Activity Discovery and Detection of Behavioural Deviations of an Inhabitant from Binary Sensors

Jérémie Saives, Clément Pianon, Gregory Faraut

Activity Discovery and Detection of Behavioural Deviations of an Inhabitant from Binary Sensors

Jérémie Saives, Clément Pianon, and Gregory Faraut, *Member, IEEE*[*]

***Abstract*—The aim of this paper is to improve the autonomy of medically monitored patients in a smart home instrumented only with binary sensors; overwatching the disease evolution, that can be characterized by behaviour changes, is helped by detecting the activities the inhabitant performs. Two contributions are presented. On one hand, using sequence mining methods in the flow of sensor events, the most frequent patterns mirroring activities of the inhabitant are discovered; these activities are then modeled by an extended finite automaton, which can then be used for activity recognition and generate activity events. On the other hand, given the set of activities that can be recognized, another automaton is built to model requirements from the medical staff supervising the inhabitant; it accepts activity events, and residuals are defined to detect any behaviour deviation. The whole method is applied to the dataset of Domus, an instrumented smart home.**

***Note to Practitioners*–This paper was motivated by the will of providing a model of the behaviour of the inhabitant at home, from which formal methods can be applied to verify some properties, namely detect behaviour deviations. The input of the framework is a log file of binary sensor events (rising and falling edges); a pre-filtering of the log might be required to get rid of repetitive, noisy events (namely spurious events generated by motion sensors, for which only the first rising edge and last falling edges are relevant). This recorded log is assumed to be recorded without sensor faults, and to be characteristic of a typical routine of the inhabitant. In order to build this model, based on the Extended Finite Automata (EFA) formalism, sequence mining techniques (first contribution of the article) are adapted: the more frequent the sequence is, the more important for the inhabitant the habit is. The second contribution of the paper is the ability to detect behaviour deviations, which is criticial for healthcare at home and disease management. The technique of "residuals", mainly used in manufacturing, has been adapted to EFAs. It requires medical requirements, defined by the medical staff, based on the activities it wants to keep an eye on. A suiting log of sensors events and medical requirements are the only prerequisite to the application of the method.**

***Index Terms*—Primary Topics: Home Automation, Discrete Event Systems, Automata, Activities of Daily Living.**

I. Introduction

SMARTNESS in daily living has become a major topic in research for a few years; large-scale applications deal with smart cities or grids, whereas small-scale applications include smart homes and buildings. The objective of smart homes is to improve the life of the inhabitant with human-centered applications. These applications can be divided into three categories [1]: Emergency assistance, Autonomy Enhancement and Comfort. Autonomy Enhancement is of importance in most industrial countries; indeed, life expectancy has continuously increased over the last decades. Moreover, according to Eurostat [2] and World Health Organization [3], the percentage of people aged 60 or more will reach 30% in many countries in 2050. These considerations about ageing lead to new issues regarding the autonomy and the independence of elderly or disabled people. Furthermore, when these people need medical assistance, the resources in hospital are limited. Houses are therefore considered as a continuum of the hospital to healthcare, and are a solution to improve the management of elderly people diseases.

In order to 'smarten' a home, numerous varieties of sensors are available. The choice of technology mainly depends on a compromise between the level of information and privacy. For instance, some studies based on cameras [4] can recognize activities, but cameras are intrusive and often refused by the inhabitant. Some other devices like wearable sensors [5] require the inhabitant to be equipped and move, whereas others have very low level information like binary sensors [6], [7], [8], [9], [10], but are the less intrusive. The choice of technology directly impacts the methods used to discover or recognize activities, and binary sensors are chosen in this work. Activity Recognition (AR) is mainly a classification task [11]. Most of studies need an expert knowledge at the beginning to label the input data. To reduce the need for expert knowledge, we use Activity Discovery (AD) techniques [12]. However, activity discovery techniques, like activity recognition techniques, still depend on the technology of the sensors. Regarding deviations, medical staff often uses questionnaires [13] or informal techniques, requiring the participation of the monitored person. Some works using formal techniques do not detect deviations of behaviour but are focused on the detection of inactivity [14]. Our long term objective is to help the formalization of the deviation detection, namely using informations issued of AD, AR, and its automation to improve disease management.

The approach proposed in this paper is divided in two contributions. The first one, based on previous work [15], is oriented towards providing a formal model of habits and activities discovered in a dataset of sequences, without *a priori* knowledge, hence in an unsupervised way [16]. It provides an automated method of building a map of the habits and activities of a monitored inhabitant. Such a model can be devoted to online real-time recognition of activities. The

[*]J.Saives, C.Pianon and G.Faraut are with LURPA, ENS Cachan, Univ. Paris-Sud, Université Paris-Saclay, 61 avenue du Président Wilson, F-94235 Cachan France, e-mail: firstname.lastname@ens-cachan.fr

second part proposes a method to build a model representing recommendations from the medical staff, so that the recognized activities from the first part can be compared to the recommendations, and behaviour deviations of the inhabitant can be detected. This model, based on the recommendations only, is independent of the behaviour of the inhabitant, and it handles all activities detected in the first part of the approach.

