EAI Endorsed Transactions
on Scalable Information Systems

Research Article EAI.EU

Notational Modeling for Model Driven Cloud Computing using Unified Modeling Language

Muhammad Faisal, Shoaib Nazir, Muhammad Babar*

Iqra University, Islamabad, Pakistan
faisal.khan16@hotmail.com, shoaibnazir100@gmail.com, muhammad.babar@iqraisb.edu.pk

Abstract

Cloud Computing and Model-Driven Engineering (MDE) are two of the most dominant paradigms nowadays. Models are considered as major parts in the MDE, and concentrates on the formation and then conversion of models to the implementation. Cloud Computing is also getting reputation as the standard approach for designing and organizing software applications over the internet, especially for distributed and e-commerce applications. In recent times, Cloud Computing has become known as a latest opportunity that how software and other resources can be provided to the consumers as a service. The cloud aspects need to be well engineered for the software engineering methodologies particularly modeling the cloud aspects to provide logical tested solution prior to implementation. On the other side, general-purpose language UML, provides modeling and designing notions to symbolize software, platforms and architectural artifacts from diverse viewpoints of object-oriented paradigm. UML can also be extended to model and visualize the non objectoriented systems. Lately, few cloud modeling methodologies have emerged, however, useful support for designing cloud application is still missing. As a result, we propose UML-based framework using UML extension mechanism for modeling cloud computing paradigm aspects.

Keywords: Cloud Computing, Model-Driven Engineering, Unified Modeling Language

Received on 15 October 2017, accepted on 11 January 2018, published on 13 April 2018

doi: 10.4108/eai.13-4-2018.154475

1. Introduction

Cloud computing has brought the innovative period of potential computing, change a huge branch of IT trade, redesign the buy and use, and get significant consideration worldwide and local society of IT, national and intercontinental organizations [3] [12]. Cloud computing is a processing model where IT capital are offered as services to users. Cloud computing provides new-fangled potentials for software engineering researchers to advance the software development process with regard to multilateral approach. Generally, there are 2 possibilities to appreciate the prospective association between software engineering and cloud; (i) to improve the software development methods to suite designing software applications for the cloud (ii) the utilization of cloud to support the process of software development [1]. Cloud computing benefits software engineering concepts like agility, availability and cost efficiency. These need to be well engineered for cloud platforms using the software engineering methodologies particularly modeling the cloud aspects to provide logical tested solution prior to implementation to improve the quality. In addition, cloud computing also plays a very important role in the Big Data analytics and smart cities planning [21]. MDE treats models as major parts, and concentrates on the formation and then conversion of models to the implementation. In most of the engineering disciplines, models are used to offer generalization and abstraction [1]. They are utilized for various functions such as: providing logical solution, documenting the software artifacts, validation and testing, static analysis, and for the generation of code automatically. The models assist in representing the problem and its solution (logically) in a methodical mode. They also demonstrate the precise details for different viewpoints and at various stages of development [2]. MDE techniques facilitates the designers to model the cloud aspects and automate or partially automate the development of cloud aspects particularly SaaS applications [2]. Cloud computing is an attractive and very appealing domain for MDE [2]. Object Management Group (OMG) is a syndicate, responsible for setting standards for modeling the different business processes, programs, systems, and other model-based standards. OMG defines a number of modeling languages as standard, among which Unified Modeling Language (UML) is probably the one most extensively accepted and used. The UML provides a standard method to mark the project drawing that includes theoretical parts such as the business processes and system functions. Moreover, it also focuses on the design and modeling of concrete things of a specific system such as the statements of programming language, the schemas of database, and the software components that can be reused [7]. However, being a general purpose language, it lacks to model and design the elements of specific systems (other than Object-Oriented). To model the elements of such system, UML provides the concept of UML profiling. The light weight extension of UML is called UML profile. It is used to specify the peculiarities of the modeled system. UML profiles are formulated by means of UML extension method for systems other than object-oriented. The goals of modeling and design of cloud paradigm, are to support all the activities of cloud modeling and design to fit into an overall model-based development from both business and IT perspectives. Unfortunately, there is not yet a consensus on the right set of models, modeling languages, model transformation and software process for model-driven cloud paradigm. To date, comparatively slight research has been done with regard to the utilization of MDE to support the development of cloud applications in a better way. The possible ways towards the model-based development of cloud paradigm could be UML profiling (extending UML using UML standard mechanism) and domain specific modeling language (defining specific meta-model like UML). We believe MDE can bring many benefits to cloud computing; in this sense we would like to propose UML-based models for modeling and designing the cloud aspects. The rest of the paper is organized as follow: section 2 illustrates the detail literature review, section 3 introduces different service models of cloud computing, section 4 details the proposed work, and finally section 5 include the conclusion and future work.

