Formal Modeling IoT Systems on the Basis of BiAgents* and Maude

Souad Marir, Belala Faiza, Hameurlain Nabil, St Souad

Formal Modeling IoT Systems on the Basis of BiAgents* and Maude

1st Marir Souad
LIRE Laboratory ; LIUPPA Laboratory
Constantine 2 University; UPPA University
Constantine, Algeria
Pau, France
souad.marir@{univ-constantine2.dz; univ-pau.fr}

2nd Belala Faiza
LIRE Laboratory
Constantine 2 University
Constantine, Algeria
faiza.belala@univ-constantine2.dz

3rd Hameurlain Nabil
LIUPPA Laboratory
UPPA University
Pau, France
nabil.hameurlain@univ-pau.fr

***Abstract*—The Internet of Things is an emerging worldwide ecosystem in which smart devices interact to build smart homes, smart cities, etc. In such context, plethora of research efforts are oriented to networking and devices; analyzing and formalizing IoT systems are still in their earliest state. In this paper, we propose a BiAgents* (Bigraphical Agents) model to provide a formal description for IoT systems structure and behavior. In addition, we encode the BiAgents* specification into Maude language to enable an autonomic execution of the IoT systems' behaviors. The proposed approach is illustrated and evaluated through an example (an intelligent case of collision avoidance system).**

***Index Terms*—IoT, Formalism, BiAgents*, Maude**

I. Introduction

The internet of things (IoT) is a world full of sensors, actuators, robots and computers that are able to communicate in a larger network than the internet. This technology is entering in many fields as transportation, health care, smart homes and even industry, making them adaptable and responsive systems and increase their efficiency. The systems involved are composed of heterogeneous but interacting components (cars, smartphones, gatewaysetc) both individually and collectively providing original services [1]. Despite a decade of research, the IoT is still into an emergent phase, actually, there are some IoT systems/devices that provide conventional computing services mainly designed for static environments with some interactions [1]. Developing IoT systems, in the right way, may be difficult, because of their complexity in a matter of low-level communication protocols, dynamic updates and preservation of the data security.

Nowadays, IoT systems are evolving in a matter of structure and behavior; thus, proposing a model assuring the quality of IoT systems remains a challenging task. Formal methods are mathematical techniques used for developing reliable and verified systems. They are proved especially useful to build models of systems that are correct by construction. Developing reliable and safe smart systems is necessary for good smart city solutions [2]. In this regard, and to help the construction of valid configurations of an IoT system, we propose a formal, executable, model based on the theory of *bigraphs* [3] and formal agents. We note this combination of these formalisms **BiAgents***. We choose to model the physical part of an IoT system using the formalism of Bigraphs for its suitability to abstract locality (place graph) and connectivity (link graph) of complex systems. In addition, we can express the evolution of Bigraphs using Reaction Rules. This allows defining the behavior and the states of a dynamically evolving system as an IoT one. Moreover, Bigraphs have simplicity of understanding that permits interdisciplinary exchange between experts in IoT and in formal methods [4]. An IoT system has also a virtual aspect, an intelligent part that manages the different physical components to ensure an intelligent execution. To model this part, we base our approach on the *BiAgents* [5] (Bigraphical agents) and adapted it to IoT systems by allowing the *agents* to analyze the observed environment and to interact with each other before making any decision. Using the formalism of **BiAgents***, we model an Intelligent Collision Detection System and execute this model using **Maude**. We choose this case of study for being a typical IoT system, where all the parts (physical and virtual) are solicited. We encode the BiAgents* specifications into Maude language to make an automatic execution of the BiAgents* specification. The existing tools based on BRS such as BigraphER and BPL Tool are not suitable for this BRS extension. Particularly, BRS model-checker BigMC permits the formal verification of safety property. Nonetheless, possible verifications rely on very limited predefined predicates [6]. We also implement this specification with the aim of the validation of relevant properties as correctness and safety in the future works. We use Maude, first of all, because of its **simplicity**. Indeed, basic programming expressions in the Maude language are simple and easy to understand. These expressions which have a simple rewrite interpretation are either equations or rewrite rules in which, the left side represents a pattern that can be replaced by another pattern in the right side. Secondly, for its **expressiveness**. the Maude language makes it possible to naturally represent deterministic and non-deterministic systems through functional modules and system modules. The deterministic calculation is implemented using the equations in functional modules. On the other hand, the non-deterministic calculation is represented with rewriting rules in system modules. Finally, The **performance** aspect of Maude has been supported by successive Maude implementations. Maude is competitive in

terms of run time, where several rewrite rules can be run per second [7].

