

A Discrete Event Model for Multiple Inhabitants Location Tracking

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

► **To cite this version:**

Mickaël Danancher, Jean-Jacques Lesage, Lothar Litz, Gregory Faraut. A Discrete Event Model for Multiple Inhabitants Location Tracking. 9th IEEE International Conference on Automation Science and Engineering - CASE 2013, Aug 2013, Madison, WI, United States. pp.922-927, 10.1109/CoASE.2013.6653946 . hal-00855156

HAL Id: hal-00855156

<https://hal.archives-ouvertes.fr/hal-00855156>

Submitted on 29 Aug 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 Discrete Event Model for Multiple Inhabitants Location Tracking

Mickael Danancher^{1,2}, Jean-Jacques Lesage¹, Lothar Litz², Gregory Faraut¹

Abstract—Smart Home technologies are aiming to improve the comfort and safety of the inhabitants into their houses. To achieve this goal, online indoor location tracking of the inhabitants is often used to monitor the air conditioning, to detect dangerous situations and for many other applications. In this paper, it is proposed an approach to build a model allowing dynamic tracking of several persons in their house. A method to construct such a model by using finite automata and Discrete Event System (DES) paradigms is presented. An approach to reduce the size of the model is also introduced. Finally, an efficient algorithm for location tracking is proposed. For the sake of better understanding, an illustrative example is used throughout the paper.

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

I. INTRODUCTION

Smart Home technologies are aiming to help people to live in a comfortable and safe environment. A smart home is a dwelling equipped with sensors and actuators. It is often also composed of a communication network connecting the key electrical devices and allowing them to be remotely monitored or controlled. 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 shut down of dangerous devices or health problem detection, for instance).

Indoor location tracking is most often required to adapt 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 signals generated by the different sensors of the house.

In most approaches, location tracking is performed by using data mining techniques [1]–[3]. 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 previously proposed an approach aiming at the systematic construction of a DES model for the location tracking of an inhabitant into his home [4].

We assumed in this previous work that at most one inhabitant was present in the instrumented house at each time. In the present work, we relax this hypothesis and propose an approach to create a finite automaton model and an algorithm for location tracking of multiple inhabitants.

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 for single inhabitant location tracking is briefly recalled. In section IV, the procedure to create a model of the detectable motion of several inhabitants is presented. The location tracking algorithm based on this model is given in section V and illustrated on a real scenario of motion of two inhabitants in their home. 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 are always at most N inhabitants in the home.

In order to help the users to accept the observation of their every movements and to guarantee the respect of their 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 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 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 an inhabitant may or not switch the light on, depending on the sun light or his life habits.

Moreover, it is assumed that each inhabitant has a totally free behavior and each inhabitant behaves independently from the other. Consequently, adopting a Discrete Event System (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

¹M. Danancher, J.-J. Lesage and G. Faraut are with Automated Production Research Laboratory (LURPA), ENS Cachan, 61 av. du President Wilson, 94235 Cachan, France, {danancher, lesage, faraut}@lurpa.ens-cachan.fr

²M. Danancher and L. Litz are with the Institute of Automatic Control, University of Kaiserslautern, P.O. Box 3049, 67653 Kaiserslautern, Germany litz@eit.uni-kl.de

the signal emitted by each binary sensor of the house. 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} .

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, the inhabitants are non-distinguishable by the sensors i.e. a signal generated by a sensor means someone is moving in its proximity but gives no information about who this person is.

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

B. Case study

Throughout the paper, the application of this approach will be illustrated on a case study. An example of smart home is given in Fig. 1. It has three rooms: an open space for the kitchen and the living room, a bedroom and a bathroom. Three motion detectors are installed: MD_1 in the open space of the living room and the kitchen, MD_2 in the bedroom and MD_3 in the bathroom. Moreover a door barrier sensor DB (detecting an inhabitant crossing the door) is installed on the front door of the house. The chosen case study has a small size (only three rooms and four sensors) for the sake of better understanding. However, our approach has been successfully applied to bigger apartments.

