A Formal Model for Interaction Specification and Analysis in IoT Applications

Souad Marir, Faiza Belala, Nabil Hameurlain

HAL Id: hal-02507242
https://hal.science/hal-02507242

Submitted on 26 Nov 2020

A Formal Model for Interaction Specification and Analysis in IoT Applications

Souad MARIR[1,2], Faiza BELALA[1], and Nabil HAMEURLAIN[2]

[1] University of Constantine 2 –Abdelhamid Mehri, LIRE Laboratory, Constantine, ALGERIA {souad.marir,faiza.belala}@univ-constantine2.dz
[2] University of PAU, LIUPAA, Pau, FRANCE nabil.hameurlain@univ-pau.fr

Abstract. The Internet of Things (IoT) is a concept where connected entities can work and interact with each other in order to facilitate daily life. Although, many research efforts in the IoT realm have been to date devoted to device, networking and application service perspectives, formalization and analysis of IoT systems are still in their infancy. This paper introduces a new BRS-based approach aiming to support specification and verification of interaction and interoperability aspects in IoT systems. The proposed approach is based on a bigraphical-agent model that investigates the spatial structure of the IoT system and its logical structure defining the behaviour and interactions of its different entities. The Tree Query Logic (TQL) is used to formally express and verify some properties inherent to IoT systems

Keywords: IoT · BRS · Formal specification · Interaction model · TQL

1 Introduction

The Internet of Things (IoT) is the connection of heterogeneous objects, geographically separated, consuming and producing data, in order to offer services. Generally considered as intelligent, IoT systems facilitate considerably every day life of human beings, with a minimum of intervention, working on interoperable networks and reacting to their environment. Developing IoT systems become more and more challenging; due to the ever growth of technologies, IoT systems are wider, distributed, heterogeneous, and involve different types of systems communication. Thus, the interaction between elements of the same system as well as the interaction between different systems is primordial. Modelling and developing IoT systems can be particularly very challenging for many reasons. For instance, the architecture of IoT systems relies on many overlapping factors that should be considered earlier in the design phase. The hardware and software components dependencies along with the complexity of the IoT systems architectures, increase significantly the difficulty of modelling these systems. The aim of this paper is twofold: in one hand, we propose a comprehensive and a generic formal model for IoT applications, reducing their complexity and design, using a judicious combination of BRS model and software agents. Our

BCAM4IoT model (Bigraphical Communicating Agent Model for the Internet of Things) gives precise semantics to possible interactions that govern the complex behaviour of these systems. On the other hand, we show how the spatial logic TQL is used to express and analyse more conveniently some relevant properties such as: communication, context awareness and interoperability, which represents a huge interest in the scientific community, we can cite the INTER-IoT project [1] that aims to allow the interoperability among different IoT platforms, the VICINITY project [2], that proposes to define the "interoperability as a service" for IoT by guaranteeing it in the communication part of an IoT system.

Bigraphical Reactive Systems (BRS) via the bigraph model, defined by Milner in [3], emphasize on both locality and connectivity in IoT systems. They propose two types of graphs: a place graph expressing physical/logical location of IoT components, and a link graph describing their interconnections. Additionally, BRS are expressive enough to be adopted for representing IoT system dynamics; in terms of reaction rules. Besides, software agents describe the possible reasoning of each entity and how it self-adapts to context changes. The spatial logic TQL [4] has emerging as an interesting tool to describe properties of several structures. Models for this logic include computational structures such as heaps, trees, graphs [4], concurrent objects [4] as well as process calculi and the Ambient Calculus [4].

The remainder of the paper is structured as follows: in the Section 3, we present a multi-levels architecture for IoT, separating physical components (hardware of software) of IoT systems from logical ones. In this section we also explain the most important properties of IoT systems, namely interoperability and context-awareness. In Section 4, we introduce the BRS formalism together with the TQL logic. In Section 5, we propose BCAM4IoT (Bigraphical Communicating Agent Model for the Internet of Things), a formal model extending the bigraphical agent one for the IoT in order to deal with interaction aspects in IoT systems. In Section 6, we explain how the two most challenging properties, interoperability and context-awareness could be expressed and checked in terms of TQL. Section 7 summarises the paper and discusses ongoing work.

