

# SLAs for Cross-layer Adaptation and Monitoring of Service-Based Applications: A Case Study

Konstantinos Bratanis  
South-East European  
Research Centre,  
CITY College,  
International Faculty of the  
University of Sheffield,  
24 Proxenou Koromila Street,  
54622, Thessaloniki, Greece  
kokratanis@seerc.org

Dimitris Dranidis  
Department of  
Computer Science  
CITY College,  
International Faculty of the  
University of Sheffield,  
3 Leontos Sofou Street,  
54626, Thessaloniki, Greece  
dranidis@city.academic.gr

Anthony J.H. Simons  
Department of  
Computer Science  
University of Sheffield  
Regent Court  
211 Portobello Street  
Sheffield S1 4DP, UK  
a.j.simons@sheffield.ac.uk

## ABSTRACT

Cross-layer adaptation and monitoring (CLAM) is an approach to the run-time quality assurance of service-based applications (SBAs). The aim of CLAM is to monitor the different layers of an SBA and correlate the monitoring results, such that in the event that a problem occurs an effective adaptation strategy is inferred for enacting a coordinated adaptation across all layers of the SBA. An important aspect of CLAM is the definition of the appropriate Service-Level Agreements (SLAs) for third party services utilised in the different layers of the SBAs. In this paper, we present insights into how to define SLAs for CLAM, by analysing SBAs in order to differentiate the third party business, software and infrastructure services utilised by the SBA. As a case study, we apply the analytical approach to an existing platform-as-a-service framework, which has been developed as an SBA and could benefit from CLAM. The analysis reveals the different third party services and their characteristics, as a precursor to defining SLAs. The case study successfully demonstrates how distinct SLAs for business, software and infrastructure services may be applied respectively in the BPM, SCC and SI layers of an SBA, to provide a flexible monitoring and adaptation response across layers.

## Categories and Subject Descriptors

D.2.11 [Software Engineering]: Software Architectures—*Service-oriented architecture (SOA)*; D.2.9 [Software Engineering]: Management—*Software quality assurance (SQA)*

## General Terms

Reliability, Verification, Design

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

QASBA '11, September 14, 2011, Lugano, Switzerland  
Copyright 2011 ACM 978-1-4503-0826-7/11/09 ...\$10.00.

## Keywords

Service-Based Application, Quality Assurance, Monitoring, Adaptation, Service-Level Agreements, Case study

## 1. INTRODUCTION

One of the main barriers to the adoption of service-based applications (SBAs) is the concern raised over the trustworthiness and reliability of third party services utilised in an SBA. The third party software services are often implemented as Web services that realise business activities, such as paying with a credit card or shipping purchased goods, and are beyond the control of the SBA provider. The problem of reliability becomes more complex when third party cloud computing services are utilised as the underlying infrastructure for provisioning the SBA. Given that the SBA provider does not have control over the quality of the third party services, unreliable third party services could threaten the quality of the SBA and result in lower business performance, software faults, and performance degradation that could consequently lead to the total collapse of the SBA. Therefore the dependability of the third party business, software, and infrastructure services utilised in an SBA becomes a principal concern for the SBA provider, who will require to adopt mechanisms within the SBA for quality assurance during run-time.

An approach to the run-time quality assurance of SBAs is the cross-layer adaptation and monitoring (CLAM), which aims on detecting problems early in the SBA layers and coordinating effective corrective actions across the SBA layers, such that problems are compensated for, or even prevented from occurring [9]. The functional layers of an SBA has been introduced in [6] and comprise the business process management (BPM), service composition and coordination (SCC), and the service infrastructure layers (SI). Based on the aforementioned separation of the SBA layers, in this paper we suggest that each layer concerns a different type of services. For example, the BPM layer concerns business services, the SCC layer concerns software services, and the SI layer concerns infrastructure service. It is necessary to perform analysis of the SBA to identify the business, software, and infrastructure services and their characteristics, in order to define appropriate Service-Level Agreements (SLAs) for such services. We present ideas for defining SLAs, used in CLAM approaches, by performing analysis of SBAs in

order to identify the third party services and their characteristics utilised in each SBA layer. We do not focus on a concrete reference architecture for CLAM. We describe a new case study for applying CLAM related to a platform-as-a-service (PaaS) offering, which has been implemented as an SBA. We analyse the PaaS offering, in order to identify the third party business, software, and infrastructure services and their characteristics for the definition of SLAs.

The rest of the paper is organised as follows. In Section 2 we present related work on CLAM. In Section 3 we describe insights for defining SLAs in SBAs. In Section 4 we present a new case study for CLAM related to a PaaS offering and we perform analysis for defining SLAs. Finally, in Section 5 we discuss conclusions and we provide an outlook for future research.