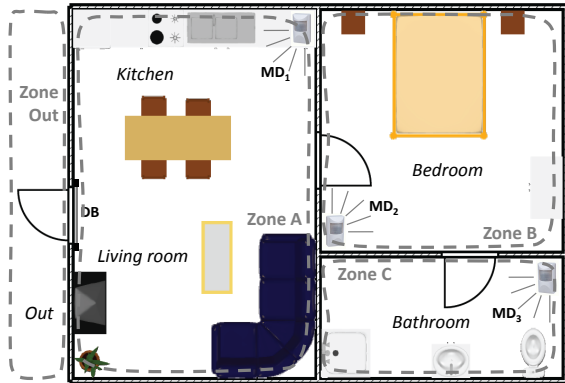


Fig. 1. Topology and instrumentation of the case study

III. A FINITE AUTOMATON MODEL FOR SINGLE INHABITANT LOCATION TRACKING

In previous work [4], we proposed an approach to systematically build a model for location tracking of a single inhabitant. This approach is briefly recalled and illustrated on the case study.

The home is divided into four zones *Out*, *A*, *B* and *C* (see gray dashed lines in Fig. 1). The topology of this zone partition (i.e. the direct paths between zones) and

the description of the zones observed by each sensors are used to systematically generate a Finite Automaton model representing the detectable motion of a single inhabitant. This finite automaton is called Detectable Motion Automaton (*DMA*) and is defined as $DMA = (Q, \Sigma, \delta, Q_0)$ with:

- Q a set of states (one state for each zone of the house),
- Σ an alphabet of events (the rising and falling edges generated by the sensors),
- $\delta : Q \times \Sigma \rightarrow 2^Q$ 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 proposed an algorithm to generate *DMA*. The result of this algorithm on the case study is represented in Fig. 2.

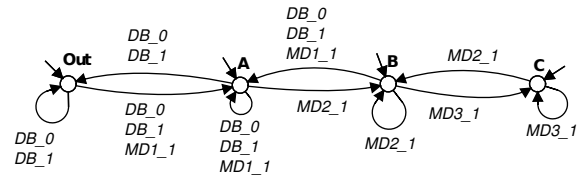


Fig. 2. Detectable Motion Automaton *DMA*

A strong semantics is associated to the states of this automaton since each state represents the location of the inhabitant in a zone of the house. The transitions and associated events represent observable motion between two zones or within a zone in case of a self-loop.

It is assumed that the initial location of the inhabitant is unknown. This 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 hardly depends on his initial location. If for some smart home applications it is mandatory to know the initial location of the inhabitant, some techniques (for instance in [5]) can be used to determine the initial state of an automaton after observing a more or less long sequence of events.

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 DB_1 , having *A* as source state, one reaching state *A* and one reaching *Out*).

Based on the *DMA*, the aim of location tracking is to estimate the reached state for an observed sequence of events. Since *DMA* is not deterministic, there are two possible procedures to perform online location tracking:

- The estimated current location (set of current states of *DMA*) is directly computed online based on this non-deterministic model.
- A state estimator is built offline in a first step and then the location tracking is performed online using this state estimator.

IV. A MODEL OF THE DETECTABLE MOTION OF MULTIPLE INHABITANTS

Based on the model of the detectable motion of a single inhabitant, we propose an approach to model the detectable motion of multiple inhabitants. An overview of this approach is given in Fig. 3. It is composed of 3 steps where the first one is the creation of an extended model of the detectable motion of each one of the inhabitants living in the instrumented house with the $N - 1$ other inhabitants. The second step consists in performing the synchronous composition of this models in order to get a model of the detectable motion of the N inhabitants being together in the house. The third step consists in reducing this model in order just to have a model of the number of inhabitants in each zone and not a model of the location of each inhabitant because it is assumed that the inhabitants are not distinguishable by the sensors. These three steps are detailed in the three following subsections.

