Towards a Formal Model for Cloud Computing

Zakaria Benzadri, Faiza Belala, and Chafia Bouanaka

LIRE Laboratory
Department of Software and Information Systems Technology
University of Constantine 2, Constantine, Algeria
benzadri@gmail.com, belalafaiza@hotmail.com,
c.bouanaka@umc.edu.dz

Abstract. The use of formal methods is an effective means to improve complex systems reliability and quality. In this context, we adopt one of these methods to formalize cloud computing concepts. We focus on modeling interactions between cloud services and customers. Based on Bigraphical Reactive Systems, the formalization process is realized via the definition of a Cloud General Bigraph (CGB) obtained by associating; primarily, a CCB (Cloud Customers Bigraph) to cloud customers. Then, a Cloud Services Bigraph (CSB) is proposed to formally specify cloud services structure. Finally, juxtaposing these two bigraphs (CSB and CCB) gives rise to the suited CGB. In addition, a natural specification of cloud deployment models is specified. This paper also addresses cloud service dynamics by defining a set of reaction rules on bigraphs in a way that is amenable to reconfigure the designed cloud system.

Keywords: Cloud Computing, Bigraphical Reactive Systems, Formal Methods, Cloud Model, Cloud General Bigraph, Cloud Customers Bigraph, Cloud Services Bigraph.

1 Introduction

Software reusability has permanently triggered researchers and practitioners of software engineering. From the notion of modules defined by Djikstra to the well-known web services, we are still aiming on maximizing software reusability and thus reducing development cost. Based on service oriented paradigm and service oriented architectures and putting forward reduction of not only software development cost but also deployment effort, cloud computing [1] generalizes service reuse to all computer resources. The main principle behind this model is offering computing, storage, and software "as a service". It implies dynamic provisioning with on demand shared computing resources, and provides computing resources as services in an attempt to reduce IT capital and operating costs. Nevertheless, cloud computing is actually changing software design and development practices and involves revisiting and redefining some fundamentals and concepts. Much as service-oriented architecture (SOA), cloud architecture must be defined, governed, and managed independently [2].

A.R. Lomuscio et al. (Eds.): ICSOC 2013 Workshops, LNCS 8377, pp. 381–393, 2014.

Since it has emerged from the industry, a hard work of formalization is still needed to overcome one of cloud computing main obstacles; namely bugs in Large-Scale Distributed Systems – "*one of the difficult challenges in cloud computing is removing errors in these very large scale distributed systems*" [3]. The main issue that still needs to be addressed is the crucial absence of an appropriate model for cloud computing. This model might be able to support major cloud computing concepts specification and allows formal modeling of high level services provided over the cloud computing architecture.

In this work, we adopt Bigraphical Reactive Systems (BRS) proposed by Milner et al. [4] to formally specify cloud services and customers and their interaction schemes. The formalization process is realized via the definition of a Cloud General Bigraph (CGB); obtained by primarily associating a CCB (Cloud Customers Bigraph) to cloud customers. We enrich bigraphs signature with new controls (kinds of node) EU and ISV representing respectively End User and Independent Software Vendor. Then, we associate a Cloud Services Bigraph (CSB) to cloud services that also needs an enrichment of bigraphs signature to support all service types; IaaS, PaaS and SaaS. Finally, the juxtaposition of these two bigraphs (CSB and CCB) gives rise to the suited CGB. Besides, the model allows a natural specification of cloud deployment models. Cloud systems dynamics is specified via a set of reaction rules on both CSB and CCB bigraphs.

This paper is presented in a coordinated and integrated manner, starting with some fundamentals recall followed by presenting necessary definitions and rules that constitute the Cloud General Bigraph. It is organized as follows. In section 2, we present related work. A brief description of Bigraphical Reactive Systems and their essential concepts is introduced in section 3. In section 4, our cloud computing model is presented. Section 5 illustrates the proposed cloud formalization approach through a well-known case study of the Cloud-Health system. Finally, conclusion and future work are addressed in section 6.

2 Related Work

Nowadays researches on cloud computing are mainly focused on technical aspects, yet a modest attention is devoted to the formalization of cloud computing fundamental concepts.