2   Related work

Recently, some approaches in the literature try to apply existing formalisms to model IoT systems while favouring certain aspects to the detriment of others. In this section, we cite some distinctive approaches that deal with formal specification of IoT systems.

Authors of [5] used a model inspired from the nature to specify the behaviour of distributed and dynamic systems while considering the interaction unities as a basic concept. IoT applications has been shown just as a case of study.

In [6], authors propose a runtime model-driven approach focusing on sensor device models description, while our approach focuses on modelling the behaviour of an IoT system with its physical, logical and interaction parts support-

ing the modelling of IoT systems and thus they proposed a generic architecture for IoT systems.

On the other hand, the research work in [7] focused only on the physical part of an IoT system. Their model supports the heterogeneity of objects and may be applied to all types of IoT systems. Their contribution lies in the definition of an ontology that captures the basic considered concepts and models the dynamical process in a static way.

On another side, there is a few number of works that attempt to check formally models destined to IoT Systems, we can cite authors of [8] that use the SPIN tool (based on the Linear Temporal Logic LTL), to check whether a smart house system modelled in petri nets is working properly. In this work, they focused more on the model checking than on the IoT aspects, it is not specified how their model supports the IoT systems characteristics, in the opposite of our proposition, where we try to model IoT systems aspects.

Contrary to the above cited work, our approach consists in modelling the behaviour of an IoT system, combining the software agents and the bigraphs, which are formal tools, while emphasizing the aspect of interaction between their elements. Moreover, we use appropriate spatial logic in order to check some of their properties.

3   A multi-levels architecture for IoT

At a high level of abstraction, IoT applications may be considered as a set of physical and logical components connected via interaction elements. To master their complexity, we propose a generic architecture (Figure 1) dealing with the separation of the physical part from the logical one, and even the physical part can be structured hierarchically. Thus, entities of the system will be distributed between the layers, and charges of each layer are divided through the components of these layers.

Through this section, we will give, in an incremental manner, the three essential parts of the proposed architecture: Physical part, Logical part and Interaction part.

3.1   Physical part

Elements belonging to this part may be classified in four different layers :

1. **Hardware**: contains input hardware identifying the objects that capture data from the environment; output hardware for the objects that actuate on the environment and input/output hardware for the components used to interact in an eventual network;
2. **Abstraction**: constitutes an interface, containing all the protocols needed by the top layers to interact with the hardware components;
3. **Communication**: contains protocols of rooting information from the hardware to the software components (and *vice-versa*), or from one system to another;
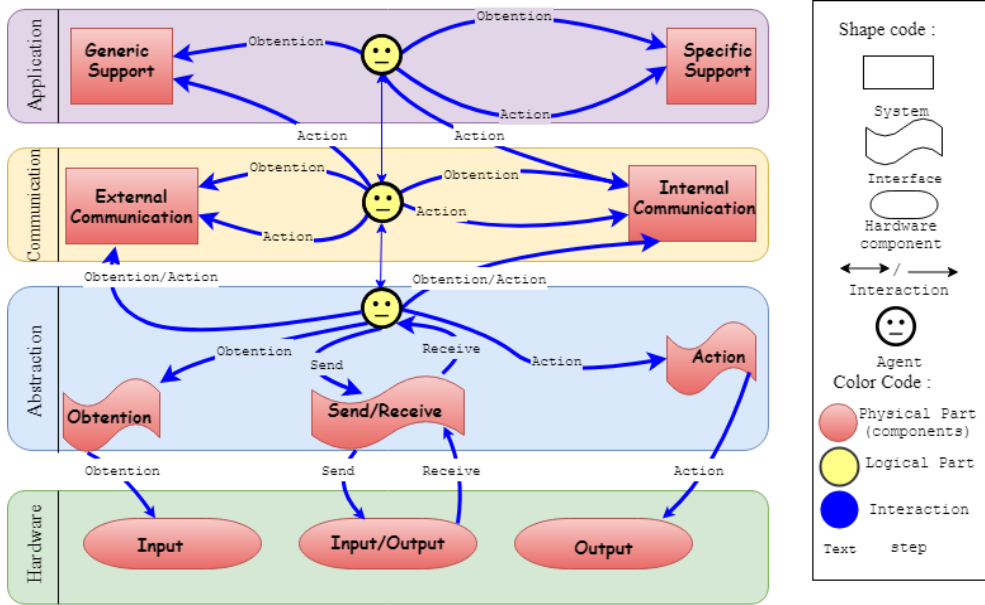
Fig. 1. Multi-layered architecture for an IoT application

4. **Application**: divided into two sub layers: the former recognizes the type of information received, extracts data and gives it an analysable format; the second sub layer is needed to process the desired IoT application.

