



Online Location Tracking of a Single Inhabitant based on a State Estimator

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

► To cite this version:

Mickaël Danancher, Jean-Jacques Lesage, Lothar Litz, Gregory Faraut. Online Location Tracking of a Single Inhabitant based on a State Estimator. Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics - SMC 2013, Oct 2013, Manchester, United Kingdom. pp.391-396, 10.1109/SMC.2013.73 . hal-00873247

HAL Id: hal-00873247

<https://hal.science/hal-00873247>

Submitted on 15 Oct 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.

Online Location Tracking of a Single Inhabitant based on a State Estimator

Mickael Danancher^{*,†}, Jean-Jacques Lesage^{*}, Lothar Litz[†] and Gregory Faraut^{*}

^{*} Automated Production Research Laboratory (LURPA), ENS Cachan, 61 av. du President Wilson, 94235 Cachan, France

Email: {danancher, lesage, faraut}@lurpa.ens-cachan.fr

[†] Institute of Automatic Control, University of Kaiserslautern, P.O. Box 3049, 67653 Kaiserslautern, Germany

Email: litz@eit.uni-kl.de

Abstract—Ambient Assisted Living (AAL) is a field of research aiming to improve the comfort and the safety of aged or disabled people living in their own house. Several AAL approaches require the location of the inhabitant to be tracked or could be improved by considering the real time location of the inhabitant. In this paper, it is proposed to use finite automata to build a model of dynamic tracking of a single person in his house. A method for systematic generation of such a finite automaton, starting from a description of the partition of the house in zones and of the instrumentation of the house, is presented. The evaluation of the relevance of a given instrumentation, based on the analysis of the generated finite automaton, is discussed. Finally, an efficient algorithm for location tracking based on a state estimator is proposed. For the sake of better understanding, an illustrative example is used throughout the paper.

Index Terms—Discrete Event Systems, Finite Automata, Ambient Assisted Living, Location Tracking.

I. INTRODUCTION

Life expectancy has continuously increased in most countries over the last decades; in 2050, 33% of people from the European countries will be at least 60 years old [1]. This leads to new issues concerning the autonomy and the independence of elderly or disabled people. Ambient Assisted Living (AAL) technologies are aiming to help them to live autonomously in a comfortable and safe environment.

Indoor location tracking is among the most active research areas of AAL. This consists in finding the location of inhabitants based on the information given by the different sensors of the house. The knowledge of the location of inhabitants in real time can then be used to improve their comfort or to detect health problems for instance.

In most approaches, location tracking is performed by using data mining techniques [2]–[4]. Consequently, a more or less long learning phase is required before the location tracking 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. For these reasons, we are interested in an approach aiming at the systematic construction of a Discrete Event System (DES) model for the real-time location of an inhabitant into his home.

A new approach to systematically generate a finite automaton model is presented in this paper. This model shows better

performances as the one proposed in [5]. A new location tracking algorithm, based on the construction of a state estimator, as well as a procedure to evaluate the relevance of a given instrumentation for location tracking are also proposed in this paper.

The problem statement and the description of an illustrative case study are given in the following section. In the third section, the approach for systematic generation of a finite automaton is described. In section IV, the location tracking algorithm is presented and illustrated using a real scenario of motion of an inhabitant in his home. The evaluation procedure of an instrumentation is given in section V. In the last section, the contributions are summarized and an outlook for future work is given.

II. PROBLEM STATEMENT AND CASE STUDY

A. Assumptions and problem statement

In this paper, some assumptions are made in order to solve the location tracking problem.

It is first assumed that there is always at most one inhabitant in the home.

In order to help the user to accept the observation of his every movements and to guarantee the respect of his privacy and the reduction of cost, the instrumentation is considered as being composed of 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 pressure sensor for instance).

It is also considered that information given by the sensors does not depend on the ability or the willingness of the inhabitant to provide this information. For instance, if a door is equipped with a door barrier sensor and a door contact sensor, the inhabitant crossing the door will systematically be detected by the barrier sensor but will be detected by the contact sensor only if the inhabitant opens or closes the door in addition to crossing it. Consequently, in our approach, door contact sensors will not be used. For similar reasons, switch sensors are also not considered because while entering a room the inhabitant may or not switch the light on, depending on the sun light or his life habits.