In this paper, we make the following research contributions:

- We describe the extension of Bigraphs noted BiAgents* to allow the specification of IoT systems.
- We specify a BiAgents* model for IoT systems and apply it to a case study (case of collision detection).
- We implement this specification into Maude and execute the proposed model.

The remainder of the paper is structured as follows. In Section II, we define what an IoT system is and we describe the system taken as a case of study. In Section III, we summarize different approaches of modeling IoT systems to position our contributions compared to others. In Section IV, we define the different parts of the BiAgents* formalism and describe the specification of the case study. In Section V, we describe the Maude language and explain how we guarantee the passage from the BiAgents* specification to Maude. Finally, we conclude this work in section VI.

II. Running example

In this section, we define what is an IoT system and present the example of the Intelligent Collision Detection System (ICDS). This example is used to illustrate the proposed model. The encoding of its specification is the Maude input for the execution of the BiAgents* model.

A. Definition of IoT Systems

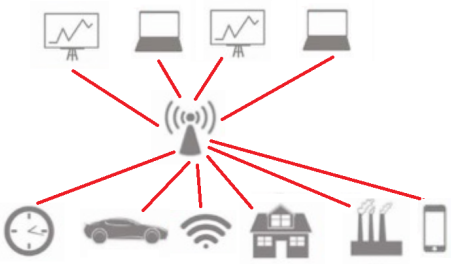

Fig. 1. IoT architecture

The IoT paradigm is based on various types of smart devices with communication and networking capabilities, using and affecting the environment around us. According to CISCO predictions, the number of Internet-connected devices overtook the human population in 2010, and will be about 30 billion by 2023 [8]. Figure 1 illustrates the common architecture of today's IoT systems from a high level perspective. Three main components involved are: sensing devices (things), the communication network and back-end IoT applications. Sensing devices capture data from the actual environment. These data are later used by IoT applications (e.g. smart transportation, agriculture, video surveillance, healthcare etc.) to provide a desirable service to end users.

Once the raw data generated by a sensing device are captured, the information are formatted and sent through the network to the application part of the system. After an appropriate analysis, the useful information is finally delivered to the end user. That may be commercial or an industrial user, or another device in M2M (Machine-to-Machine) workflow.

Unfortunately, such systems are not always well designed, which may result in poor behavior, damaging the used infrastructure. This must be resolved in order to fully exploit the opportunities offered by heterogeneous access networks in IoT environments.

B. ICDS example

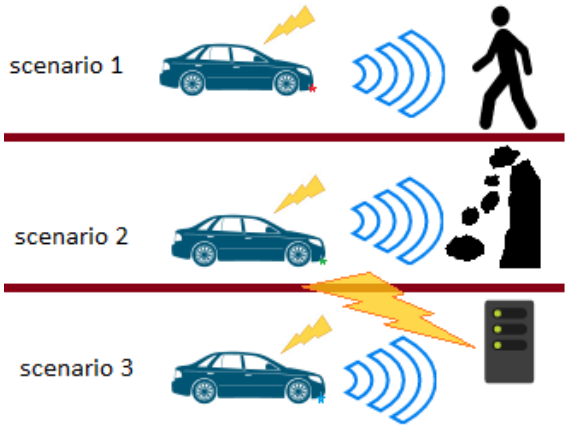

Fig. 2. Illustration of the ICDSystem

Smart transportation is one of the most significant IoT systems. There are many reasons for the adoption of IoT in this field, from social to economic.

Some common services refer to traffic management, safety and infotainment. Traffic management systems examine traffic behavior and events to smartly coordinate the different vehicles on the road. Safety services aim to minimize accidents for pedestrians and vehicles. Infotainment services focus on classic IP applications like video streaming, e-mail and social networks. Tesla Motors and Google are making plenty of endeavors in developing a software provided with a "hands free" operation of the car. These cars will not need human contribution in the use of the primary driving features such as acceleration, brakes and steering [9].

In this paper, we choose one of these systems to model as a case of study, The Intelligent Collision Detection System (ICDSystem).