3.2   Logical part

The logical part of this architecture is supported by a set of reactive agents which behave according to a control loop (observation, analysis, control), each layer of the physical part (except the hardware one) is managed by one agent type.

- In the abstraction layer, the agent controls each solicitation of the hardware entities by elements of the layers above and *vice-versa*. It also decides if the interaction can occur or not, according to the result of its observation's analysis.
- The agent that manages the communication layer is responsible of making the decisions to send or not information for internal or another system.
- Finally, in the application layer, the agent makes the decisions to make a perceptible action after analysing the data received and the data produced by the specific support.

These agents are responsible of the various communications that may exist between the systems or their components.

3.3   Interaction part

In an IoT system, the interactions between objects are the most important concept requiring a particular attention. In this architecture, unidirectional arrows are used to illustrate these interactions, that may be of two types:

– **Agent/Component**: In the three high levelled layers of the architecture, the only interaction defined is between agents and physical components. At each time, agents obtain some information from a component, analyse it, and decide to obtain other information from the same component, or to actuate on another one. Modelling a system with a set of agents involve specifying the communications between them using interaction protocols. Specifying the way they interact is interesting and challenging because of their heterogeneity and autonomy [9]. In [10], we modelled the architecture proposed with its different aspects, using the formalism of bi-agents, but, we didn't take care of the communication between the agents which was the weak point of this model.

– **Component/Component**: This kind of interaction exists only between the abstraction layer entities and the hardware ones. It represents the hardware's solicitations of the system's application side and *vice versa*. It is shown that the abstraction agent doesn't interact directly with the hardware components, but through interfaces (this allows the heterogeneity of objects needed in real IoT systems).

The separation of IoT system components using a layered architecture facilitates its modelling. The interaction between the elements of the system becomes the most important part to define in order to reduce more this complexity. In the present paper, by extending an existing architecture [10], we are interested in modelling and checking the two following properties:

- Interoperability: The definition of interoperability in IoT is currently being developed, so a standard definition is not yet established. In the following sections, we consider that interoperability [11] is to deal with the fact that devices must support interoperable communication protocols and can interact with other devices and also with the infrastructure;

- Context awareness [11]: It is the ability of the sensor nodes to gain knowledge about the surrounding context, based on the sensed information about the physical and environmental parameters. The decisions that the sensor nodes take thereafter are context-aware.

In order to model this architecture formally, according to its hierarchical aspects, the different types of connections between its entities, and the management aspect made by agents, the formalism of bigraphical agents seems to be the most adequate. Hence, we choose the Tree Query Logic to check the considered properties using this resulted model. We are motivated by its adaptability to tree structured models as bigraphical ones. Being a spacial logic, it can be used for the analyses of a model's structure as it is given by the properties of interoperability and context awareness,that are independent of the execution scenarios. Some aspects related to the formalism of biagent and the TQL are presented in the next section.

4   Basic concepts

In this section, we briefly recall the formal concepts used to model and check an IoT system.

4.1   Bigraphical Reactive Systems

A Bigraphical Reactive System (BRS) [3] is the set of bigraphs representing the states of the system obtained from an initial bigraph, and the reaction rules applied successively on it. A bigraph, as shown in the Figure 2 is composed of two graphs, the place graph used to model the location of physical and logical components, and the link graph can express the connectivity between the components:

- The place graph with a set of nodes $\{v_0, v_1\}$ and a parent function $prnt$:$sites \uplus nodes \rightarrow nodes \uplus regions$, the roots are called regions (as 0 and 1 in Figure 2), and some special nodes are called sites (dashed boxes 0,1 and 2 in Figure 2).

- The link graph is defined also with a set of nodes $\{v_0, v_1\}$, hyperedgese_0 and the link function $X \uplus ports \rightarrow hyperedges \uplus Y$, where X are the internal names (oriented to the bottom) and Y the external ones (oriented to the top).

