



**HAL**  
open science

## A DES Simulator for Location Tracking of Inhabitants in Smart Home

Mickaël Danancher, Gregory Faraut, Jean-Jacques Lesage, Lothar Litz

► **To cite this version:**

Mickaël Danancher, Gregory Faraut, Jean-Jacques Lesage, Lothar Litz. A DES Simulator for Location Tracking of Inhabitants in Smart Home. Proceedings of the 8th EUROSIM Congress on Modelling and Simulation - EUROSIM 2013, Sep 2013, Cardiff, Wales, United Kingdom. pp.330-335. hal-00862386

**HAL Id: hal-00862386**

**<https://hal.science/hal-00862386>**

Submitted on 16 Sep 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A DES Simulator for Location Tracking of Inhabitants in Smart Home

Mickael Danancher<sup>\*,†</sup>, Gregory Faraut<sup>\*</sup>, Jean-Jacques Lesage<sup>\*</sup> and Lothar Litz<sup>†</sup>

<sup>\*</sup> Automated Production Research Laboratory (LURPA), ENS Cachan, 61 av. du President Wilson, 94235 Cachan, France  
Email: {danancher, lesage, faraut}@lurpa.ens-cachan.fr

<sup>†</sup> Institute of Automatic Control, University of Kaiserslautern, P.O. Box 3049, 67653 Kaiserslautern, Germany  
Email: litz@eit.uni-kl.de

**Abstract**—In previous works we showed that, in return for some assumptions, indoor Location Tracking (LT) can be formulated as a Discrete Event Systems (DES) problem. We proposed a method for constructing Finite Automata (FA) models on which LT is performed online for single or multiple inhabitants. However, the accuracy of LT strongly depends on the choices of the designer concerning topology (in particular the splitting of the dwelling in zones) and the number and placement of sensors. To evaluate the impact of these choices onto the efficiency of the LT, we proposed an analytical approach that allows determining asymptotic performance criteria. This performance evaluation does not take into account the human behavior (real time moving of inhabitants) which is very difficult to model. In order to include the inhabitants behavior, and thus to improve the performance evaluation by considering dynamic criteria, we propose in this paper a discrete event simulation approach that allows to emulate the smart home and to immerse the human operator into this environment without having to model the human behavior.

**Index Terms**—Discrete Event Systems, Smart Home, Location Tracking, Discrete Event Simulation, Finite Automata.

## INTRODUCTION

Smart Home technologies are aiming to help people to live in a comfortable and safe environment. A *smart home* can be defined as a dwelling equipped with sensors, actuators and communication devices; based on the information given by the sensors, the actuators can be controlled in order to improve comfort (heating or air conditioning for instance) or to guarantee the safety of the inhabitants (automatic shutdown of dangerous devices or health problem detection for instance) [1].

*Location tracking* (LT) is most often required for adapting services to the habits or to the behavior of the inhabitants. This task consists in finding in real time the location of one or several inhabitants, based on the observation of the signals generated by the different sensors of the house [2]. Different sensor technologies like video cameras [3], radio frequency identification (RFID) tags [4], ultrasonic badges [5]... are used for performing LT. In order to help the users to accept the observation of their every movement and to guarantee the respect of their privacy and the reduction of cost, we choose to only consider non-wearable, non-intrusive and low-cost sensors. Such sensors are mostly binary sensors (door barrier sensors, motion detectors...) or sensors delivering a signal that can be interpreted as binary using a threshold (electricity consumption, water flow or floor pressure sensor for instance).

In most approaches, LT is performed by using data mining techniques [2], [6], that need a more or less long learning phase before the LT can be performed. Furthermore, this phase has to be performed again as soon as the instrumentation is modified (i.e. if new sensors are added or if existing sensors are removed or if their placement is modified). Last but not least, such learning techniques lead to the lack of a formal and explicit model of the location of inhabitants. For these reasons, we proposed an approach allowing the systematic construction of Discrete Event Systems (DES) models (Finite Automata) on which is performed the real-time LT of single [7] or multiple [8] inhabitants into a smart home.