We define the ICDSystem as a system of prevention for drivers against different types of collision. In this system, data is captured from two cameras and radar. After that, the application part of the system treats the data and acts accordingly. In the end, if the system detects a danger, a sound alert specifying the type of danger will warn the driver.

There are three possible scenarios of execution of the system as illustrated in the Figure 2:

- **Scenario 1** If the danger concerns only the actual driver (presence of a pedestrian), the system will only alert the local driver.
- **Scenario 2** If the danger is more general (natural disaster), the system alerts the actual driver and sends to a distant server the alert in order to warn the systems around.
- **Scenario 3** If the driver may be concerned by a danger afar, the system receives a distant signal that warns him.

III. RELATED WORK

During the last few years, research interest for IoT systems has grown increasingly important, resulting in numerous research works covering different aspects of IoT systems. In this section, we review and discuss some research works that have addressed IoT systems design and their correctness.

Authors of [10] proposed a formalism called IoT-SEC framework. They made a formal model that they transform into Prism for model checking. Then they use PCTL to express functional requirements. After that, using probabilistic model checking, they verify some properties of security at different levels according to an example. Authors of [11] suggested a scalable model for the verification of the property of safety, using the StateMate model for the description of an IoT system. They transform the resulting model into a stamped colored Petri net and then, the result is translated into xml code in order to be usable as an input to the CPN tool which is based on the LTL logic. Authors of [12] proposed a face identification and resolution framework using the Fog computing paradigm. They proposed three schemes: the identity authentication, the data encryption, and the data integrity checking one for Fog computing, verifying the properties of confidentiality, integrity and availability using the BAN [1] (Burrows-Abadi-Needham) logic. In [13], authors came up with an abstract mathematical meta model for IoT using the formal method Event-B, and execute their model using the Rodin tool. In [14], authors proposed a modeling language called IoTGolog to define and evaluate IoT systems execution scenarios. Using this language, they modeled the communication between cars in a radius of twenty kilometers. In [15], authors supplied a network-based multidimensional structure. This structure is a layered one, with an object usage graph, with all the advantages of a layered model as heterogeneity. They experimentally verify the properties of scalability, heterogeneity and dynamicity by simulation. In [16], authors contributed with an agent based model in order to increase security, reliability and credibility in IoT models. Their approach is to install on each sensor node a TAEC (Trustworthy Agent Execution Chip), which provides an execution environment for agents. In [17] authors proposed an MDD based methodology (Model Driven Development) and a Service Oriented Architecture (SOA) in the purpose of reducing IoT system's development's time and cost. In [18], authors proposed a detailed layered IoT architecture based on the transparent computing, in order to build manageable and scalable applications. This architecture includes a large variety of resources: operating systems, applications and data allowing to define a manageable, interoperable and scalable IoT systems in a minimum amount of time. In [19], authors proposed the middleware ACOSO (Agent-based COoperating Smart Object) that supports interoperable and autonomous systems. This middleware uses JADE based agents, a capture and action manager on the environment, a knowledge base and a communication manager. Using OMNET ++, their simulation proved that a big number of objects may cause a lot of interferences. Finally, in [20], authors proposed a Fog architecture modeled using Bigraphical Reactive Systems (BRS). They showed the utility of this formalism in modeling complex systems as a Fog used system. This contribution is the closest to ours considering the formalism used. However, we add more precision using the sorting of bigraphs as presented in section IV. Regarding to these approaches, we propose a formal model based on BRS and agents paradigm allowing to specify both physical and virtual aspects of an IoT system. Bigraphs model the structure and the behavior of an IoT system with all its constraints. The agents' part manages the whole execution of such system, providing the intelligent side of a smart system. In order to facilitate a future verification, we execute automatically this model using Maude language [21].

[1]This logic determines if exchanged information is trustworthy and secured against eavesdropping

IV. BIAGENT* BASED APPROACH FOR MODELING

In this section, we present first the BRS (Bigraphical systems) to be familiarized with some basic concepts needed to understand the proposed model. Secondly, we introduce the BiAgents* model. We define it with its different aspects (**physical** and **virtual**). Finally, we present its semantics with the different interpretations of an IoT system using the proposed model.

A. BRS Overview

A BRS (*Bigraphical Reactive System*) [3] is a formalism used for modeling the temporal and spatial evolution of complex systems. It gives a graphical model that focuses on both connectivity and locality. A BRS is defined by a set of bigraphs and a set of reaction rules, which describes the dynamic evolution of the system by specifying the succession of bigraphs.