In addition to that, dynamics of an specified system is defined in terms of reaction rules as explained in [3].
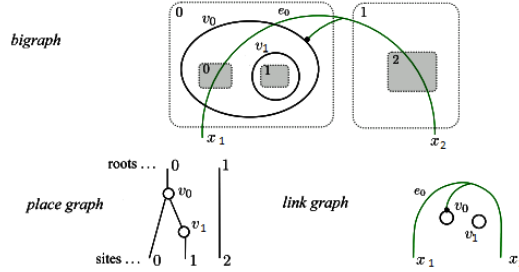

Fig. 2. Example of a bigraph [3] $\langle 3, \{x, y\} \rangle \rightarrow \langle 2, \emptyset \rangle$

4.2   The Tree Query Logic

The Tree Query Logic (TQL) [4] is a spatial logic used in order to reason about eventual updates of tree structures and also permits the model checking of some relevant properties. The TQL is inspired by the Ambient logic, that describes a finite unordered labelled trees called *information trees*, noted, i.t.'s. This logic permits the use of the quantifiers $\forall$ and $\exists$. To understand well a TQL statement, The Table 1 contains some notations:

Table 1. Some TQL Logical Formula [4]

0	The empty information tree.
n[A]	Single-edge i.t.'s labelled n and leading to a subtree satisfying A.
A|B	i.t.'s splittable into two i.t.'s satisfying A and B respectively.
T	All information trees.
¬A	All i.t.'s not satisfying A.
A∧B	i.t.'s satisfying A and B.
A∨B	i.t.'s satisfying A or B.

The empty information tree **0** represents the concept of nothing, having an edge leading to nothing means that there is nothing after. n[A] means that there is an edge of label n leading to something that is between [ ]. The | is the most important operator in a spatial logic, it makes the separation between the elements of the model, if the representation of the model contains at least one |, that means that the model can be separated into at least two parts. **T** helps avoiding repetition. ∧, ∨ and negativity as in all logics are needed for the expression of most properties.

We can express a formula F and checking it by deciding if a tree T satisfies this formula, we note $F(T \vDash F)$. For validating some property, its formula F is valid, if and only if, non F is not satisfiable, we note $\neg(\exists T.T \vDash \neg F)$.

The formalism of bigraphs is used to **model** the behaviour and the structure of a system. In order to **verify** structural and dynamical properties, spatial and temporal logics are used respectively. We notice that the structure of bigraphs can be represented as trees which permits us to use the TQL as a logic of verification of some structural properties. We see in the section 6 how this can be done.

5   BCAM4IoT model

In this model, we focus on the importance of defining the interaction that exists between the different elements of an IoT system. Agents in Figure 1 are responsible of the communication in an IoT system, our proposed model should deal with these communication types. This model is formally defined as: BCAM4IoT=$a_{IoT*}^{(i)} \bullet \mathcal{B}_{IoT}$ (BCAM for Bigraphical Communicating Agent Model), where $\mathcal{B}_{IoT}$ its structural part is given by the Definition 1, and $a_{IoT*}^{(i)}$ its virtual part, by the Definition 2

Definition 1 (The physical structure).

$$\mathcal{B}_{IoT} = (\mathbb{B}, \mathcal{R}, \mathbb{U}, B_0, F)$$

– $\mathbb{B}$ *is the set of bigraphs*
– $\mathcal{R}$ *is the set of reaction rules.*

- $\mathbb{U} \subset V_{\mathbb{B}} \times \mathcal{R}$ *is the set of controls, which is the application of a reaction rule to a specific node in the bigraph with $V_{\mathbb{B}}$ the set of nodes of every bigraph $B \in \mathbb{B}$.*
- $B_0 \in \mathbb{B}$ *is the first bigraph of $\mathbb{B}$.*
- *F is the control function defining the transition between a bigraph and another according to a control $u_i \in \mathbb{U}$.*

In the Definition 1, the chosen IoT application is modelled by the set of bigraphs $\mathbb{B}$. Each bigraph represents a state of the system, and the transition from a state to another is defined by a control. In order to avoid the non-determinism of a particular reaction rule, there are controls $\mathbb{U}$ that are the application of a reaction rule to a specific node, in other words, they represent an action that changes the system's state according to a specific condition. $B_0 \in \mathbb{B}$ represents the initial state of the IoT system.

Definition 2 (Virtual structure of BCAM4IoT).

