Vol. 7(24), Apr. 2017, PP. 3354-3362

Contents list available at IJMEC

International Journal of Mechatronics, Electrical and Computer Technology (IJMEC)

Journal Homepage: www.aeuso.org

PISSN: 2411-6173 - EISSN: 2305-0543

A New Automata Based Approach to Ambient Intelligence Formal Modeling

Mostafa Kaardaan and Zohreh Bateni*

Department of Computer Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran

***Phone Number:** +98-9194898658*

***Corresponding Author's E-mail:** Zoh.Bateni@iauctb.ac.ir*

Abstract

In recent years, in particular, discussion about the ambient intelligence has been widely spread. Intelligent environment is a rich environment of multifunction devices which its main objective is to create a comfortable and convenient environment for the users. In this paper a new automata based approach in Ambient Intelligent Formal Modeling is proposed which investigates the system behavior. Finally after the implementation and verification of the case study, this method is compared with event calculus and ambient calculus methods, to investigate more advantage compared to them.

***Keywords:** Ambient Intelligence, Formal Modeling, Hierarchical Timed Automata, HTA, UPPAAL.*

1. Introduction

Intelligent environments, is distributed network of smart devices that provide us information, communication, basic requirements and even produce entertainment. In fact, smart environment, is rich environment of Multifunction Devices that the main objective of these devices is to create a comfortable and convenient environment for the users. In today life, the population of the elderly and people who suffer from unpleasant situation is increasing, therefore, the existence of such areas in community is desperately needed ant it will help these people in their daily activities and strengthen their independence. So far, different models has been made such as Petri nets based modeling, calculus methods, methods based on formal grammars, multifactorial systems, Fault tree analysis, Evidential reasoning, Evidential Ontology networks, Temporal Logics, hidden Markov models and formal methods based on Theorem Proving, graphics, object-oriented, etc. Despite the performed task in the past, we still cannot implement systems to gain users trust, because the lack of verification, which is the most important element of ensuring the correct implementation of demands on the environment, is quite evident. In fact, the reason that yet few methods has been raised in relation to the verification of this issue, is significant deficiencies in terms of Formal and rational descriptions of this area [7, 19]. So a new hierarchical timed automata based approach in Ambient Intelligent Formal Modeling is proposed which investigates the system behavior and it can model its behavior. The rest of the paper is organized as follows. Section 2 talks about related works; Section 3 introduces some theoretical background; Section 4 illustrate our approach on a case study; Section 5 describes the verification. Section 6 compares with Ambient Calculus and Event Calculus approaches and concludes our paper.

Article History:
Received Date: Oct. 12, 2016
Accepted Date: Feb. 05, 2017
Available Online: Mar. 02, 2017

2. Related Works

Augusto and Hornos [10] using formal methods in modeling and verification reliable behavior at the early stages of system design, and offered improved reliability in smart environments. BenGhazi et al. [11] provided approach to investigation of evaluating satisfaction non-functional requirements such as safety. They have used of Real-time design and analytical methods based on UML and CSP formal models in their approach. Bernardeschi et al. [6] provided formal verification environment to ensure the desired behavior with the safety of their spontaneity. They simulated railway line lock system which control smart monitor communication through Calculus of Communicating Systems formalism.

Bonhomme et al. [18] have used engineering standards for the design process. Design process is used the UML2 diagrams for homes energy management systems. The system will be configured to work and detects user comfort patterns, habits, and the current temperature of the house and tries to control the functions of various components to create a comfortable environment. Boytsov and Zaslavsky [4] have presented procedure to check the area and the position in ubiquitous computing environments. Values changed in the sensors can be non-numeric or (e.g. on and off in lamp) or can be numeric (such as light intensity that can be expressed as a percentage) that is very important during checkout that what would depend on what kind of values.