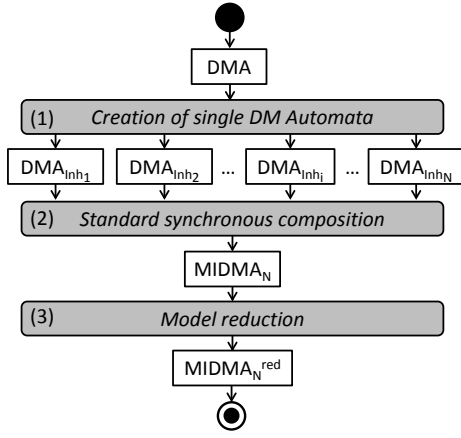


Fig. 3. Overview of the approach to get a model of the detectable motion of multiple inhabitants

A. Single models of the detectable motion

The model representing the detectable motion of each inhabitant of the set $Inh = \{Inh_1, Inh_2, \dots, Inh_N\}$ is created in two steps. It is a Finite Automaton, denoted as DMA_{Inh_i} for the i^{th} inhabitant, defined as $DMA_{Inh_i} = (Q_{Inh_i}, \Sigma_{Inh_i}, \delta_{Inh_i}, Q_{0_{Inh_i}})$ with:

- $Q_{Inh_i} = Q$ the set of states (associated to the zones of the house),
- Σ_{Inh_i} an alphabet of events,
- $\delta_{Inh_i} : Q_{Inh_i} \times \Sigma_{Inh_i} \rightarrow 2Q_{Inh_i}$ the transition function,
- $Q_{0_{Inh_i}} = Q_{Inh_i}$ the set of initial states.

The motion of each single inhabitant is represented by the previously described DMA (Fig. 2). Consequently, each DMA_{Inh_i} has the same structure (states and transitions) as DMA . Each state q_{Inh_i} of Q_{Inh_i} has the following semantics: "inhabitant Inh_i is located in zone q ".

The motion of the inhabitants (moving between two different zones or within a zone) is observed by the sensors of the house. When observing multiple inhabitants, each time a sensor detects motion and emits an event, this event represents either one inhabitant moving alone, or one inhabitant

moving at the same time as 1, 2, ..., $N - 1$ other inhabitants. Different events are introduced in order to represent these different cases.

For instance, considering $N = 3$ inhabitants with $Inh = \{a, b, c\}$ and focusing on the observation of the motion of the inhabitant a , 4 events are defined for each observable sensor event $\sigma \in \Sigma$:

- σ_a represents the observation of the motion of the inhabitant a alone
- σ_{ab} represents the observation of the motion of both inhabitant a and inhabitant b
- σ_{ac} represents the observation of the motion of both inhabitant a and inhabitant c
- σ_{abc} represents the observation of the motion of all the inhabitants a, b and c

In this case, the alphabet of events of DMA_a is therefore $\Sigma_a = \bigcup_{\sigma \in \Sigma} \{\sigma_a, \sigma_{ab}, \sigma_{ac}, \sigma_{abc}\}$ with Σ the set of sensor events of DMA . In a similar manner, $\Sigma_b = \bigcup_{\sigma \in \Sigma} \{\sigma_b, \sigma_{ba}, \sigma_{bc}, \sigma_{bac}\}$ and $\Sigma_c = \bigcup_{\sigma \in \Sigma} \{\sigma_c, \sigma_{ca}, \sigma_{cb}, \sigma_{cab}\}$. This is illustrated on the case study for 3 inhabitants with $Inh = \{a, b, c\}$ in Tab. I.

TABLE I
ALPHABETS OF EVENTS OF DMA AND OF THE DIFFERENT DMA_{Inh_i}

Σ ($ \Sigma = 5$)	Σ_a ($ \Sigma_a = 20$)	Σ_b	Σ_c
MD_{1-1}	MD_{1-1a} MD_{1-1ab} MD_{1-1ac} MD_{1-1abc}	MD_{1-1b} MD_{1-1ab} MD_{1-1bc} MD_{1-1abc}	MD_{1-1c} MD_{1-1ac} MD_{1-1bc} MD_{1-1abc}
\vdots	\vdots	\vdots	\vdots
DB_0	DB_0a DB_0ab DB_0ac DB_0abc	DB_0b DB_0ab DB_0bc DB_0abc	DB_0c DB_0ac DB_0bc DB_0abc