A bigraph has algebraic notations that are equivalent to the graphical ones. The parallel product (noted $\|$) depicts the juxtaposition of two bigraphs, the fusion (noted $|$) is the fusion of the elements of two bigraphs, the imbrication (noted .) is the insertion of a node into another and the identity (noted id) is the elementary bigraph (i.e., a region and a site). More details about Bigraphs can be found in [22].

Sorts are used to classify controls and links. A sorting discipline is a triple $\Sigma = \Theta, \mathcal{K}, \Phi$, where Θ is a non-empty set of sorts, K is a signature, and Φ is a set of formation rules. A formation rule is a set of properties satisfied by a bigraph.

Disjunctive sorts are written as $\widehat{ab}$, expressing that a node can either be of sort a or sort b.

A *Bigraphical Reactive System* (BRS) is a set of bigraphs and a set of reaction rules. Each bigraph represents a state of the system. Reaction rules define the execution of the system (by going from one state to another). A reaction rule R_i is a pair (R, R'), where R is called redex and R', reactum. R and R' are bigraphs that have the same interface. The execution of a system S is made by checking if R is present in the state St and by replacing it with R' to reach to a new state of the system St'. This is possible with applying the appropriate reaction rule R_i. This state transition is noted $St \rightarrow St'$.

There are many BRS extensions as *binding bigraphs* [23], *stochastic bigraphs* [24], *bigraphs with sharing* [25], *directed bigraphs* [26] and ***biAgents*** [5]. This formalism and its extensions can be used to model many systems in many fields as Cloud systems [27] [6], context aware systems [28], Fog systems [20],the deployment of software architecture [29], etc.

In this paper, we enrich the extension of BiAgents in order to fit well with the modeling of an IoT system with its aspects of smartness and communication. In a previous work [30], we defined the extension BCAM*4*IoT which extends the BiAgent formalism. In the present work, we add the notion of **buffer** for sending and receiving messages between agents, which is closer to an executable model.

B. Definitions

A part of our contribution consists in modeling the different aspects of locality, connectivity and behavior among other aspects of an IoT system. The model proposed is based on the formalism of BiAgents* which is an enriched version of the existing extension BiAgent.

Definition

An IoT system S_IoT can be defined by a BiAgent* $S_{IoT} = \mathcal{A}_{IoT} \bullet \mathcal{B}_{IoT}$ where:

- $\mathcal{B}_{IoT} = (\mathbb{B}, \mathcal{R}, \mathbb{U}, B_0, F)$:
  - $\mathbb{B}$ is the set of bigraphs representing the IoT system's states.
  - $\mathcal{R}$ is the set of reaction rules modeling the different possible transitions from a state to another.
  - $\mathbb{U}$ is the set of actions $R \times V_{\mathbb{B}}$ with $V_{\mathbb{B}}$ the set of nodes of each bigraph $B_i \in \mathbb{B}$ according to a specific part of the system (any element of the different layers of the system).
  - B_0 is the initial state of the system.
  - F is the transition function that the model uses to reach a new state.
- $\mathcal{A}_{IoT}$ is a set of a_{IoT} agents with :

$$a_{IoT} = (\mathcal{O}, \mathcal{U}, \mathcal{D}, obs, an, ctr, mgrt, int, buf, host_0)$$

  - $\mathcal{O}$ is the set of the different system's states (bigraphs) that the agent a_IoT observes.
  - $\mathcal{U}$ is the set of reaction rules that a_{IoT} can apply according to a node of the current bigraph, it represents the set of actions that a_{IoT} can do in a specific part of the system.
  - $\mathcal{D}$ is the set of decisions that the agent a_{IoT} can take after an analysis.
  - buf is a buffer which will be the holder of the received messages.
  - obs is the function of observing a system's state.
  - an is the function of analyzing an information.
  - ctr is the application of a decision taken by an agent after analysis.
  - $mgrt$ is the function of migration. This agent can move from any part of the system to another, and even though the layers.
  - int is the function of an interaction between agents.
  - $host_0$ is the initial host of the agent (a bigraphical node). There are three types of agents: Abstraction, Communication and Application one. According to the modeled IoT system, there can be many instantiations of each agent to make each one focus in a specific task.