Whatever the LT technique used, the accuracy of the estimated location strongly depends on the topology of the dwelling and of the number and placement of sensors (for instance a lack of instrumentation in some areas may lead to a partial observation of the motion of the inhabitants). To evaluate the impact of these two parameters onto the efficiency of the LT, we proposed an analytic approach [7] that allows determining asymptotic performance criteria such as *unlocationable zones* or strong and weak *accurate-location-ability*. Nevertheless, this performance evaluation does not take into account the human behavior (real time moving of inhabitants) which is very difficult to model.

In order to include the inhabitants behavior, and thus to improve the performance evaluation of LT by proposing dynamic criteria, we describe in this paper a *discrete event simulation approach* that allows to emulate the considered smart home and to immerse the human operator into this simulation environment without having to model the human behavior.

The rest of the paper is structured as follows: in section I the main results of our previous works are recalled thanks to a case study. Our simulation approach and the developed simulator are presented in the second section. The third section presents and discusses the relevance of practical results.

## I. PROBLEM STATEMENT AND CASE STUDY

### A. Context and assumptions

As stated before, in our approach only binary sensors are considered. It is also considered that information given by the sensors do not depend on the ability or the willingness of each inhabitant to provide this information. For instance,

if a door is equipped with a door barrier sensor and a door contact sensor, an inhabitant crossing the door will systematically be detected by the barrier sensor but will be detected by the contact sensor only if this inhabitant opens or closes the door while he crosses it. Consequently, in our approach, door contact sensors will not be used. For similar reasons, light-switch sensors are also not considered because while entering a room an inhabitant may or not switch the light on, depending on the luminosity or his life habits.

Moreover, it is assumed that each inhabitant of a dwelling has a totally free behavior and behaves independently from the others. Consequently, adopting a DES point of view, each inhabitant living in an instrumented environment is seen as a spontaneous event generator. These events are the rising and falling edges of the signals emitted by the binary sensors of the smart home during the motion of the inhabitant. As a convention, the rising edge and the falling edge of a sensor  $s_1$  are respectively denoted as  $s_{1\_1}$  and  $s_{1\_0}$ . The simultaneous occurrence of several events is also considered as not being possible.

Considering the topology of an apartment and a potential lack of instrumentation in some areas, we also have to make the assumption of partial observation of the behavior of each inhabitant. Moreover since we do not consider wearable sensors, the inhabitants are non-distinguishable by the sensors, i.e. a signal generated by a sensor means someone is moving in front of it but gives no information about who this person is.

Based on these considerations, the problem of multiple inhabitants online LT can be solved by using DES techniques: the set of all the possible locations and observable motions of inhabitants is modeled by a finite automaton; the real time location of the inhabitants is estimated by playing this automaton with the sequence of observed events, that are generated when the inhabitants move. The structure of such automata and their use for LT are now described by using an example.

### B. Case Study

Even if our approach has been successfully applied to complex instrumented apartments, for the sake of better understanding a small size example of smart home has been chosen (Fig. 1). It is composed of three rooms: an open space for the kitchen and the living room, a bedroom and a bathroom. This smart home is equipped with three motion detectors ( $MD_1$  in the open space of the living room and the kitchen,  $MD_2$  in the bedroom and  $MD_3$  in the bathroom) and a door barrier sensor  $DB$  (detecting an inhabitant crossing the door) on the front door of the house.

All the possible motions (i. e compatible with the topology of the apartment) of a single inhabitant between the different zones of this apartment that are observable by the considered instrumentation (i. e. the number, the technology and the placement of the chosen sensors) are represented by the Finite Automaton (FA) given in Fig. 2. The systematic construction of this FA (that is called the Detectable Motion

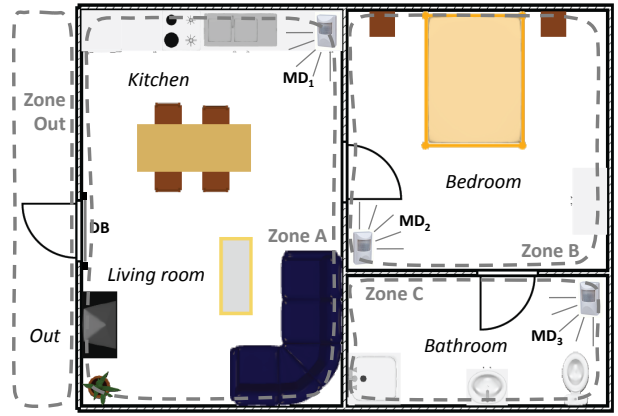


Fig. 1. Topology and instrumentation of the case study

Automaton - *DMA*) from the only knowledge of the topology of the apartment and the instrumentation is described in detail in [7].