Note that $\forall (i, j) \in \{1, 2, \dots, N\}^2$ the events $\sigma_{Inh_i Inh_j}$ and $\sigma_{Inh_j Inh_i}$ both represent the observation of the motion of both inhabitants Inh_i and Inh_j and can be merged. The same is true for the events representing the motion of n inhabitants with $n \in \{2, \dots, N\}$ (e.g. $\sigma_{abc} = \sigma_{bac} = \sigma_{cab}$)

Each set of events can be obtained systematically using the following procedure. Let us denote as $Part(Inh)$ the set of partitions of the set Inh . For $Inh = \{a, b, c\}$, $Part(Inh) = \{\emptyset, a, b, c, ab, ac, bc, abc\}$. To construct the set of events related to the inhabitant a , only the partitions including a are considered i.e. only the partitions not being composed of only other inhabitants are considered. The set of partitions of only other inhabitants is equal to $Part(Inh \setminus \{a\}) = Part(\{b, c\}) = \{\emptyset, b, c, bc\}$. Thus, the set of only the partitions including a is equal to $Part(Inh) \setminus Part(Inh \setminus \{a\}) = \{\emptyset, a, b, c, ab, ac, bc, abc\} \setminus \{\emptyset, b, c, bc\} = \{a, ab, ac, abc\}$. The alphabet of events Σ_a can therefore be generated by duplicating the events of Σ as follows: $\Sigma_a = \bigcup_{\sigma \in \Sigma} \left(\bigcup_{l \in Part_a} \{\sigma_l\} \right)$ with $Part_a = Part(Inh) \setminus Part(Inh \setminus \{a\})$.

This can be generalized to N inhabitants with $Inh = \{Inh_1, Inh_2, \dots, Inh_N\}$. For each automaton DMA_{Inh_i} , the set of events $\Sigma_{Inh_i} = \bigcup_{\sigma \in \Sigma} \left(\bigcup_{l \in Part_{Inh_i}} \{\sigma_l\} \right)$ with $Part_{Inh_i} = Part(Inh) \setminus Part(Inh \setminus \{Inh_i\})$.

The number of events of each automaton $|\Sigma_{Inh_i}|$ is equal to the product of the cardinal of $Part_{Inh_i}$ and the number of sensor events $|\Sigma|$ with:

$$\begin{aligned} Card(Part_{Inh_i}) &= Card(Part(Inh) \setminus Part(Inh \setminus \{Inh_i\})) \\ &= Card(Part(Inh)) - Card(Part(Inh \setminus \{Inh_i\})) \\ &= 2^N - 2^{N-1} \\ &= 2^{N-1} \end{aligned}$$

Thus $|\Sigma_{Inh_i}| = |\Sigma| \times 2^{N-1}$

Finally, to obtain DMA_{Inh_i} , each transition defined in DMA labeled by a sensor event σ is duplicated 2^{N-1} times with the same source state and destination state but labeled with one of the 2^{N-1} events determined as described above.

For the case study with 3 inhabitants $Inh = \{a, b, c\}$, the automaton DMA_a representing the motion of the inhabitant a in the same house as the two other inhabitants is given in Fig. 4.

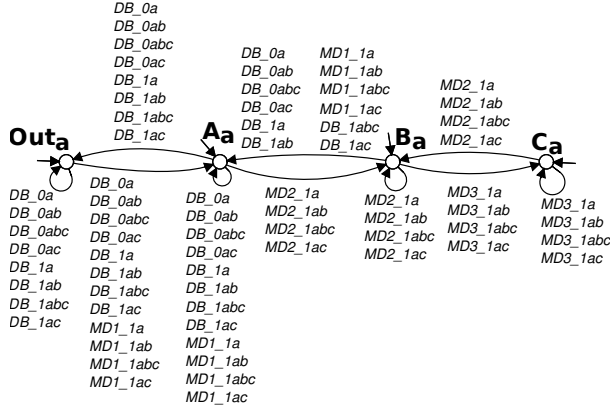


Fig. 4. Detectable Motion Automaton of inhabitant a (DMA_a) among three inhabitants $\{a, b, c\}$

In this model, the events have been duplicated in order to represent the motion of one inhabitant alone or with 1, 2, ..., $N-1$ persons. However, online location tracking is based on a sequence of observed events. This set of sensor events is Σ , previously defined for single inhabitant location tracking.

