Towards a multi-scale modeling for architectural deployment based on bigraphs

Amal Gassara, Ismael Bouassida Rodriguez, Mohamed Jmaiel

Towards a multi-scale modeling for architectural deployment based on bigraphs

Amal Gassara[1], Ismael Bouassida Rodriguez[1,2,3], and Mohamed Jmaiel[1]

[1] ReDCAD, University of Sfax, B.P. 1173, 3038 Sfax, Tunisia
{amal.gassara,bouassida}@redcad.org
mohamed.jmaiel@enis.rnu.tn
[2] CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France
[3] Univ de Toulouse, LAAS, F-31400 Toulouse, France

Abstract. With the evolution of distributed systems in size and complexity, software deployment remains a challenging task. Despite the existence of several approaches, most of them use informal models that lack a solid mathematic foundation. In this paper, we propose a bigraphical based approach for modeling and formalizing the deployment of distributed applications. This approach relies on multi-scale modeling. So, we start by modeling the first scale with a bigraph. This bigraph is enriched, through a series of reaction rules, until reaching the last scale that represents the deployment architecture.

Keywords: Deployment, Multi-scale modeling, Bigraphs

1 Introduction

With significant advances in software development, software applications become more and more complex, and distributed over a large network. These applications need to deal with hardware constraints and user requirements during the execution. Thus, software components must be placed on the suitable hosts among the distributed target environment to run the application properly. We called this process software deployment.

By placing software components on hardware nodes, we can have several deployment architectures. So, it is necessary to select the more efficient one in terms of QoS. Consequently, the designer needs to specify the possible deployment architectures in order to select the suitable one. This process remains a challenging task. In this paper, we focus on the modeling of the deployment architectures. In future work, we will analysis these models in qualitative and quantitative way allowing the selection of the appropriate deployment architecture.

Despite the efficiency of existing models, most of them are informal and lack of a solid mathematic foundation. Since bigraphs have a highly logical algebraic language, we use it as a formal model. So, the aim of this paper is to propose a bigraphical based approach for modeling and formalizing the deployment for distributed applications. We follow a multi-scale modeling approach. So, we start with modeling the first scale which is defined by a bigraph. This bigraph is enriched using bigraphical reactive system (BRS) to reach the scales one by one. At the last scale, we obtain bigraphs that represent the deployment architectures. The latter includes hosts and software components. The enriching rules are defined through a series of reaction rules.

The remainder of this paper is organized as follows. In Section 2, we explain our proposal for the formal modeling of deployment architecture. In this section, we present also overviews of bigraphs and multi-scale modeling. Then, we present in the section 3 a case study called "Smart Home" which illustrates the feasibility of our approach. We briefly review most related work in Section 4. Finally, Section 5 concludes this paper and presents future work.

2 The proposed approach

We propose an approach aiming the modeling and the formalizing of deployment for distributed applications. This approach performs three steps to generate deployment architectures. In this paper, we focus only on the two first steps.

- **Step 1: Modeling** The designer starts with defining the different scales. Then, he models the first scale using bigraphs.
- **Step 2: Enriching** The models specified by the designer are enriched by applying reaction rules to reach the scales one by one. At the end of this step, we obtain the set of the possible deployment architectures.
- **Step 3: Selecting** Each deployment architecture obtained at the previous step is quantified in order to select the suitable one.

2.1 Overview of bigraphs

The Static Structure Bigraphs [1], proposed by Milner, formalise distributed systems by emphasising both locality and connectivity. We use Fig. 1 to introduce bigraphs informally. A bigraph consists principally of hyperedges and nodes which can be nested and have ports. Each hyperedge can connect many ports on different nodes (for example, $v_0, v_1\, and\ v_2$ are joined by e_1). A bigraph combines two graphical structures, a *place graph* and a *link graph*, hence the term bigraph. Fig 2 depicts the *place* and *link graphs* of the bigraph G.