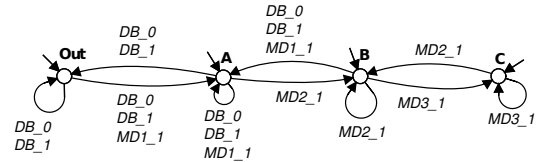


Fig. 2. Detectable Motion Automaton *DMA* for one inhabitant

In the FA of Fig. 2, each state represents a zone of the house where the inhabitant can be. All the states are considered as initial because the initial location of the inhabitant is assumed to be unknown. Each transition between a state  $q_i$  and a state  $q_j$  labeled with an event  $s_{k\_1}$  (or  $s_{k\_0}$ ) expresses that the motion between these two zones is topologically possible and is observable through the sensor  $s_k$ .

In case of multiple inhabitants, a FA describing the possible locations and motions can be constructed in the same way [8]. The case of two inhabitants living together in the apartment of Fig. 1 is given in Fig. 3.

In the FA of Fig. 3, each state represents the location of the two inhabitants in the different zones of the house. Each transition between a state  $q_i$  and a state  $q_j$  labeled with an event  $s_{k\_1}$  (or  $s_{k\_0}$ ) expresses that the motion of one or two inhabitants between these two sets of zones is topologically possible and is observable through the sensor  $s_k$ .

The online LT of the inhabitants is performed onto the *DMA* thanks to Algorithm 1. After each observed event generated by the sensors, this algorithm allows playing the *DMA* for determining the new location reached by each inhabitant.

Of course, the result of the LT is more or less accurate depending on the choices of the zones partition and of the instrumentation. It is therefore important to be able to evaluate this accuracy for choosing the best combination zones/instrumentation. The analytic approach we proposed in [7] allows determining asymptotic performance criteria

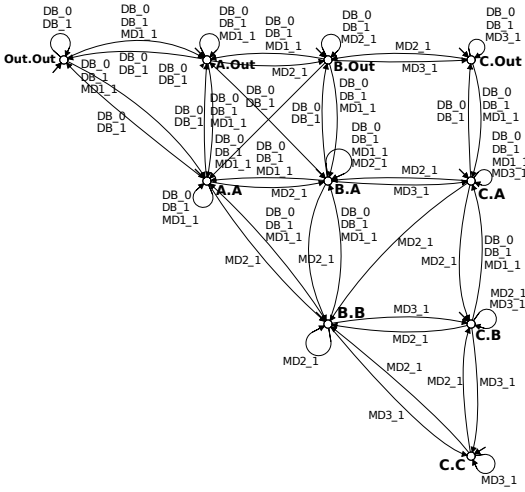


Fig. 3. Detectable Motion Automaton for 2 inhabitants

---

**Algorithm 1** Online location estimation algorithm

---

**Require:**  $DMA = (Q, \Sigma, \delta, Q_0)$

- 1: Initialization: Current location  $L_{Est} = \text{set of states } Q_0$
  - 2: **while** location tracking is active **do**
  - 3:   Wait for a new event  $e$
  - 4:   New event  $e$  is observed
  - 5:   **if**  $\exists q \in L_{Est}$  such that  $\delta(q, e) \neq \emptyset$  **then**
  - 6:      $L'_{Est} = \bigcup_{q \in L_{Est}} \delta(q, e)$
  - 7:     Update current location  $L_{Est} = L'_{Est}$
  - 8:   **else**
  - 9:     The location remains  $L_{Est}$
  - 10:   **end if**
  - 11: **end while**
- 

such as the set of unlocationable zones (if they exist) and the *accurate-location-ability* (that characterizes the ability to estimate accurately the location of the inhabitants). For this case study, the analytical evaluation gives the following result: there are no unlocationable zones and we can guarantee a *weak accurate-location-ability* (it means that depending on the behavior of the inhabitant, the location may become and remain accurate after a finite time). Nevertheless, this static performance evaluation does not take into account the human behavior, which is obviously complex, since neither deterministic nor stochastic but rather arbitrary and potentially irrational.