In practice, events $DB.1_{ab}$, $DB.1_c$ or $DB.1_{abc}$ for instance will be observed in a non-distinguishable manner by the sensor DB through the event $DB.1$.

However, this "artificial" duplication of the labeled transitions allows obtaining the finite automaton representing the detectable motion of the N inhabitants together in the instrumented house by performing the standard synchronous composition as described in the following subsection.

B. Finite Automata Composition

The definition of the standard synchronous composition of 2 finite automata is recalled below [6] and can be generalized to N automata.

Let G_1 and G_2 be two automata such that $G_1 = (Q_1, \Sigma_1, \delta_1, Q_{0,1})$ and $G_2 = (Q_2, \Sigma_2, \delta_2, Q_{0,2})$, the automaton $G_{sc} = (Q_{sc}, \Sigma_{sc}, \delta_{sc}, Q_{0,sc})$ is computed by synchronous composition of G_1 and G_2 such that:

$G_{sc} = G_1 || G_2 = (Q_{sc}, \Sigma_{sc}, \delta_{sc}, Q_{0,sc})$ with:

- $Q_{sc} \subseteq Q_1 \times Q_2$;
- $\Sigma_{sc} = \Sigma_1 \cup \Sigma_2$;
- $\delta_{sc}((q_1, q_2), \sigma) = \begin{cases} (\delta_1(q_1, \sigma), \delta_2(q_2, \sigma)) & \text{if } \delta_1(q_1, \sigma)! \text{ and } \delta_2(q_2, \sigma)! & (1) \\ (\delta_1(q_1, \sigma), q_2) & \text{if } \delta_1(q_1, \sigma)! \text{ and } \sigma \notin \Sigma_2 & (2) \\ (q_1, \delta_2(q_2, \sigma)) & \text{if } \delta_2(q_2, \sigma)! \text{ and } \sigma \notin \Sigma_1 \\ \text{undefined} & \text{otherwise} \end{cases}$

- $Q_{0,sc} = Q_{sc} \cap (Q_{0,1} \times Q_{0,2})$

In our approach, the model of the detectable motion of multiple inhabitants is called $MIDMA_N$. Its structure is given by the standard synchronous composition in the following way: $MIDMA_N = (Q_N, \Sigma_N, \delta_N, Q_{0,N}) = DMA_{Inh_1} || DMA_{Inh_2} || \dots || DMA_{Inh_N}$.

The set of events Σ_N is equal to the union of the sets Σ_{Inh_i} . Some events are common to two or more automata DMA_{Inh_i} and some are specific to one DMA_{Inh_i} . The composition is performed in a synchronous manner (case (1)) on common events and in an asynchronous manner (case (2)) on non-common events. Considering the case study with 3 inhabitants and focusing on the event MD_{1-1} :

- The three events MD_{1-1_a} , MD_{1-1_b} , MD_{1-1_c} are specific to one DMA_{Inh_i} , consequently, the case (2) of the definition of the transition function of the composition is applied. The transition is defined in an asynchronous manner on these non-common events.
- The three events $MD_{1-1_{ab}}$, $MD_{1-1_{ac}}$ and $MD_{1-1_{abc}}$ are each shared by 2 automata and the event $MD_{1-1_{abc}}$ belongs to all the DMA_{Inh_i} , thus the case (1) of the definition of the transition function of the composition is applied. The transition is defined in a synchronous manner on these common events.

After performing the composition, the resulting automaton describes all the possible motion of all inhabitants, alone or with all or part of the $N-1$ other inhabitants.