A more general discussion about related work is stated in section II. The problem statement, and assumptions made, are given in section III. Section IV shortly presents the proposed approaches. Section V and section VI respectively detail the two contributions: the first one focuses on the identified model of habits allowing then to recognize activities of the inhabitant, and the second one focuses on the detection of behaviour deviations. Finally, a case study based on the DOMUS dataset is proposed in section VII. Conclusions and future works are detailed in the last section.

II. Related Work

The goal of our approach is to detect behaviour deviations of an inhabitant in a smart home. The first problematic concerns the kind of sensors that equip the smart home.

A. Sensors technology

Amongst the equipments not dealt with in this work are cameras, wearable devices or RFID sensors. For instance, in [17], cameras are used to detect whether the monitored inhabitant has fallen. In [18], wearable sensors are used to determine the posture and movement of the equipped inhabitant using accelerometers. In [19], RFID sensors are used in order to detect which objects are being used by the inhabitant wearing an RFID reader. However, all those approaches are either intrusive or require the inhabitant to wear a special device. In the latter case, the inhabitant must be willingly participating; should he suffer from Alzheimer disease for instance, or simply forget to equip the devices, and the approaches would be inefficient.

Therefore, a solution would be to use a Wireless Sensor Network, using a collection of various sensors with a binary output, such as the universal switch sensors defined in [20]. The network provides a flow of sensor events, each event containing very little information, but activities can be recognized from a sequence issued of the flow ([6][21]).

Regardless of sensors technology, the main problematic is to detect behaviour deviations. Before that, it is required to be able to recognize the current behaviour (Activity Recognition) and have the knowledge of a 'normal' behaviour (Activity Discovery).

B. Activity Recognition

Recognizing current behaviour is the area of Activity Recognition (AR) which is considered as a classification task [22], [11]. Supervised classification approaches share the same strategy: a learning phase on a set that has been studied by an expert, then the recognition phase, during which new sets of data are classified and parameters. The most widespread classification models would be the Hidden Markov Model (HMM) or Support Vector Machines (SVM) ([23], [24], [25], [26], [27]). Different models have also been designed in the literature. For instance, expert-designed boolean functions on the statuses of the sensors are used in [28], with an emphasis on the adaptation part. Expert models of activities are also designed in [29], each activity being described by an ordered set of action sequences, and the adaptation being achieved by learning automata.

However, these techniques need, at the beginning, expert knowledge to define a list of activities or relations between activities and sensors data. This task is not an easy one and it is not always possible to have access to this information. To obtain these relations, an activity discovery step can be conducted.

C. Activity Discovery

Activity Discovery aims at getting the knowledge of the normal behaviour of an inhabitant, represented by patterns. Some unsupervised approaches, coming from activity recognition, discover these patterns to build the activity recognition model without expert knowledge. These approaches have mainly been developed for wearable devices ([30], [31], [32]) because determining relations becomes easy with high-level information. In the case of binary sensors technology, sequence mining approaches are more adapted to discover patterns ([33], [11]). For instance, data mining techniques are exploited in [34], [35] to find frequent patterns in a sequence of events and compare them to expert models. Nevertheless, all those works require expert knowledge; we propose to discover habits and leave the task of defining activities to the very end.

Once the normal behaviour has been observed and defined, we can proceed to the detection of deviations.

D. Behaviour deviation detection

Most of the works previously cited are efficient in activity recognition, but provide no indication on behaviour deviations. Indeed, in disease management, the medical staff does not focus on which activities are performed but rather on deviations of some particular activities, symptomatic of improvements or deteriorations of the health of the inhabitant. Behaviour deviations have been studied in some works like [36],[14] or [37], where the authors use the localization of the inhabitant to estimate a potential risk if he stays too long in a room. The authors of [38], [39], [40] can detect behaviour deviations but use mainly multi-context information. Finally, very few studies based on only binary sensors focus on the detection of deviations of complex habits or activities [41], but they still need expert knowledge at the beginning.

Much of the works do not focus on binary sensors, despite their advantages of cheap cost and low instrusiveness. The global goal of this paper is to achieve behaviour deviations from binary sensors, with no initial expert knowledge on the activities. To get such a result, it proposes firstly a new Activity Discovery method based on data mining techniques, leading to the construction of original finite-state models of the

discovered habits. With such a model of habits, online Activity Recognition could be performed. Then, a residual method proven in the field of automation, but seldom used outside of it, has been applied to such models to detect deviations. The whole method of this paper, from the binary sensors to the behaviour detection requires only two expert interventions, which are not a prerequisite at the beginning: once the map of discovered habits has been built, to cluster it into activities (which is eased by the readability of the model), and to design the medical requirements from which the deviations are defined. Two perspectives of this work are to limit or even remove the need for these interventions.