The problem we investigate in this paper can therefore now be reformulated as follows: How to evaluate the relevance of a Detectable Motion Automaton (constructed for a given couple zones partition/instrumentation) for performing LT and with what resulting accuracy? For that, we propose to use the simulation approach that is presented below.

## II. SIMULATION APPROACH

### A. Overview of the approach

An overview of the approach we propose is given in Fig. 4. The simulation of location tracking is performed using the algorithm 1, i.e. the simulator is based on exactly the same algorithm as the one used for location tracking in the real home. Moreover, this simulation is aiming at evaluating the relevance of the FA model (*DMA*) as it will be used for performing the location tracking. At this point, for a given sequence of sensor events, there is no deviation between the location tracking done by simulation and the expected location tracking in real homes. The goal of the simulation is to evaluate *a priori* the relevance of the *DMA* for LT by generating a lot of relevant sequences of events that represent accurately the moving of inhabitants, without having to perform tests in the real instrumented environment.

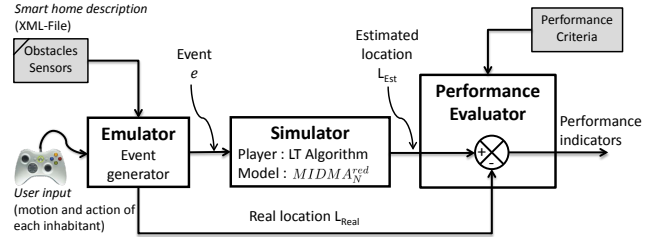


Fig. 4. Performance evaluation of LT model by simulation

We chose to develop an emulator which allows immersing a user (thanks to a keyboard or a joystick) in a virtual smart environment reproducing the topology and the instrumentation of the dwelling. The sensors are reacting to the inhabitants' motion and action and provide the according sequence of events. Furthermore, in the emulator, the exact location of each inhabitant at each time is known. It is then possible to evaluate the dynamic performance of the accuracy of the LT model by comparing the estimated with the real location. The comparison criteria are detailed in the next section.

### B. Simulator implementation

We chose to develop the emulator, the simulator and the performance evaluator using *Python 2.7*<sup>1</sup> and its object-oriented paradigms because this programming language is well appropriated to quickly develop proof-of-concept. In addition, the module *pygame*<sup>2</sup> was used in order to simulate the interaction of the inhabitants with their environment and to provide a graphical representation. The user inputs (motion and action inhabitants) are transmitted to the program via a keyboard or a joystick.

The program is strongly object-oriented, the UML Class Diagram of the smart home emulator is shown in Fig. 5. Three main classes are defined:

<sup>1</sup><http://www.python.org>

<sup>2</sup><http://www.pygame.org>

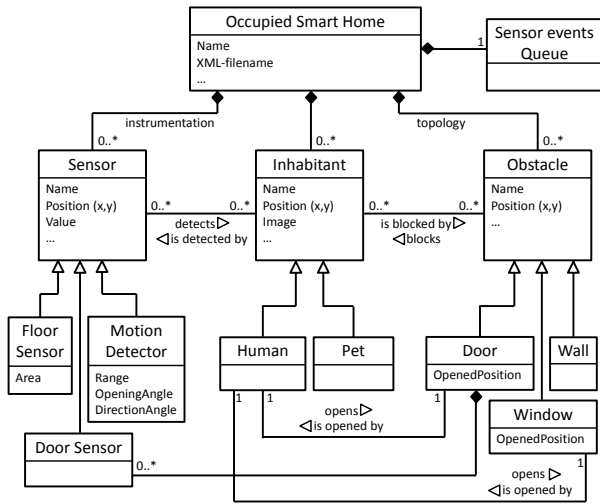


Fig. 5. UML diagram of the smart home emulator

- The class *Obstacle* defines the different possible obstacles composing the topology of the smart home like the walls. We defined the classes *Wall*, *Door* and *Window* which are inheriting from the class *Obstacle*.
- The class *Sensor* defines the different possible sensors that can be used in the smart home. Since there are different types of sensors, additional classes inheriting from *Sensor* have been defined, for instance *Floor Sensor*, *Door Sensor* or *Motion Detector*. Their technology is defined through a specific model of functioning and specific attributes.
- The class *Inhabitant* defines the different inhabitants living in the smart home. Different kind of inhabitants can be defined, using the classes *Human* or *Pet* that both inherit from the class *Inhabitant*