Gazit et al [9] have been worked on the presence of a person or find him by robots in different parts of an environment with using LTL logic. Initially, they were divided studied environment into several parts. Hoogendoorn et al. in [13] and [14] have suggested a formal model for interactions a multifactorial environment. To behavioral modeling, used propositional logic and in order to verification descriptions have taken advantage from TTL and have used of TTL Checker and SMV. Leelaprute et al. [15] have worked on modeling and verification of a smart home with inside services. They used the concepts of object-oriented modeling and presented a descriptive language, then expression of the environmental requirements using CTL and through famous SMW and have paid to the evaluation of the model.

Ranganathan et al. [1] have presented theory approach with using AC in order to describe and verification pervasive computing environments. They have considered control software components, devices, users, environment and other objects in their work environment. Coronato and Pietro in [2] using real-time logic have added time discussion to the AC and later these authors in the article [3] with additions that had done on the AC, have designed a tool with a graphical environment to pay smart environments Description and its Verification. Korento et al. also have used of provided task by Mardare et al in [16] and suggested, in addition to their utility also can be used NuSMV tool for Verification. Ishikawa et al. [8] using Algebra events to provide a framework to describe the official description of smart environments and had special focus on the concepts of ubiquitous computing environments. In this paper, the authors have tried to work on the user interaction with the environment devices and their behaviors (during the specified period).

3. Backgournd

3.1. Hierarchical Timed Automata

Hierarchical structures are a powerful mechanism to describe complex systems [5]. Modeling languages such as UML uses hierarchical structures to organize the design and different specifications of a system and meet the requirements of developers, customers and performers. State-diagram liked models are more popular, because they describe the behavioral approach of system and also provide execution of the model at high-level.

The general combination of Hierarchical Timed Automata is as follow [5]:

The tuple of Hierarchical Timed Automata is in form of $A = \langle S, S_0, \delta, \sigma, V, C, Ch, T \rangle$ and includes following elements:

- S is finite set of locations.
- $S_0 \in S$ is the set of initial locations.
- δ: S → 2S is necessary to create a tree structure. δ maps the super state to all of its sub-states.
- σ : S → {AND,XOR,BASIC,ENTRY, EXIT, HISTORY}: associates a type to every location.
- V, C and Ch are integer variable, clocks and channel respectively which are used in guards and invariants.
- $T \subseteq S \times (Guard \times Sync \times Reset \times \{true, false\}) \times S$ is the set of edges or transitions.

3.2. Event Calculus

As we know, every event occurs at a specific moment of time. Event Calculus is a logic based language which aims to show reasons for events and their effects [17]. In fact, events occur in an instant of time with their consequences. Event Calculus has been introduced to express and argument the effect of events. In terms of artificial intelligence, Fluent is referred to conditions which created due to an event and can be changed over time. Event Calculus discusses about the effect of works on the fluent. In EC, the works are on the basis of behavior. All events and modeling of environment are conducted based on existence behaviors, perform each event, and the response to the event by taking into account their time. EC is very suitable for behavioral modeling but it suffers from some shortages. For example, it has no mention on the place and movement or it has no approach for control tasks of the environment.

3.3. Ambient Calculus

Ambient Calculus was firstly introduced in 1998 by Cardelli and Gordon for concurrent systems [12]. This model exists in the realm of logical modeling. The main discussion of AC is about movement and in fact, it has provided an opportunity to describe hierarchy places and their movement. In AC structure, all entities are known as a process. The most important type of processes is environment. Each environment has border and boundary and the computational events occur within it. The boundaries of each environment are clear and specified, its processors and what events occur in it are completely clear and has a unique name. Environments can be moved and placed inside and outside from each other. ACs are proper to hierarchical modeling and communications, but they don't support constraints and time factor.

4. Modeling with Hierarchical Timed Automata

The main idea of such an approach is modeling and verification of a system after designing state diagram of the system, i.e. after designing state diagram of the system it would be possible to model the system in timed automata form and then carry out the verification operation to verify the system. In fact, UML diagrams can be used to design a system in standard form. Among this, the state diagram can visually represent the relations and manners dominating on the system and so it can be a behavioral descriptor of things. This stage can be considered as one of the most effective steps for modeling system.