H. Dong et al. [5]. and T. Grandison et al. [6], gave some discussion and exploration on establishing relationships between virtualization and Cloud Computing. Throughout their work, they attempt to give out a formal definition of cloud computing from a virtualization viewpoint using its theoretical basic concepts. S.-X. LUO et al. [7], propose an access control model to achieve a fine-grained data confidentiality and scalability via a formal definition of the HABAC model (Hierarchy Attribute-Based Access Control). A. Adamov and V. Hahanov [8] define a security model for individual cyberspace (ICS) protection as a means to ensure a secured user's virtual environment. They establish an analysis of security issues related to ICS and propose a conceptual model for modern security environments. L Freitas et al. [9], present an abstract formalization of federated cloud workflows using the Z notation. They define various rule based properties to restrict valid options with respect to: security and cost constraints. T. Binz et al. [10], propose Enterprise Topology Graphs (ETG) as a formal model to describe an enterprise topology. Based on the established graph theory, ETG is used to both formalize and verify cloud systems. Besides, authors have shown how ETG can improve the environmental impact of IT enterprise. R. He et al. [11], propose a trust model to specify trustworthiness and uncertainty of trust relationships between peers, namely cloud-model. Their model is strange and unspecified; it cannot be directly applied to model trust, and needs to be extended.

Up to now, however, cloud computing paradigm lacks a standard and formal definition of its basic concepts; service and deployment models, only some technological attempts are realized; for virtualization as it has been done in [5] and [6], for security as in [7], [8] and [9], or for IT enterprise as in [10].

Albeit, various models were adopted (Petri Nets [12] and [13], Semantic Technology [14], MDA [15], Agent-Based [16], or Component Model [17]), they do not show an efficient adequacy to cloud computing. Particularly, they deal with only one problem at a time. In this paper, BRS [4] will be adopted for two reasons. On the one hand, the model emphasizes on both locality and connectivity that can be used to specify cloud entities location and interconnection. On the other hand, bigraphical reaction rules are very useful to formalize cloud services elasticity providing them the ability to reconfigure themselves.

3 Bigraphical Reactive Systems

A bigraph as an ordinary graph is composed of nodes and edges, unlike nodes in a bigraph can be nested giving rise to hierarchical and larger bigraphs. Additionally, a bigraph is the result of composing a link graph; representing interconnection between nodes, and a place graph; expressing physical locations of theses nodes, hence the prefix 'bi' in bigraph.

3.1 Concrete Place Graph

The place graph consists of a forest of trees; each with its own root and servers to model locality or containment of entities. The formal definition of a place graph is:

Definition 1 (Place Graph [4]).
A place graph is a 3-tuple (V, ctrl, prnt): $m \rightarrow n$ having an inner interface m and an outer interface n, both are finite ordinals, used to index place graph sites and roots respectively. Where: V is a finite set of nodes, ctrl:$V \rightarrow S$ is a control map assigning controls to nodes. Each node has a control, which is an identifier belonging to a set that is called a signature (usually denoted as *S*). Each control indicates how many ports the node has, which controls are *atomic* (empty node), and which of the non-atomic controls are *active* (node permitting reaction inside) or *passive*. Finally, prnt: $m \uplus V \rightarrow V \uplus n$ is a parent map indicating the parent of each node.

3.2 Concrete Link Graph

The link graph models system connectivity or assembly and is composed of a set of nodes and a set of hyper-edges; meaning that each edge has multiple tentacles to connect different nodes; tentacles are called ports. Formal definition of a link graph is as follows:

Definition 2 (Link Graph [4]).
A link graph is a quadruple (V, E, ctrl, link): X →Y having an inner interface *X* and outer interface *Y*; called respectively the inner and outer names of the link graph. Where: *V* and *E* are sets of nodes and edges respectively, ctrl: V→S is a control map, and link: $X \uplus P \rightarrow E \uplus Y$ is a link map with *P* a set of ports. We shall call $X \uplus P$ the link graph points, and $E \uplus Y$ its links.

3.3 Concrete Bigraph

Having defined the place and link graph independently, we combine them to obtain a bigraph formal definition:

Definition 3 (Bigraph [4]).
A bigraph is a 5-tuple (V, E, ctrl, prnt, link): (m, X) → (n, Y), also written <GP, GL>, consisting of a concrete place graph GP = (V, ctrl, prnt):m→n and a concrete link graph GL = (V, E, ctrl, link): X →Y.