2 Literature Review

In this section the comprehensive and detail literature review is given. It includes two parts; first part gives the background of the cloud computing and MDE, and the second part highlights some related work for the modeling of cloud computing paradigm.

2.1 Background

Cloud computing and MDE could be associated in a number of ways [2]:

- MDE use for the providers, where it is utilized to design the cloud-based software and cloud environment.
- MDE utilization for consumers, where MDE techniques can be used to access/consume cloud services.
- MDE use in the cloud, where the cloud is utilized to provide models which is known as Maas (modeling as a service)
- MDE for cloud providers, where MDE techniques are utilized to generate code and offer code generation as a service [9].

MDE can manage variability in the cloud. One of the possible strength of MDE is the capability to abstract and deal with variability [5] [6]. According to [1], the cloud computing can benefited with MDE in the following ways:

- Model-management services could be offered in the cloud with the help of model-based development.
- The scalability can be easily ensured with the help of modeling and designing the services in the cloud.
- Facilitating model execution and evolution in the cloud. The models can be transformed into code and the code-generation services to the cloud will assist the use and growth of cloud aspects particularly SaaS applications.
- MDE in cloud can solve the inter-operability issues in cloud computing.
- MDE can facilitate the global model management. By default the cloud is a very complex environment. Complex structure can be categorized into different models and MDE can assist the handling of all the model artifacts in a scattered setting.

Furthermore, the author [2] also identifies the following opportunities of the MDE in the cloud computing paradigm.

- Run time modification
- Support economies of cloud
- Enabling new business opportunities in the cloud

Unified Modeling Language (UML) is a modeling language to visualize, identify, build, and document the artifacts of a system [7]. UML is general purpose language. UML is used to document and model the elements of object-oriented system for the products visualization. One of the most important strength of UML is that it provides light weight extension mechanisms that permit the language to be tailored to other various systems, domains, methods, paradigms and processes [14]. UML can be customizing for a specific platform (J2EE, .NET) or domain (e.g., aerospace, healthcare, financial), through a proper and standard way [7]. The light weight extension of UML is called UML profile. It is used to indicate the peculiarities of the system that is to be modeled. UML profiles are designed using expansion method provided by OMG. The UML extension mechanism is composed of the stereotypes, constraints and tagged values.

2.2 Related Work

As far as the related work is concerned, there are a few approaches have been presented with regard to modeling and design of cloud aspects. M Abu Matar et al has surveyed that there has been extremely little research found on the support and utilization of MDE in the cloud applications [10]. It has been noticed that even modeling the software processes can also be made available via clouds [11]. They have not proposed anything related to the modeling the aspects of cloud. Moreover, a domain specific modeling language is proposed to describe the MDE services to generate repositories [16]. Furthermore, there are a number of researches found which talk about the utilization of modeling and MDE in the cloud paradigm. A Bergmayr et al highlights the needs of MDE in cloud and also describe its manifesto [15]. Similarly, a model is proposed which offers a foundation for an architecture-driven modeling method for designing cloud-based software [18]. Likewise, work has been taken into consideration to bring the MDE and cloud together and to discover the strategies that would permit a wider use of MDE techniques in the field of cloud computing [20]. Even the MDA concepts are also utilized at implementation level for RESTfull Services development and deployment [17]. In addition, how to model the message passing between different models is also presented but it lacks modeling the technical aspects of the cloud paradigm [4]. G C Silva et al presented another domain specific modeling language [13]. But the presented language is only used for supporting cloud portability and did not provide modeling notations and concepts for modeling the technical cloud aspects. The most relevant modeling approached presented by A Bergmayr et al [8]. They proposed a specific modeling language called Cloud Application Modeling Language (CAML) to support the cloud deployment. The CAML is used only to facilitate the deployment and lack the other technical aspects. Similarly, a UML profile is proposed but that is only focusing on how UML can be extended for the cloud application [19]. However to conclude, comparatively slight research has been done to utilize the MDE that can best hold up in the development of cloud applications. The possible ways towards the model-based development of cloud paradigm could be UML profiling (extending UML using UML standard mechanism) supported by a huge number of CASE tools. We believe MDE can bring many benefits to cloud computing; in this sense we would like to propose UML-based models for modeling and designing the cloud aspects.