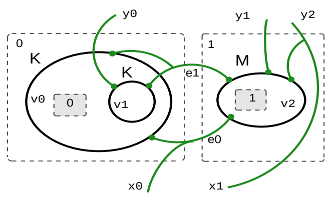

Fig. 1: A Bigraph G

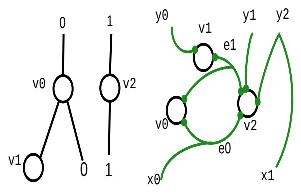

Fig. 2: *place* and *link graphs*

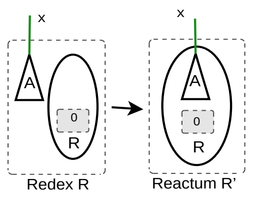

Fig. 3: A reaction rule

Place graph: The *place graph* is a hierarchical tree that describes the locality of the nodes. In this graph, branches establish the nesting relationship of nodes in the bigraph. Trees of are rooted by regions represented by dashed rectangle. Within the *place graph*, in addition to nodes and regions, there can also be sites, represented as grey rectangles. A site is a hole that can host new nodes.

Link graph: The *link graph* is a hyper-graph that describes the connectivity of nodes. Within this graph, there can be outer names (Fig. 2 y_0, y_1, y_2) and inner names (Fig. 2 x_0, x_1) represented as open links. These names define the connection points at which coincident names may be fused to form a single link. So, they give bigraphs the possibility to be composed by joining the inner names of one bigraph with the corresponding outer names of another bigraph.

Control and Signature: Each node in the bigraph is assigned a control. Controls (in this example K and M) indicate the node ports' number through the arity and how it behaves dynamically through the status which is either active or passive or atomic. An atomic node cannot contain any node and a non atomic node can be active or passive which means whether reactions may take place within the node. We can use the notation "X-node", which means a node that has been assigned the control X. The set of controls forms the signature.

The Dynamic Structure A BRS is a set of bigraphs and a set of reaction rules that may be applied to rewrite these bigraphs. Each reaction rule consists of two bigraphs: a *Redex R* and a *Reactum R'*. The application of the rule consists of identifying the image of *R* in a bigraph and replacing it by the corresponding *R'*. For example in Fig. 3, the rule allows an A-node to enter a R-node which is placed in the same region. The site (grey rectangle) in the *Redex* represents all other possible occupants of the R-node which are unchanged after applying this rule. The graphical representation used above is handy for modeling, but unwieldy for reasoning. Fortunately, bigraphs have an associated term language [2]. The corresponding algebraic expression (using details in table 1) of this rule is:

$$A_x \mid R.d_0 \rightarrow R.(A_x \mid d_0)$$

Table 1: The term language for Bigraphs

Algebraic expression	Meaning
U||V	Juxtaposition of roots
U|V	Juxtaposition of nodes
U.V	Nesting (U contains V)
K_x	K-Node linked to an outer name x
d_i	Site numbered i
1	The *barren* (empty) root
/x.U	U with outer name x replaced by an edge

2.2 Multi-scale modeling

Our approach is based on multi-scale modeling [3]. In fact, a scale is a generic model that provides additional details of the design and describes a level or a layer in a system. Multi-scale methodologies are based on the fundamental principle: model each phenomenon across the most relevant. For this, these methodologies have two key points: the first is to distinguish between different scales and the second is to model the relationships between these different scales.

Multi-scale modeling with Bigraphs In our approach, a scale is represented as a bigraph, where nodes correspond to deployment nodes (i.e., physical environment, hosts, devices, etc) or software components, edges represent interaction between linked nodes. Moreover, the transition from one scale to another is considered as bigraphical reactive system. This transition is an enriching performed through a series of **meta-reaction** rules. In fact, a meta-reaction rule contains nodes having a variable control (i.e., a variable can represent any control from the signature). Thus, the meta-reaction rule can be instantiated to several ones with different controls.

3 Case study: Smart Home

In order to illustrate our approach, we consider an example of an M2M application named "Smart Home" denoted in the Fig. 4. A smart home is composed of rooms. Each room can be equipped with heterogeneous devices (sensors like thermometer, presence sensor, light sensor, etc and actuators like air conditioner, lamp, etc). Between these devices, there is a need of communication. Sensors monitor and record information related to the environment such as rooms luminosity, human presence, temperature, etc. These information are transmitted to a control unit. Once received, the control unit, by analyzing them, makes the appropriate decisions to configure the devices and propagate these decisions. Now we apply our approach on Smart Home.