Moreover, it is assumed that the inhabitant has a totally free behavior. Consequently, adopting a DES point of view,

the inhabitant living in an instrumented environment is seen as a spontaneous event generator. These events are the rising and falling edges of the signal emitted by each binary sensor of the house. As a convention, the rising edge and the falling edge of a sensor s_i are respectively denoted as s_{i_1} and s_{i_0} .

Finally, considering the topology of an apartment and a potential lack of instrumentation in some areas, we have also to make the assumption of partial observation of the motion of the inhabitant.

Based on these considerations, the problem of online location tracking can be reformulated in terms of a DES problem: how to estimate in real time the current location of the inhabitant, considered as a spontaneous event generator, based on a potentially incomplete observed sequence of sensor events?

B. Case study

Throughout the paper, the application of our approach will be illustrated on a case study. Several apartments in Kaiserslautern (Germany) have been instrumented with binary sensors. One of them has been described in detail in [6]. Our case study is inspired by this instrumented apartment. Its plan is depicted in Fig. 1. It is composed of five rooms: a vestibule, an open space for the kitchen and the living room, a closet, a bathroom and a bedroom.

Concerning the instrumentation, we consider only four motion detectors (MD_1 in the bedroom, MD_2 in the bathroom, MD_3 in the open space of the living room and the kitchen, MD_4 in the vestibule) and one door barrier sensor (DB placed on the door between the kitchen and the bedroom).

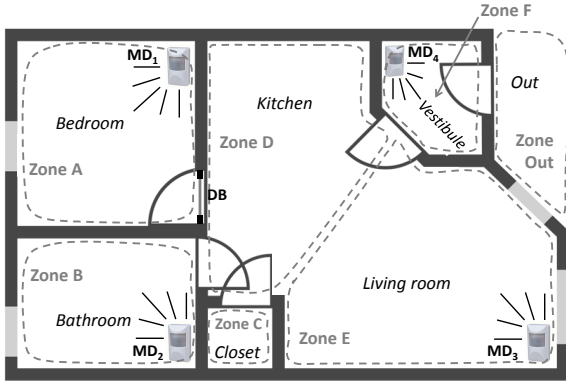


Fig. 1. Topology and instrumentation of the case study

III. SYSTEMATIC GENERATION OF A DES MODEL FOR LOCATION TRACKING

A. Formalization of a zone partition and of an instrumentation

The first step of our approach consists in formalizing both the partition of the apartment in zones and its instrumentation. This is performed offline and requires not much expert knowledge.

A zone partition is denoted as $P^l = (\mathbf{Z}^l, Top^l)$ with $l \in \mathbb{N}$ for the l^{th} partition. \mathbf{Z}^l is a vector of $|\mathbf{Z}^l|$ elements (each element is a zone) and Top^l is a $|\mathbf{Z}^l| \times |\mathbf{Z}^l|$ -matrix

(representing the topology of the apartment). Different choices of zone partition can be made, thus different P^l can be defined. To save space, in this paper we propose only one partition of the apartment of Fig. 1 in 7 zones (6 zones are representing inside the house and 1 is representing outside). This partition, represented by dotted lines in Fig. 1, is defined by $P = (\mathbf{Z}, Top)$ described below. The only 2 rules that are needed to define the zone partition are that the whole house should be covered by zones and that there can not be overlapping zones. This means that the inhabitant is always in exactly one zone: location tracking consists in estimating this zone. Based on this zone partition, the topology has to be described using the matrix Top where $Top_{(i,j)} = 1$ if there exists a direct path between zone \mathbf{Z}_i and zone \mathbf{Z}_j . For instance, there exists a direct path between the zone $\mathbf{Z}_1 = A$ and the zone $\mathbf{Z}_4 = D$ and this can be seen in the matrix Top because $Top_{(1,4)} = 1$. On the contrary, there is no direct path between the zone $\mathbf{Z}_1 = A$ and the zone $\mathbf{Z}_2 = B$ and this can be seen in the matrix Top because $Top_{(1,2)} = 0$. It can be noticed that Top is always a symmetrical matrix although this is not mandatory from a formal point of view.

$$\mathbf{Z} = \begin{pmatrix} A \\ B \\ C \\ D \\ E \\ F \\ Out \end{pmatrix} \quad Top = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