4.1. Case Study

Different scenarios can be considered in intelligent environments, which smart homes are one of the most well-known examples in this area. One of the scenarios of smart homes and smart shops is smart electronic door which has been selected as a case study in this section. In this case study, the smart electronic door detects the presence of user and opens the door for his entrance and after a while gets the door in Close mode and the cycle continues with detecting another user again and again.

The design specification of the door is in a way that there is a door, there is a controller to manage the modes and there is a sensor to detect user's presence which is one of the elements of controller.

The door behavior has been considered in such a way that it is in completely Close or Open mode, is in opening or closing mode, it gets to opening mode when a user is approaching, then it is opened and remains in Open mode for a while allowing the user to pass and gets in to Closing mode after the user passed.

Since the ambient intelligent systems must operate in real time, in this case study a number of temporal constraints are needed which is as follow:

- When a user is approaching it takes 3 time units to door be completely opened.
- It remains in completely open mode for 5 time units allowing the user to pass.
- When the user moves away, it longs 3 time units to be completely closed.

4.2. Modeling the case study using Hierarchical Timed Automata

The user's behavior is considered so that either the user is not near to the door or the user is approaching to the door and passes it. The Hierarchical Timed Automata model of user has been shown in Figure 1 . Z is the clock to consider the time factor. To start, it is considered that the user is not near. For this reason, the starting point has been adjusted on Far state at the user's range. When the user approaches, the sensor detects him and the Approach signal is activated and the user gets to the near state.

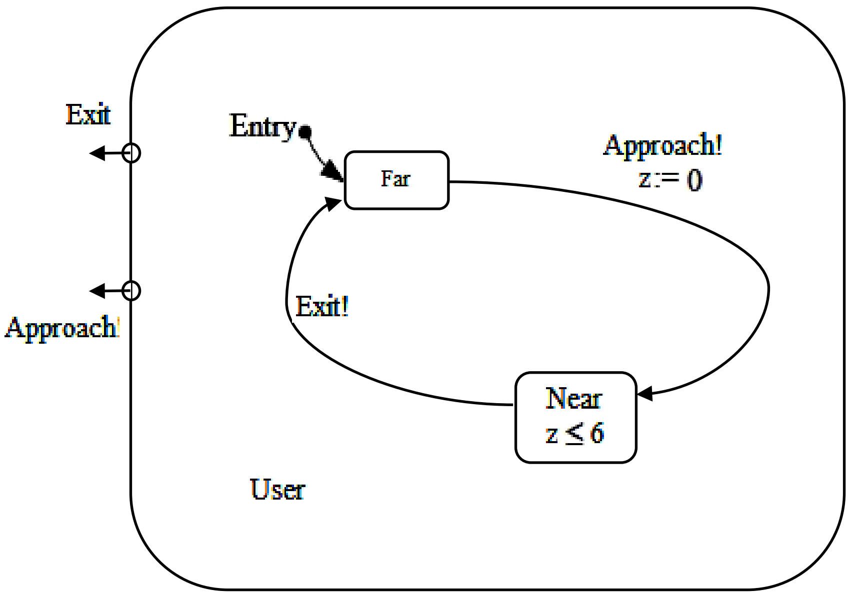

Figure 1: User Hierarchical Timed Automata Model

In the first state, the controller is sensing the approaching user by sensor. After detecting a user approaching, the last state checks the door which is in closed state and sends it the signal of opening. Since the door opening includes a time constraint, so a state must be considered for it. After the specified period, the door is fully open allowing the user to pass through it. While the user passes the door, the controller receives the user exit signal through sensor detection and gets into Exiting state, in which the controller sends a signal to the door for closing and gets into closing state. After a specified time, , the door is completely closed and the controller gets to closed state and then gets to sensor state after receiving a signal by door to detect the next approaching user. The stages have been represented in Figure (2). In the Figure, Sensing indicates sensor state of controller, CL is closed door, OPG is the state of opening door, OP is the mode of opened door, Exiting is the exiting user, CLG represents the mode of closing door and y is the clock of system.