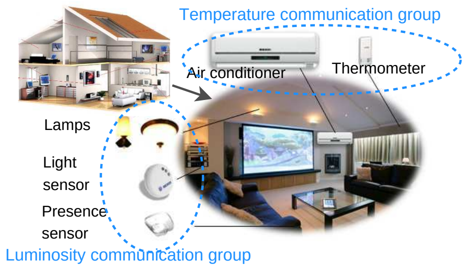
Temperature communication group
Air conditioner
Thermometer
Lamps
Light sensor
Presence sensor
Luminosity communication group

Fig. 4: Smart Home

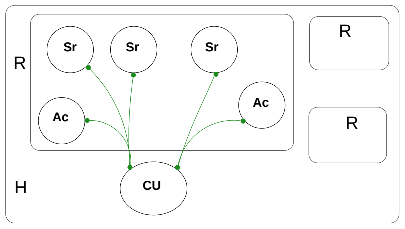

Fig. 5: Deployment infrastructure scale

3.1 Step 1: Modeling

This step allows the designer to determine the application scales. Until now, our approach enables the deployment of the software entities ensuring the communication of an application. So, we propose three scales: the deployment infrastructure scale, the communication scale and the deployment scale. After that, the designer models the first scale using a bigraph.

Scale 0: Deployment infrastructure scale This scale includes physical environment, hosts, devices, etc. The bigraph defined in the Fig. 5 illustrates this scale for a smart home. The nodes H and R represent the home and a room respectively. Whereas, the nodes CU, Sr and Ac represent the control unit, a sensor and an actuator respectively. Initially, these nodes are empty. A hyperedge depicts a need of communication. For example, the hyperedge at right depicts the temperature communication group between the air conditioner (Ac-node) and the thermometer (Sr-node) and the control unit. The other hyperedge depicts the luminosity communication group.

3.2 Step 2: Enriching

Scale 1: Communication scale This scale represents explicitly the entities that take part in the communication. Each communication group is formed by a set of senders and receivers. So, the bigraph defined in the first step (Fig. 5), is enriched by applying three meta-reaction rules: R1.1, R1.2 and R1.3.

Rule to add a sender: R1.1 This rule consists of nesting a sender (S-node) in each empty sensor (Sr-node depicting a thermometer, a presence sensor or a light sensor) having an outer name x. In fact, a communication group is defined by an hyperedge that links nodes with the same outer names (i.e., outer names are not represented explicitly in the Fig. 5 because they are joined to form hyperedges). Then, we nest in this S-node an x-node to mark its communication group. For lack of space, we present the reaction rules only with algebraic expressions.
R1.1: $Sr_x.(1) \rightarrow Sr_x.(S.x)$

Rule to add a receiver: R1.2 This rule consists of nesting a receiver (R-node) in each empty actuator (Ac-node depicting a lamp or an air conditioner) having an outer name x. Like with the rule R1.1, we nest in the R-node an x-node.
R1.2: $Ac_x.(1) \rightarrow Ac_x.(R.x)$

Rule to add a pair of sender and receiver: R1.3 This rule allows to add a pair of sender and receiver in the empty control unit (CU-node) for each communication group (i.e., outer names x and y). We also nest an x-nodes in the first pair and y-nodes in the second one.
R1.3: $CU_{x,y}.(1) \rightarrow CU_{x,y}.(S.x \mid R.x \mid S.y \mid R.y)$

The meta-reaction rules R1.1, R1.2 and R1.3 are instantiated by changing the name of the control x according to the outer name. In our example, we instantiate R1.1 and R1.2 twice by replacing x with $g0$ then $g1$ for the temperature and the luminosity communication group respectively. These rules are applied several times until there are no more empty devices (Sr and Ac nodes). We instantiate R1.3 once by replacing x with $g0$ and y with $g1$ and the intantiated rule is applied one time.