C. BiAgents* semantics

The sorting discipline associated to S_IoT is a triple $\Sigma = \Theta, \mathcal{K}, \Phi$, where Θ is a non-empty set of sorts. KS_{IoT} is its signature, and Φ is a set of formation rules associated to the BiAgent* elements. Table I associates to each IoT concept the equivalent abstraction in the BiAgent* model. This consists of the control associated to each entity, its arity (number of ports) and its associated sort. Sorts are used to distinguish node types for structural purposes and constraints while controls identify states and parameters a node can have. We categorize the elements of an IoT system as follows: all interfaces are of sort e, all information are of sort I, formats are of sort F, protocols of sort P, subsystems of sort SS and layers of sort L. We present this categorization in Table I. We note here RI for Raw Information, this information can be an image or a sound. The action A is the triggering of an actuator as playing a sound.

Table II contains the formation rules Φi that draw constraints in the bigraphical specification construction. These rules precise the structural constraints over the bigraphical part of the model. Rule $\Phi 0$ specifies that we can only format an information I. Rule $\Phi 1$ permits to an action A to move through different parts of the system. Rules $\Phi 6$, $\Phi 4$ and $\Phi 0$ express the fact that an information must be formatted to be treated in the upper subsystems.

The behavior of the IoT system is specified as Bigraphical Reaction Rules that express the structural dynamicity of the system. Here, in Tables III and IV, we define a set of reaction rules followed by their algebraic form.

TABLE I
CONTROLS AND SORTS OF THE BIGRAPHICAL PART

IoT Element	Control	Arity	Sort
Input interface	In	1	e
Output interface	Out	1	e
In/out interface	InOut	1	e
Raw Information	RI	1	I
Extracted Information	EI	2	I
Communication Format	CF	2	F
Action Format	AF	2	F
Action	A	2	A
Internal Communication System	ICL	1	SS
External Communication System	ECL	1	SS
Generic Support	GS	1	SS
Specific Support	SS	1	SS
Image Recognition System	IRS	1	SS
Sound Recognition System	SRS	1	SS
Formatting System	FS	1	SS
Data Matching System	DMS	1	SS
External Communication Protocol	ECP	1	P
Packet Verification System	PVS	1	P
Internal Communication Protocol	ICP	1	P
Abstraction Layer	Ab	0	L
Communication Layer	Com	0	L
Application Layer	App	0	L

TABLE II
CONDITIONS OF FORMATION RULES Φi FOR THE BIGRAPHICAL PART

Rule	Description
$\Phi 0$	All children of F nodes have sort I
$\Phi 1$	All parents of P nodes have sort SS
$\Phi 2$	Parents of A nodes can have sort SS or e or L only
$\Phi 3$	All parents of e nodes have sort L
$\Phi 4$	All parents of F nodes have sort SS
$\Phi 5$	All parents of SS nodes have sort L
$\Phi 6$	Nodes of I nodes can't have a parent of sort SS

V. MODEL EXECUTION

To execute the proposed model and to observe its behavior, we provide an executable solution for the BiAgent* specification. Bigraphical Reactive Systems improved by agents are supposed to provide good meta-modeling bases to specify IoT systems' physical structure with its behavioral parts in addition to the virtual one. As far as we know, there exists no tool built around neither BRS, nor BiAgents, that enables to express necessary conditions in order to choose the right reaction rule. In this work, we opt to use the Maude language to tackle the limitation.

Maude [21] is a high-level formal specification language based on rewriting logic. It has been used to formally specify and execute context aware systems [28], Cloud systems [6], System of Systems (SoS) [31] and more

The defined BiAgents* model for IoT systems' structure can be encoded in a functional module, where the declared operations and equations define constructors that build the system's elements. Similarly, BiAgents* model for describing the behavior of an IoT system can be encoded in a system module. The execution of the system is represented by a set of rewrite rules R expressing bigraphical reaction rules

TABLE III
ICDSYSTEM CASE 1 REACTION RULES

Reaction Rule ID	Description
R1	The RI is sent to the ICL (Internal Communication Layer) in the purpose of being formatted.
R2	The RI is formatted according to the Internal Communication Protocols (ICP).
R3	The formatted information is transmitted to the General System (GS).
R4	The system sends the formatted information to the specific support (SS)
R5	The specific support (SS) looks for corresponding data from known dangers to the formatted captured information (CF).
R6	An action depending on the detected danger is requested.
R7	The action is transmitted to the Internal Communication System (ICS) to be carried out on the Current System (CS).
R8	An action is played in the different actuators (vibrator, screen and speaker).

