Towards a Variability Modeling Approach for Service-Oriented Cloud Architectures

Mohammad Abu-Matar
Eitsalat British Telecom Innovation Center
Khalifa University of Science, Technology and Research
Abu Dhabi, United Arab Emirates
Mohammad.abu-matar@kustar.ac.ae

Abstract— Cloud computing has emerged as a model for distributed computing that promotes on-demand scalability, flexible application deployment and reuse. Software product lines (SPL) promote reusable application development for product families. Cloud systems change to respond to changing clients’ requirements. Cloud systems can be modeled as Software-as-a-Service (SaaS) families similar to the SPL concept. Some SPL development techniques rely on feature models to describe the commonality and variability of member applications. In this position paper, we extend our previous work, on service-oriented variability modeling, to introduce a platform independent approach to model cloud systems variability based on feature modeling and SPL. In addition, our approach is accompanied by a meta-model that formalizes the multiple views of cloud systems. We believe that such an approach facilitates variability modeling of cloud software families in a systematic and platform independent way.

Keywords— Cloud Computing, Software as a Service, Software Product Lines, Service Oriented Architecture, Variability Modeling, Service Families.

I. INTRODUCTION

Cloud computing has emerged as a model for distributed computing that promotes on-demand scalability, flexible application deployment and reuse [1]. One of the major benefits claimed for cloud computing is the flexible deployment of IT solutions that can react to changing business requirements quickly and economically. The cloud computing ecosystem consists of service providers, service consumers, and service brokers [2].

Cloud services are consumed by many clients that have different requirements. Thus, clients that consume the same services usually exhibit varying requirements needs. Varying requirements usually necessitate varying software architectures. In other words, when applications’ requirements are changed, the software architectures of these applications are modified to satisfy the changed requirements. Therefore, both requirements and architectures have intrinsic variability characteristics.

Software architecture can describe application designs from different perspectives [3]. In other words, the same application architecture consists of multiple depictions of different perspectives, also called views, that address specific architectural concerns. Managing variability in cloud systems is complicated by the presence of interrelated service views in these systems. A change in one view necessitates a change in other views. Therefore, manual development of variable cloud systems is unwieldy and error-prone. Some of cloud services variability types include:

- **Functional Variability** – different types of application domains, i.e. Software as a Service (SaaS).
- **Platform Variability** – Operating systems, Programming languages, Frameworks, and Libraries, i.e. Platform as a Service (PaaS).
- **Provisioning Variability** – Hardware, Networking, Virtual Machines, and Elasticity, i.e. Infrastructure as a Service (IaaS).
- **Deployment Variability** – Public, Private, Community, and Hybrid clouds [2].
- **Provider Variability**, i.e. Amazon Web Services (AWS), Google Application Engine (GAE), and Salesforce to name a few.

Software Product Lines (SPL) are families of software systems that share common functionality, where each member has variable functionality [4]. The main goal of SPL is the agile and rapid development of family members by using reusable assets from all phases of the development lifecycle. This goal is similar to the goal of Service Oriented Architecture (SOA), which underpins cloud computing, where agile and flexible application development is a common theme. SPL and Commonality/Variability Modeling model the variability of application families. Since cloud systems vary in much the same as application families, i.e. they have common and variable features; this paper describes the use of SPL concepts to model the variability concerns of cloud systems by extending our previous work on multiple-view variability modeling of SOA systems [5].

This position paper describes an approach for the modeling of variable service-oriented cloud product lines. The approach integrates feature modeling [4], [6] with service views using UML and SoaML [7], the service oriented standardized modeling language. Such an approach facilitates variability modeling of cloud service family architectures in a systematic and platform independent way. At the heart of the approach is a meta-model [8] that describes variability in requirements and architectural meta-views of service oriented systems.
The rest of the paper is structured as follows. Section 2 motivates the need for this research. Section 3 briefly provides background information about variability modeling in SPLs. Section 4 describes our research approach at a high level, section 5 discusses related work, and section 6 concludes the paper.