Σ_N the set of events of $MIDMA_N$ has now to be redefined as being equal to the set of observable sensor events Σ . For each transition of the composition labeled with an event σ_N of Σ_N , the transition is redefined with a label being the sensor event of Σ corresponding to σ_N . For instance, a transition labeled with the event $DB.1_{ab}$ will be transformed into the same transition labeled with the event $DB.1$. In the same manner, another transition labeled with the event $DB.1_c$ will be transformed into the same transition labeled with the event $DB.1$. Once this operation is performed, $MIDMA_N$ is usable for multiple inhabitants location tracking.

Proposition 1: $MIDMA_N$ has exactly Z^N states, with Z the number of zones and N the number of inhabitants.

Proof: Each automaton DMA_i has the same number of states Z (Z being the number of zones of the house). Moreover, using the approach for systematic generation of

DMA , each state of each automaton DMA_{Inh_i} is accessible with a transition labeled with one of the events σ_{Inh_i} which is particular to the automaton i.e. $\forall(i, j) \in \{1, 2, \dots, N\}^2$ such that $i \neq j$, $\sigma_{Inh_i} \in \Sigma_{Inh_i}$ and $\sigma_{Inh_i} \notin \Sigma_{Inh_j}$. Thus, the set of states of $DMA_{Inh_i} \parallel DMA_{Inh_j}$ is exactly the set $Q_{Inh_i} \times Q_{Inh_j}$ whose cardinal is Z^2 . By extension, the set of states of the composition $MIDMA_N$ of all the DMA_{Inh_i} is equal to $Q_{Inh_1} \times \dots \times Q_{Inh_N}$ and has a cardinal Z^N . ■

Note that the number of states of this synchronous composition is the same as the number of states of the asynchronous composition of the N automata.

The automaton $MIDMA_2$ represents the detectable motion of $N = 2$ inhabitants with $Inh = \{a, b\}$. This automaton is given in Fig. 5 for the case study. $MIDMA_2$ is composed of every black and gray colored states and transitions. Several points can be highlighted on this model.

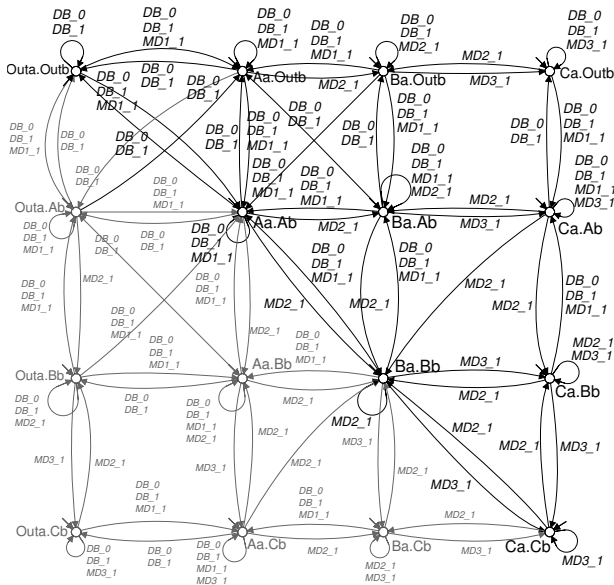


Fig. 5. Detectable Motion Automaton for 2 inhabitants $MIDMA_2$ (black and gray) and reduced DMA $MIDMA_2^{red}$ (black only)

A strong semantics is associated to the states of $MIDMA_N$. Each state represents the location of the N inhabitants within the zones of the house. For instance the state $(A_a.Out_b)$ means that the inhabitant a is in the zone A and the inhabitant b is in the zone Out (meaning he is outside the house). The transitions and events represent the observable motion of the inhabitants i.e. one or several inhabitants changing of location or moving in the same location in case of a self-loop.

A sensor can detect the motion of one or several inhabitants. This can be seen in the model because $MIDMA_N$ is a non-deterministic Finite Automaton (e.g. three transitions starting from $(A_a.A_b)$, labeled with the same event $MD_{2,1}$, one reaching state $(A_a.B_b)$, one reaching state $(B_a.A_b)$ and one reaching $(B_a.B_b)$).

