AGREEING ON AND CONTROLLING SERVICE LEVELS IN SERVICE-ORIENTED ARCHITECTURES

Benjamin Heckmann[1], Andrew D. Phippen[2], Ronald C. Moore[1] and Christoph Wentzel[1]
[1]*University of Applied Sciences Darmstadt, Haardtring 100, 64295, Darmstadt, Germany*
[2]*University of Plymouth, Drake Circus, PL4 8AA, Plymouth, U.K.*

Keywords: Availability, Business Processes, Monitoring, Reliability, SOA.

Abstract: Business Service Level Agreements (BSLAs) are introduced as a generalised concept to agree on feasibility and workload of business processes hosted in service-oriented architectures as an alternative to technical SLA. Based on BSLAs an according approach to control feasibility at runtime is presented.

1 INTRODUCTION

In Service-oriented Architectures (SOA) Service Level Agreements (SLAs) (Group et al., 2011) are used to specify technical thresholds, corresponding actions to keep them and penalties when failing. From a business point of view, only the feasibility of business processes is of interest for economical success. Business processes are feasible if a given workload is processed in a certain maximum time frame. Workloads of processes are the sum of all current actively processed process instances. Determining whether a business process based on a SOA is feasible can be complex for highly meshed service cascades including redundant alternate service offers (e.g., for load balancing).

This research is conducted in cooperation with secco[1] as business partner, introducing the problem statement. The applied research aims at providing a concept to monitor the feasibility and workload of business processes, hosted on multi-tier IT service provision infrastructures for SOA services, without the need for active technical monitoring. To elaborate an solution approach, a typical multi-tier IT service provision infrastructure operated in the context of our business partner is analysed in section 3. A generalised concept to track feasibility and workload of business processes hosted based on multi-tier infrastructures is elaborated in section 4. In section 5, a BSLA monitoring framework is implemented as a proof-of-concept.

[1]secco advanced GmbH, Grossostheim, Germany, http://www.seccoadvanced.de.

2 RELATED WORKS

This paper offers an approach to technically converge the quality-related ontologies of service, experience, and business as introduced in (Van Moorsel, 2001; Dobson and Sanchez-Macian, 2006). Most other authors address technical perspectives on SLA in SOA. From the IT architecture point of view, authors deal with SLA descriptions of performance modelling (Brebner, 2008), SLA-driven development (Muthusamy et al., 2009) or dependability throughout the life cycle (Stantchev and Malek, 2010). In operations management SLA are mostly enforced through an distribute-and-enforce tactic. By (Hsu et al., 2008; Raibulet and Massarelli, 2008; Chen et al., 2009; Muthusamy and Jacobsen, 2008) highly detailed SLAs are defined, distributed and then enforced on each member of a service cascade. The complexity to manage such approaches increases with the complexity of the given cascade. (Stantchev and Schroepfer, 2008) decouples SLA operations management from the complexity of a service cascade. This paper presents a similar approach and advances it by embedding BSLA in a whole life cycle concept (Heckmann and Phippen, 2010). Technical operations is focused on the *technical monitoring* of technical resource thresholds. Three types can be distinguished: active, passive and agent-based (Utlik and Alexeyev, 2010). Other authors propose the *workflow monitoring* of business process workloads. It is focused on the workflow state rather than underlying technical measurements (Ou et al., 2008). In contrast, (Moser et al., 2008) aims to provide a non-intrusive workflow mon-

Heckmann B., D. Phippen A., C. Moore R. and Wentzel C..
AGREEING ON AND CONTROLLING SERVICE LEVELS IN SERVICE-ORIENTED ARCHITECTURES.
DOI: 10.5220/0003871702670270
In *Proceedings of the 2nd International Conference on Cloud Computing and Services Science* (CLOSER-2012), pages 267-270
ISBN: 978-989-8565-05-1

itoring approach combined with active SLA management. This paper broadens this approach to incorporate technical monitoring data and address general IT services based on IP networks.

3 MULTI-TIER INFRASTRUCTURE ANALYSIS

In the context of secco, SOA infrastructures consist of 7 horizontal layers, shown in Figure 1. Business processes are represented by technical workflows acting as service consumers on the top layer. Business functionality is provided by the orchestration (Andrews et al., 2003) of application layer services, for example web services (Haas and Brown, 2004). These service instances are hosted on the application infrastructure layer (e.g., within database systems or application servers). All software components from upper layers are deployed on the operating system layer, each instance running in a virtual machine on the virtual infrastructure layer. The virtual hardware is mapped to resources on the physical systems layer. As the final layer, the network services connect these systems relying on resources such as routers, switches or domain name services. Complementary to the previously described horizontal multi-tier SOA infrastructure, there is the vertical technical monitoring layer. It evaluates technical measuring points of the horizontal layers, such as network availability, CPU load, memory consumption or storage usage.