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

- $\mathcal{O} \subset \mathbb{B}$ *is the observation space.*
- *the function of observation obs which provides an observation $o \in \mathcal{O}$ using a bigraph and the host of the observant agent: $obs(b, h) = o$.*
- *the analysis function that, with an observation or a set of observations and a host, analyses this host or its sons and returns a decision: $an(o, h) = \alpha \in \mathcal{D}$.*
- *The control function which gives the next succession of rules to be executed, according to a node, using the result of an analysis: $ctr(\alpha) = u \in \mathcal{U}$.*
- *The migration function that provides the next host of the agent according to the current host and an observation: $mgrt(o, h) = h'$.*
- *the interaction function int, using the decision made $\alpha \in \mathcal{D}$ and the host of the agent with whom it interacts, the agent interacts with another one using this function: $int(h, \alpha)$*

$$int : V_{\mathbb{B}} \times \mathcal{D}$$
$$(h, \alpha)$$

- $\mathcal{U}$, $\mathcal{D}$ *and $host_0$ are respectively the space of controls, of decisions and the initial host of the agent.*

The Definition 2 shows how the semantic of an agent in BCAM4IoT is defined, each one has:

- A set of observations,
- A set of controls,
- A set of decisions,
- An initial host.
- It observes bigraphs,
- It analyses its observation,
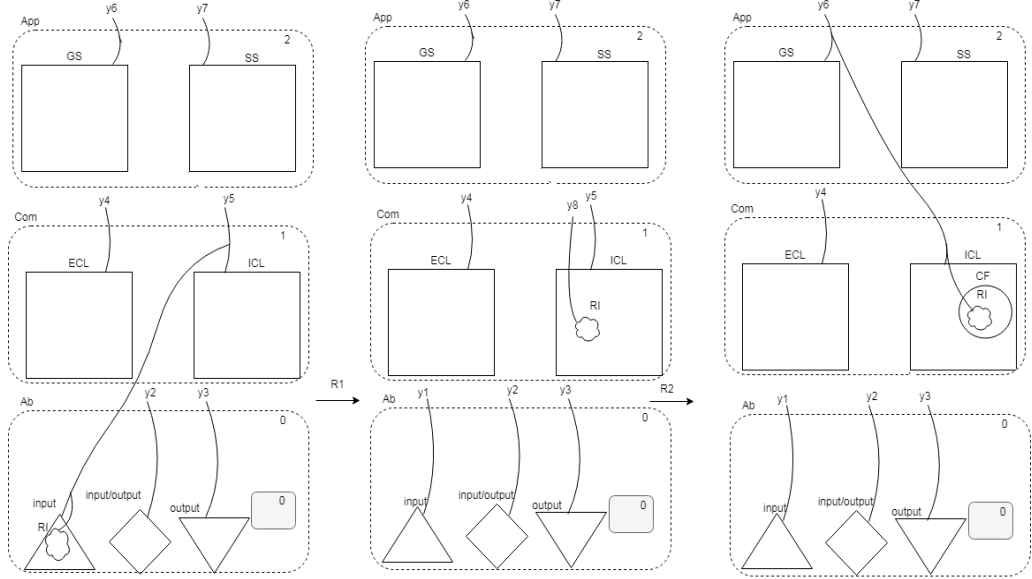- It interacts with other agents giving them its decision

Fig. 3. Example of a trace

The regions App, Com and Ab represent respectively the application, the communication and the abstraction layers. The nodes GS, SS, ECL and ICL represent the generic support, the specific support, the external communication layer and the internal communication layer In BCAM4IoT, we emphasis that an agent makes actions according to its context, it analyses its observations and launches a trace of actions after interacting with agents likely to be influenced by its decision.

5.1   Example

We consider a case of a collision avoidance system as example, provided with a radar, a vibrator and a wifi card. We present a simple scenario showing the contribution of BCAM4IoT to support the definition of the interaction semantics between agents. For instance, we present the sequence of executions materialized by the trace: $T = B_0 \overset{R1}{\prec} B_1 \overset{R2}{\prec}$ in Figure 3. Each state of the system is represented by a bigraph (containing the layers Ab, Com and App), Each layer is represented as a region (dotted boxes) of the bigraph and sub systems, as ECL (External Communication Layer) and ICL (Internal Communication Layer) are represented by simple squares. (In the figure 3, the agent is not visible). The radar (input) captures a suspect sound, the abstraction agent captures the sound through the interface of the radar $\overset{R1}{\longrightarrow}$. After its observation $\overset{obs}{\longrightarrow}$, it migrates in the information node representing the sound (RI) $\overset{mgrt}{\longrightarrow}$ in order to