The ability of an instrumentation to detect motion in the zones of a partition P^l is defined as $I^{(m,l)} = (\mathbf{S}^m, \mathbf{Z}^l, Obs^{(m,l)})$ where \mathbf{S}^m is a vector of $|\mathbf{S}^m|$ elements (each element is a sensor) and $Obs^{(m,l)}$ is a $|\mathbf{S}^m| \times |\mathbf{Z}^l|$ -matrix (representing the zones observed by each sensor). In this paper, we define only one instrumentation and its ability to detect motion in zones of P , denoted as $I = (\mathbf{Z}, \mathbf{S}, Obs)$. \mathbf{Z} is described above, \mathbf{S} and Obs are described below for the case study. There are 4 motion detectors and one door barrier sensor composing the vector \mathbf{S} . The observed zones of each sensor are represented by the matrix Obs where $Obs_{(i,j)} = 1$ if the sensor \mathbf{S}_i can observe motion in zone \mathbf{Z}_j . For instance, the sensor MD_2 can observe motion or presence in the zone $\mathbf{Z}_2 = B$ and this can be seen in the matrix Obs because $Obs_{(2,2)} = 1$. A sensor can also observe several zones, for instance the sensor MD_3 allow to observe motion in zones $\mathbf{Z}_4 = D$ and $\mathbf{Z}_5 = E$ and this can be seen in the matrix Obs because $Obs_{(3,4)} = Obs_{(3,5)} = 1$.

$$\mathbf{S} = \begin{pmatrix} MD_1 \\ MD_2 \\ MD_3 \\ MD_4 \\ DB \end{pmatrix} \quad Obs = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

B. Algorithm for systematic generation of a DES model

Once a zone partition and the instrumentation have been defined, a finite automaton representing the detectable motion of the inhabitant can be built. This finite automaton is called Detectable Motion Automaton (*DMA*) and is formally defined as $DMA = (Q, \Sigma, \delta, Q_0)$ with:

- Q a set of states,
- Σ an alphabet of events,
- $\delta : Q \times \Sigma \rightarrow 2^Q$ the transition function,
- $Q_0 \subseteq Q$ the set of initial states.

We also adopt the notation that $\delta(q, \sigma)!$ means that $\delta(q, \sigma) \subseteq Q$ i.e. at least one transition from state q labeled with the event σ is defined.

We propose Algorithm 1 to systematically generate the *DMA*. The first step consists in defining the set Q of the states of *DMA* (line 3). Each state represents a zone of the house. Furthermore, each state of the automaton is defined as being an initial state of the automaton (line 4) because the initial location of the inhabitant is assumed to be unknown. Then, the events are created (line 6 to 10). Both the falling edges and rising edges are considered for the sensors like door barrier sensors of floor pressure sensors since both edges are indicating a movement of the inhabitant. On the contrary, only the rising edge is considered for motion detectors because a falling edge is not representative of a change of location but it can just be symptomatic of someone staying motionless in a zone. Finally, the transitions are defined (line 12 to 19). There exists a transition between a state q_i and a state q_j labeled with the event S_{k_1} (and there also exists a transition labeled with the event S_{k_0} for sensors not being motion detectors) if it is topologically possible ($Top_{(i,j)} = 1$) and if the zone related to the destination state q_j is observed by the sensor S_k ($Obs_{(k,j)} = 1$).

For the case study, the resulting *DMA* is depicted on Fig. 2. Some particular points have to be highlighted.

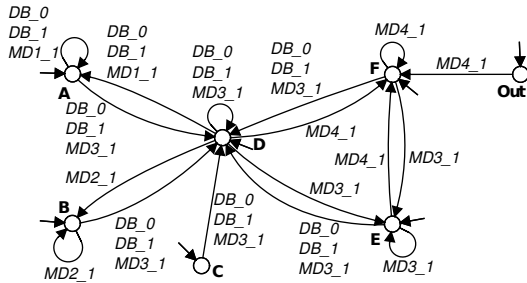


Fig. 2. Detectable Motion Automaton *DMA*

As stated before, it is assumed that the initial location of the inhabitant is unknown. This is the most permissive hypothesis and can be seen in the model where each state is initial. Knowing accurately the initial location is not necessary to perform online location tracking because the current estimation of the location of the inhabitant does not strongly depends on his initial location. If for some smart home applications it is mandatory to know the initial location of the inhabitant, some