Analysed service cascades include redundant service offers and the support for dynamic coupling[2] between service consumer and provider should be considered. The technical monitoring solutions Amberpoint, Progress Actional, SOA Manager Service Manager, Oracle Enterprise Manager SOA Management Pack and OpTier CoreFirst do not offer sufficient information to gather quantifications of failure impacts and reliable conclusions on the feasibility of the implemented business processes in the given SOA infrastructures. Specifically analyses of the current feasibility of business processes in scenarios with redundant service offers fail due to the evaluation of secco[3].

4 SOLUTION DESIGN

To agree on feasibility and workload of business processes Business Service Level Agreements (BSLA) are proposed as abstraction layer for the contracting of service quality between service consumer and service provider. BSLAs are focused on the description of the estimated usage behaviour, extended by the declaration of the maximum allowed response time for service requests and can optionally be enriched by the declaration of maintenance windows, maximum unplanned downtimes, fines, pricing or other non-functional properties. BSLAs are aimed at replacing SLAs. In BSLAs the consumer's usage behaviour is described by *Usage Patterns* (Heckmann and Phippen, 2010), which offer an approach for the description of the quantitative consumer-provider-relation in terms of request frequency and processing complexity. BSLAs enable analyses on business process feasibility and workload by specifying the *contracted usage*. In opposite, the *monitored usage* reflects the current request amount and resource utilisation within IT infrastructures. The business process's workload is determined by comparing its contracted and monitored usage, assuming all infrastructure components are technically available. To enable monitoring of the request amount this paper proposes the use of a centralised request routing component, named Service Broker[4] (see Figure 1). The Service Broker provides a measuring point for request amounts per business process, which represent the process's workload, taking the contracted usage as reference. The business process's feasibility is lead back from its workload combined with information about the technical availability of all infrastructure components hosting the process.

The aggregation of technical monitoring information in service cascades hosting business processes is addressed by a *topology graph*. The term topology graph is introduced to reflect the functional dependencies between the components in an IT infrastructure. To build the topology graph infrastructure components can be retrieved from a configuration management database (CMDB) (Group et al., 2011). To represent redundant service offers within a topology graph *service lines* are introduced. A service line is a logical group of infrastructure components that are necessary to provide an application layer service. To aggregate resource utilisation of service line spanning resources the term *component category* is introduced. Component categories logically group infrastructure components that provide similar functionalities (e.g., application servers, which provide hosting of application layer services), see Figure 1 as example. To calculate the resource utilisation, each topology graph node is enriched with interpreted technical monitor-

[2]Intermediate logic that changes the invocation target of a service request at runtime.

[3]Based on an internal technical report in June 2009.

[4]The Service Broker acts as a economical load-balancer for cloud infrastructures (Heckmann, 2007).

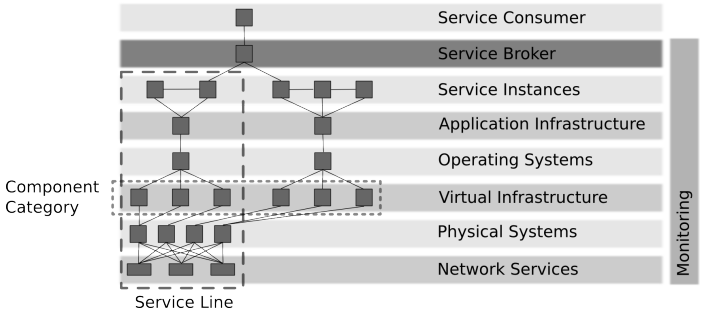

Figure 1: Multi-tier architecture including service broker, service line and component category as example.

ing information. The enriched graph is introduced as *availability graph*. Considered are two levels of monitoring data interpretation.

In case of *state-based* analyses the availability of a graph node is lead back by interpreting state-related technical monitoring data of the represented infrastructure component, such as ping states[5] retrieved from a technical monitoring system. Interpretation of this data limits results to two simple states: available and non-available. This variant is simpler to impose, but is less significant when deducting the feasibility of constitutive business processes. For *load-based* analyses the availability of a graph node is estimated comparing current resource utilisation with its maximum capacity. The utilisation is calculated based on load-related technical monitoring data like CPU load. This enables the provision of proportional metrics reflecting the current availability of the represented resource (e.g., 20 % utilisation of a DNS server).

The Service Broker estimates a business process as feasible if in the availability graph all state-based nodes of at least one service line are available and the resource utilisation of all load-based component categories offer sufficient reserves to process the *estimated usage*. The estimated usage for a given time frame is calculated by subtracting the monitored usage for a given business process from its contracted usage. In strictly state-based availability graphs only the process workload is taken into account when calculating the estimated usage, otherwise also the resource utilisation is incorporated.