analyse its content $\xrightarrow{\alpha}$. It interacts with the communication agent giving the resulting decision $\xrightarrow{int}$: the sound will be sent to the ICL in order to be formatted, and returns to the abstraction layer $\xrightarrow{mgrt}$. After that, the ICL formats the information $\xrightarrow{R2}$ and the communication agent observes the bigraph representing this state of the system $\xrightarrow{obs}$. Before migrating into this format $\xrightarrow{mgrt}$ and analysing the packet $\xrightarrow{\alpha}$, the communication agent interacts with the application agent $\xrightarrow{int}$ saying that the packet will be sent to the general support and returns to its layer $\xrightarrow{mgrt}$. After this sequence of executions T, the application agent manages the highest layer and the instructions cascade down to the hardware.

From T, two projection of traces can be defined:

$$t^{Ab_{Ag}} = (B_0, 0) \overset{\text{R1}}{\prec} (B_1, 0) \overset{obs}{\prec} (B_1, 0) \overset{\text{mgrt}}{\prec} (B_1, I) \overset{\alpha}{\prec} (B_1, I) \overset{\text{int}}{\prec} (B_1, I) \overset{\text{mgrt}}{\prec} (B_1, 0)$$

$$t^{Com_{Ag}} = (B_1, 1) \overset{\text{R2}}{\prec} (B_1, 1) \overset{obs}{\prec} (B_1, 0) \overset{\text{mgrt}}{\prec} (B_1, CF) \overset{\alpha}{\prec} (B_1, CF) \overset{\text{int}}{\prec} (B_1, CF) \overset{\text{mgrt}}{\prec} (B_1, 1)$$

The trace rules defined in this section may serve to model various evolution scenarios of IoT systems. Each scenario consists of a sequence of trace rules applications. In the following section, we proceed to the checking of BCAM4IoT, expressing the considered properties of interoperability and context-awareness, using the TQL logic, and move on to execute the checking process.

6   Formal analysis

The hierarchical aspect of BCAM4IoT allows us to use the TQL in order to express structural properties of some models. These expressions are made in the purpose of checking our model in terms of a query and its verification. Here, we are interested by two properties: (i) Interoperability guaranteed by the verification of three atomic ones PICP, FRI and AI. (ii) Context-awareness.

Presence of Interoperable Communication Protocols (PICP).
This is defined in BCAM4IoT by the existence of the node ICP (interoperable communication protocols) in the node ICL (internal communication layer) in the region 1 (communication layer) of the bigraph representing a state of the system. This property is represented in the TQL by an information tree (see figure 4) and it is noted:

$$B[\,region\,[\,1\,[\,subsystemNode\,[\,ICL\,[\,ICP\,[\,\mathbf{0}\,]\,]\,]\,|\,\mathbf{T}\,]\,]\,|\,\mathbf{T}\,]$$
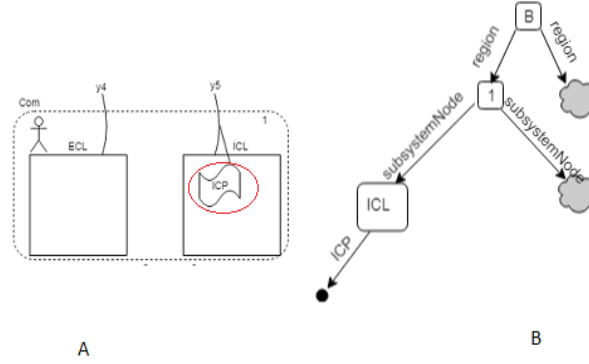
Fig. 4. Formal notation of PICP property