TABLE IV
ICDSYSTEM CASE 1 REACTION RULES ALGEBRAIC FORM

RR ID	Algebraic form
R1	R1 ((C1.RI)|wifi|v|sc|sp)|id || ICL.ICP|id → (C1|wifi|v|sc|sp)|id || ICL.(ICP|RI)|id
R2	R2 ICL.(ICP|RI)|id → ICL.(ICP|(CF.RI))|id
R3	R3 ICL.(ICP|(CF.RI))|id ||(GS.(IRS|FS|SRS))|id → ICL.ICP|id ||S.(IRS|FS|SRS|(CF.RI)))|id
R4	R4 (GS.(IRS|FS|SRS|(CF.RI)))|(SS.DMS)→ (GS.(IRS|FS|SRS))|(SS.(DMS|(CF.RI)))
R5	R5 (GS.(IRS|FS|SRS|EI))|(SS.DMS)→ (GS.(IRS|FS|SRS))|(SS.(DMS|AF.EI|SI))
R6	R6 (GS.(IRS|FS|SRS|EI))|(SS.(DMS|(AF.EI)|SI)) → (GS.(IRS|FS|SRS|EI))| (SS.DMS) | (A.EI)
R7	R7 (ICL.ICP)| id ||(A.EI)|id →(ICL.(ICP|(A.EI)))|id || id
R8	R8 (ICL.(ICP|(A.EI)))|id ||(C1|wifi|v|sc|sp)|id → (ICL.ICP)|id ||(C1|wifi|v.IA1|sc.IA2|sp.IA3)|id

in addition to state change of each agent. Every change is triggered by a decision of an agent.

In this contribution, Maude executes the model by reuniting the principle of rewriting and executing the agents' way of thinking. As presented in Section IV, the BiAgent* model is divided into two principle parts: **Physical** and **Virtual** one. Formal agents specify the virtual one. We can represent the life cycle of an agent by the view illustrated in Figure 3. In this figure, the different states stand for: obs = observing; mgrt = migrating; An = analyzing; int = interacting; D = making a decision and RR is the triggering of a reaction rule.

At each step, the concerned agent executes this loop to master the structural evolution of the system. We encode the different states of each agent as shown in Table V. Precisely, we want to focus on the interaction operation by defining it as sending a message from an agent to the buffer of the receiver. The operations of sending and receiving are expressed in Maude as presented in Table V.

Regarding to the physical part of the BiAgent* Model, we encode its specification into Maude language as operations that

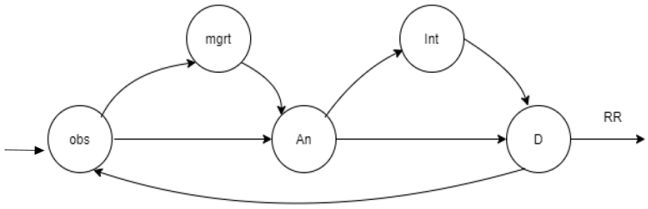

Fig. 3. View of an agent behavior

TABLE V
MAUDE SPECIFICATION OF BIAGENTS* AGENTS

BiAgent* Model	Maude Specification
Agents' states	
Observing	op abObs:->Agent[ctor].
	op comObs:->Agent[ctor].
	op appObs:->Agent[ctor].
Migrating	op abMgrt:->Agent[ctor].
	op comMgrt:->Agent[ctor].
	op appMgrt:->Agent[ctor].
Analysing	op abAn:->Agent [ctor].
	op comAn:->Agent [ctor].
	op appAn:->Agent [ctor].
Interacting	op abInt:->Agent [ctor].
	op comInt:->Agent[ctor].
	op appInt:->Agent[ctor].
Deciding	op abD:->Agent [ctor].
	op comD:->Agent [ctor].
	op appD:->Agent [ctor].
Agents's Interaction	
Sender	op int.Mes[_+_] : Agent Message Agent -> Agent [ctor] .
Receiver	op opAgent.Buf[_+_] : Agent Message Agent -> Agent [ctor] .
Agents' Messages	
op "sending" : ->Message [ctor] .	
op "ordering": ->Message [ctor] .	
op "executing":->Message [ctor] .	