The Fig. 6 shows the application effect of the instantiated rules on the bigraph given on the Fig. 5. In this bigraph, there are S-node in sensors, R-node in actuators and both of them in the control unit. The S-nodes and the R-nodes belonging the temperature communication group, contain $g0$-node. Whereas, those belonging the luminosity communication group, contain $g1$-node .

Scale 2: Deployment scale This scale represents the middleware components that ensure the communication between the application components. Here, we use the Event-Based Communications (EBC) which provides three types of entities: *event producers* (EP), *event consumers* (EC) and *channel managers* (CM). The EP and EC are connected to CM. The EP sends data to the CM to which they are connected. The CM returns a copy of the received data to all the EC which are connected to it. Therefore, we enrich the bigraph obtained at the previous scale (defined in Fig. 6) by applying the following meta-reaction rules:

Rule to add a CM: R2.1 This rule enables to add a CM-node to each communication group (e.g. a set of juxtaposed nodes linked by one hyperedge). The CM-node is placed in one node belongs to the communication group. It contains also a nested node to mark its communication group.
R2.1: $/x\ Y1_x \mid\mid Y2_x \mid\mid ... \mid\mid Yn_x \rightarrow /x\ Y1_x \mid\mid Y2_x \mid\mid ... \mid\mid (Yn_x \mid CM.x)$

Rules to add an EP and an EC: R2.2 and R2.3 These rules enable to add an EP-node to each sender and EC-node to each receiver. The x-node indicates the communication group to which the receiver or the sender belongs.
R2.2: $S.x \rightarrow S.EP.x$
R2.3: $R.x \rightarrow R.EC.x$

Rules to link EPs and ECs with CM: R2.4 and R2.5 The meta-reaction rule R2.4 allows to link an EP and a CM that belong to the same communication group (i.e, having a nested x-node). whereas R2.5 allows to link an EC and a CM.
R2.4: $EP.x \mid\mid CM.x \rightarrow /y\ EP_y \mid\mid CM_y.x$
R2.5: $EC.x \mid\mid CM.x \rightarrow /y\ EC_y \mid\mid CM_y.x$

The above meta-reaction rules are instantiated by replacing x with $g0$ then $g1$ for the temperature and the luminosity communication group respectively. The Fig. 7 shows the application effect of the set of these instantiated rules on the bigraph given on the Fig. 6. For the luminosity communication group, we find EP-node nested in each sender and EC-node nested in each receiver belonging this group. All of them are connected to the CM-node which is nested in the CU-node. Also, for the temperature communication group, all the EP-nodes and the EC-nodes belonging this group are connected to the CM-node which is nested in the Ac-node. This bigraph is one of a set of possible bigraphs obtained after applying these rules due to the choice of the channel manager placement (i.e., the CM-node is deployed on one node belongs to the communication group).

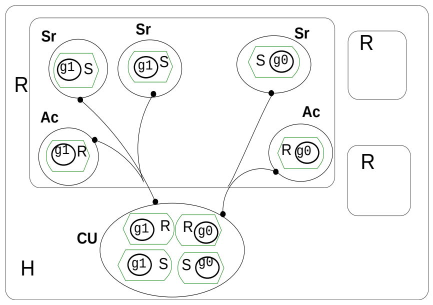

Fig. 6: The communication scale

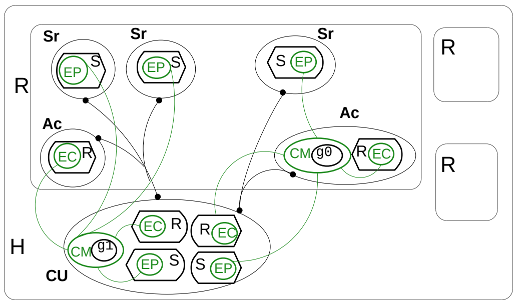

Fig. 7: The deployment scale

4 Related Work

Current research studies on component deployment propose various approaches.

Architecture-based approaches These approaches use Architecture Description Langauge (ADL). The work[4] presents an ADL extension for specifying a context-aware deployment. This deployment is performed in a propagative way and is driven by constraints put on the resources of the target hosts. However, it does not perform the automatic deployment planning and optimization.

