Modeling Accountable Cloud Services Based on Dynamic Logic for Accountability

Jun Zou, Department of Computing, Macquarie University, Sydney, Australia
Yan Wang, Department of Computing, Macquarie University, Sydney, Australia
Mehmet A. Orgun, Department of Computing, Macquarie University, Sydney, Australia

ABSTRACT

Cloud computing services have been increasingly considered by businesses as a viable option for reducing IT expenditure. However, there are often associated problems with unmanaged accountability. This paper first analyses the accountability properties of a cloud service and then proposes the accountable cloud service (ACS) model to address those problems. In addition, the authors argue that from an accountability perspective a cloud service is a proactive system that needs to be modeled differently from the traditional reactive systems. They extend traditional structural operational semantics to cater for modeling of actors as well as scenarios of inaction and exception in state transitions. This leads to the creation of a new form of a process algebra called Accountable Process Algebra (APA). They also propose an Obligation Flow Diagram (OFD) as a simple method for conflict resolution and verification for the ACS model. The ACS model enables accountability in cloud service consumption. Using Amazon S3 service as a case study, they show how to address those known accountability problems by using our ACS model. Finally the authors discuss the applicability of their model to cloud services in general.

Keywords: Accountability, Accountable Cloud Service (ACS), Accountable Process Algebra (APA), BPMN2.0, Dynamic Logic, Proactive System, Process Algebra

1. INTRODUCTION

1.1. Background

Recently cloud computing has emerged as a new business model that turns IT capabilities into e-services. The sustainability of this business model largely depends on whether the accountability of cloud services can match that of the traditional IT services. By accountability, we mean a clear disclosure of service obligations; faithfully honoring disclosed obligations, or otherwise assuming the liability for the unsatisfactory performance of the obligations (Zou, Wang & Lin, 2010). Traditionally, accountability in service is achieved through the enforcement of a legal and paper-based contract. In a cloud service context, using a paper-based contract is no longer effective. The current practice is for service providers to publish a terms and
conditions page and a text-based SLA for their offerings on their web-site like Amazon’s S3 (see https://aws.amazon.com/en/s3-sla) does. This form of contract is called a web-enabled paper-based contract. In its plain-text form, a web-enabled paper-based contract can neither be interpreted by software agents, nor used as a basis for monitoring the execution of a contract. Although policy management tools, such as Ponder and K Ao S, can be adopted to represent obligations in a formal manner, their strengths lie in specifying policies on fine-grained objects, lacking generality (Phan et al., 2008) and an overarching process view that is required in dealing with service participants’ obligations in service collaborations.

While Service Level Agreement (SLA) is an extensively researched topic and it can be represented by existing approaches such as WSLA, WS-Agreement, SL Ang (Skene, Raimondi, and Emmerich, 2009), in essence, the service level only covers non-functional requirements, missing the crucial functional requirements for business. Thus, the existing approaches neither enable the disclosure of service obligations, nor allow a software agent to decide which party is responsible for what action, and which party is liable for what result. This is evident in today’s cloud market, where there are no formal policy based or SLA languages used in representing service contracts. Hence the consumer has no effective means to detect the violation of service obligations, and service provider can hardly be held accountable. As such, currently the accountability of cloud services on the market is a serious concern. This may become a major obstacle for enterprise customers to take up those cloud services. Therefore it is critical to build an accountability model for cloud services.

Such an accountability model should provide:

1. A representation language that is declarative and machine interpretable so that the obligation statements in a cloud service contract can be checked and interpreted automatically;
2. A collaboration process and diagram notation that clearly illustrate the interaction behavior between the service provider and the consumer, and therefore facilitating reasoning about obligation fulfillment and liability assignment; and
3. A tool for enabling the validation of the consistency of the model.

We call this model an accountable cloud service (ACS) model.

1.2. A Motivating Example

We here use a traditional storage service and a typical cloud service - Amazon S3 storage service as an example to compare and contrast different accountability issues. Suppose a service provider offers a data storage managed service to enterprise consumers. The regular process for this kind of an off-line service provision is described as follows:

1. The provider prepares a statement of work (SOW) and a service-level agreement (SLA) based on the requirements submitted by the consumer. The SOW describes the rights and responsibilities of the provider and the consumer respectively, specifying each party’s permitted activities, obligated activities including deliverables and acceptance criteria. For instance, the provider’s permitted activities may include delivering data in either plain or compressed format; obligated activities can be storage and retrieval of data objects while making sure that the data are not lost, damaged or leaked to other people. The SLA stipulates the committed service levels for the service activities and deliverables, i.e., a 99.9% availability of the overall storage service. Sometimes penalty clauses may be attached if an obligation is violated, for example, not achieving the 99.9% of availability target.
2. The provider negotiates with the consumer on the SOW and SLA. If a deal is reached, normally an overarching contract is signed between the provider and the consumer,
Related Content

A Fast Shapelet Discovery Algorithm Based on Important Data Points
Cun Ji, Chao Zhao, Li Pan, Shijun Liu, Chenglei Yang and Lei Wu (2017).
International Journal of Web Services Research (pp. 67-80).
www.igi-global.com/article/a-fast-shapelet-discovery-algorithm-based-on-important-data-points/181300?camid=4v1a

Interoperability Among Heterogeneous Services: The Case of Integration of P2P Services with Web Services
Aphrodite Tsalgatidou, George Athanasopoulos and Michael Pantazoglou (2010).
Web Services Research for Emerging Applications: Discoveries and Trends (pp. 390-421).
www.igi-global.com/chapter/interoperability-among-heterogeneous-services/41531?camid=4v1a
Early Capacity Testing of an Enterprise Service Bus
[www.igi-global.com/article/early-capacity-testing-enterprise-service/34106?camid=4v1a](www.igi-global.com/article/early-capacity-testing-enterprise-service/34106?camid=4v1a)

Mining Lifecycle Event Logs for Enhancing Service-based Applications
[www.igi-global.com/chapter/mining-lifecycle-event-logs-enhancing/69474?camid=4v1a](www.igi-global.com/chapter/mining-lifecycle-event-logs-enhancing/69474?camid=4v1a)