permit the construction of an IoT system with all its layers and sub systems. Using these operations and the definitions of the different aspects of the agents, we were capable of executing the first scenario of the case study ICDS system. Given an initial state, where a camera captures an image as shown in Figure 4 the rewriting results show the result, which is emitting an alert according to the danger detected at the beginning.

```
rewrite in SpecModel : opIoTSystem< opLayer< Applayer >.P[opSubSys< GS >.P[IRS
    | FS | SRS] | opSubSys< SS >.P[DMS]].Ag[appObs],opLayer< ComLayer >.P[
    opSubSys< ICL >.P[ICP] | opSubSys< ECL >.P[ECP | PVS]].Ag[comObs],opLayer<
    AbLayer >.P[(wificard | speakers) |opInterface< camera >.P[Image]].Ag[
    abObs] > .
rewrites: 36 in 0ms cpu (0ms real) (121212 rewrites/second)
result IoTSystem: opIoTSystem< opLayer< Applayer >.P[opSubSys< GS >.P[IRS | FS
    | SRS] | opSubSys< SS >.P[DMS]].Ag[appObs],opLayer< ComLayer >.P[opSubSys<
    ICL >.P[ICP] | opSubSys< ECL >.P[ECP | PVS]].Ag[comObs],opLayer< AbLayer
    >.P[camera | wificard |opInterface< speakers >.P[opAction< Sound,Image
    >]].Ag[abObs] >

Maude>
```

Fig. 4. Rewrite execution of ICDS example

VI. CONCLUSION

In this paper, structural and behavioral aspects of an IoT system are modeled using BiAgents*. The judicious combination of the *Bigraphical Reactive System* formalism and formal Agents with the ability of observation, analysis and communication, permits to model different parts of an IoT system. Therefore, we showed how the BiAgents* model may specify both the virtual and structural parts of the IoT system. Moreover, for a better understanding, we bring closer an abstract model to reality by instantiating it into a concrete scenario of collision detection system. Finally, using Maude, we have executed the proposed model in order to prototype and analyze an IoT system. In the present work, we have been interested by showing how Maude syntax permits the natural definition of agents' states and their transition. In the next step, we plan to enlarge our BiAgents* model to provide the execution of more than one scenario (many strategies) and to turn to the concept of *Fog systems* [32].

REFERENCES

[1] R. Casadei, G. Fortino, D. Pianini, W. Russo, C. Savaglio, and M. Viroli, "Modelling and simulation of opportunistic iot services with aggregate computing," *Future Generation Computer Systems*, vol. 91, pp. 252–262, 2019.

[2] E. Alkhammash, "Formal modelling of owl ontologies-based requirements for the development of safe and secure smart city systems," *Soft Computing*, pp. 1–14, 2020.

[3] R. Milner, *The space and motion of communicating agents*. Cambridge University Press, 2009.

[4] B. Archibald, M.-Z. Shieh, Y.-H. Hu, M. Sevegnani, and Y.-B. Lin, "Bigraphtalk: Verified design of iot applications," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2955–2967, 2020.

[5] E. Pereira, C. Kirsch, and R. Sengupta, "Biagentsa bigraphical agent model for structure-aware computation," *Cyber-Physical Cloud Computing Working Papers, CPCC Berkeley*, pp. 1–13, 2012.

[6] K. Khebbeb, N. Hameurlain, F. Belala, and H. Sahli, "Formal modelling and verifying elasticity strategies in cloud systems," *IET Software*, vol. 13, no. 1, pp. 25–35, 2018.

[7] H. Sahli, *Cloud Systems Elasticity Modeling : Towards Formal Verification of their Behavior*. PhD thesis, 2017.

[8] Cisco, "Cisco annual internet report (20182023) white paper," March 2020. Available at https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html, [updated on March,9 2020].

[9] S. Tomovic, K. Yoshigoe, I. Maljevic, and I. Radusinovic, "Software-defined fog network architecture for iot," *Wireless Personal Communications*, vol. 92, no. 1, pp. 181–196, 2017.

[10] S. Ouchani, "Ensuring the functional correctness of iot through formal modeling and verification," in *International Conference on Model and Data Engineering*, pp. 401–417, Springer, 2018.