3. Cloud Computing Service Models

The cloud provides different IT resources to the consumers as services. The services are categorized into three different categories also known as cloud service models. These models are describes below:

3.1 Infrastructure as a Service (IaaS) Model

IaaS is the basis of cloud computing and the lowest tier. In this model the IT infrastructure is provided as a service to the consumers. The consumers need not to purchase, lease, maintain and deploy the complex IT infrastructure. The IT infrastructure may include the physical resources such as computer servers, CPU cores, storage, and data transfer and network. As a replacement for having to procure hardware utter, consumers can acquire IaaS based on utilization, alike to electricity or other utility billing. In this model only the infrastructure is provided as a service, the rest of the IT resources are the responsibility of the consumers such as building the platform (Virtual Machines, Operating Systems and web servers etc) and then the application development

3.2 Platform as a Service (PaaS) Model

PaaS model is one tier above than IaaS model. In this service model of the cloud computing, the cloud providers also provide the platform as a service addition to the physical resources. PaaS model provides a platform to the consumers to design, develop, deploy, and control the software without the difficulty of constructing and sustaining the platform. Platform may include the VM, OS, DBMS, runtimes, security and integration and other web servers. In this model the user is ready to develop and maintain only the applications without the overhead of platform and infrastructure

3.3 Software as a Service (SaaS) Model

SaaS is the top tier of service models of cloud computing. In this model with addition to infrastructure and platform, the applications are also provided as service. This is the model where the consumers get the readymade applications without the overhead of infrastructure, platform and application development and maintenance. It is a software distribution model in which the cloud provider hosts software and makes them available to the users. SaaS is actually a means of distributing applications via Internet as a service. Rather than installing and maintaining the applications, user just simply access it over the Internet, releasing themselves from complex and complicated software and hardware management.

4 Proposed Framework

In this section, the proposed work is elaborated in detail. We propose extension of different UML models to model, visualize and symbolize the aspects of the service models of cloud computing.

4.1 UML Representation for Infrastructure as a Service (IaaS) Model

In this part, extensions for the elements of IaaS model are proposed and explained. We propose extension of UML specification level deployment diagram to visualize the elements of IaaS model.

Use of UML Specification Level Deployment Diagram

The UML Specification Level Deployment Diagram is used to represent the IaaS model of cloud computing. Deployment diagram is used to identify the physical elements of a system where the software elements are deployed in the ObjectOriented environment. This diagram is used for IaaS model of cloud computing because the IaaS model allows organizations to outsource the physical computing equipment and resources such as server, storage etc. on the other hand the deployment diagram in ObjectOriented environment is also used for the representation of the physical deployment of the system. The deployment diagram is composed of different nodes where the node represents specific physical equipment or resource. In the IaaS model of cloud computing, the node will be representing the infrastructure unit or the resource that is going to be on the cloud to provide as a service. The stereotype definition and description of the PhysicalResource is given in table 1. There could be the possibility to communicate two or more nodes such as a server communicating with other servers etc. This connection and association between these equipments are denoted with a simple straight line. Moreover, the different PhysicalResource examples are shown in the fig 1.

Table 1–PhysicalResource Stereotype Description

NAME	PhysicalResource
UM MODEL ELEMENT	node
DESCRIPTION	representations of this stereotype represents physical equipments in IaaS model
CONSTRAINTS	A node can only be associated with another node

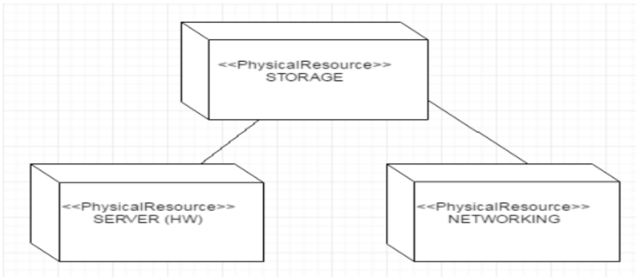

Fig 1–Stereotype of PhysicalResource

4.2 UML Representation for Platform as a Service (PaaS) Model

In this part, extensions for the elements of PaaS model are proposed and explained. We propose extension of UML instance level deployment diagram and component diagram to visualize the elements of PaaS model.

Use of UML Instance Level Deployment Diagram

The UML Instance Level Deployment Diagram is used to represent the PaaS model of cloud computing. Instance level deployment diagram is used to represent the instances of physical resources of a system where the software elements are deployed in the Object-Oriented environment This diagram is used for PaaS model of cloud computing because the PaaS model provides the VMs which are actually the instances of a specific machine (such as a server).

