BiCloud-2M: A Combined Bigraph Maude-based Tool for Cloud Specification and Analysis

Zakaria Benzadri
LIRE Laboratory
University of Constantine 2
Constantine, Algeria

benzadri@gmail.com

Chafia Bouanaka and Faiza Belala
LIRE Laboratory
University of Constantine 2
Constantine, Algeria

c.bouanaka@umc.edu.dz and
belalafaiza@hotmail.com

Abstract – Service availability is a challenging issue in Cloud Computing. It implies continuous reconfiguration of cloud architecture by adding or removing different resources (virtual machines, services...) to ensure the suited quality of service. Thus a main goal in Cloud systems design is to model and analyse cloud architecture and its dynamic reconfiguration. Based on Bigraphical Reactive Systems (BRS) theory as a semantic framework and Maude language as an executable specification language, we propose a tool called BiCloud-2M offering a formal support for specifying and analysing cloud architecture systems. In this paper, we describe the BiCloud-2M implementation, and how it can be used to verify some cloud inherent properties.

Keywords – Cloud Computing; Bigraphical Reactive Systems; Maude; Model checking.

1. INTRODUCTION

Cloud Computing is a recent paradigm for information technology that enables remote, on-demand access to a set of configurable computing resources as internet-based services. Although this cloud model promotes service availability, emphasizes on resources reuse rationalization and provides opportunities for reducing software development costs, there are still many open issues. One major challenging topic is to formally model cloud-based architecture and analyse its shape shifting. The main objective of this paper is to propose a combined bigraph Maude-based tool (BiCloud-2M) to specify cloud systems and offer analysis support to model-check their inherent properties. To ease BiCloud-2M exploitation, the tool offers a java user interface that assists designers to specify cloud systems and analyse them.

Bigraphical Reactive Systems (BRS) [1] were adopted as a semantic basis to specify fundamental aspects of cloud computing [2]. BRS seem adequate for two reasons. Firstly, the model emphasizes on both locality and connectivity that can be used to specify location and interconnection of cloud systems. Secondly, a set of reaction rules, providing to bigraphs the ability to reconfigure themselves, are very useful to formalize cloud system dynamics. A nice consequence of this axiomatization is that relationships between cloud services and cloud customers have been exploited to formally analyse some cloud inherent properties. This would not have been possible without a mapping of the bigraphical model to a Maude-based specification [3]. Maude [4] is a high-level language and a high-performance system that supports rewriting logic specification and programming of systems.

BiCloud-2M implements our bigraphical modelling methodology in which both the cloud architecture elements and their interactions are modelled explicitly. It has an efficient Maude-based rewriting engine, able to execute and analyse the cloud architecture specifications.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 serves as a brief introduction to the proposed modelling methodology. Then, section 4 describes the main principles of our tool, and its uses. Finally, section 5 compares our work with other related proposals and section 6 draws some conclusions and outlines some future research activities.

2. RELATED WORK

In the literature, several frameworks have been recently provided but do not deal with all fundamental concepts of cloud computing. They particularly focus on cloud computing financial and technological aspects; MobiCloud (Cloud Framework for Mobile Computing and Communication) [5], is a new cloud framework for MANETs that focuses on interrelated system components including resource and information flow isolations. It enhances communication by addressing trust management, secure routing, and risk management issues in the network. The work in [6] aims to present technical and business challenges for organizational Cloud adoption, and describes four key areas to be addressed: Classification; Organizational Sustainability Modelling; Service Portability and Linkage. The Cloud Computing Business Framework (CCBF) has been proposed to help organizations achieving Cloud design, deployment, and service migration. It has been used in several organizations offering added values and positive impacts. In [7], the authors identify the necessary perspectives to capture benefits of cloud computing. Then, they propose a conceptual framework for cloud computing benefits. Their framework accounts for the different business areas and organizational levels where each of the benefits manifests. Ricardo J. et al [8] propose an ontology-enriched framework for cloud-based Enterprise Interoperability. The proposed framework allows knowledge, decisions, and responsibilities to be exchanged about negotiations. It is supported by a reference ontology and uses cloud-computing as the paradigm to deploy services on the network. Chandrakumar T. and Parthasarathy S. [9] explore the available literature on cloud ERP systems, suggest factors to be considered in cloud ERP, and propose a framework for evaluating cloud ERP systems. Their framework is grounded on software engineering parameters involved in the development of cloud ERP.

The aforementioned works are limited and focus solely on enhancing the financial and technological aspects of cloud computing. Consequently, cloud computing lacks a theoretical framework that associates a clear semantics to its basic concepts: service delivery and deployment models. This framework might be able to support major cloud computing concepts specification and allows formal analysis of high level services provided over the cloud computing architecture. Within this perspective, we have proposed a formal semantic framework for specifying cloud systems [2]. In this paper, we develop a tool called BiCloud-2M that offers analysis support to model-check their inherent properties.