In addition to these classes, the class *Occupied Smart Home* is defined. It is characterized by a list of instances of the class *Obstacle*, an instrumentation which is a list of instances of the class *Sensor* and a set of inhabitants which is a list of instances of the class *Inhabitant*. Finally, a queue of sensor events (FIFO type) is generated by the smart home emulator and used by the simulator.

It can also be noticed that the different classes are linked. The motions of the inhabitants are blocked by the different obstacles of the house, this is represented by the link between the classes *Inhabitant* and *Obstacle*. In a similar way, the sensors are reacting to the position and to the motion of the inhabitants; this is represented by the link between the classes *Sensor* and *Inhabitant*. Each inhabitant may also interact with the obstacles e.g. doors or windows can be opened but only by humans (not the pets). Consequently, there is a link between the class *Human* and the class *Door* and a link between the class *Human* and the class *Window*. This allows to take into account different possible behaviors for different types of inhabitants.

To perform the simulation of the location tracking and its evaluation, three processes are running in parallel: the smart

home emulator, the simulator and the performance evaluator. Each process gives a graphical result in a separate window on the screen. The global result of the simulation screen is shown in Fig. 6. On Fig. 6(a) is drawn the smart home and the position, motion and action of the different inhabitants (in this case, there is one inhabitant in the bathroom). Fig. 6(b) shows the estimated location, i.e. the active state(s) of the *DMA* (in this case, the location is accurately estimated and there is only one active state: *C*). Fig. 6(c) shows the results of the performance evaluator through the confusion matrix and other calculated criteria that are detailed in the next section. The content of the three windows is updated online according to the inputs given by the user. When the user decided to close the program, the results are saved in a log-file.

### III. PRACTICAL RESULTS

Several criteria are defined in order to evaluate the results of the simulated Location Tracking. These criteria and their interpretation are illustrated using the case study and a scenario involving a single inhabitant for the sake of clarity. However, the emulator and the performance evaluator are also usable and perform well for multiple inhabitants location tracking.

The evaluation criteria are based on the confusion matrix [9]. A confusion matrix represents the number of times a prediction corresponds or not to a real situation. It is mainly used to evaluate data-driven learning approaches e.g. activities of daily living recognition in smart homes [10].

We propose a confusion matrix called  $CM_{LT}$  to evaluate the accuracy of the estimated location compared to the real location. For the case study and considering a single inhabitant, there are 4 possible locations. Consequently  $CM_{LT}$  is a  $4 \times 4$ -matrix and each element ( $CM_{LT(i,j)}$ ) is defined as follow:

$$CM_{LT(i,j)}(T) = \frac{1}{T} \int_0^T \frac{\mathbf{1}_{\{(L_i=L_{Real}(t)) \wedge (L_j \in L_{Est}(t))\}}}{|L_{Est}(t)|} dt$$

where:

- $T$  is the duration of the simulation,
- $\mathbf{1}_{\{predicate\}} = 1$  if *predicate* is true and 0 else,
- $L_i$  (resp.  $L_j$ ) represents the  $i^{th}$  (resp. the  $j^{th}$ ) possible location of the inhabitant,
- $L_{Real}(t)$  is the real location (in one zone) of the inhabitant at the date  $t$ ,
- $L_{Est}(t)$  is the estimated location (in a set of zones, possibly containing only one zone) of the inhabitant at the date  $t$ ,
- $|L_{Est}(t)|$  is the number of zones composing the estimated location (for instance, if the estimated location  $L_{Est}(t_1) = (A, Out)$ , then  $|L_{Est}(t_1)| = 2$ ).