Table 2–Virtual Machine Stereotype Description

NAME	VM
UML MODEL ELEMENT	Instance node
DESCRIPTION	representations of this stereotype represents VM in PaaS model
CONSTRAINTS	An instance of node can only be associated with a node

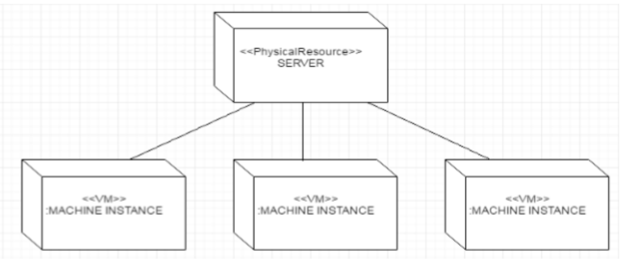

Fig 2–Stereotype of VM

Use of UML Component Diagram

The UML Component Diagram is also used to represent the PaaS model of cloud computing. Component diagram is used to describe the executable artifacts of a system. Component diagram does not depict the functionality of the software but it depicts and explains the elements used to make those functionalities. Therefore, it is used to describe the elements of the PaaS model because the OS and web servers are the elements to which make functional the applications. The elements of PaaS model like runtimes, middleware and load balancing are considered as components. The detail description of the stereotype of PaaS element is given in table 3. The representation of PaaS elements via UML component diagram is shown in fig 3.

Table 3– PaaS Element Stereotype Description

NAME	PaaS element
UML MODEL ELEMENT	component
DESCRIPTION	representations of this stereotype represents PaaS element in PaaS model
CONSTRAINTS	A component can only be used to represent the PaaS element

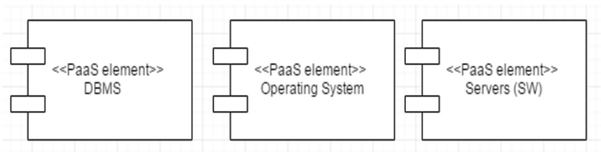

Fig 3–Stereotype of PaaS element

4.3 UML Representation for Software as a Service (SaaS) Model

In this part, extensions for the elements of SaaS model are proposed and explained. We propose extension of UML class diagram and package diagram to visualize the elements of PaaS model.

Use of UML Class Diagram

This UML diagram is used to symbolize the SaaS model of cloud computing. Class diagram is used to visualize the characteristics and functionalities of an object in object-oriented paradigm. This diagram is used for SaaS model of cloud computing because the SaaS model allows organizations to access the readymade services from vendor. The readymade services provide the overall functionalities of a specific application. The class diagram is composed of three different compartments where first one illustrate the name, second highlights all the characteristics and properties, and the last compartment highlights the operations and functionalities performed by the class. In the cloud environment with the tailored definition, the first place shows the name of the service, the second place visualizes the characteristics of that particular service and the last place shows the functionalities of that particular service depicted in fig 4. The stereotype definition for SaaS is given in table 4.

Table 4– Service Stereotype Description

NAME	Service
UML MODEL ELEMENT	class
DESCRIPTION	representations of this stereotype represents service in SaaS model
CONSTRAINTS	A class can only be used to represent the service

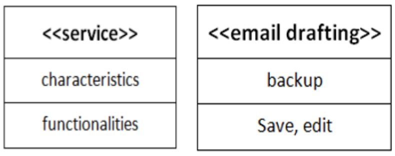

Fig 4–Stereotype of Service

Use of UML Package Diagram

The UML package Diagram is also used to represent the services in the SaaS model of cloud computing. Package diagram is used to describe and visualize a group of similar classes in the object-oriented paradigm. Similarly, in the cloud computing paradigm, the package diagram is used to represent the group of similar services or a complete service which is composed of different similar sub services. The detail description of this stereotype is given in table 5 and the representation via UML package diagram is shown in fig 5.

Table 5– meta-service Stereotype Description

NAME	meta-service
UML MODEL ELEMENT	package
DESCRIPTION	representations of this stereotype represents service which is composed of different services in SaaS model
CONSTRAINTS	A package can only be used to represent the meta-service

Fig 5–Stereotype of meta-service

5 Conclusion

Model-Driven Engineering (MDE) and Cloud Computing are two of the most dominant paradigms nowadays. Models are considered as major parts in the MDE, and concentrates on the formation and then conversion of models to the implementation. Cloud Computing is also getting reputation as the standard approach for designing and organizing software applications over the internet. It is observed that MDE can play a vital role in the cloud computing paradigm. Therefore, standard and verifiable notation are required to model the aspects o cloud computing to support the MDE in cloud computing. UML provides modeling and designing notions to symbolize software, platforms and architectural artifacts from diverse viewpoints of object-oriented paradigm. UML can also be extended to model and visualize the non object-oriented systems. In this work we propose the extensions of UML to visualize the cloud aspects. We used different UML models for the elements for different cloud modeling services. We use extension of UML specification diagram to model IaaS elements, UML instance level deployment and component diagram for modeling PaaS elements, and UML class and package diagram for symbolizing the SaaS elements of cloud computing serve models.