It is assumed that the initial location of the inhabitant is unknown. This can be seen in the model where each state is initial. However, just as explained previously for

single inhabitant, knowing accurately the initial location is not necessary to perform online location tracking because the current estimation of the location of each inhabitant does not hardly depends on their initial location. If for some smart home applications it is mandatory to know the initial location of each inhabitant, some techniques (for instance in [5]) can be used to determine the initial state of an automaton after observing a more or less long sequence of events.

C. Model reduction

Since the different inhabitants are not distinguishable by the sensors of the house, $MIDMA_N$ can be reduced using the symmetry of this model. Using a reduced model will allow to perform more efficiently the online location tracking by having a smaller model without loss of information.

The symmetry of the model can be illustrated on the case study. If the inhabitant a is in A and the inhabitant b is in B , it is exactly the same as the inhabitant a is in B and the inhabitant b is in A because the inhabitants are not distinguishable by the sensors. Consequently, the state $(A_a.B_b)$ and $(B_a.A_b)$ have exactly the same meaning. One of them is thus redundant and can be removed from the model. Formally for N inhabitants, the state $(q_{Inh_1}.q_{Inh_2} \dots q_{Inh_N})$ is equivalent to the state $(q_{Inh_2}.q_{Inh_1} \dots q_{Inh_N})$ and to all the other permutations of $\{q_{Inh_1}, q_{Inh_2}, \dots, q_{Inh_N}\}$ (e.g. $(A_a.B_b) = (B_a.A_b)$ for the case study). Finally, without loss of information, the redundant states and their related transitions are deleted from the model and the obtained automaton is called $MIDMA_N^{red}$. For the case study and for two inhabitants, $MIDMA_2^{red}$ is composed only of black colored states and transitions in Fig. 5.

The number of states of $MIDMA_N^{red}$ can be determined using the notion of multiset (see [7] for more details about multisets and other enumerative problems).

Proposition 2: $MIDMA_N^{red}$ has exactly $\binom{Z}{N}$ states,

where $\binom{Z}{N}$ is the number of multisets of cardinality N , with elements taken from a finite set of cardinality Z .

Proof: Each state of $MIDMA_N^{red}$ represents a location of the N inhabitants in the Z zones i.e. each state represents one repartition of the N inhabitants among the Z zones. The number of these repartitions is equal to the number of multisets of cardinality N , with elements taken with repetition from a finite set of cardinality Z . This number is denoted $\binom{Z}{N}$ and is equal to the binomial coefficient
$$\binom{Z+N-1}{N} = \frac{(Z+N-1)!}{N!(Z-1)!}.$$
 ■

The number of states of $MIDMA_N^{red}$ is smaller than the number of states of $MIDMA_N$ which is equal to Z^N as proved in Proposition 1. For instance, for $Z = 8$ zones and $N = 4$ inhabitants $MIDMA_N$ has 4096 states and $MIDMA_N^{red}$ has only 330 states.

Despite its apparent complexity, the proposed modeling approach remains scalable since we consider only instrumented apartments or houses and not a whole smart building (like for instance in [8]). Consequently, both the number of zones Z and the number of inhabitants N remain small.

$MIDMA_N^{red}$ represents at each time the number of inhabitants in each of the Z zones which is exactly the same information as the one represented by the automaton $MIDMA_N$ if the inhabitants are not distinguishable. Both models can be used for location tracking but using the reduced one leads to a decreased complexity of online location tracking.

V. MULTIPLE INHABITANTS LOCATION TRACKING

A. Algorithm for online location estimation

Online multiple inhabitants location tracking consists in providing at each time the estimation of the state of the model and thus of the location of each of the N inhabitants based on the observed sequence of events. As for single inhabitant, this can be done either using a previously offline computed state estimator or directly estimating online the state of the model $MIDMA_N^{red}$. We propose Algorithm 1 to compute online the estimation of the state of $MIDMA_N^{red}$ and thus the location of the N inhabitants.