II. MOTIVATION

In SOA, service providers are usually decoupled from service requesters, thus requesters and providers vary independently of each other. This variation manifests itself in several ways, i.e. in changing requirements, changing architectures, and changing execution environments. Architectures change because in SOA, the architecture is not fixed, because the main elements of the architecture are services usually provided by external providers.

The advent of cloud computing necessitated even more variability in the form of service types, deployment types, and different cloud participant roles. Thus, variability modeling is necessary to manage the inherent complexity of service-oriented cloud systems.

It is hard to model complex and reasonably sized software systems from one perspective, e.g., structural. To manage this complexity, the software engineering community has used multiple-view modeling to model software systems from different perspectives [10]. In essence, the same application can be modeled from different perspectives, where each perspective models a specific concern, e.g. requirements, architectures, and physical environments. Multiple-view modeling techniques can be applied to model variability concerns of service cloud systems for the same reasons mentioned above. Cloud systems can be segregated into multiple views such as contract/business workflow requirements views, service interface/coordination architectural views, provisioning views, and deployment views. It should be noted, that these views depend on each other. In other words, a change in one view could necessitate a change in a different view.

For example, a service-oriented cloud E-Commerce system could have a view that describes service contracts, service providers, and consumers. Another view could be a business process view that describes the workflow of order fulfillment. Yet another view could be a service interface view that describes an Ordering service’s operations and parameters. Furthermore, another view could be a deployment view that specifies a public or private deployment model. If a task changes in the business process view, say a task is added to allow electronic check payments; a new service interface for electronic check payment has to be added to the service interface view. Likewise, if a credit check contract is added in the service contract view, a credit check service provider gets added to provide this capability. For reasonably sized applications, changes in the interdependent views of service cloud systems can quickly become unwieldy and difficult to manage.

It becomes evident from the aforementioned discussion, that requirements and architectural variability concerns are dispersed among the multiple views of cloud systems. To have a full picture of variability in service cloud systems, it would be necessary to have one view that only describes variability in the entire system. However, current cloud services development practices lack a systematic approach for managing variability and are typically platform-dependent.

III. MODELING VARIABILITY IN SPLS

In model-based software design and development, models are built and analyzed prior to the implementation of the system, and are used to direct the subsequent implementation. A better understanding of a system or SPL can be obtained by considering the multiple views [9, 10] such as requirements models, static models, and dynamic models of the system or SPL. A key view in the multiple views of a SPL is the feature modeling view [4, 6], which explicitly models SPL commonality and variability in terms of features, which are requirements or characteristics that differentiate among product line members.

A SPL development process has two main processes, as depicted in Fig. 1: a) Domain Engineering. A SPL multiple-view model, which addresses the different views of a SPL, is developed. The SPL multiple-view model, SPL architecture, and reusable components are developed and stored in the SPL reuse library. b) Application Engineering. The application developer selects the required features for the individual SPL member. Given the features, the SPL model and architecture are adapted and tailored to derive the application architecture.

The proposed research in this position paper extends our previous work on multiple-view SOA variability modeling in which the unifying view is the feature model. In particular, feature modeling provides the added dimension of modeling variability in cloud SPL. The approach uses the multi-view service variability modeling approach advocated by our earlier work [5, 9], but defines four new views and meta-views which are needed for cloud product lines. The new cloud views are: the Service Provisioning and Topology views for requirements modeling, and the Service Mediation and Deployment views for architectural modeling. Fig. 2 lists the aforementioned views.
Current cloud services development practices lack a systematic approach for managing variability and are typically platform-dependent. Furthermore, existing cloud services modeling approaches do not address the multiple-view nature of variability in cloud systems.