Formatting Raw Information into packets (FRI).
This is defined in BCAM4IoT by the ability of the node ICL (internal communication layer) to format (CF) raw information (RI) into packets. This property is represented in the TQL by an information tree (figure 5) and it is noted:

$$B[\,region\,[\,1\,[\,subsystemNode\,[\,ICL\,[\,ICP\,[\,\mathbf{0}\,]\,]\,|\,[\,CF\,[\,RI\,]\,]\,]\,|\,\mathbf{T}\,]\,]\,|\,\mathbf{T}\,]$$
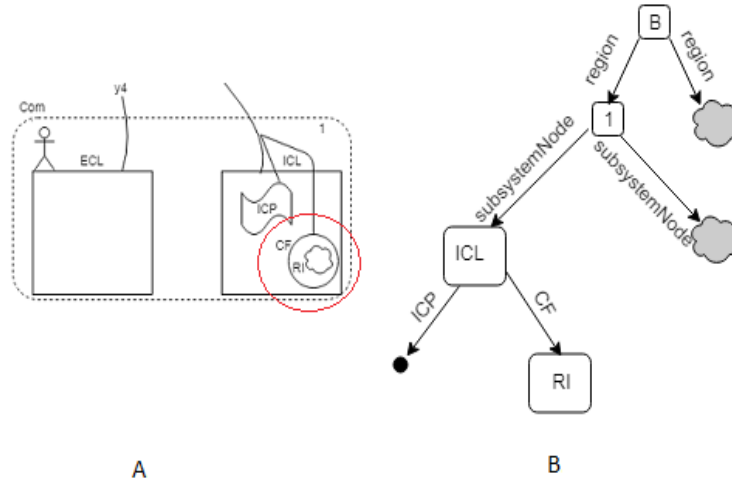

Fig. 5. Formal notation of FRI property

Ability of interacting (AI).
This is defined in BCAM4IoT by the fact that all the elements of the physical layer interact with each other, by the capacity of the abstraction agent to communicate its decision to the communication agent; from the abstraction layer (0), to the communication layer (1); and by its influence on the interfaces (input, input \ output and output) of these elements. This property is represented in the TQL by an information tree as in figure 6 and it is noted:

$$B[\,region\,[\,0\,[\,Agents\,[\,Ag^{Ab}\,[\,Operations\,[\,int\,[\,decision\,[\,content]]\,|\,[\,0\,[\,1\,]]]]\,|$$
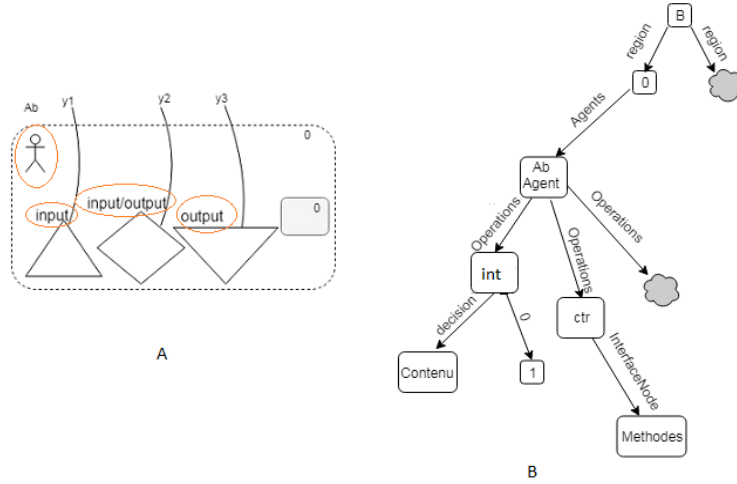$$[\,ctr\,[\,interfaceNode\,[\,methodes\,]]]]\,|\,\mathbf{T}]]]]\,|\,\mathbf{T}]$$

Fig. 6. Formal notation of AI property

In the bigraphical representation of AI property, the operations of the agents are not visible; the following definition of the abstraction agent, especially the functions *int* and *ctrl*, complete the structural definition of BCAM4IoT with the details used to express this property.

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

Note 1. We notice that the interoperability between the hardware components of the system is established by making the software components, and their representation in the model, interoperable.

The context awareness property: in BCAM4IoT, this property is defined as: every Action is triggered in the application layer according to an extracted

information EI made in the general support sub system GG from the raw information RI formatted with the communication format CF received from the communication layer and captured previously. So the system must have these three states to own this property.