Note that the sum of all the  $CM_{LT(i,j)}$  is equal to 1 i.e.

$$\forall T \sum_{i,j} CM_{LT(i,j)}(T) = 1 \quad (1)$$

Based on this matrix, several performance criteria can be defined. The first one is named *accuracy* and gives the proportion of time for which the estimated location was the

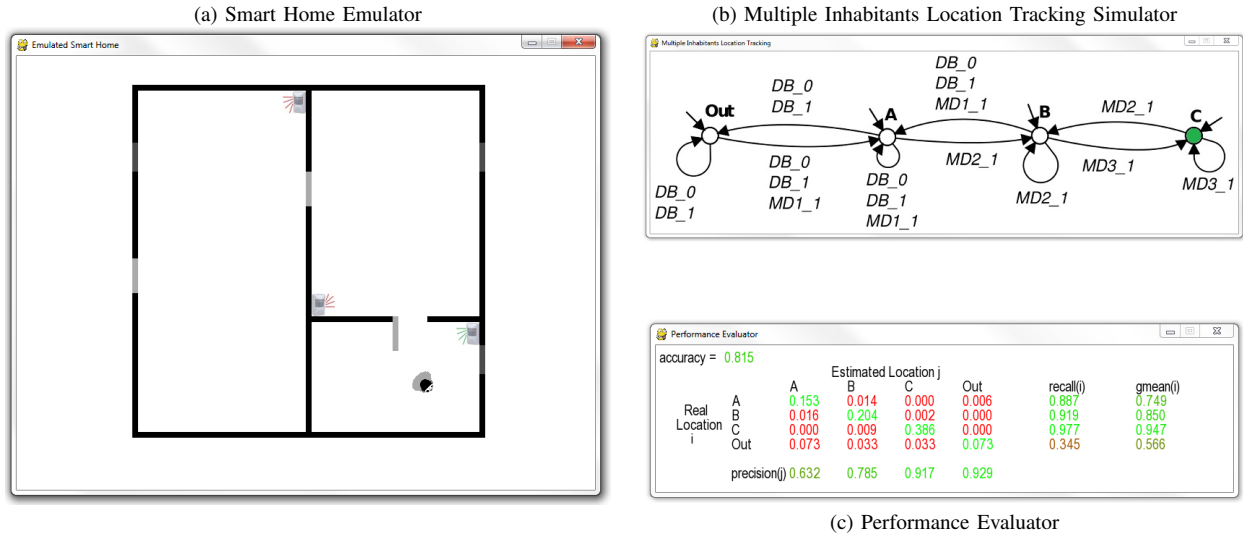


Fig. 6. Screenshot of the developed program

real one during the simulation (e.g. if the estimated location was correct during 6 minutes for a simulation of 10 minutes, then the accuracy is equal to 60%). Formally it is the sum of the diagonal elements of the confusion matrix:

$$accuracy(T) = \sum_i CM_{LT(i,i)}(T) \in [0, 1] \quad (2)$$

The *accuracy* gives a global measure of the location tracking performance. Complementary to this indicator, two criteria concerning each possible location are defined. The *precision*  $p$  is the proportion of time for which a real location is correctly estimated and the *recall*  $r$  is the proportion of time for which the estimated location correctly represents the real location. Formally, for a given location  $i$ :

$$p(i, T) = \frac{CM_{LT(i,i)}(T)}{\sum_j CM_{LT(j,i)}(T)} \in [0, 1] \quad (3)$$

$$r(i, T) = \frac{CM_{LT(i,i)}(T)}{\sum_j CM_{LT(i,j)}(T)} \in [0, 1] \quad (4)$$

The criteria *precision* and *recall* can be combined using a geometric mean called *gmean*:

$$gmean(i, T) = \sqrt{r(i, T) \cdot p(i, T)} \in [0, 1] \quad (5)$$

All of these criteria are calculated and displayed online while simulating the behavior of one or several inhabitants in the smart home. For instance, in Fig. 6(c) representing the performance evaluator, the current *accuracy* is equal to 0.815. The *precision* for the zone  $A$  is equal to  $\frac{0.153}{0.153 + 0.016 + 0 + 0.073} = 0.632$  while the *recall* for this zone is equal to  $\frac{0.153}{0.153 + 0.014 + 0 + 0.006} = 0.887$ .

As an example, a scenario involving a single inhabitant inside the house has been simulated. The location tracking

has been performed using the model of Fig. 2. The evolution of the different criteria are given in Fig. 7 and Fig. 8 for the same duration of simulation.