IV. PROBLEM STATEMENT AND RESEARCH APPROACH

This research addresses the lack of systematic approaches to handling variability modeling concerns in cloud service-oriented systems by developing a multiple-view variability modeling and meta-modeling approach. Furthermore, this research addresses cloud services variability modeling concerns in a systematic and platform-independent way. The high-level research approach is summarized as follows:

1. Extend the multiple-view variability model [5] with new views relevant to cloud systems (Fig. 2), and define the relationships between the multiple views. Specifically, the Provisioning and Mediation views model the elastic characteristics of cloud systems. In addition, the Topology and Deployment views model the deployment options of cloud systems, i.e. Public, Private, and Hybrid. This multiple-view model uses Feature Modeling to construct a unifying view that describes variability concerns, which are dispersed in the multiple cloud views.

2. Extend the multiple-view service variability meta-model [8] that formalizes the multiple-view variability model in the previous step. A meta-model is needed to serve as the underlying representation of the multiple-view model described in the previous step. Fig. 3 is a high-level depiction of the service variability meta-model that will be extended. The meta-model depicts how features in SPLs are mapped to service views and how the service views are related.

3. Develop rules for consistency checking and mapping between the multiple views in the meta-model to ensure that these views are consistent with each other as cloud service systems change.

4. Extend our Service Oriented Architecture Product Line Engineering (SoaSPLE) framework [8] and tool prototype to include the new cloud based views.

5. Validate this research by applying the multiple-view modeling approach and proof-of-concept prototype to case studies.

V. RELATED WORK

The authors in [11] employ aspect-oriented software...
treats variable cloud systems in a multiple-view variability.

The authors in [12] use SPL and Feature Modeling to create a decision support tool to help cloud providers manage the different application requirements of SaaS systems. Unlike our approach, the approach in [12] is platform-dependent and does not offer any multiple-view architectural treatment of cloud services variability.

The authors in [13] discuss customization opportunities for cloud-based software product lines. They list advantages and disadvantages of software customization approaches for the three prominent layers of cloud computing, i.e. IaaS, PaaS, and SaaS. While the authors present no specific SPL treatment to realize the customization problem, they merely compare approaches and conclude that PaaS is the most cloud layer that could benefit significantly from SPL approaches.

The authors in [14] argue that because of the intrinsic heterogeneity of cloud based systems, SPL approaches are suitable for cloud-based development. The authors present SPL techniques to aid in the variability management of cloud systems. Specifically, feature models are adapted to cater for specific cloud computing concerns like provisioning, customization, and price calculation. The work in [14] is the closest to our work in that it exploits feature models to manage the variability of cloud systems. However, our approach differs in that it uses feature models as a unifying view to describe variability in multiple views of cloud systems, and provide mapping, based on meta-models, between features and cloud services artifacts.

To the best of our knowledge, there exists no research that treats variable cloud systems in a multiple-view variability modeling and meta-modeling approach.

VI. CONCLUSION

In this position paper, we have introduced a multiple-view modeling approach that addresses service-oriented cloud systems variability concerns in a unified manner. In particular, we have described the integration of SPL concepts of feature modeling and commonality/variability analysis techniques with service modeling approaches (such as SoaML) to model cloud systems variability. We believe that our approach has several benefits:

- Treatment of cloud systems variability concerns in a unified, systematic, multiple-view manner.
- A Multiple view meta-model for describing cloud based software product lines.

- Facilitates variability modeling of cloud service families in a platform independent way.
- The use of established SPL Feature Modeling to manage variability in the Cloud.

Future research will elaborate our work by utilizing and extending our SoaSPLEx framework. We will add the new cloud-based Requirements and Architectural views, mentioned above, to our meta-model. We will be augmenting our meta-model consistency rules, using the Object Constraint Language (OCL). In addition, we will extend the SoaSPLEx tool prototype [5] that automates the mapping of features in feature models to cloud service elements in cloud environments. Finally, we plan to validate our approach by conducting industrial case studies.

REFERENCES