3. BiCloud-2M FORMAL BASIS

The BiCloud-2M modelling methodology is based on a judicious coupling of BRS theory as a semantic framework and Maude language as an executable specification language. We have proposed [2] a formal model for cloud computing architecture design and its shape shifting specification using Bigraphical Reactive Systems. A mapping to Maude executable specifications was also defined in order to analyse the obtained specifications [3].

A cloud service is modelled by a node representing an abstraction of three different service delivery models (IaaS, PaaS and SaaS) that collaborate to ensure front end requests. Controls attached to nodes allow distinguishing between the three service delivery models. Cloud customers are also modelled as nodes equipped with specific controls that enable determining both types of Cloud customers (End users and Independent Software Vendors). Cloud architecture dynamics in terms of interactions between cloud services and customers are established via reaction rules. In Maude language, bigraphical nodes are defined as a specific sort, and reaction rules are implemented by rewrite rules. The following correspondence table summarizes our Bigraphical Cloud computing architecture concepts and their equivalent in Maude language.

Table 1. Correspondence table Cloud / Bigraph / Maude concepts

Cloud Computing	**BRS**	**Maude Language**
Service and Customer	Nodes	Operators
Service models	Controls	Sorts
Customers types	Controls	Sorts
Dynamics	Reaction rules	Rewrite rules
Properties	———	LTL Specifications

The use of rewriting logic via its implementation language Maude, takes advantage of the model-checker tool for LTL properties verification [4]. Thus, BiCloud-2M offers a verification module for model checking some cloud inherent properties.

4. BiCloud-2M TOOL PRESENTATION

Our model is implemented in BiCloud-2M tool as system modules. Figure 1 presents a basic overview of the defined modules and their sub-module dependencies.

4.1. Principle

The current BiCloud-2M tool is composed of a JAVA-frontend for various editing tasks as designing new cloud architecture and introducing

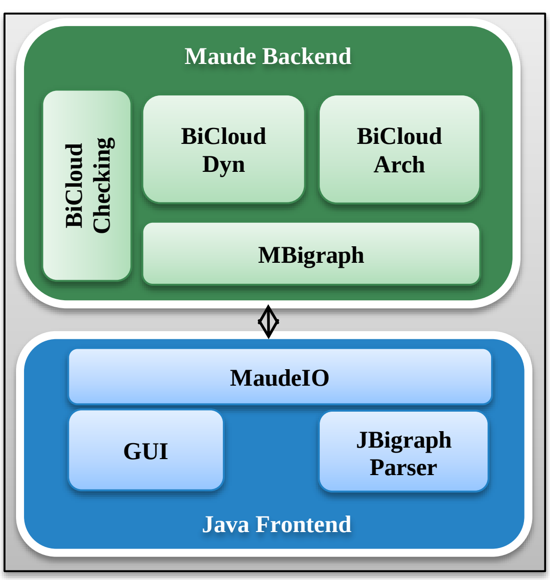

Figure1. BiCloud-2M modules

a property to verify. Additionally, it is responsible of parsing the BiCloud architecture and generating the corresponding Maude-based specifications (using the defined *JBigraph Parser and MaudeIO modules*).

The graphical user interfaces of BiCloud-2M are implemented in the *GUI module*. Figure 2 presents a snapshot of the proposed tool. The Menu Bar contains a list of shortened commands to be executed in BiCloud-2M. The left hand panel (area 1 in Figure 2) is a toolbar for cloud architecture specification. The upper right panel in area 2 of figure 2 is a toolbar for cloud execution and analysis. The lower right pane (area 3 in figure 2) is a Maude console for various BiCloud-2M results and outputs.

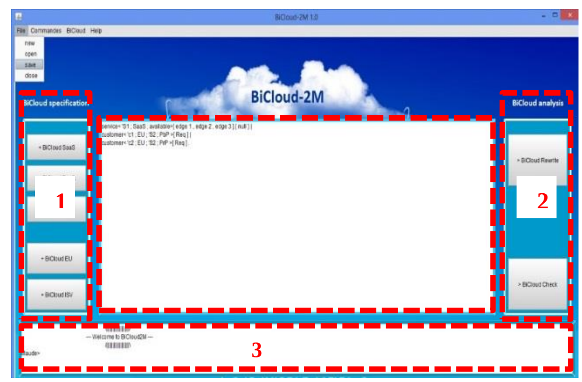

Figure 2. A snapshot of BiCloud-2M Tool

4.2. Core BiCloud-2M

The Maude-based backend component is a rewriting engine for executing BiCloud specifications and verifying their inherent properties. It is composed of four defined modules (see Figure 1), each one with a specific role: the *MBigraph* module includes sorts and operators declaration for Bigraphs theory concepts definition. The *BiCloud-Arch* module defines the syntax and semantics of cloud architecture elements. The *BiCloud-Dyn* module specifies Cloud system dynamics, it contains a set of rewrite rules expressing cloud architecture possible reconfigurations. Finally, the *BiCloud-Checking* module allows checking LTL formula by introducing a given initial state.

