ruler: it takes the BLA entered through business interface and convert it to business rule that can be monitored thought the framework. It guarantees the consistency of the rules and their parameters and attributes.

Data presenter: This component concerned with presenting the monitoring results and data to business analyst in human understandable way though the business interface.

Engine Controller: This component in the heart of the framework, it controls other component within the framework. It reads and understands rules from business rules. Also it communicates with data collector and verifies if the business rules happened.

Run Time Environment Interceptor:

This container deals with the running environment; it logs SOA services, collects the required data and discovers the environment state. Below are the contained components.

Data Collector: it collects the data being monitored within the topology and web service logger. This data is being communicated to Engine controller to decide if it match the business rules.

Web Service Logger: it logs the data at web service level to ensure the technical characteristics of a web service.

Topology discovery: This component informs the state of the run time environment and what web service exits and running. It also provides general information about the environment.

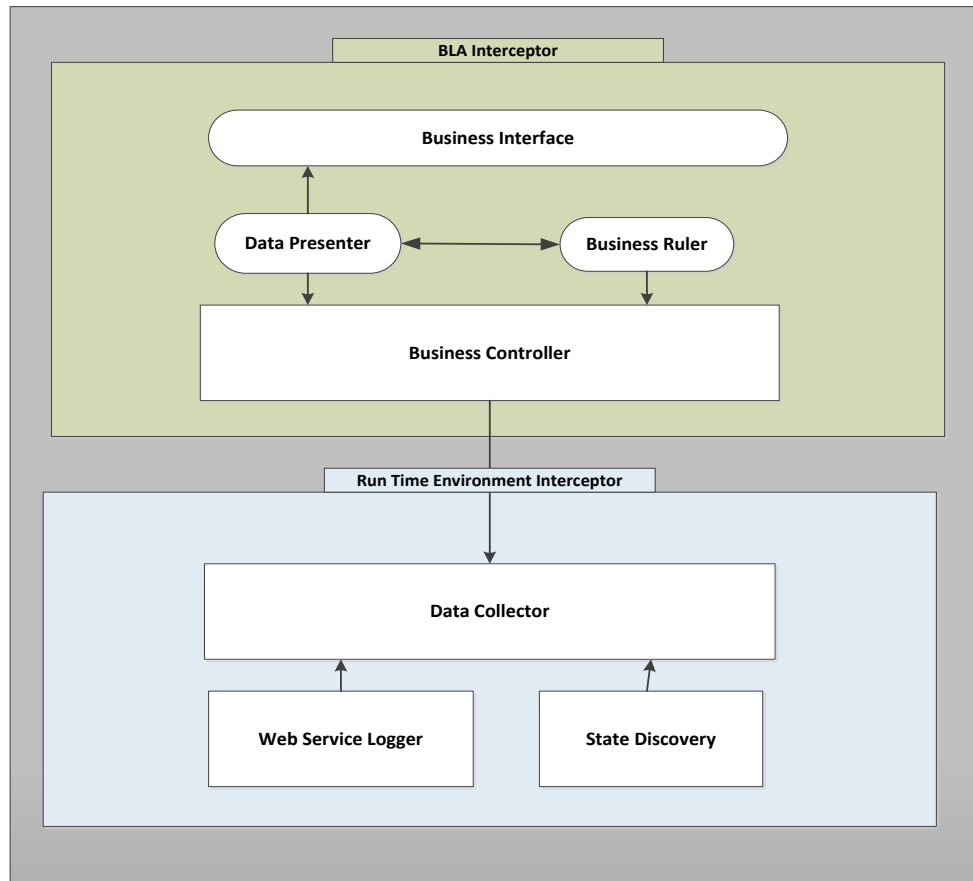


Figure 1: BLA Monitoring Framework

VIII. Monitoring Mechanism

Our monitoring mechanism is based on two major containers; BLA interceptor and run-time environment interceptor. Run Time Environment interceptor looking after the SOA run time environment and provide classic data of the environment such as environment state, web services running in place, web services executed environment errors and other information help monitoring the run time environment. BLA interceptor, on the other side, is looking after the BLA and business rules if they are executed and elect specific monitored data to compare with its business rules. BLA interceptor components are dealing with out layer of the framework through business interface, at the same time also with run time environment interceptor through business controller. Business controller is the corner stone of the BLA interceptor, it is responsible of handling the whole data in the framework and deciding which data to be collected and reported.