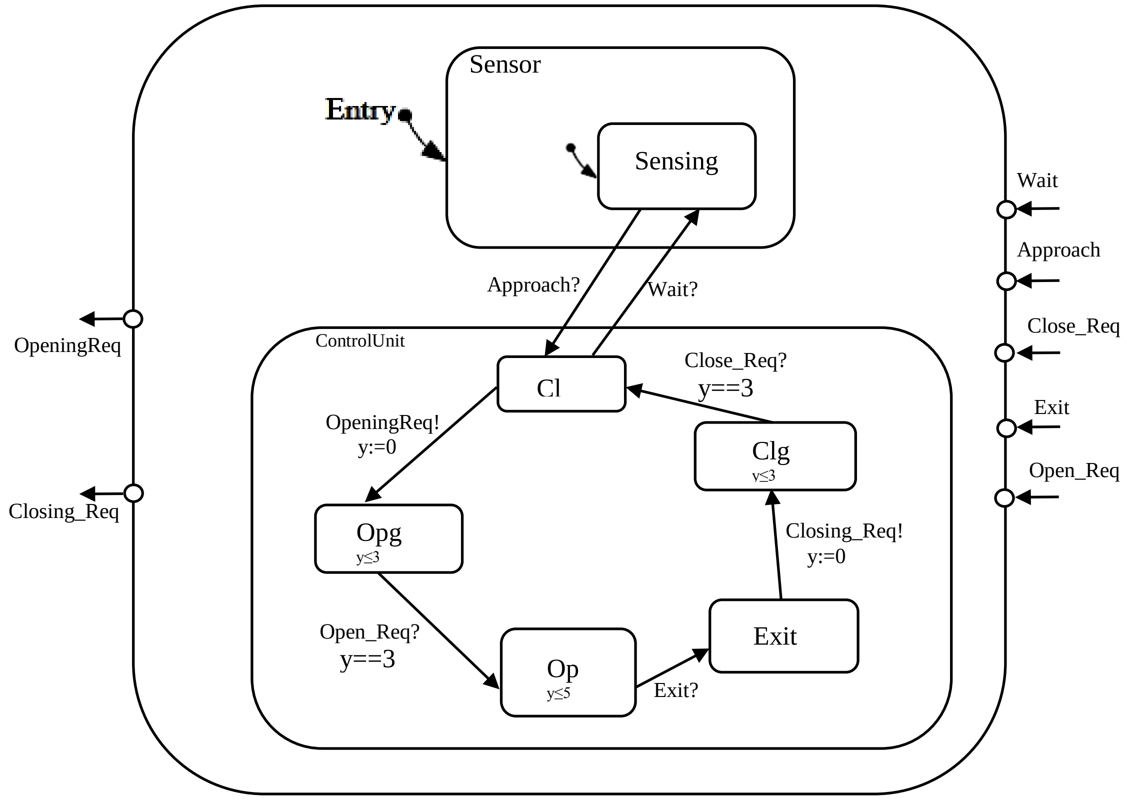

Figure 2: Controller Hierarchical Timed Automata Model

The door can have two main states of without moving and moving. The two states have themselves some sub-states including Close, Open and waiting for without moving mode and Closing and Opening for moving mode.

In general, the door is closed and doesn't move; therefore it is in waiting state until it receives a signal by the controller on opening. After receiving a signal by controller, the door gets to moving state and opening sub-state. After spending the corresponding period, the door is completely opened and therefore gets to without moving state and Open state. The door reminds opened for some time unit allowing the user to pass. Then, it gets to moving mode and Closing sub-mode. After spending the corresponding period, the door is completely closed and therefore gets to without moving state and Close state. In this mode, the door sends Waiting signal to controller and gets to waiting state and waits for next command. The stage has been shown in Figure 3.