The proposed BiCloud-2M tool will be illustrated via the cloud health system that allows doctors to exchange patient's information. At each appointment, the doctor needs to consult medical information of the patient by editing state history and make sure of the performed treatments. We consider that the SaaS (S1) allows doctors to consult medical information of every patient.

BiCloud-2M Architecture: As initial state, we consider the SaaS (S1) is started in the cloud and two customers (doctor's) (C1 and C2) are requesting it. This initial state is edited in BiCloud-2M using commands available in cloud architecture specification toolbar.

Figure 3 shows the cloud service input-box that allows the specification of a cloud service. It takes as arguments:

- a Quoted Identifier to specify its name;
- a Control specifying its service delivery model;
- an Attribute specifying a service state (available, unavailable, and cloned);
- a set of Edges connected to its three ports.

Figure3. Cloud Service Input-box

Figure 4 represents the input-box to specify a cloud customer. It takes as arguments:

- a Quoted Identifier to specify its name;
- a Control specifying its customer type;
- a Quoted Identifier to specify the requested service;
- a Port specifying on which the requested service will be delivered.

Figure 4. Cloud Customer Input-box

BiCloud-2M Dynamics: We defined a set of rewriting rules that express the cloud architecture dynamics**.** Table 2 illustrates the proposed rewriting rules.

Table 2. Cloud rewriting rules

Rewriting rule	Maude code
Cloud service Allocation	rl : service< i ; cs:Scsb ; available >[e1 , e2 , e3].{ b2 } | customer< i1 ; cc1:Sccb ; i ; p:Port >[**Req**] | b1 **=>** service< i ; cs:Scsb ; available >[e1 , e2 , e3].{ b2 } | customer< i1 ; cc1:Sccb ; i ; p:Port >[**e1**] | b1 .
Cloud service deallocation	rl: customer<i1 ; cc:Sccb ; i ; p:Port >[edge na] | b1 **=>** customer<i1 ; cc:Sccb ; i ; p:Port >[null] | b1 .
Cloud Scaling Up	rl: service< i ; cs:Scsb ; unavailable >[e1 , e2 , e3].{ b2 } | customer< i1 ; cc1:Sccb ; i ; p:Port >[Req] | b1 **=>** service< i ; cs:Scsb ; unavailable >[e1 , e2 , e3].{ b2 } | service< i ; cs:Scsb ; cloned >[e1 , e2 , e3].{ null } | customer< i1 ; cc1:Sccb ; i ; p:Port >[e1] | b1 .
Cloud Scaling Down	rl : service< i ; cs:Scsb ; cloned >[e1 , e2 , e3].{ b2 } | customer< i1 ; cc1:Sccb ; i ; p1:Port >[null] | b1 **=>** customer< i1 ; cc1:Sccb ; i ; p1:Port >[null] | b1 .

Allocating the cloud service (S1), is realized via the BiCloud-rewrite command, which is available in the cloud architecture execution toolbar. Figure 5 shows the BiCloud-rewrite input-box that allows the execution of the underlying specifications. It takes as arguments the rewrite command type and the number of rewrites.

Figure 5. BiCloud-Rewrite Input-box

The result will be shown in the output pane (see figure 6):

Figure 6. BiCloud-Rewrite result

BiCloud-2M Checking: One major benefit of adopting Maude as an implementation language of the BiCloud-2M tool is the exploitation of Maude model-checker for cloud system analysis. We deal here with a significant property such as service availability.

Service availability is fulfilled if system model do not contain an unsatisfied customer request. It is formally specified with the following LTL formula: "O [] not (requester)".

Figure 7 represents the BiCloud-check input-box that takes as arguments the LTL property specification and its relevant state predicates.

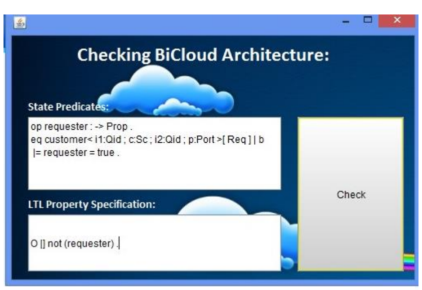

Figure 7. BiCloud-check Input-box

Figure 8 shows model checking results. The result is that the model is concluded to possess the desired property.