Algorithm 1 Generation of *DMA*

Require: Z, Top, S, Obs

```

1:  $DMA := \langle Q, \Sigma, \delta, Q_0 \rangle$ 
2: for  $i := 1$  to  $|Z|$  do
3:   Create state  $q_i = Z_i$  in  $Q$ 
4:   Set state  $q_i$  initial,  $q_i \in Q_0$ 
5: end for
6: for  $k := 1$  to  $|S|$  do
7:   Create event  $S_{k\_1}$  in  $\Sigma$ 
8:   if  $S_k$  is not a motion detector then
9:     Create event  $S_{k\_0}$  in  $\Sigma$ 
10:  end if
11: end for
12: for  $(i, j, k) := (1, 1, 1)$  to  $(|Z|, |Z|, |S|)$  do
13:   if  $(Obs_{(k,j)} = 1) \wedge (Top_{(i,j)} = 1)$  then
14:     Define  $\delta(q_i, S_{k\_1}) = q_j$ 
15:     if  $S_k$  is not a motion detector then
16:       Define  $\delta(q_i, S_{k\_0}) = q_j$ 
17:     end if
18:   end if
19: end for
20: return  $DMA$ 

```

techniques (for instance in [7]) can be used to determine the initial state of an automaton after observing a more or less long sequence of events.

Some zones are not observed by any sensor. This can be seen in the model: certain states of *DMA* are reachable only by the fact that they are initial (*Out* and *C*). This is symptomatic of a lack of instrumentation in these zones.

Some sensors are observing more than one zone. This can be seen in the model: *DMA* is a non-deterministic Finite Automaton (e.g. two transitions labeled with the same event $MD3_1$, having *D* as source state, one reaching state *D* and one reaching *E*).

Based on the *DMA*, the aim of location tracking is to estimate the reached state(s) for an observed sequence of events. Since *DMA* is not deterministic, the construction of a state estimator of *DMA* is a way to perform location tracking. The procedure to obtain this estimator is detailed in the next subsection.

C. Offline computation of the state estimator

Since each state of *DMA* represents the location of the inhabitant in exactly one zone of the house, the semantics of the states of the automaton is important and has to be kept while building a state estimator. This estimator is obtained by computing the equivalent deterministic automaton, by using a standard algorithm proposed in many works e.g. [8] or [9].

The result is a Finite Automaton denoted as $Est(DMA) = (Q_{Est}, \Sigma, \delta_{Est}, q_{Est_0})$ with:

- $Q_{Est} \subseteq 2^Q$ a set of states (each state is a subset of the set of states Q of *DMA*),
- Σ the same alphabet of events as the alphabet of *DMA*,

- $\delta_{Est} : Q_{Est} \times \Sigma \rightarrow Q_{Est}$ the deterministic transition function,
- $q_{Est_0} \in Q_{Est}$ the unique initial state.

For the case study, $Est(DMA)$ is represented in Fig. 3. It is a Deterministic Finite Automaton and each of its state represents an estimation (more or less accurate) of the state of DMA i.e. of the location of the inhabitant.

The state estimator $Est(DMA)$ is computed offline. This operation is theoretically of exponential complexity $O(2^{|Z|})$ with $|Z|$ the number of zones, but for practical applications, the number of zones is not so large and the complexity is limited.

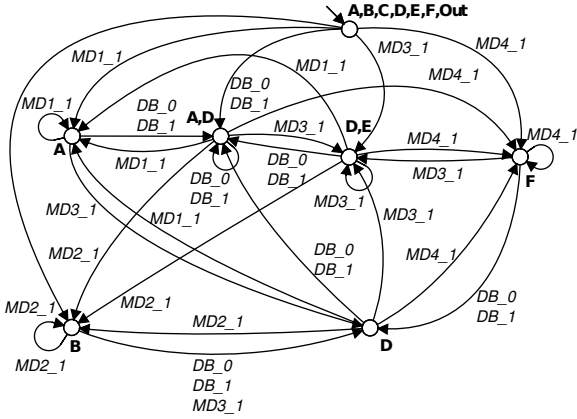


Fig. 3. State estimator for Location Tracking $Est(DMA)$

IV. MODEL-BASED ONLINE LOCATION TRACKING

From the state estimator, we propose Algorithm 2 to compute the online location tracking that gives in real time an estimation (denoted as L) of the zone(s) where the inhabitant is.