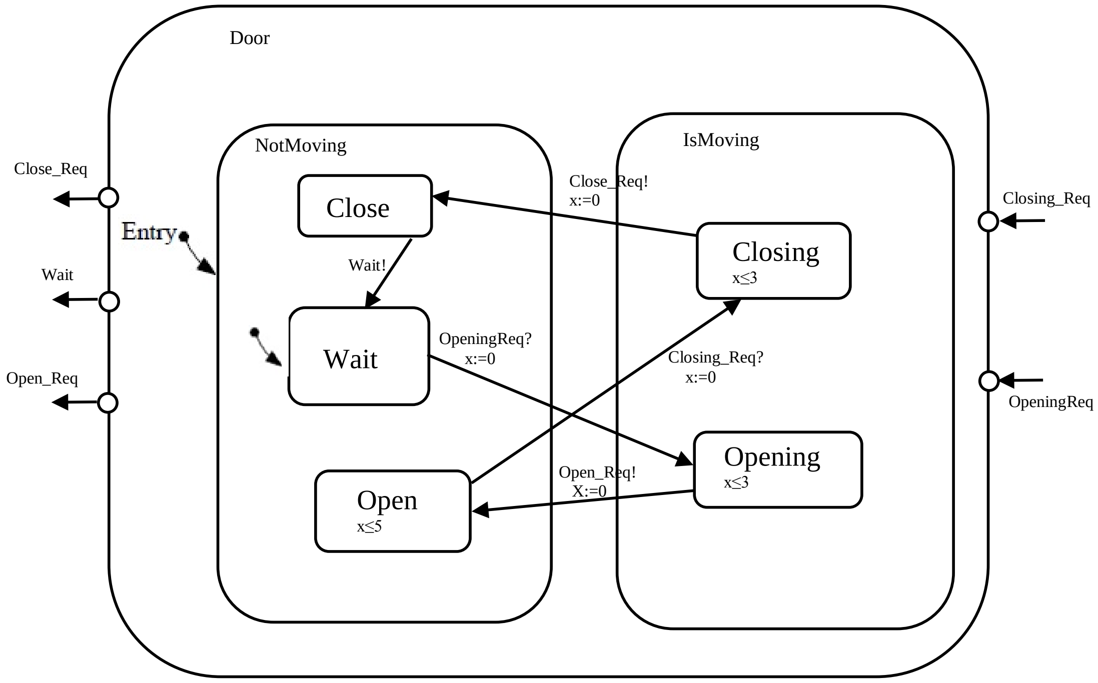

Figure 3: Door Hierarchical Timed Automata Model

5. Verification

In this section, the Hierarchical Timed Automata model has been verified through computational tree logic using UPPAAL Software. Since, UPPAAL Software works with timed automata concept, the hierarchical timed automata is flattened into timed automata and implemented in UPPAAL Software which have been shown in Figures 4, 5 and 6.