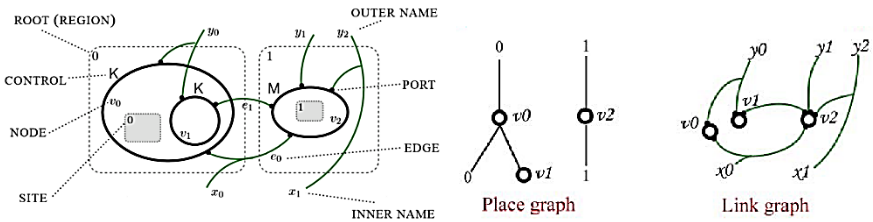


Fig. 1. Anatomy of bigraphs (Source: [4])

These definitions make precise the bigraph anatomy illustrated in Figure 1.

Additionally to graph theory based definitions that are insufficient to formally reason on bigraphs, an algebraic representation is also proposed offering various forms of bigraphs composition.

The language is summarized in Table 1:

Table 1. Terms language for bigraphs

Term	Meaning
U.V	Nesting. U contains V
U|V	Prime product
A⊗B	Juxtaposition
A◦B	Composition.

While nesting a bigraph within another bigraph is realized via the composition operation, placing them side-by-side is achieved using the juxtaposition operation which is a useful way for combining bigraphs.

3.4 Bigraphical Reactive Systems

Once bigraph structure presented, its dynamics will be defined through a BRS, consisting of a category of bigraphs and a set of reaction rules to be applied on them and describing bigraphs structural dynamics.

Definition 4 (Reaction Rule [4]).
A reaction rule takes the form (R, R', O) where R: m→J is a bigraph called redex (the pattern to be changed), R': m'→J is also a bigraph called reactum (the changed pattern), and O: m'→m is a map of ordinals establishing the correspondence between inner interfaces of R and R'.

BRS basic concepts introduced here will be exploited to formalize both cloud structure and dynamics in the following sections.

4 A Model for Cloud Computing

A bigraph represents orthogonal notions of locality and connectivity through the use of two separate graph structures (place graph and link graph), so it is an elegant solution and formal approach to describe cloud computing actors and their relationships. Cloud computing organization can be divided into two essential parts: the front-end and the back-end; usually connected via internet. The Front-end encloses customer's computer and necessary interface to access the cloud and the back-end contains the cloud services. In the present work, we adapt bigraphs to specify customers, services and their deployment models, and eventual interactions between them. Such formalization defines a precise semantics to the considered concepts. Our cloud model is called CGB (Cloud General Bigraph) which is a juxtaposition of two independent bigraphs: the Cloud Services Bigraph (CSB) defining the back-end part and the Cloud Customers Bigraph (CCB) modeling the front-end part. Additionally, a set of reaction rules is defined to formalize dynamics of the cloud computing architecture.

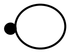
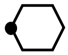
4.1 Cloud Customers Bigraph

We propose a formal definition of a Cloud Customers Bigraph that captures essential concepts identifying both, End users accessing only to SaaS and ISV (Independent Software Vendor) accessing to IaaS and PaaS types of customers. We model cloud customers as nodes equipped with specific controls to distinguish the two types of cloud customers; End user and ISV. Both are atomic nodes and have many ports to send their requests. We use the notation "*a: (x, act)*" where '*a*' is a control with arity (number of ports) 'x' and activity 'act'. We also use the ar(-) map to identify the arity of a given control, and we suggest a suitable graphical representation for each control (see table 2).

Definition 6 (Cloud Customers Bigraph).
The Cloud Customers Bigraph formalizing customers model takes the form (V_{CCB}, E_{CCB}, $ctrl_{CCB}$, GP_{CCB}, GL_{CCB}): $<m_{CCB},X_{CCB}>\rightarrow<n_{CCB},Y_{CCB}>$, with V_{CCB} representing all cloud customer nodes, E_{CCB} is a finite set of edges, $ctrl_{CCB}$: $V_{CCB}\rightarrow S_{CCB}$ is a control map that assigns a control to each cloud customer. The signature S_{CCB} is defined by S_{CCB}= {EU, ISV}. The map ar: $S_{CCB}\rightarrow$N assigns an arity to each control, where ar(EU)= ar(ISV)=x and x>0**.** X_{CCB} is the inner face and Y_{CCB} is the outer face. GP_{CCB} represents the corresponding place graph and GL_{CCB} represents the link graph. Therefore, the link map $link_{CCB}$:$X_{CCB}\uplus P_{CCB}\rightarrow E_{CCB}\uplus Y_{CCB}$, with a set of ports P_{CCB}= {(v,i) | i ∈ ar($ctrl_{CCB}$(v))}, i.e., a port is represented as a pair consisting of a node (from V) and an index.