Algorithm 2 Location tracking algorithm

Require: $Est(DMA) = (Q_{Est}, \Sigma, \delta_{Est}, q_{Est_0})$

- 1: Initialization of the location tracking:
Current location L = Initial state q_{Est_0}
 - 2: **while** location tracking is active **do**
 - 3: Wait for a new event e
 - 4: New event e is observed
 - 5: **if** $\delta_{Est}(L, e) \neq \emptyset$ **then**
 - 6: $L' = \delta_{Est}(L, e)$
 - 7: Update current location $L = L'$
 - 8: **else**
 - 9: The location remains L
 - 10: **end if**
 - 11: **end while**
-

At the beginning, the estimation L , representing the current location of the inhabitant, is equal to the initial state of the estimator i.e. to the set of states of DMA . Then the algorithm waits for the occurrence of a new event e emitted by a sensor. It is assumed that two events can not simultaneously occur.

When an event e occurs, the algorithm computes, with respect to the transition function, the state L' , successor of L , such that $L' = \delta_{Est}(L, e)$. When the new state is computed, the state L is updated by the new state L' . Then, the program waits for another new event to compute again the new location of the inhabitant.

The complexity of this algorithm is linear in the number of sensors $O(2|S|)$, where $|S|$ is the number of sensors. This can be demonstrated by calculating the maximal number of transitions having L as the source state that have to be explored before finding the one involving the observed event e . There are at most $2|S|$ transitions (one involving the rising edge + one involving the falling edge for each sensor) having L as a source state. Thus, algorithm 2 is efficient and the real-time computation of the online location tracking can be easily performed.

The practical use of this algorithm can be illustrated on a real scenario of motion of the inhabitant for the case study (see table I).

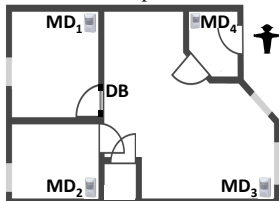
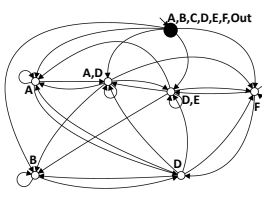
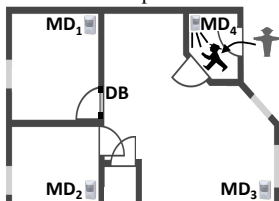
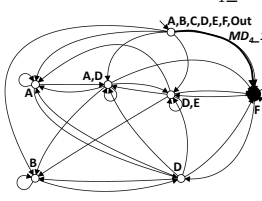
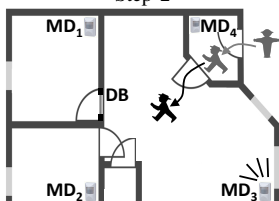
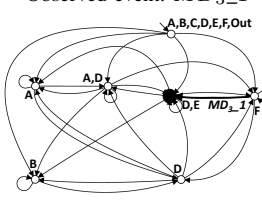
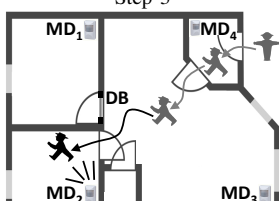
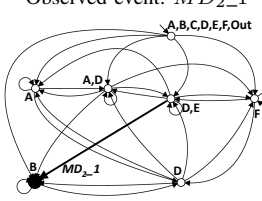
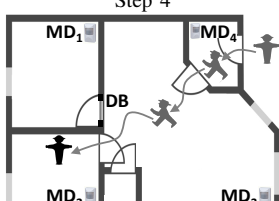
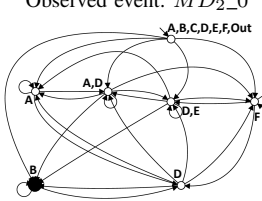
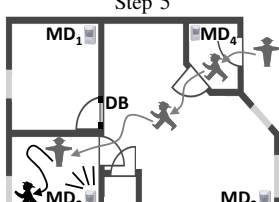
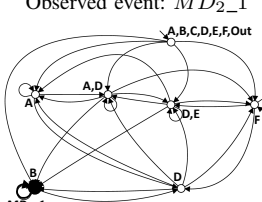
The scenario is the following: the inhabitant is entering the house by the front door and is going to the bathroom (crossing the vestibule, the living room and the kitchen).