Figure 8. BiCloud-check result

5. DISCUSSION

A set of tools is proposed in the literature for BRS edition, execution and verification; BPL Tool [10] is the first implementation of BRS with binding, Bigredit (for "bigraph editor"), Big Red [11] is a visual editor for bigraphs, and BigMC [12] is a model-checker designed for BRS properties model checking. However, during their exploitation, we have been confronted with several limitations, because they remain restrictive and do not correspond to our expectations. Although BigMC is the unique tool for model checking bigraphs, it remains very restrictive since it enforces designers to adapt their model to the desired property.

BiCloud-2M offers the possibility to execute and formally analyze cloud bigraphical specifications, by: (i) simulating the behavior from a given initial state; (ii) checking that all terminal states reachable from the initial state satisfy a linear temporal logic property. Additionally, BiCloud-2M is more expressive in terms of properties specification and its response time is reduced considerably. Figure 9 highlights BiCloud-2M performance, where the y-axis is the time spent to rewrite the initial state, and the x-axis is the number of rewrites.

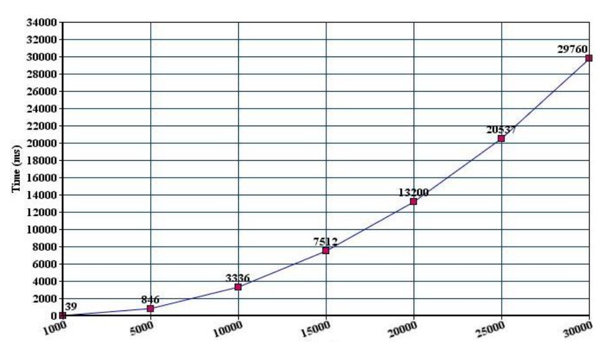

Figure 9. BiCloud-2M Performance

6. CONCLUSION

BiCloud-2M is a tool for cloud systems design and verification that supports a bigraphical modelling methodology for both cloud architecture elements and their interactions modelling. It has an efficient rewriting engine, able to execute and analyse the cloud architecture specifications.

BiCloud-2M tool is open and extensible where new features and properties can be easily in traduced as generalizing its use to any bigraph and the corresponding inherent properties.

REFERENCES

[1] Milner, R.: The Space and Motion of Communicating Agents. Cambridge University Press (2009).

[2] Benzadri, Z., Belala F., and Bouanaka, C.: Towards a Formal Model for Cloud Computing. ICSOC workshops, 2013, (2014).

[3] Benzadri, Z., Bouanaka, C., and Belala F.: Verifying Cloud Systems using A Bigraphical Maude-Based Model Checker. CLOSER workshops, 2014, (2014).

[4] Clavel, M., Durn, F., Eker, S., Lincoln, P., Mart-Oliet, N., Meseguer, J., Talcott, C.L., eds.: All About Maude - A High-Performance Logical Framework, How to Specify, Program and Verify Systems in Rewriting Logic. In Clavel, M., Durn, F., Eker, S., Lincoln, P., Mart-Oliet, N., Meseguer, J., Talcott, C.L., eds.: All About Maude. Volume 4350 of Lecture Notes in Computer Science., Springer (2007).

[5] Dijiang Huang; Xinwen Zhang; Myong Kang; Jim Luo. (2010). "MobiCloud: Building Secure Cloud Framework for Mobile Computing and Communication," Service Oriented System Engineering (SOSE), 2010 Fifth IEEE International Symposium on , vol., no., pp.27,34, 4-5

[6] Chang, Victor, Walters, Robert John and Wills, Gary. (2014). The development that leads to the Cloud Computing Business Framework. International Journal of Information Management, June, 33, (3), 524-538.

[7] Nattakarn P., Xiaofeng W., and Pekka A. (2013). Towards a Conceptual Framework for Assessing the Benefits of Cloud Computing, 4th International Conference, ICSOB 2013, Potsdam, Germany, June 11-14, 2013. Proceedings

[8] Ricardo J., Adina C., Carlos C, Moisés D. , Parisa G. (2013) Ontology Enriched Framework for Cloud-based Enterprise Interoperability. Concurrent Engineering Approaches for Sustainable Product Development in a Multi-Disciplinary Environment

[9] Chandrakumar T. and Parthasarathy S. (2014). A Framework for Evaluating Cloud Enterprise Resource Planning (ERP) Systems. Computer Communications and Networks. pp 161-175

[10] Glenstrup, A.J., Damgaard, T.C., Birkedal, L., and jsgaard, E. H. An implementation of bigraph matching. 2007.

[11] Faithfull, AJ, Perrone, GD & Hildebrandt, T 2013, 'Big Red: A Development Environment for Bigraphs' E A S S T Electronic Communications, vol 61.

[12] Perrone, G., Debois, S., Hildebrandt, T.T.: A model checker for bigraphs. In Ossowski, S., Lecca, P., eds.: SAC, ACM (2012) 1320–1325.