In this definition, the signature of CCB is S_{CCB}**= {**EU: (x, atomic), ISV: (x, atomic)}. A suitable graphical representation of each control type is presented in table 2.

Table 2. Cloud Customers Signature

Control	Activity	G. Representation
EU (End User)	Atomic	
ISV(Independent Software Vendor)	Atomic	

4.2 Cloud Services Bigraph

Three different service models are deployed within a cloud architecture, to ensure front end requests: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Albeit, all available cloud services are modeled by nodes in the Cloud Services Bigraph, controls attached to each nodes allow us to distinguish between the three categories of cloud services (see table 3). Additionally, services stack is naturally modeled via the hierarchy of nodes within the place graph (as shown in Figure 2) with respect to cloud services constraints. A node of control IaaS (infrastructure) can only contain nodes of control PaaS (platform). Also, a node of control PaaS can only contain nodes of control SaaS (software). Finally, a node of control SaaS does not contain any node. Consequently, nodes of control IaaS and PaaS are active, while nodes of control SaaS are atomic. Besides, we suggest a suitable graphical representation to each control (see table 3).

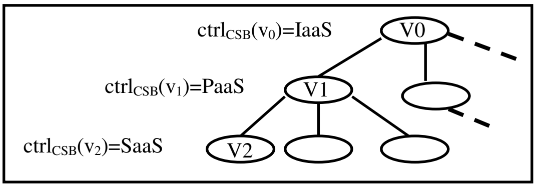


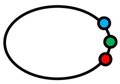
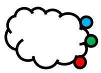
Fig. 2. Cloud Services Place Graph

To offer several access modes to cloud services, we propose that each cloud service (node) gets three different ports: Public (blue color in table 3), Private (red color in table 3), and Community (green color in table 3). As with Cloud Customers Bigraph, we now give a formal definition of the Cloud Services Bigraph enriched with a new map tp_{CSB} (-) assigning a type to each node port.

Definition 7 (Cloud Services Bigraph).
A Cloud Services Bigraph is associated to cloud service models and takes the form **CSB** = (V_{CSB}, E_{CSB}, $ctrl_{CSB}$, GP_{CSB}, GL_{CSB}, tp_{CSB}): $<m_{CSB},X_{CSB}>\rightarrow<n_{CSB},Y_{CSB}>$. Where V_{CSB} represents all cloud service nodes, E_{CSB} is a finite set of edges, $ctrl_{CSB}$: $V_{CSB}\rightarrow S_{CSB}$ is a control map that assigns a control to each cloud service. Controls range over the signature S_{CSB}= {IaaS, PaaS, SaaS} with a map ar: $S_{CSB}\rightarrow N$ assigning an arity to each control. Since each cloud service has three types of ports, ar(IaaS)=ar(PaaS)=ar(SaaS)=**3.** X_{CSB} is the inner face and Y_{CSB} is the outer face. GP_{CSB} represents its place graph and GL_{CSB} represents a link graph, such that, $link_{CSB}$: $X_{CSB} \uplus P_{CSB} \rightarrow E_{CSB} \uplus Y_{CSB}$, is a link map, with $P_{CSB}= \{(v,i) \mid i \in ar(ctrl_{CSB}(v))\}$ is the set of ports. Also, we define a new map assigning a type $t \in PT_{CSB}$={PbP, CmP, PrP}, to each node port $p \in P_{CSB}$, tp_{CSB}: $P_{CSB} \rightarrow PT_{CSB}$. So: tp_{CSB} (p)=PbP if p is Public Port, tp_{CSB} (p)=CmP if p is Community Port, tp_{CSB} (p)=PrP if p is Private Port.