5 PROOF-OF-CONCEPT IMPLEMENTATION

As proof-of-concept an application was implemented representing the state-based availability graph of an exemplary business process. For technical monitoring the implementation bears on Zabbix[6]. The application uses a given XML configuration file representing the topology graph. Its topology graph consists of nine nodes representing two service lines. Each node description is enriched with a reference to its Zabbix database identifier. The application extracts the state values for all nodes from the monitoring system. It autonomously determines the service lines and aggregates their availability states. As first outcomes, the ability to realise state-based availability graphs based on technical monitoring data is presented. The validation of the determination reliability is subject of future research.

6 CONCLUSIONS

This paper introduces a alternate abstraction layer in order to agree on the feasibility and workload of a business process instead of technical thresholds of the underlying technology as known from common SLA. This layer is called Business Service Level Agreements (BSLA) and establishes a black box around service capacity and technical implementation, thus loosening the coupling between technical service provision and business service consumption on the level of service agreements. Based on the identified qualified technical indicators, the paper evolves that BSLA approach. Corresponding, an approach for the technical monitor and enforcement of BSLAs during operations is presented. Concluding, a proof-of-concept implementation demonstrates the capabilities of these approaches.

REFERENCES

Andrews, T., Curbera, F., Dholakia, H., Goland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D.,

[5]Ping enables the monitoring of the technical network interface of a remote system and is specified in the Internet Control Message Protocol (ICMP) (Postel, 1981).

[6]http://www.zabbix.com

Thatte, S., Trickovic, I., and Weerawarana, S. (2003). *BPEL4WS, Business Process Execution Language for Web Services Version 1.1*. IBM, BEA Systems, Microsoft, SAP AG, Siebel Systems.

Brebner, P. C. (2008). Performance modeling for service oriented architectures. In *Companion of the 30th international conference on Software engineering*, pages 953–954, Leipzig, Germany. ACM.

Chen, Y., Iyer, S., Milojicic, D., and Sahai, A. (2009). A systematic and practical approach to generating policies from service level objectives. In *Integrated Network Management, 2009. IM '09. IFIP/IEEE International Symposium on*, pages 89 –96.

Dobson, G. and Sanchez-Macian, A. (2006). Towards unified QoS/SLA ontologies. In *IEEE Services Computing Workshops, 2006. SCW '06*, pages 169–174. IEEE.

Group, A., TSO, and Office, C. (2011). ITIL. http://www.itil-officialsite.com.

Haas, H. and Brown, A. (2004). Web services glossary. http://www.w3.org/TR/ws-gloss/.

Heckmann, B. (2007). Service provision in a utility computing environment. In *Proceedings of the Third Collaborative Research Symposium on Security, E-Learning, Internet and Networking*, pages 185–198, Plymouth, UK. Lulu.com.

Heckmann, B. and Phippen, A. (2010). Quantitative and qualitative description of the consumer to provider relation in the context of utility computing. In *Proceedings of the Eighth International Network Conference (INC 2010)*, pages 335–344, Heidelberg, Germany.

Hsu, C., Liao, Y., and Kuo, C. (2008). Disassembling SLAs for follow-up processes in an SOA system. In *2008 11th International Conference on Computer and Information Technology*, pages 37–42, Khulna, Bangladesh.

Moser, O., Rosenberg, F., and Dustdar, S. (2008). Non-intrusive monitoring and service adaptation for WS-BPEL. In *Proceeding of the 17th international conference on World Wide Web*, pages 815–824, Beijing, China. ACM.

Muthusamy, V. and Jacobsen, H. (2008). SLA-driven distributed application development. In *Proceedings of the 3rd workshop on Middleware for service oriented computing*, pages 31–36, Leuven, Belgium. ACM.

Muthusamy, V., Jacobsen, H., Chau, T., Chan, A., and Coulthard, P. (2009). SLA-driven business process management in SOA. In *Proceedings of the 2009 Conference of the Center for Advanced Studies on Collaborative Research*, pages 86–100, Ontario, Canada. ACM.

Ou, T., Sun, W., Guo, C., and Li, J. (2008). Visualized monitoring of virtual business process for SOA. In *Proceedings of the 2008 IEEE International Conference on e-Business Engineering*, pages 767–770. IEEE Computer Society.

Postel, J. (1981). Internet control message protocol - RFC 792. http://tools.ietf.org/html/rfc792.

Raibulet, C. and Massarelli, M. (2008). Managing non-functional aspects in SOA through SLA. In *2008 19th International Conference on Database and Expert Systems Applications*, pages 701–705, Turin, Italy.

Stantchev, V. and Malek, M. (2010). Addressing dependability throughout the SOA life cycle. *IEEE Transactions on Services Computing*, 99(PrePrints).

Stantchev, V. and Schroepfer, C. (2008). Techniques for service level enforcement in web-services based systems. In *Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services*, pages 7–14, Linz, Austria. ACM.

Utlik, A. and Alexeyev, N. (2010). Comparative analysis of service level agreement monitoring methods. In *Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET), 2010 International Conference on*, pages 346 –346.

Van Moorsel, A. (2001). Metrics for the internet age: Quality of experience and quality of business. *5th Performability Workshop*.