- 0) The location tracking algorithm is initialized. Current location is $L = (A, B, C, D, E, F, Out)$ which is the initial state of $Est(DMA)$. It is a very inaccurate location because it is assumed the initial location of the inhabitant is unknown.
- 1) A rising edge of the motion detector in the vestibule (MD_{4_1}) is observed. The current location is updated and is now $L = F$. The location of the inhabitant is now accurate.
- 2) A rising edge of the motion detector in the living room (MD_{3_1}) is observed. The location is updated and is equal to $L = (D, E)$. The location is inaccurate, it can not be decided whether the inhabitant is in the kitchen or in the living room.
- 3) A rising edge of the motion detector in the bathroom (MD_{2_1}) is observed. The location is updated and $L = B$. The location is accurate again.
- 4) A falling edge of the motion detector in the bathroom (MD_{2_0}) is observed. The falling edges of motion detectors are not taken into account for location tracking, consequently the event MD_{2_0} does not belong to the alphabet of events of the automaton. Thus, the estimation of the current location is not updated and $L = B$.
- 5) A rising edge of the motion detector in the bathroom (MD_{2_1}) is observed. Since there are self loops in the model representing the inhabitant moving within a zone and not changing of location, the location is updated and $L = B$. The location is accurate.

We applied this model-based algorithm on longer real scenarios from real case studies in Kaiserslautern (Germany). It is efficient and shows good performances to locate a single inhabitant in an instrumented home. However, these experiments showed that the accuracy of location is greatly connected to the choices of a zone partition and of an instrumentation. To quan-

tify how good a combination zone partition - instrumentation ($P^l, I^{(m,l)}$) is for location tracking, we propose in the next section an evaluation procedure based on the systematically generated Finite Automaton.

TABLE I
REAL SCENARIO OF MOTION OF AN INHABITANT

Real location	Estimated location
<p>Step 0</p> 	<p>Initialization</p> 
<p>Step 1</p> 	<p>Observed event: MD_{4_1}</p> 
<p>Step 2</p> 	<p>Observed event: MD_{3_1}</p> 
<p>Step 3</p> 	<p>Observed event: MD_{2_1}</p> 
<p>Step 4</p> 	<p>Observed event: MD_{2_0}</p> 
<p>Step 5</p> 	<p>Observed event: MD_{2_1}</p> 

V. MODEL-BASED EVALUATION OF A COUPLE ZONE PARTITION - INSTRUMENTATION

This performance evaluation is performed *a priori* i.e. offline and before performing location tracking. Thus, different combinations can be evaluated and compared before choosing the best combination zone partition - instrumentation and installing the sensors.

For a given combination zone partition - instrumentation, two criteria are proposed:

- The *unlocationable zones* (inability of the sensors to detect the presence of the inhabitant in certain zones).
- The *strong and weak accurate-location-ability*, based on the accuracy of location estimation (guarantee to estimate accurately the location of the inhabitant).

A. Unlocationable zones

The ability of $(P^l, I^{(m,l)})$ to detect the presence of the inhabitant in each zone can be quantified by computing the set of states representing unlocationable zones Q_{UZ} . They are the states of DMA being reachable only by the fact that they are initial. It is impossible to detect the presence of the inhabitant in these zones because they do not belong to the set of states of $Est(DMA)$ except to its initial state.

Definition 1: Unlocationable Zones

The number of Unlocationable zones (if they exist) can be quantified by the cardinal of Q_{UZ} with:

$$Q_{UZ} = \{q' \in Q \mid \nexists (q, \sigma) \in Q \times \Sigma, \delta(q, \sigma) = q'\}$$

For the case study: $Q_{UZ} = \{C; Out\}$ and $|Q_{UZ}| = 2$. This result shows that the inhabitant will never be located in zone C or in zone Out . This is not critical for zone C since it only represents a closet in the house. On the contrary, this is critical for the zone Out since the location of the inhabitant in this zone is meaning that the inhabitant is not at home. This particular information is mandatory for some applications, for instance applications aiming to detect health problem when the inhabitant is at home. Such an application should be deactivated each time the inhabitant is outside in order to avoid false alarm. Thus, the ability to give the information “inhabitant is out of the home” is required. Consequently, this combination (P, I) does not have the ability to guarantee that the apartment is empty.