In this definition, we summarize a suitable signature for CSB as follows: S_{CSB}={IaaS: (3, active), PaaS: (3, active), SaaS: (3, atomic)}.

Table 3. Cloud Services Signature

Control	Activity	G. representation	Conditions
IaaS	Active		∀ U, N in $V_{CSB,}$ (U.N ∧ $ctrl_{CSB}$(U)=IaaS) => $ctrl_{CSB}$(N)=PaaS.
PaaS	Active		∀ U, N in $V_{CSB,}$ (U.N ∧$ctrl_{CSB}$(U)=PaaS) => $ctrl_{CSB}$(N)=SaaS.
SaaS	Atomic		∀ U in $V_{CSB,}$ $ctrl_{CSB}$(U)=SaaS => {v in V_{CSB} ∣ U.N} = ∅.

4.3 Cloud General Bigraph

Once structural concepts of both cloud services and customers bigraphs have been separately defined, their juxtaposition defines the cloud general bigraph. Its place graph formally expresses cloud services and customers location. Its link graph formally expresses interconnections, in terms of service request/response relationship, between cloud services and cloud customers. Formally, we have the following definition.

Definition 8 (Cloud General Bigraph).
A Cloud General Bigraph formalizing cloud computing, takes the form $CGB=CSB\otimes CCB: I_{CSB}\otimes I_{CCB}\rightarrow J_{CSB}\otimes J_{CCB}$, where:
CSB⊗CCB $=<GP_{CSB}\otimes GP_{CCB}, GL_{CSB}\otimes GL_{CCB}>$, with:

- $GP_{CSB}\otimes GP_{CCB}: m_{CSB}+m_{CCB} \rightarrow n_{CSB}+n_{CCB}$ is defined by

  $GP_{CSB}\otimes GP_{CCB} = (V_{CSB}\uplus V_{CCB}, ctrl_{CSB}\uplus ctrl_{CCB}, prnt_{CSB}\uplus prnt_{CCB})$.
- $GL_{CSB}\otimes GL_{CCB}: X_{CSB}\uplus X_{CCB} \rightarrow Y_{CSB}\uplus Y_{CCB}$ is defined by

  $GL_{CSB}\otimes GL_{CCB} = (V_{CSB}\uplus V_{CCB}, E_{CSB}\uplus E_{CCB}, ctrl_{CSB}\uplus ctrl_{CCB}, link_{CSB}\uplus link_{CCB})$.

4.4 Cloud Deployment Models

Four cloud deployment models are identified in cloud computing. DM= {Public, Private, Community, Hybrid}. To take in charge such models, a formal description is done thanks to a meaningful interpretation of Cloud Services Bigraph. To identify service deployment models, we propose a function depm: $V_{CSB} \rightarrow DM$, where V_{CSB} represents cloud services. Since interconnections between cloud services and cloud customers are well defined via the link graph ($GL_{CSB}\otimes GL_{CCB}$), whenever a cloud service is connected to a cloud customer via a unique port, then service deployment model corresponds to the type of the port being used. Otherwise, if a cloud service is connected to cloud customers via various ports, then the cloud service is deployed as a hybrid cloud. Figure 3 represents cloud deployment models, using a link graph that is independent from locality, with C1, C2, C3 being Cloud customers and S1, S2, S3, S4 Cloud services. For instance, the cloud customer C2 is relied to S1 cloud service via its community port (CmP), so S1 is deployed as a community cloud (green color). Also the cloud customers C1 and C3 are relied to the cloud service S4, the first one with a public port (blue color) and the second one with a private port (red color), then S4 is deployed as a hybrid cloud (orange color).

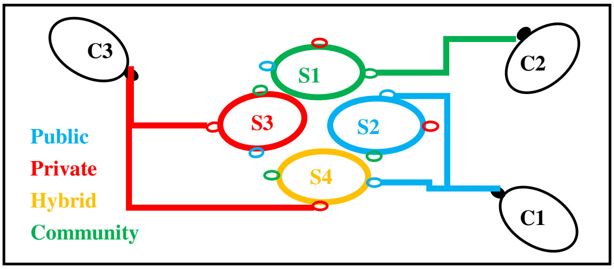


Fig. 3. Cloud Deployment Models Link Graph

Formally, $\forall$ s $\in V_{CSB}$, Ps={(s,i) | i $\in$ ar($ctrl_{CSB}$(s))} represents the set of ports of s and $\exists$ pb, pr, cm $\in$ Ps, such that: tp(pb)=PbP, tp(pr)=PrP, and tp(cm)=CmP. $\exists$ e $\in$ E, $\exists$ c $\in V_{CCB}$ and Pc={(c,i) | i $\in$ ar($ctrl_{CCB}$(c))} represents the set of ports of c and p $\in$ Pc. Such that:

- `link(s,`**`pb`**`)=e and link(c,p)=e → depm(s)=` **`Public.`**
- `link(s,`**`pr`**`)=e and link(c,p)=e → depm(s)=` **`Private.`**

- `link(s,`**`cm`**`)=e and link(c,p)=e →  depm(s)=` **`Community.`**
- `(depm(s) = Public and (depm(s) = Private or depm(s) = Community)) OR (depm(s) = Private and (depm(s) = Public or depm(s) = Community)) OR (depm(s) = Community and (depm(s) = Private or depm(s) = Public)) → depm(s) =` **`Hybrid.`**

4.5 Cloud Reaction Rules

We have now defined a Cloud General Bigraph in terms of its static structure, and being, expressive enough to model cloud services and customers connectivity and locality. To be moreover able to specify cloud system dynamics, CGB will be equipped with a set of reaction rules.

Locality reconfiguration is effected by shifting a cloud service from a parent cloud service to another one of the same control. Thereby, we propose two reaction rules defining the dynamics of bigraphs in this context.

- Rule PLR (PaaS Locality Reconfiguration). It expresses the fact that a platform may migrate from one infrastructure to another one. It changes the placing; a PaaS (P1) inside an IaaS (I1) shifts to another IaaS (I2), so we write: [I1.P1|I2 --> I1|I2.P1].
- Rule SLR (SaaS Locality Reconfiguration). Instead of migrating a platform to another infrastructure, rule SLR changes the placing of a service from one platform to another. A SaaS (S1) inside a PaaS (P1) shifts to another PaaS (P2), so we write: [P1.S1|P2-->P1|P2.S1].

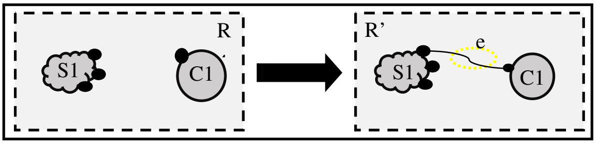


Fig. 4. Allocation Cloud Service Reaction Rule

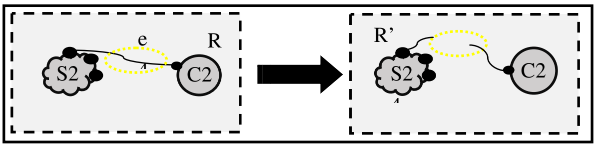


Fig. 5. Liberation Cloud Service Reaction Rule

Connectivity reconfiguration changes only the linking—not the placing—in a bigraph. We suggest that the redex (R) —the left-hand pattern—can match any cloud service control (IaaS, PaaS, and SaaS), and we propose two reaction rules, defining the dynamics of bigraphs in terms of service allocation. While, figure 4 bellow shows a cloud customer (C1) allocating a cloud service (S1), figure 5 represents a cloud customer (C2) liberating a cloud service (S2).

5 Case Study

Cloud-Health is a cloud system, allowing doctors to exchange information concerning their patients. We present this example in order to illustrate how the Cloud General Bigraph model is able to capture and formally represent all cloud computing aspects.

Let's suppose that we have the following cloud services: three SaaS (S1, S2, S3) in two PaaS (P1, P2) within only one IaaS (I1), see figure 6 for more details.

- S1 allows consulting doctors directories by supplying multiple information (name, address, telephone, specialty), that can be used by everyone.
- S2 allows supplying administrative or medical information of every patient. It can be used by doctors. Only a private access is allowed to patients in order to modify their administrative information.
- S3 allows every doctor to manage his medical office. This service is only used by the concerned doctor.