In the TQL, it is interpreted as follows:

$$B[\,region\,[\,2\,[\,subsystemNode\,[\,GS\,[\,CF\,[\,RI\,]\,]\,]\,]\,]\,|\,\mathbf{T}\,]\,|\,\mathbf{T}\,]$$
$$\wedge\, B[\,region\,[\,2\,[\,subsystemNode\,[\,GS\,[\,EI\,[\,\mathbf{0}\,]\,]\,]\,]\,]\,|\,\mathbf{T}\,]\,|\,\mathbf{T}\,]$$
$$\wedge\, B[\,region\,[\,2\,[\,subsystemNode\,[\,A\,[\,EI\,[\,\mathbf{0}\,]\,]\,]\,]\,|\,\mathbf{T}\,]\,|\,\mathbf{T}\,]$$

We propose a representation of the initial state in the TQL noted as follows

$$B[region[0[input]||[in\backslash out]||[output]||[site]||[Agent[Ab[...]]]]]||[1[subsystemNode[ICL$$
$$[ICP[\mathbf{0}]]]||[ECL[ECP|PVS]]]||[Agent[com[...]]]]]||[2[GS[...]]||[SS[...]]||[Agent[app[...]]]]]$$

For example, if we proceed to the validation of the property PICP in the initial state, we find that it is true; but if we try to check for FRI in the same state, it is not valid, because in the initial state of the system, there are no raw information captured yet.

7   Conclusion

Development of IoT applications becomes each year more complex and challenging, this is due to, the highly dynamic cooperation among heterogeneous things, the need of ensuring interconnectivity of software and hardware (physical) entities, and the support of a huge scale of interconnected devices. This paper focused on the use of formal methods to design and specify such applications by supporting the characteristics of complex, distributed and heterogeneous systems. We have proposed an extended Biagent model (called BCAM4IoT), that inherits the benefits of each of the implied concepts (BRS and software agents). The BRS have been adopted to specify the physical part of an IoT application dealing with the spatial distribution, the mobility and the heterogeneity of its constituents. The Agents allowed to describe the logical reasoning of IoT elements, a behavioural semantics of an agent in BCAM4IoT considers various action types, including observation (of the context), analysis (of possible situations), control and communication (with each other).

A nice consequence of using a formal model is that trace rules have been exploited to formally specify IoT systems behaviour, dealing with context-aware feature and interoperability one. We also showed the adequacy of the TQL logic to express and validate these important properties. As a future work, we plan to investigate the formal verification of other IoT inherent properties and also, implementing a tool that permits the edition of our model BCAM4IoT, provided with a model checker which checks the structural and dynamical properties of an IoT system using the bigraphical logic and the TQL.

References

1. PROJECT, I.I.: Inter-iot - interoperability internet of things (2016) http://www.inter-iot-project.eu.
2. INITIATIVE, E.P.: Vicinity — open virtual neighbourhood network to connect iot infrastructures and smart objects (2016) https://vicinity2020.eu/vicinity.
3. Milner, R.: The space and motion of communicating agents. Cambridge University Press (2009)
4. Conforti, G., Ghelli, G., Flesca, S., Greco, S., Saccà, D., Zumpano, E.: Spatial tree logics to reason about semistructured data. language **17** (2003) 16
5. Ikram, A., Anjum, A., Hill, R., Antonopoulos, N., Liu, L., Sotiriadis, S.: Approaching the internet of things (iot): a modelling, analysis and abstraction framework. Concurrency and Computation: Practice and Experience **27**(8) (2015) 1966–1984
6. Chen, X., Li, A., Zeng, X., Guo, W., Huang, G.: Runtime model based approach to iot application development. Frontiers of Computer Science **9**(4) (2015) 540–553
7. Bermudez-Edo, M., Elsaleh, T., Barnaghi, P., Taylor, K.: Iot-lite: a lightweight semantic model for the internet of things. In: Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), 2016 Intl IEEE Conferences, IEEE (2016) 90–97
8. Yamaguchi, S., Tsugawa, S., Nakahori, K.: An analysis system of iot services based on agent-oriented petri net pn2. In: Consumer Electronics-Taiwan (ICCE-TW), 2016 IEEE International Conference on, IEEE (2016) 1–2
9. Amit K. Chopra, M.P.S.: Multiagent systems (2011)
10. Marir, S., Kitouni, R., Benzadri, Z., Belala, F.: Biagent-based model for iot applications. In: International Conference on Service-Oriented Computing, Springer (2017) 111–123
11. Ray, P.P.: A survey on internet of things architectures. Journal of King Saud University-Computer and Information Sciences (2016)