## 2. RELATED WORK

Recent research into run-time quality assurance has focused on implementing CLAM techniques for SBAs [9], by integrating the existing fragmented work in the field of adaptation and monitoring of service-based systems. Gjørven et al. [5] introduce a middleware for supporting the implementation of cross-layer self-adaptation of SBAs. Kazhamikin et al. [6] describe a conceptual framework comprising the definition of the SBA layers and a set of requirements needed to be addressed by mechanisms and techniques for CLAM of SBAs. Popescu et al. [10] present a methodology for cross-layer adaptation using adaptation templates. Latest research has focused on SLAs for CLAM. More particularly, Fugini et al. [4] describe an SLA contract that comprises parameters from user goals, business service and IT infrastructure for CLAM of SBAs. Schmieders et al. [11] propose the combination of SLA prediction, which uses assumptions about the characteristics of the execution context, and cross-layer adaptation mechanisms for preventing SLA violations.

SLAs for third party services utilised in each SBA layer are an important element in such approaches, since SLAs specify the expected characteristics of each third-party service, named Service-Level Objectives (SLOs), to be monitored, and possibly adaptation strategies for compensating or even proactively preventing violations of SLOs. We support the research directions towards the run-time quality assurance of SBAs using CLAM techniques, and we believe that such techniques could greatly benefit from an analysis approach of SBAs for identifying third party services and their characteristics across the SBA layers for the definition SLAs.

To the best of our knowledge, we are aware only of one recent work that is related to the definition of SLAs for CLAM [4]. Although this work describes a methodology for creating SLAs, it focuses on the dependencies between the characteristics of services, and it does not follow the SBA layers as they have been defined in [9, 6]. The authors focus more on how KPIs and IT infrastructure metrics impact the goals of a service user, and they introduce a new indicator named Key Goal Indicator. Our work is different since we present ideas for defining SLAs, by performing analysis of SBAs in order to identify the third party services and their characteristics utilised in each SBA layer.

In the next section, we present insights into how to define SLAs for CLAM by analysing SBAs, in order to differentiate the third party business, software and infrastructure services utilised by an SBA.

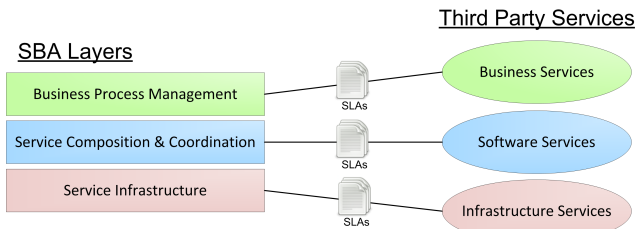
## 3. SBA ANALYSIS FOR DEFINING SLAS

In the context of Service-Oriented Computing, an SLA specifies the exact functionality and the desired quality of service to be delivered by a software service [7]. An SLA includes a set of metrics and a behavioural specification that could be used to determine whether the service provider is delivering the service as agreed. An SLA could also include compensation actions in the event that the agreement was violated. Machine-readable SLAs for the third party services, used in an SBA, are utilised in CLAM. CLAM approaches monitor the third party services for detecting violations of the agreed service characteristics, in order to perform compensation actions across all layers of an SBA.

The three functional layers of an SBA are defined in [6, 9]. The top layer of an SBA is the business process management (BPM) layer and it concerns the business level aspects of an SBA, such as process workflows, service networks, key performance indicators, and process performance metrics. The BPM layer focuses mostly on monitoring business activities and manages the performance of the business. The middle layer of an SBA is the service composition and coordination layer (SCC), which concerns the composition of individual services into new services, the functional (e.g. service behaviour) and non-functional quality of service (QoS) (e.g. responsiveness and availability) characteristics of the individual services or the composed services. The SCC layer focuses mostly on both run-time verification and testing of the service behaviour, and monitoring the QoS of the individual or the composed services. The bottom layer of an SBA is the service infrastructure (SI) layer and concerns the software (e.g. service middleware, service registry) and the hardware (e.g. compute, storage, bandwidth) resources utilised in an SBA.

Based on the description of the SBA layers, we suggest that that each layer concerns different types of services. The BPM layer concerns business services or business activities realised through software services. For instance, a shipping provider exposes a Web services API for shipping goods. The shipment of goods is a business activity provided through a Web service. The SCC layer concerns software services that implement a specific functionality or a business activity. For instance in the case of the shipping provider the Web service is the software service. The SI layer concerns the infrastructure services used by an SBA. For instance, an SBA could be running on a third party Cloud Computing infrastructure and rely on shared computing, storage, and networking resources. Based on the aforementioned suggestions, we argue that an SBA is a software application that outsources business activities, consumes software services, and uses infrastructure services.