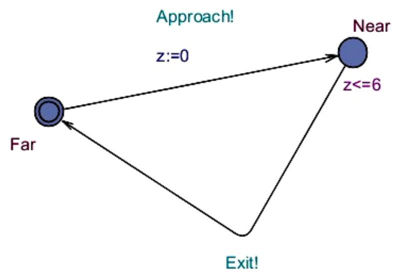

Figure 4: User Template in UPPAAL

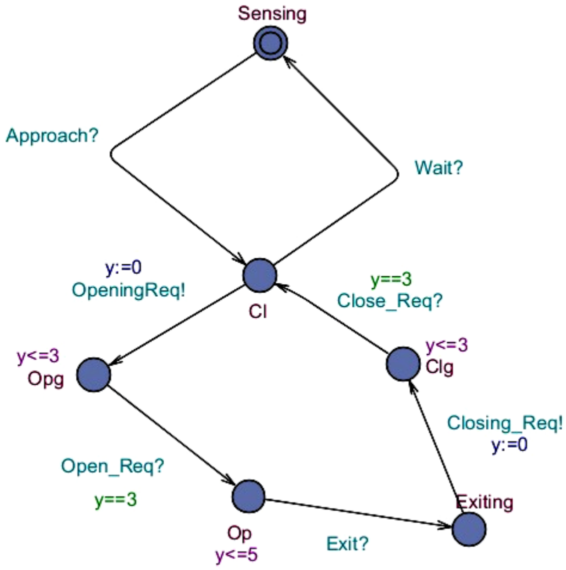

Figure 5: Controller Template in UPPAAL

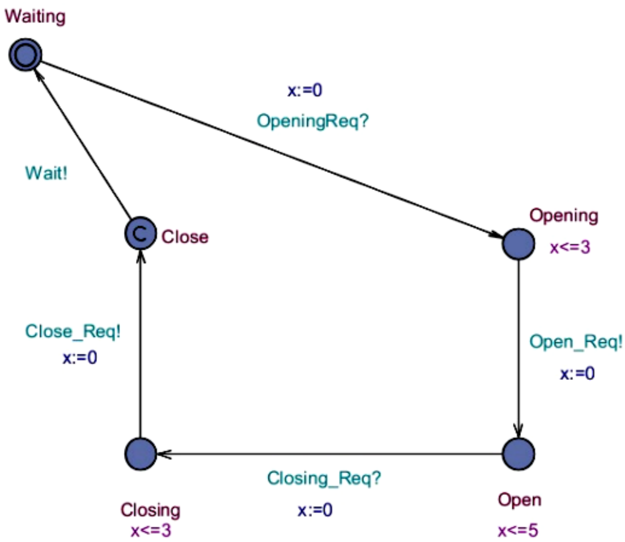

Figure 6: Door Template in UPPAAL

In this section, the following formulas of computational tree logic have been used to evaluate the system to check the reachability, liveness and safety of system which confirm system's integrity.

1. The system must be deadlock free. This feature is inspected in the software using the following formula.

- A[] not deadlock

2. In each path, there is no case in which simultaneously the door is closed, the user is approaching and the controller is in closed state.

- A[] !(ProcessDoor.Close && ProcessUser.Near && ProcessController.Cl)

3. In each path, there is no case in which simultaneously the door is open, the user is far and the controller is in open state.

- A[] !(ProcessDoor.Opening && ProcessUser.Far && ProcessController.Opg)

4. If a user is approaching, the door opens.
- ProcessUser.Near --> ProcessDoor.Open

All these Properties are satisfied by CTL in UPPAAL.

Comparison and Conclusion

Various modeling have been conducted on ambient intelligence until now, despite of conducted works, it cannot be yet possible to implement systems that gain the user's full trust because of the lack of verification which is the main support to ensure the correct implementation of demands on the environment. The method has been used in present study is based on the Hierarchical Timed Automata concept which was implemented on under study case using UPPAAL Software and successfully verified by CTL logic. The Hierarchical Timed Automata are able to support different criteria such as time, communication, hierarchical and constraints. In this fields, the event calculus and ambient calculus methods cannot support all of these criteria. Table (1) presents a comparison between event calculus and ambient calculus methods and the proposed Hierarchical Timed Automata method of present study.

Table 1: Comparison between proposed method, event calculus and ambient calculus.

Method / Criteria	EC	AC	HTA
Time	√	×	√
Communication	√	√	√
Hierarchy	×	√	√
Constraints	√	×	√

Acknowledgement

This paper is extracted from the M.Sc. Thesis sponsored by the Islamic Azad University Central Tehran Branch.

References

[1] A. Ranganathan and R. H. Campbell,(2008) "Provably correct pervasive computing environments," in Pervasive Computing and Communications, 2008. PerCom 2008. Sixth Annual IEEE International Conference on, pp. 160–169.