Algorithm 1 Online location estimation algorithm

Require: $MIDMA_N^{red} = (Q_N^{red}, \Sigma, \delta_N^{red}, Q_{0N}^{red})$

- 1: Initialization: Current location $L =$ set of states Q_{0N}^{red}
- 2: **while** location tracking is active **do**
- 3: Wait for a new event e
- 4: New event e is observed
- 5: **if** $\exists q \in L$ such that $\delta_N^{red}(q, e)!$ **then**
- 6: $L' = \bigcup_{q \in L} \delta_N^{red}(q, e)$
- 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 inhabitants, is equal to the set of the initial states. Then the algorithm waits for the occurrence of the new event e emitted by a sensor. It is assumed that two events cannot simultaneously occur. When an event e happens, the algorithm computes, with respect to the transition function, the set of states L' , successors of each state of L , such that $L' = \bigcup_{q \in L} \delta_N^{red}(q, e)$. When the new set of states is computed, the set of state L is updated by the new set of states L' . Then, the program waits for another new event to compute again the new locations of the inhabitants.

The complexity of this algorithm is $O\left(\binom{Z}{N} \times 2S\right)$ each time a new event is observed. However, since the number of zones Z , the number of inhabitants N and the number of sensors S remain small in practice, this algorithm is efficient and can be performed online.

B. Illustration on a real scenario

We illustrate the location tracking algorithm on a real scenario for the case study (Fig. 1) with $N = 2$ inhabitants and using the reduced model $MIDMA_N^{red}$ (Fig. 5). The

scenario is the following: the two inhabitants are initially in the bedroom. Then, one is entering the bathroom while the other is going to the living room.

- 1) The location tracking algorithm is initialized. Current location is $L = \{(Out.Out), (Out.A), (Out.B), (Out.C), (A.A), (A.B), (A.C), (B.B), (B.C), (C.C)\}$ which is the set of initial states of $MIDMA_2^{red}$. It is an inaccurate location.
- 2) A rising edge of the motion detector in the bedroom ($MD_2.1$) is observed. The current location is updated and is now $L = \{(Out.B), (A.B), (B.B), (B.C)\}$. The location of the two inhabitants is inaccurate. One is in the zone B and the other is in any zone.
- 3) A rising edge of the motion detector in the bathroom ($MD_3.1$) is observed. The current location is updated; $L = \{(Out.C), (A.C), (B.C), (C.C)\}$. The location is inaccurate. One is in C and the other is in any zone.
- 4) A rising edge of the motion detector in the living room ($MD_1.1$) is observed. The current location is updated and is now $L = \{(A.C)\}$. The location is now accurate. One is in A and the other in C .

We applied this model-based algorithm on longer scenarios. It is efficient and shows good performances to locate multiple inhabitants in an instrumented home.

VI. CONCLUSION

In this paper, we proposed an approach for model-based location tracking of multiple inhabitants in an instrumented home. We developed a procedure to obtain a Finite Automaton model based on the synchronous composition and a method to reduce its size without loss of information under certain hypotheses. Finally we presented an algorithm for model-based online location tracking of multiple inhabitants.

Future work will be devoted to extend this approach to distinguishable inhabitants using wearable sensors and to consider the possible occurrence of sensor faults.

REFERENCES

- [1] 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 *Proceedings of the IEEE Conference on Systems, Man and Cybernetics*, 2006.
- [2] Y. Rahal, H. Pigot, and P. Mabileau, "Location Estimation in a Smart Home: System Implementation and Evaluation Using Experimental Data," *International Journal of Telemedicine and Applications*, 2008.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Failure diagnosis using discrete-event models," *IEEE Trans. on control systems technology*, vol. 4 (2), pp. 105–124, 1996.
- [7] R. P. Stanley, *Enumerative Combinatorics*, Cambridge University Press ed., 2012, vol. 1, second edition.
- [8] J. P. Boyer, K. Tan, and C. A. Gunter, "Privacy Sensitive Location Information Systems in Smart Buildings," in *Security in Pervasive Computing, Lecture Notes in Computer Science*, vol. 3934, 2006, pp. 149–164.