IX. Monitoring scenario

BLA is entered (normally by business analyst) through business interface. Business ruler takes the entered BLA and converts it to business rules with check of consistency of the business rules.

Data collector is continuously collecting data from state discovery and web service logger in reconfigurable predefined patterned.

Then business controller reads the rules from business ruler and find if matched with data being collected by data collector. If business rules verified, business controller push the collected data with match business rule to data presenter which present it to business analyst in human understandable format.

X. Conclusion

This paper highlighted the Business Level Agreement (BAL) and compares it with Service Level Agreement. We show the importance of BLA how it can be beneficial to the organization when monitoring SOA-based applications according to BAL. Also we introduced a framework for monitoring BLA on SOA-based applications and discussed its architecture components, monitoring mechanism and monitoring scenario. Related work on SLA had been also discussed.

As future work, we will implement the framework and test it on SOA environment and discuss the results.

References

- [1]. Safy, Farag Zakaria, Mohammad El-Ramly, and Akram Salah. "Runtime Monitoring of SOA Applications: Importance, Implementations and Challenges." In SOSE, pp. 315-319. 2013.
- [2]. Spiess, Patrik, et al. "SOA-based Integration of the Internet of Things in Enterprise Services." Web Services, 2009. ICWS 2009. IEEE International Conference on. IEEE, 2009.
- [3]. Kosinski, Jacek, et al. "SLA monitoring and management framework for telecommunication services." Networking and Services, 2008. ICNS 2008. Fourth International Conference on. IEEE, 2008.
- [4]. Bratanis, Konstantinos. "Business-Level Agreements for the Future Internet of Services."
- [5]. Wei, Wei, et al. "Effective detection of sophisticated online banking fraud on extremely imbalanced data." World Wide Web (2013): 1-27. **23
- [6]. Winkler, M. , J. Cardoso, and G. Scheithauer, "Challenges of Business Service Monitoring in the Internet of Services". iiWA S2008, ACM, Austria. 2008. P P : 6 1 3-6 1 6
- [7]. Bratanis, Mr Konstantinos, Dimitris Dranidis, and Anthony JH Simons. "Conformance to Business-Level Agreements."
- [8]. Kwon, Sungju, and Jaeyoung Choi. "An agent-based adaptive monitoring system." Agent Computing and Multi-Agent Systems. Springer Berlin Heidelberg, 2006. 672-677.
- [9]. Chen, Anan, Tang, Yazhe, Liu, Yuan, and Li, Ya, MAGMS: Mobile Agent-Based Grid
- [10]. Schmietendorf, Andreas, Reiner Dumke, and Daniel Reitz. "SLA management-challenges in the context of Web-service-based infrastructures." Web Services, 2004. Proceedings. IEEE International Conference on. IEEE, 2004.
- [11]. Al-aaidroos, Mohammed, Norleyza Jailani, and Muriati Mukhtar. "Agent-based negotiation framework for web service's SLA." Information Technology in Asia (CITA 11), 2011 7th International Conference on. IEEE, 2011. [11]
- [12]. Baresi, L., S. Guinea, and P. Plebani, "WS-Policy for Service Monitoring ". springer, LNCS 3 8 1 1 , pp. 72-8 3 , 2006.
- [13]. Monitoring System, Lecture Notes in Computer Science, Vol. 3841, Springer-Verlag
- [14]. P. Bhuyan, C. Prakash, D. Mohapatra, "A survey of regression testing in SOA", International Journal of Computer Applications, vol. 44, no. 19, pp. 0975 – 8887, April 2012.
- [15]. Safy, Farag Zakaria, Mohammad El-Ramly, and Akram Salah. "Runtime Monitoring of SOA Applications: Importance, Implementations and Challenges." In SOSE, pp. 315-319. 2013.
- [16]. Zhang, Liang-Jie LJ. "EIC editorial: Introduction to the body of knowledge areas of services computing." *IEEE Transactions on Services Computing* 2 (2008): 0062-74.
- [17]. Psiuk, Marek, and Tomasz Bujok. "Enterprise Service Bus Monitoring Framework for SOA Systems." Services Computing, IEEE Transactions on 5.3 (2012): 450-466.
- [18]. Chen, Cuiting, Andy Zaidman, and Hans-Gerhard Gross. "A framework-based runtime monitoring approach for service-oriented software systems." In Proceedings of the International Workshop on Quality Assurance for Service-Based Applications, pp. 17-20. ACM, 2011.