[2] A. Coronato and G. De Pietro,(2011) "Formal specification and verification of ubiquitous and pervasive systems," ACM Trans. Auton. Adapt. Syst., vol. 6, no. 1, pp. 1-9.

[3] A. Coronato and G. De Pietro,(2012) "Tools for the rapid prototyping of provably correct ambient intelligence applications," Softw. Eng. IEEE Trans., vol. 38, no. 4, pp. 975–991.

[4] A. Boytsov and A. Zaslavsky,(2013) "Formal verification of context and situation models in pervasive computing," Pervasive Mob. Comput., vol. 9, no. 1, pp. 98–117.

[5] A. David and M.O. Möller, (2001)" From HUppaal to Uppaal: a translation from hierarchical timed automata to flat timed automata," Aarhus: BRICS.

[6] C. Bernardeschi , A. Fantechi, S.Gnesi, S.Larosa, G.Mongardi, D.Romano,(1998) “A formal verification environment for railway signaling system design,” Formal Methods System Design 12(2):139–161.

[7] F. Sadri,(2011) “Ambient intelligence: A survey,” ACM Comput. Surv., vol. 43, no. 4, Article .36, pp. 1–66.

[8] F. Ishikawa, B. Suleiman, K. Yamamoto, and S. Honiden,(2009) “Physical interaction in pervasive computing: formal modeling, analysis and verification,” in Proceedings of the 2009 international conference on Pervasive services, pp. 133–140.

[9] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas,(2007) “Where’s Waldo? Sensor-based temporal logic motion planning,” in Robotics and Automation, 2007 IEEE International Conference on, pp. 3116–3121.

[10] J. C. Augusto and M. J. Hornos,(2013) “Software simulation and verification to increase the reliability of Intelligent Environments,” Adv. Eng. Softw., vol. 58, pp. 18–34.

[11] K. Benghazi, M. V Hurtado, M. J. Hornos, M. L. Rodríguez, C. Rodríguez-Domínguez, A. B. Pelegrina, and M. J. Rodríguez-Fórtiz,(2012) “Enabling correct design and formal analysis of Ambient Assisted Living systems,” J. Syst. Softw., vol. 85, no. 3, pp. 498–510.

[12] L. Cardelli and A. D. Gordon,(1998) “Mobile ambients,” in Foundations of Software Science and Computation Structures , pp. 140–155.

[13] M. Hoogendoorn, M. C. A. Klein, Z. A. Memon, and J. Treur,(2013) “Formal specification and analysis of intelligent agents for model-based medicine usage management,” Comput. Biol. Med., vol. 43, no. 5, pp. 444–457.

[14] M. Hoogendoorn, M. C. A. Klein, Z. A. Memon, and J. Treur,(2009) “Formal verification of an agent-based support system for medicine intake,” in Biomedical Engineering Systems and Technologies, Springer, pp. 453–466.

[15] P. Leelaprute, M. Nakamura, T. Tsuchiya, K. Matsumoto, and T. Kikuno,(2005) “Describing and verifying integrated services of home network systems,” in Software Engineering Conference, 2005. APSEC’05. 12th Asia-Pacific, pp. 1-10.

[16] R. Mardare, C. Priami, P. Quaglia, and O. Vagin,(2005) “Model checking biological systems described using ambient calculus,” in Computational Methods in Systems Biology,pp. 85–103.

[17] R. Kowalski and M. Sergot,(1989) “A logic-based calculus of events,” in Foundations of knowledge base management, Springer, pp. 23–55.

[18] S. Bonhomme, E. Campo, D. Esteve, J. Guennec,(2008) “Methodology and tools for the design and verification of a smart management system for home comfort,” In: 4th International conference on intelligent systems, IEEE. pp 24–2–24–7.

[19] T. Strang and C. Linnhoff-Popien, (2004) “A context modeling survey,” in Workshop Proceedings.