Due the fact that BPM, SCC, and SI layers concern the business, the software, and the infrastructure services respectively, and considering that such services in each layer could be provided by third parties, it is necessary to have separate SLAs for all services in each layer. The SLAs in each of the three layers are required for monitoring the conformance of the services to the agreements. Given the possibility that there are dependencies between business, software, and infrastructure services, it is necessary for the monitoring activity to correlate the monitoring results from all services in each layer, such that the real cause of a problem is diagnosed, in order to conclude and enact an effective adaptation strategy.



**Figure 1: The SBA layers concern the quality of third party business, software, and infrastructure services utilised in an SBA.**

In the following section, we present a new case study showing how the analysis of different types of service supports the introduction of CLAM in an existing platform-as-a-service offering, which has been developed as an SBA. The platform is analysed in order to identify the business, software, and infrastructure services and their characteristics for the definition of SLAs for such services in the three SBA layers.

## 4. CASE STUDY

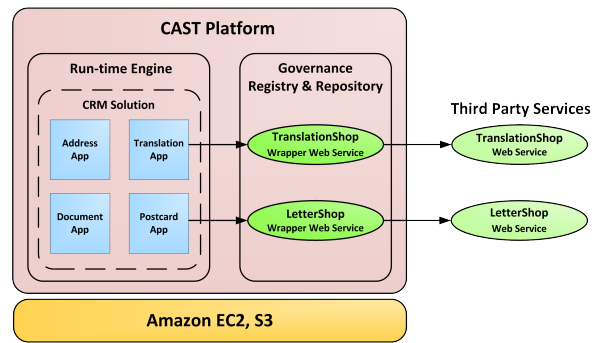
In this section, we present a new case study showing the analysis of an SBA in preparation for CLAM. We provide a brief description of the SBA used in the case study. We continue by identifying the distinct business, software and infrastructure services and their characteristics as a precursor to defining SLAs for these services.

### 4.1 The CAST Platform

The case study concerns a platform-as-a-service (PaaS) offering for enabling the customisation of software-as-a-service (SaaS) applications by third parties, developed during the European project CAST<sup>1</sup>. The platform was developed using Java, OSGI, Web services, and other service-oriented technologies. A brief summary of the CAST Platform follows based on the work [8, 1, 2] carried out during the CAST Project.

The CAST platform offers the foundations for developing software ecosystems of domain specific solutions, which comprise apps and external Web services. A developer is able to create his own solution by implementing new apps or by extending the functionality of existing community apps, which are made available by other developers. In the context of the CAST Platform, a solution is a collection of multiple apps that are the building blocks of the platform for implementing a particular functionality. Figure 2 depicts the case of a Customer Relationship Management (CRM) solution in the CAST Platform. The CRM solution comprises an address app for managing customer addresses, a document app for managing documents, a translation app for translating documents, and a postcard app for sending postcards to customers. An app can interact with external Web services, which have been registered to the Governance Registry & Repository system of the CAST Platform. The main function of the registry is the lifecycle management and quality assurance of solutions, apps, and external Web services. Two examples of apps that use external Web services are the translator app, which uses the TranslationShop Web service for automated or human-expert translation of

<sup>1</sup>CAST project website - <http://www.cast-project.eu/>



**Figure 2: The CRM solution in the CAST Platform, which is deployed on a cloud infrastructure.**

text, and the postcard app, which uses the LetterShop Web service for sending digital and paper postcards to customers.

The provider of the CAST Platform implements a strict quality assurance policy for external Web services used in the platform, in order to minimise the risk of the external services to threaten not only the reliability and performance, but also the reputation and the business value of the platform. Thus, before a Web service becomes available to be used in an app, it has to be registered in the Governance Registry & Repository system. During the registration of a new Web service, the WSDL file of the service and a machine-readable SLA for the service are stored. The SLA is an agreement between the provider of the external service and the platform provider and it specifies the expected response time and availability of the provided Web service.

Due the high costs involved for creating a private owned infrastructure, the platform provider has decided to use a cloud provider, such as Amazon Web Services, for the provisioning of the CAST Platform. The fact that the CAST Platform uses computational, storage, and networking resources provided by a third party raises concerns about the reliability of the underlying third party infrastructure. Thus, the platform provider has an SLA with Amazon for the provided Amazon EC2 and Amazon S3 services, which provide compute, storage, and networking resources to the CAST platform. The expected characteristics of these services that have been included in the SLA comprise the availability, performance, and error rates regarding the I/O requests to these services.