MDA-based approaches These approaches use usually OMG D&C specification [5] which offers three models. The *component model* defines descriptors for components and configurations, the *target model* defines descriptors for the target site on which applications can be deployed and the *execution model* defines the *DeploymentPlan*, which describes deployment decisions. It defines an ExecutionManager which executes application according to this deployment plan.

Some frameworks have been developed on the top on this approach to support component deployment like DAnCE [6], Dacar [7] and Deployment factory [8]. DAnCE is a QoS-enabled Component Deployment and Configuration Engine targeted for DRE systems. This framework deals only with CORBA Component Model. Whereas Deployment Factory is an unified environment for deploying component based applications. This framework does not deal with reconfigurable systems. However, Dacar is a model-based framework for deploying autonomic software distributed systems. Some MDA-based approaches use UML including [9]. This work proposes a UML extension named “DM profile” ensuring a high-level description for modeling the deployment and its management in distributed application. All these research activities do not focus on deployment planning and deployment optimization.

5 Conclusion and future work

In this paper, we have presented an approach for modeling and formalizing the deployment. We have proposed a multi-scale modeling approach based on bigraphs and bigraphical reactive system. It performs three scales: deployment infrastructure scale, communication scale and deployment scale. The first scale is given by the designer. Whereas the other scales are obtained using a BRS. So, we define a series of meta-reaction rules ensuring the transition between scales. These meta-reaction rules are instantiated according to the bigraphical models specified by the designer.

In future work, we aim to generalize this contribution and alter the enriching rules to generic ones. Besides, we plan to define more scales for modeling and accomplish the third step of our approach (selecting a deployment architecture).

Acknowledgments. This research is supported by the ITEA2's A2NETS (Autonomic Services in M2M Networks) project [4].

References

1. Jensen, O.H., Milner, R.: Bigraphs and mobile processes. Technical Report UCAM-CL-TR-580, University of Cambridge, Computer Laboratory (February 2004)
2. Birkedal, L., Debois, S., Elsborg, E., Hildebrandt, T., Niss, H.: Bigraphical models of context-aware systems. In Aceto, L., Ingólfsdóttir, A., eds.: Proceedings of the 9th International Conference on Foundations of Software Science and Computation Structure (FoSSaCS'06). Volume 3921 of Lecture Notes in Computer Science., Springer-Verlag (March 2006) 187–201
3. Khlif, I., Hadj Kacem, M., Drira, K.: Une approche de description multi-échelles et multi points de vue pour les architectures logicielles dynamiques. In: Conférence francophone sur les Architectures Logicielles, Montpellier, France (May 2012)
4. Hoareau, D., Mahéo, Y.: Constraint-based deployment of distributed components in a dynamic network. In: Proceedings of the 19th international conference on Architecture of Computing Systems. ARCS'06, Berlin, Heidelberg, Springer-Verlag (March 2006) 450–464
5. Object Management Group, I.: Deployment and configuration of component-based distributed applications specification, version 4.0 (April 2006)
6. Deng, G., Balasubramanian, J., Otte, W., Schmidt, D.C., Gokhale, A.S.: Dance: A qos-enabled component deployment and configuration engine. In: Component Deployment. (November 2005) 67–82
7. Dubus, J., Merle, P.: Towards Model-Driven Validation of Autonomic Software Systems in Open Distributed Environments. In: Workshop M-ADAPT, in conjunction with ECOOP 2007, Berlin, Allemagne (July 2007)
8. Hnetynka, P.: A model-driven environment for component deployment. In: Proceedings of the Third ACIS Int'l Conference on Software Engineering Research, Management and Applications. SERA '05, Washington, DC, USA, IEEE Computer Society (August 2005) 6–13
9. Miladi, M.N., Krichen, F., Jmaiel, M., Drira, K.: A UML based deployment and management modeling for cooperative and distributed applications. In: Proceedings of the ACIS International Conference on Software Engineering, Management and Applications (SERA 2010), Montreal, Canada (May 2010) 16p.

[4] `https://a2nets.erve.vtt.fi/`