[11] V. Phartchayanusit and S. Rongviriyapanish, "Safety property analysis of service-oriented iot based on interval timed coloured petri nets," in *2018 15th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pp. 1–6, IEEE, 2018.

[12] P. Hu, H. Ning, T. Qiu, H. Song, Y. Wang, and X. Yao, "Security and privacy preservation scheme of face identification and resolution framework using fog computing in internet of things," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1143–1155, 2017.

[13] A. Jarrar, T. Gadi, and Y. Balouki, "Modeling the internet of things system using complex adaptive system concepts," in *Proceedings of the 2nd International Conference on Computing and Wireless Communication Systems*, pp. 1–6, 2017.

[14] S. R. Humayoun, Y. Dubinsky, and R. AlTarawneh, "Using iotgolog to formalize iot scenarios," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, pp. 234–238, IEEE, 2015.

[15] J. Jung, S. Chun, and K.-H. Lee, "Hypergraph-based overlay network model for the internet of things," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, pp. 104–109, IEEE, 2015.

[16] X. Xu, N. Bessis, and J. Cao, "An autonomic agent trust model for iot systems," *Procedia Computer Science*, vol. 21, pp. 107–113, 2013.
[17] C. M. Sosa-Reyna, E. Tello-Leal, and D. Lara-Alabazares, "Methodology for the model-driven development of service oriented iot applications," *Journal of Systems Architecture*, vol. 90, pp. 15–22, 2018.
[18] H. Guo, J. Ren, D. Zhang, Y. Zhang, and J. Hu, "A scalable and manageable iot architecture based on transparent computing," *Journal of Parallel and Distributed Computing*, vol. 118, pp. 5–13, 2018.
[19] C. Savaglio, G. Fortino, and M. Zhou, "Towards interoperable, cognitive and autonomic iot systems: An agent-based approach," in *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pp. 58–63, IEEE, 2016.
[20] A. Bouheroum, Z. Benzadri, and F. Belala, "Towards a formal approach based on bigraphs for fog security: Case of oil and gas refinery plant," in *2019 7th International Conference on Future Internet of Things and Cloud (FiCloud)*, pp. 64–71, IEEE, 2019.
[21] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martı-Oliet, J. Meseguer, and C. Talcott, "Maude manual (version 2.1)," *SRI International, Menlo Park*, 2005.
[22] R. Milner, "Bigraphs and their algebra," *Electronic Notes in Theoretical Computer Science*, vol. 209, pp. 5–19, 2008.
[23] T. C. Damgaard and L. Birkedal, "Axiomatizing binding bigraphs," *Nordic Journal of Computing*, vol. 13, no. 1/2, p. 58, 2006.
[24] J. Krivine, R. Milner, and A. Troina, "Stochastic bigraphs," *Electronic Notes in Theoretical Computer Science*, vol. 218, pp. 73–96, 2008.
[25] M. Sevegnani and M. Calder, "Bigrapher: rewriting and analysis engine for bigraphs," in *International Conference on Computer Aided Verification*, pp. 494–501, Springer, 2016.
[26] D. Grohmann and M. Miculan, "Directed bigraphs," *Electronic Notes in Theoretical Computer Science*, vol. 173, pp. 121–137, 2007.
[27] Z. Benzadri, C. Bouanaka, and F. Belala, "Big-caf: a bigraphical-generic cloud architecture framework," *International Journal of Grid and Utility Computing*, vol. 8, no. 3, pp. 222–240, 2017.
[28] T. A. Cherfia, F. Belala, and K. Barkaoui, "Towards formal modeling and verification of context-aware systems.," Citeseer.
[29] N. Benlahrache and F. Belala, "Towards formalising installation and reconfiguration tasks of aadl architecture," *International Journal of Communication Networks and Distributed Systems*, vol. 11, no. 4, pp. 431–452, 2013.
[30] S. Marir, F. Belala, and N. Hameurlain, "A formal model for interaction specification and analysis in iot applications," in *International Conference on Model and Data Engineering*, pp. 371–384, Springer, 2018.
[31] A. Seghiri, F. Belala, Z. Benzadri, and N. Hameurlain, "A maude based specification for sos architecture," in *2018 13th Annual Conference on System of Systems Engineering (SoSE)*, pp. 45–52, IEEE, 2018.
[32] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," *Journal of Systems Architecture*, vol. 98, pp. 289–330, 2019.