### 4.2 SLAs in the CAST Platform

The CAST platform could clearly benefit by being enhanced with CLAM. This would support more appropriate, and more timely checking of SLA violations, and the triggering of more suitable and better-coordinated adaptations in the BPM, SCC, and SI layers, whose effects would therefore be much less likely to interfere with each other across the layers. The platform provider will need to identify the external business, software, and infrastructure services and their characteristics, which could affect the quality of the platform. The characteristics could be used for defining SLOs to be included in SLAs for these services. Table 1 summarises the business, software, and infrastructure services that have been identified in the CAST Platform. The table shows the also the identified characteristics of each service that could affect the quality of the platform.

**Table 1: The results of the analysis show the identified business, software, and infrastructure services and their characteristics identified in the CAST Platform.**

SBA Layer	Service Type	Services	Service Characteristics	Service-Level Objectives
BPM	Business	automated translation of text	automated translations processed per day, average time for an automated translation	automated translations processed per day > 50, average time for an automated translation ≤ 5min
		manual translation of text	manual translations processed per day, average time for a manual translation	manual translations processed per day > 10, average time for an manual translation ≤ 1 hour
		send digital postcard	digital postcards sent per day, average delivery time of digital postcards	digital postcards sent per day ≥ 300, average delivery time of digital postcards ≤ 15min
		send paper postcard	paper postcards sent per day, average delivery time of paper postcards	paper postcards sent per day ≥ 50, average delivery time of paper postcards ≤ 3 days
SCC	Software	TranslationShop Web service	behavioural conformance, average response time, hourly availability, average error rate per hour	average response time ≤ 350ms, hourly availability ≤ 98%, average error rate per hour ≤ 0.05
		LetterShop Web service		
SI	Infrastructure	Amazon EC2/S3	average time to provision a resource, hourly availability of resources, average storage I/O per minute, average storage error rate per minute	average time to provision a resource ≤ 5min, hourly availability of resources ≥ 99%, average storage I/O per minute ≥ 10 million, average storage error rate per minute ≤ 0.02

In the BPM layer, two business services were identified. Two separate SLAs are required between the platform provider and the two service providers of the TranslationShop and LetterShop services. These services realise automated activities such as machine translation or emailing digital postcards, and manual activities that require a human to perform a task, for example, human-expert translation or posting of paper postcards. Therefore, the SLAs should contain agreements on characteristics of both automated and manual activities. The two SLAs will include agreements on the key performance indicators (KPIs) and process performance metrics (PPM) of the two services. A KPI for both services could be the customer satisfaction, which could be calculated from the ratings provided by end users of the two services. The PPMs for the TranslationShop service could comprise the number of automated and manual translations processed per day, and the average time to complete a machine or manual translation. The PPMs for the LetterShop service could comprise the number of digital or paper postcards sent per day, and the average delivery time for digital or paper postcards.

In the SCC layer, two software services were identified. Two separate SLAs are required between the platform provider and the two service providers of the TranslationShop and LetterShop services. The two SLAs will contain agreements on the technical characteristics of the TranslationShop and LetterShop services. The technical characteristics comprise some functional and non-functional elements of the two external services. The functional characteristics could comprise the behavioural specification of each service, while the non-functional characteristics could comprise the average re-

sponse time, the hourly availability, and the average error rate per hour.

In the SI layer, one infrastructure service was identified. Only one SLA is required between the platform provider and the Amazon Web Services. This SLA will contain agreements for the characteristics of the infrastructure services provisioned by Amazon. The characteristics of the infrastructure could comprise the average time to provision a resource (e.g. a virtual machine or more storage space), the hourly availability of resources, the average storage I/O per minute, and the average storage error rate per minute.

The identified SLAs for the third party business, software, and infrastructure services utilised in the CAST Platform will be used for implementing CLAM. Existing SLA frameworks for Web services, such as WSLA<sup>2</sup> or WS-Agreement<sup>3</sup>, could be employed for representing machine-readable SLAs. Each SLA could include a subset of an adaptation strategy to be used during the generation of the cross-layer adaptation strategy. For instance, a subset of an adaptation strategy for the TranslationShop service in the SLA of the BPM layer could comprise that in the event of a dramatic increase of the average time of manual translation, the manual translation business process in the Translation App will have to be adapted, such that it will use the automated translation function of the TranslationShop service, in order to provide a quick low quality translation, while waiting for the manual translation of better quality to arrive at a latter time. CLAM approaches similar to the efforts in [11], will be re-

<sup>2</sup><http://www.research.ibm.com/wsla/>

<sup>3</sup><http://www.gridforum.org/documents/GFD.107.pdf>

quired for correlating the observations of the monitoring activities, deciding effective adaptation strategies, which will take into consideration the subsets of adaptation strategies provided in the SLAs, for enacting effective adaptations in the CAST Platform for preventing or responding to SLA violations.