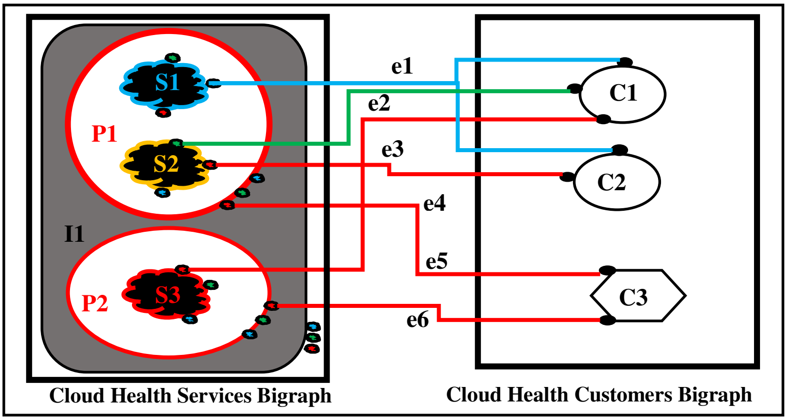


Fig. 6. Cloud Health General Bigraph

The cloud-health administrator ensures a smooth running of these three applications by supplying a private access to both PaaS (P1) and (P2).

According to our formalization approach, we can identify the CGB entities as follows (see figure 6):

- S1, S2, S3, P1, P2, I1 $\in V_{CSB}$, where: $ctrl_{CSB}$(S1)= $ctrl_{CSB}$(S2)= $ctrl_{CSB}$(S3)= SaaS, $ctrl_{CSB}$(P1)= $ctrl_{CSB}$(P2)= PaaS, and $ctrl_{CSB}$(I1)= IaaS.
- C1, C2, C3 $\in V_{CCB}$, where: $ctrl_{CCB}$(C1)= $ctrl_{CCB}$(C2)= EU (represents respectively a doctor and a patient), and $ctrl_{CCB}$(C3)= ISV (represents the administrator).

- The associated place graph in this case is represented in figure 7:

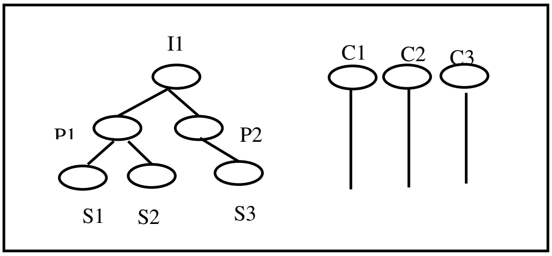


Fig. 7. Place graph of Cloud Health

- e1, e2, e3, e4, e5, e6 ∈ E_{CGB}, with each edge representing a connection between cloud customers and cloud services.
- The associated link graph, in this case, models cloud system connectivity (see figure 8):

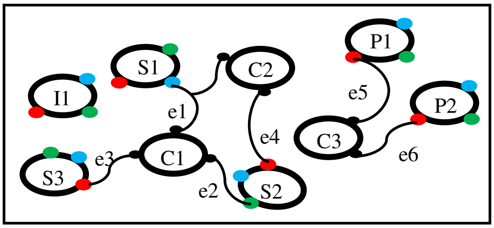


Fig. 8. Link graph of Cloud Health

Obviously, depm(-) function; defined in section 4.4, returns in our case the following values: depm(S1)=Public (edge e1 in figures 6 and 8), depm(S2)=Hybrid (edge e2 and e4 in figures 6 and 8), depm(S3)=Private (edge e3 in figure 6 and 8), depm(P1)=Private (edge e5 in figures 6 and 8), and depm(P2)=Private (edge e6 in figures 6 and 8).

Whenever a PaaS (P1) becomes unavailable for maintenance reasons, reconfiguring S1 service can be applied in order to migrate S1 to P2 using the SLR reaction rule (defined in section 4.5) which is denoted as follows:

`[I1.(P1.(S1|S2)|P2.S3)→I1.(P1.S2|P2.(S1|S3))]`

6 Conclusion

Bigraphical Reactive Systems (BRS) have been adopted as a formal model for cloud computing architecture specification. Two different bigraphs have been associated to both cloud services and customers, by enriching them with new sorts of nodes and ports. Their juxtaposition (CSB and CCB) gives rise to Cloud General Bigraph. The defined bigraphs allow developers to correctly reason about all cloud computing features, including modeling, composition, scheduling, monitoring and reconfiguration.