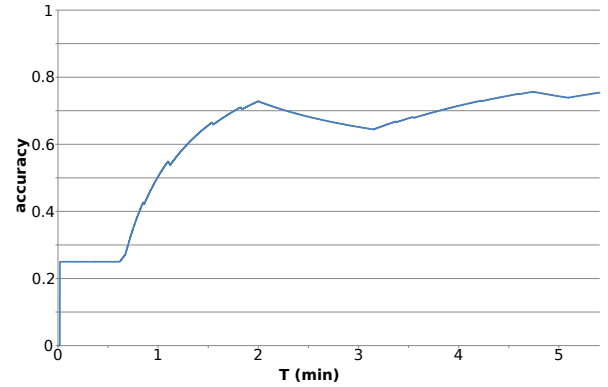


Fig. 7. Evolution of  $accuracy(T)$  along the time  $T$

The evolution of the *accuracy* of LT (Fig. 7) shows a quite good result at the end of the scenario. However, it can be seen that the *accuracy* decreased between 2 and 3 minutes. We simulated the inhabitant going outside the house during this temporal windows. Since there is no sensor outside in our case study, it is quite normal to have such a decrease of the *accuracy*.

The evolution of  $gmean(i, T)$  (Fig. 8) confirms the previous result. The criterion *gmean* shows great performances for the zones  $B$  and  $C$  (i.e. the bedroom and the bathroom) but worse performances for the zones  $Out$  and  $A$  (i.e. outside and in the living room). These results reflect a lack of instrumentation in these two zones or a bad placement of sensors. In our case, there is no sensor observing exclusively outside (lack of instrumentation) and the sensor ( $DB$ ) is observing two zones ( $A$  and  $Out$ ).

These results provide the designer with indications to modify the zone partition and/or the instrumentation. For

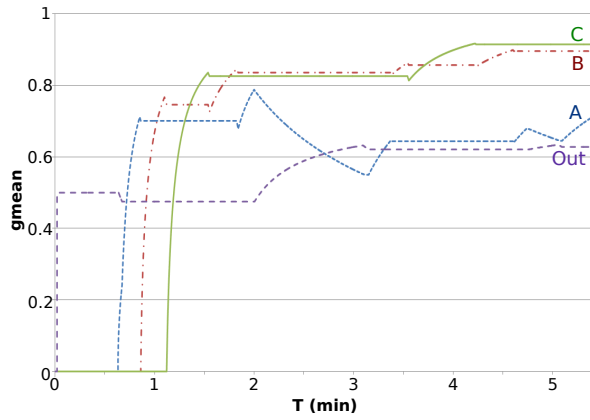


Fig. 8. Evolution of the *precision* and *recall* using the criterion  $gmean(i, T)$  along the time  $T$  and for each location

instance, it is possible to add a floor pressure sensor outside, just in front of the entrance door. By doing this modification and computing systematically the new *DMA* model (not shown here due to a lack of space), we can simulate exactly the same scenario using the new instrumentation and the new model. The results are shown in Fig. 9 for the *accuracy* and in Fig.10 for the *gmean*.

Note that the *accuracy* at the end of this scenario has increased from 75% to 90% (Fig. 9) and the different zones have an increased *gmean*, particularly the zones *A* and *Out* (Fig. 10).

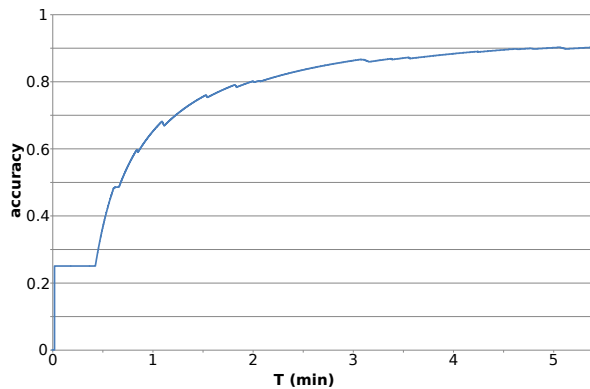


Fig. 9. Evolution of  $accuracy(T)$  along the time  $T$  for the new instrumentation

The results provided by this dynamic evaluation are complementary to those given by our previously proposed analytical approach. Considering the original and the modified instrumentation (with the additional floor pressure sensor), the analytical approach gives the same result, i.e. no *unlocatable zones* and *weak accurate-location-ability*. However, the new criteria *accuracy* and *gmean* allowed to highlight differences between these two possible instrumentations and to quantify how good the new instrumentation is, compared to the original one.