B. Accuracy of the estimated locations

The accuracy of the location estimation is computed on the state estimator $Est(DMA)$. Each state of $Est(DMA)$ is a subset of the set of states of DMA . Thus, the accuracy of each estimated location can be quantified by the cardinal of this subset, denoted as $|q_{Est}| \forall q_{Est} \in Q_{Est}$. For instance in $Est(DMA)$ of the case study, $|(D, E)| = 2$ and $|A| = 1$. Each $q_{Est} \in Q_{Est}$ such that $|q_{Est}| = 1$ represents an accurate estimation of the location (inhabitant accurately located in one zone). In opposite, each $q_{Est} \in Q_{Est}$ such that $|q_{Est}| > 1$ represents more or less inaccurate estimation of the real location (it can not be decided accurately in which zone the inhabitant is really located). Based on this consideration, the states of $Est(DMA)$ are divided in two subsets: Q_a representing the

set of accurate estimated locations and Q_i representing the set of inaccurate estimated locations. $Q_{Est} = Q_a \cup Q_i$ where:

- $Q_a = \{q_{Est} \in Q_{Est} \text{ such that } |q_{Est}| = 1\}$
- $Q_i = \{q_{Est} \in Q_{Est} \text{ such that } |q_{Est}| > 1\}$

Note that $Q_a \cap Q_i = \emptyset$ and that the initial state $q_{Est_0} \in Q_i$. For the case study, $Q_a = \{A, B, D, F\}$ and $Q_i = \{(A, B, C, D, E, F, Out), (A, D), (D, E)\}$

Based on these definitions of the sets Q_a and Q_i , two notions of accurate-location-ability are defined. They are inspired by the notions of strong and weak detectability of DES [10].

C. Accurate-location-ability

The *accurate-location-ability* is defined as the ability to estimate accurately the location of the inhabitant. Two definitions for *strong* and *weak accurate-location-ability* as well as properties to check strong or weak accurate-location-ability are proposed.

Definition 2: Strong accurate-location-ability

A combination $(P^l, I^{(m,l)})$ is strongly accurate-location-able if after a finite sequence of sensor events the location is accurate from now, whatever the inhabitant is doing.

Property 1:

A combination $(P^l, I^{(m,l)})$ is strongly accurate-location-able if Q_a is not empty and there is no loop between states of Q_i and no transition from a state of Q_a to a state of Q_i . Formally this property is written:

$$(P^l, I^{(m,l)}) \text{ is strongly accurate-location-able if:}$$

$$\left\{ \begin{array}{l} \bullet Q_a \neq \emptyset \\ \bullet \forall \sigma_1 \sigma_2 \dots \sigma_m \in \Sigma^* \mid \exists (q_1, q_2, \dots, q_m, q_{m+1}) \in Q_i^{m+1}; \\ \quad \delta_{Est}(q_1, \sigma_1) = q_2, \dots, \delta_{Est}(q_m, \sigma_m) = q_{m+1}; \\ \quad \forall (q_i, q_j) \in (q_1, q_2, \dots, q_{m+1})^2 \quad q_i \neq q_j \\ \bullet \nexists (q, \sigma) \in Q_a \times \Sigma \mid \delta_{Est}(q, \sigma) \in Q_i \end{array} \right.$$

Proof: The condition of non emptiness of Q_a and the condition of no transition from Q_a to Q_i guarantee that once a state of Q_a is reached, the subsequent states are also in Q_a and thus the location will be accurate. Moreover, if a state of Q_i is reached, the condition of no loop in Q_i guarantees that after a sequence of events of maximum length $|Q_i|$ a state belonging to Q_a will be reached. ■

Definition 3: Weak accurate-location-ability

A combination $(P^l, I^{(m,l)})$ is weakly accurate-location-able if it is possible that the location is accurate from now, depending on what the inhabitant is doing.