Particularly, we have shown how this model provides a flexible conceptual framework where cloud deployment models can be naturally defined. A nice consequence is that relationships between cloud services and cloud customers have been exploited to formally define cloud architecture reconfiguration via a set of reaction rules. Our ongoing work will focus on validating the proposed model by verifying some cloud computing inherent properties. BigMC, a Bigraphical Model Checker [18] designed to operate on Bigraphical Reactive Systems (BRS), will be used to formally model check the chosen properties.

References

1. Mell, P., Grance, T.: The nist definition of cloud computing. Technical Report 800-145, National Institute of Standards and Technology (NIST), Gaithersburg, MD (2011)
2. Marks, E.A., Lozano, B.: Executive's Guide to Cloud Computing. John Wiley and Sons Inc., Hoboken (2010)
3. Armbrust, M., Fox, A., Grith, R., Joseph, A.D., Katz, R.H., Konwinski, A., Lee, G., Patterson, D.A., Rabkin, A., Stoica, I., Zaharia, M.: Above the clouds: A berkeley view of cloud computing. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley (2009)
4. Milner, R.: The Space and Motion of Communicating Agents. Cambridge University Press (2009)
5. Dong, H., Hao, Q., Zhang, T., Zhang, B.: Formal discussion on relationship between virtualization and cloud computing. In: 2010 International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), pp. 448–453 (2010)
6. Grandison, T., Maximilien, E., Thorpe, S., Alba, A.: Towards a formal definition of a computing cloud. In: 2010 6th World Congress on Services (SERVICES-1), pp. 191–192 (2010)
7. Luo, S.X., Liu, F.M., Ren, C.L.: A hierarchy attribute-based access control model for cloud storage. In: 2011 International Conference on Machine Learning and Cybernetics (ICMLC), vol. 3, pp. 1146–1150 (2011)
8. Adamov, A., Hahanov, V.: A security model of individual cyberspace. In: 2011 9th East-West Design Test Symposium (EWDTS), pp. 169–172 (2011)
9. Freitas, L., Watson, P.: Formalising workflows partitioning over federated clouds: Multi-level security and costs. In: 2012 IEEE Eighth World Congress on Services (SERVICES), pp. 219–226 (2012)
10. Binz, T., Fehling, C., Leymann, F., Nowak, A., Schumm, D.: Formalizing the cloud through enterprise topology graphs. In: 2012 IEEE 5th International Conference on Cloud Computing (CLOUD), pp. 742–749 (2012)
11. He, R., Niu, J., Hu, K.: A novel approach to evaluate trustworthiness and uncertainty of trust relationships in peer-to-peer computing. In: The Fifth International Conference on Computer and Information Technology, CIT 2005, pp. 382–388 (2005)
12. Fitch, D.F., Xu, H.: A petri net model for secure and fault-tolerant cloud-based information storage. In: SEKE, Knowledge Systems Institute Graduate School, pp. 333–339 (2012)
13. Fang, X., Wang, M., Wu, S.: A method for security evaluation in cloud computing based on petri behavioral profiles. In: Yin, Z., Pan, L., Fang, X. (eds.) Proceedings of The Eighth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA). AISC, vol. 212, pp. 587–593. Springer, Heidelberg (2013)

14. Hu, L., Ying, S., Jia, X., Zhao, K.: Towards an Approach of Semantic Access Control for Cloud Computing. In: Jaatun, M.G., Zhao, G., Rong, C. (eds.) Cloud Computing. LNCS, vol. 5931, pp. 145–156. Springer, Heidelberg (2009)
15. Howard, F., George, S.: Formal methods in model-driven development for service oriented and cloud computing (2010)
16. Sim, K.M.: Agent-based cloud computing. IEEE Transactions on Services Computing 5, 564–577 (2012)
17. Di Cosmo, R., Zacchiroli, S., Zavattaro, G.: Towards a Formal Component Model for the Cloud. In: Eleftherakis, G., Hinchey, M., Holcombe, M. (eds.) SEFM 2012. LNCS, vol. 7504, pp. 156–171. Springer, Heidelberg (2012)
18. Perrone, G., Debois, S., Hildebrandt, T.T.: A model checker for bigraphs. In: Ossowski, S., Lecca, P. (eds.) SAC, pp. 1320–1325. ACM (2012)