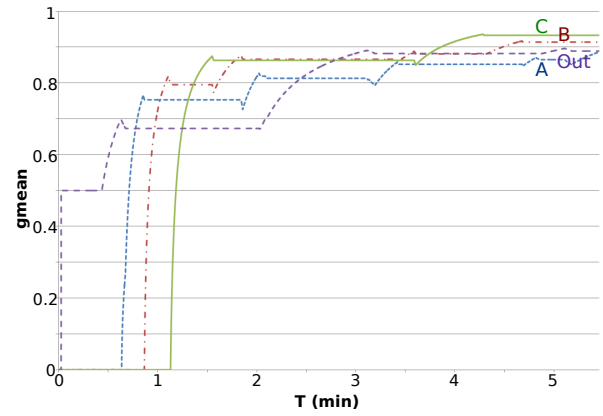


Fig. 10. Evolution of the *precision* and *recall* using the criterion  $gmean(i, T)$  along the time  $T$  and for each location for the new instrumentation

## CONCLUSION

In this paper, we proposed a simulator to evaluate *a priori* the performances of a model for location tracking in smart homes. The results of the discrete event simulation approach are complementary to those of the previously proposed analytical approach.

Future work on this topic will mainly be devoted to the improvement of the realism of the emulated smart home, by including actuators and a more sophisticated sensor model including their possible faults.

## REFERENCES

- [1] R. J. Robles and T.-h. Kim, "Applications, Systems and Methods in Smart Home Technology: A Review," *International Journal of Advanced Science and Technology*, vol. 15, pp. 37–47, Feb. 2010.
- [2] Y. Rahal, H. Pigot, and P. Mabillean, "Location Estimation in a Smart Home: System Implementation and Evaluation Using Experimental Data," *International Journal of Telemedicine and Applications*, 2008.
- [3] C.-R. Yu, C.-L. Wu, C.-H. Lu, and L.-C. Fu, "Human Localization via Multi-Cameras and Floor Sensors in Smart Home," in *proc. of the IEEE Conference on Systems, Man and Cybernetics*, 2006.
- [4] P. Bahl and V. Padmanabhan, "RADAR: An Inbuilding RF-based user location and tracking system," in *Proc. of IEEE INFOCOM*, vol. 2, 2000, pp. 775–784.
- [5] H. S. Lee, B. Song, H. Y. Youn, and K. Chung, "The Method of Location Error Detection and Correcting in Smart Home Environments," in *Proc. of the International Conference on Hybrid Information Technology, Cheju Island, Korea*, 2006, pp. 432–438.
- [6] W.-H. Liao, C.-L. Wu, and L.-C. Fu, "Inhabitants Tracking System in a Cluttered Home Environment Via Floor Load Sensors," *IEEE Trans. on automation science and engineering*, vol. 5, no. 1, pp. 10–20, January 2008.
- [7] M. Danancher, J.-J. Lesage, L. Litz, and G. Faraut, "Online Location Tracking of a Single Inhabitant based on a State Estimator," in *Proc. of the IEEE Conference on Systems, Man and Cybernetics, Manchester, United Kingdom (submitted)*, 2013.
- [8] M. Danancher, J.-J. Lesage, L. Litz, and G. Faraut, "A Discrete Event Model for Multiple Inhabitants Location Tracking," in *Proc. of the 9th IEEE international Conference on Automation Science and Engineering, Madison, U.S.A. (to appear)*, 2013.
- [9] M. Kubat, R. Holte, and S. Matwin, "Machine Learning for the Detection of Oil Spills in Satellite Radar Images," in *Machine Learning*, 1998, pp. 195–215.
- [10] B. Chikhaoui, S. Wang, and H. Pigot, "Activity Recognition in Smart Environments: An Information Retrieval Problem," in *Proc. of the 9th International Conference on Smart Homes and Health Telematics, ICOST'11, Montreal, Canada*, 2011.