## 5. CONCLUSION

In this paper, we have suggested that each SBA layer concerns different types of services used in an SBA. Our view is that the business process management (BPM) layer is concerned with business services, the service composition and coordination (SCC) layer is concerned with software services, and the service infrastructure (SI) layer is concerned with infrastructure services. We demonstrated that a CLAM approach requires a clean separation of these types of service, such that different kinds of SLA may be drawn up with different providers in the service chain. We have focused on the definition of SLAs in the BPM, SCC, and SI layers, for cross-layer adaptation and monitoring of SBAs. We have presented insights into how an SBA should be analysed, in order to identify and separate the distinct business, software and infrastructure services. We applied this technique to a new case study, based on an existing platform-as-a-service offering, which was chosen as an example for CLAM. Each of the services identified in the study was then analysed to reveal its individual characteristics, prior to drawing up appropriate SLAs. The study clearly demonstrates the utility of separating the run-time quality assurance concerns at each layer of the SBA.

We are currently investigating existing methods for representing SLAs for the business, software, and infrastructure services. As future work, we plan to extend previous work [3] related to the implementation of an extensible monitoring architecture for Web services, in order to support the development of a CLAM framework for SBAs that will use multiple SLAs for business, software, and infrastructure services.

## 6. ACKNOWLEDGMENTS

We cordially thank Dr. Iraklis Paraskakis and Dimitrios Kourtesis from the South-East European Research Centre for providing information concerning the CAST Platform.

We would like to thank the anonymous reviewers for their constructive and detailed comments on improving the paper.

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement 215483 (S-Cube).

## 7. REFERENCES

- [1] *Deliverable D1.1 Requirements Analysis*. CAST Project, 2009.
- [2] *Deliverable D2.1 CAST Platform Architecture*. CAST Project, 2010.
- [3] K. Bratanis, D. Dranidis, and A. J. H. Simons. An extensible architecture for run-time monitoring of conversational web services. In *Proceedings of the 3rd International Workshop on Monitoring, Adaptation and Beyond / ECOWS 2010*, pages 9–16. ACM, 2010.
- [4] M. Fugini and H. Siadat. SLA contract for Cross-Layer monitoring and adaptation. In S. Rinderle-Ma, S. Sadiq, and F. Leymann, editors, *Business Process Management Workshops*, volume 43, pages 412–423. Springer Berlin / Heidelberg, 2010.
- [5] E. Gjørven, R. Rouvoy, and F. Eliassen. Cross-layer self-adaptation of service-oriented architectures. In *Proceedings of the 3rd workshop on Middleware for service oriented computing, MW4SOC '08*, pages 37–42. ACM, 2008.
- [6] R. Kazhamiakin, M. Pistore, and A. Zengin. Cross-layer adaptation and monitoring of service-based applications. In *Proceedings of the 2009 international conference on Service-oriented computing, ICSOC/ServiceWave'09*, pages 325–334, Berlin, Heidelberg, 2009. Springer-Verlag.
- [7] C. Kotsokalis, editor. *D.A5a SLA Management Foundations*. SLA@SOI, 2010.
- [8] D. Kourtesis, V. Kuttruff, and I. Paraskakis. Optimising development and deployment of enterprise software applications on PaaS: the CAST project. In *ServiceWave 2010 Workshop Proceedings*. Springer Berlin / Heidelberg, 2010.
- [9] M. Papazoglou, K. Pohl, M. Parkin, and A. Metzger, editors. *Service research challenges and solutions for the future internet: S-cube - towards engineering, managing and adapting service-based systems*. Springer-Verlag, Berlin, Heidelberg, 2010.
- [10] R. Popescu, A. Staikopoulos, P. Liu, A. Brogi, and S. Clarke. Taxonomy-Driven adaptation of multi-layer applications using templates. In *Proceedings of the IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, pages 213–222, Los Alamitos, CA, USA, 2010. IEEE Computer Society.
- [11] E. Schmieders, A. Micsik, M. Oriol, K. Mahbub, and R. Kazhamiakin. Combining SLA prediction and cross layer adaptation for preventing SLA violations. In *Proceedings of the 2nd Workshop on Software Services: Cloud Computing and Applications based on Software Services*, Timisoara, Romania, June 2011.