Property 2:

A combination $(P^l, I^{(m,l)})$ is weakly accurate-location-able if Q_a is not empty and there are loops in Q_a . Formally this property is written:

$$(P^l, I^{(m,l)}) \text{ is weakly accurate-location-able if:}$$

$$\left\{ \begin{array}{l} \bullet Q_a \neq \emptyset \\ \bullet \exists \sigma_1 \sigma_2 \dots \sigma_m \in \Sigma^* \mid \exists (q_1, q_2, \dots, q_m, q_{m+1}) \in Q_a^{m+1}; \\ \quad \delta_{Est}(q_1, \sigma_1) = q_2, \dots, \delta_{Est}(q_m, \sigma_m) = q_{m+1}; \\ \quad \exists (q_i, q_j) \in (q_1, q_2, \dots, q_{m+1})^2 \quad q_i = q_j \end{array} \right.$$

Proof: The condition of non emptiness of Q_a and the fact that every states of Q_a are accessible (by construction of the estimator) guarantees that it is possible to enter at least

one loop on Q_a . Thus, there exists a sequence of events such that after a certain number of events the current location and subsequent locations become accurate. ■

For the case study, strong location-ability is not obtained because there is at least one transition between Q_a and Q_i in $Est(DMA)$ (for example the transition from F to (D, E) labeled with event MD_{3_1}). Thus there exists a sequence of events such that the location becomes inaccurate after it has been accurate. For instance the sequence of events MD_{4_1}, MD_{3_1} leads to the sequence of locations $(A, B, C, D, E, F, Out), F, (D, E)$ where F is accurate and the first and third locations are inaccurate.

However, since Q_a is not empty and there exist loops on Q_a (for instance the self loop on state A , formally written $\delta_{Est}(A, MD_{1_1}) = A$), the weak location-ability is guaranteed. There exists at least one sequence of events such that the current location becomes accurate and all the subsequent location are accurate. For instance the sequence $MD_{3_1}, MD_{1_1}, MD_{1_1}^*$ leads the location to be inaccurate in (A, B, C, D, E, F, Out) , then it is inaccurate in (D, E) and then it is accurate in A and remains accurate in A since $\delta_{Est}(A, MD_{1_1}) = A$.

VI. CONCLUSION

In this paper, we proposed an approach for model-based location tracking of a single inhabitant in his instrumented home. We developed a method for systematic generation of the model and model-based criteria to evaluate the relevance of an instrumentation for location tracking.

Our current work is devoted to relax the restrictive hypothesis of single inhabitant and thus to extend the proposed approach to multiple inhabitants location tracking.

REFERENCES

- [1] United Nations Department of Economic and Social Affairs - Population Division, "Word Population Ageing," 2009.
- [2] C.-R. Yu, C.-L. Wu, C.-H. Lu, and L.-C. Fu, "Human Localization via Multi-Cameras and Floor Sensors in Smart Home," pp. 3822–3827, 2006.
- [3] Y. Rahal, H. Pigot, and P. Mabilieu, "Location Estimation in a Smart Home: System Implementation and Evaluation Using Experimental Data," *International Journal of Telemedicine and Applications*, 2008.
- [4] 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.
- [5] M. Danancher, J.-J. Lesage, and L. Litz, "Indoor Location Tracking Based on a Discrete Event Model," in *Proceedings of ICOST 2012, Lecture Notes in Computer Science*, vol. 7251, June 2012, pp. 262–265.
- [6] M. Floeck, L. Litz, and A. Spellerberg, *Monitoring Patterns of Inactivity in the Home with Domotics Networks*, ios press amsterdam ed. Augusto, J.C. et al. (eds), 2012, pp. 258–282.
- [7] S. Shu and F. Lin, "I-Detectability of Discrete-Event Systems," *IEEE Trans. on automation science and engineering*, vol. 10, no. 1, pp. 187–196, January 2013.
- [8] C. Cassandras and S. LaFortune, *Introduction to Discrete Event Systems*, springer-verlag ed., 2009.
- [9] R. van Glabbeek and B. Ploeger, "Five Determinisation Algorithms," in *Proceedings of the 13th international conference on Implementation and Applications of Automata, CIAA*. Springer-Verlag Berlin, Heidelberg, 2008, pp. 161–170.
- [10] S. Shu and F. Lin, "Detectability of Discrete Event Systems," *IEEE Transactions on Automatic Control*, vol. 52, no. 12, pp. 2356–2359, 2007.