References

1. J. Whittle, et al, “The State of Practice in Model-Driven Engineering.” IEEE Software 31(3), 79-85, 2014.
2. H. Brunelière, et al,“Combining Model-Driven Engineering and Cloud Computing,” in 4th Workshop on Modeling, Design, and Analysis for the Service Cloud, Paris, France, June 2010.
3. Armbrust M, Fox A, Griffith R, Joseph A D, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, StoicaI, Zaharia M. A View of Cloud Computing. Communications of the ACM; 2010;53(4):50–58.
4. G Kovesdán, M Asztalos and L Lengyel, “Modeling Cloud Messaging with a Domain-Specific Modeling Language”, Proceedings of the 2nd International Workshop on Model-Driven Engineering on and for the Cloud co-located with the 17th International Conference on Model Driven Engineering Languages and Systems (MoDELS 2014), Valencia, Spain, September 30, 2014.
5. M. Abu-Matar, et al, “Towards Software Product Lines Based Cloud Architectures,” Proceedings of the IEEE International Conference on Cloud Engineering (IC2E), 2013.
6. Nicolas Ferry, et al, “Towards Bridging the Gap Between Scalability and Elasticity,” 4th International Conference on Cloud Computing and Services Science, 746-751, 2014.
7. ”UML Revision Task Force”. OMG Unified Modeling Language: Superstructure, February 2009. //www.omg.org/docs/formal/09-02-02.pdf
8. A Bergmayr, J Troya, P Neubauer, M Wimmer, and G Kappel, “UML-based Cloud Application Modeling with Libraries, Profiles, and Templates”, Proceedings of 17th International Conference on Model Driven Engineering Languages and Systems (MoDELS 2014) co-located with the the 2nd International Workshop on Model-Driven Engineering on and for the Cloud, Valencia, Spain, September 30, 2014.
9. R Crocombe and D Kolovos, “Code Generation as a Service”. Proceedings of ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems, Ottawa (Canada), September 29, 2015.
10. M Abu Matar and J Whittle, “MDE Opportunities in Multi-Tenant Cloud Applications”, Proceedings of the 2nd International Workshop on Model-Driven Engineering on and for the Cloud co-located with the 17th International Conference on Model Driven Engineering Languages and Systems (MoDELS 2014), Valencia, Spain, September 30, 2014.
11. S Alajrami , Ar Romanovsky , P Watson , and A Roth, “Towards Cloud-Based Software Process Modelling and Enactment”, Proceedings of the 2nd International Workshop on Model-Driven Engineering on and for the Cloud co-located with the 17th International Conference on Model Driven Engineering Languages and Systems (MoDELS 2014), Valencia, Spain, September 30, 2014.
12. Mell P, Grance T. The NIST Definition of Cloud Computing. Communications of the ACM; 2010;53(6):50.
13. G C Silva, L M. Rose, and R Calinescu, “Cloud DSL: A Language for Supporting Cloud Portability by Describing Cloud Entities”, Proceedings of the 2nd International Workshop on Model-Driven Engineering on and for the Cloud co-located with the 17th International Conference on Model Driven Engineering Languages and Systems (MoDELS 2014), Valencia, Spain, September 30, 2014.
14. L. Fuentes and A. Vallecillo. An Introduction to UML Profiles. The European journal for the Informatics Professional, April 2004.
15. A Bergmayr, A Rossini, N Ferry, G Horn, L Orue-Echevarria, A Solberg, and M Wimmer, “The Evolution of CloudML and its Manifestations”, Proceedings of ACM/IEEE 18th International Conference on Model

Driven Engineering Languages and Systems, Ottawa (Canada), September 29, 2015.
16. C Cl Manzanares, J S Cuadrado, J de Lara, “Building MDE cloud services with DISTIL”, Proceedings of ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems, Ottawa (Canada), September 29, 2015.
17. C Zolotas and A L Symeonidis, “Towards an MDA Mechanism for RESTful Services Development”, Proceedings of ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems, Ottawa (Canada), September 29, 2015.