III. Assumptions and problem statement

In this paper, the following assumptions are made. In order to help the users to accept the observation of their every movement, to guarantee the respect of their privacy and to reduce the costs, the instrumentation consists 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 sensors for instance). Furthermore, we assume that the sensors are fault-free.

It is also assumed that there is always at most one inhabitant in the home. Moreover, its behaviour is considered totally free. Indeed, even in the learning phase, the inhabitant is not compelled to repeat a specific action to enable the learning of his behaviour. Using a Discrete Event System (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 signals emitted by each binary sensor of the house. Furthermore, we assume events do not occur simultaneously. Even if two events are generated at the same time, they are sequentially recorded.

We also consider that an habit is always done in the same way, independently of time, i.e. the sequence of event is the same. The approach is expected to be even more efficient when time information is included. Due to the technology of the sensors chosen, there might be spurious information. For instance, while solicited, an infrared motion sensor keeps sending rising and falling edges. The data provided has to be filtered, and therefore an assumption is made that a same event can not occur twice in an habit.

To improve the interpretation and the understandability of the behaviour of the inhabitant, we consider that a graphic representation of the habits is helpful, such as an automaton. This is particularly important to help the expert to determine relations between habits and activities.

For the deviation detection phase, it is assumed that the medical staff has provided a list of textual requirements that should be satisfied for the monitored inhabitant to be considered in good health. For instance, have three meals a day. Depending on the patient, missing one meal can be considered critical (local deviation), or instead only the repetition of missed meals on a larger timescale becomes worrisome (global deviation). The result of the method should take into account both types of deviations, depending on what the requirements of the medical staff are.

Based on these considerations, the problem of activity discovery, recognition and behaviour deviations detection can be reformulated in terms of a DES problem: from observed sequences of sensor events and requirements from a medical staff, discover habit models that can recognize activities from the spontaneous event generator that is the inhabitant, with a minimal learning phase, and that can be used to detect behaviour deviations.

IV. Global overview of the proposed approach

The global structure of our approach is illustrated on Fig. 1. A smart home equipped with a binary sensor network generates a sequence of *events* representing the inhabitant doing an activity. The behaviour of a human being is however arbitrary, hence multiple sequences of events can be images of the same activity, hardening the difficulty of building an expert model. The first contribution focuses on activity discovery, and aims at building a map representing a model of all habits of inhabitant. From this map, an expert will identify corresponding activities by clustering and defining boundaries for each activity. This contribution is detailed in section V.

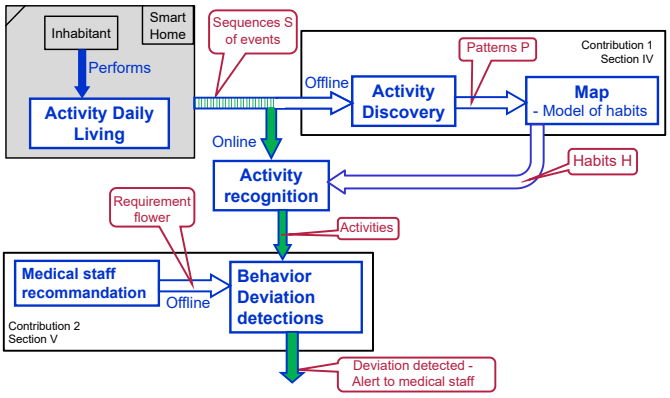


Fig. 1. Overview of the proposed approach

The second part of our contribution, based on the sequence generated by the inhabitant and by the activities previously recognized, focuses on the inhabitant's behaviour deviation detection. A model, called "requirement flower", is built, representing the deviations that the medical staff wants to focus on. Residuals techniques are used on it to detect deviations. When a deviation is detected, it provides precise information on the deviation. This contribution is detailed in section VI.

V. Map habits generation and recognition

In this first part, a possibility is to use a training set (a few sequences that have already been observed), within which the sequences are still very different. Numerous observations would be required to depict all the possible sequences that depict the activity. Nevertheless, the sequences share subsequences, which are the fundamental basis of the activities, because they are very often played. Those frequent sequences would thus represent fundamental habits of the inhabitant.

The first part revolves around the discovery of these habits, which can be achieved by data mining methods (**Mining Step** of Figure 2). The dataset can stand in two forms. On one hand, it might be a unique sequence of events, within which frequent episodes can be found [42],[43]. A few days worth of observation could lead to a single sequence of events. On the other hand, one could use distinct sequences of events, as long as they represent the same temporal window, and compare each other in order to extract the habits. This last solution has been chosen. For the remainder of this work, it is also supposed that a single inhabitant is being monitored. Once the patterns have been found, habits can be identified within the set of patterns (**Identification Step** of Figure 2).

Then, the second part consists in the automated modeling of the discovered habits, which then leads to the building of a *map* of the habits of the inhabitant (**Automated Building Step** of Figure 2). This map allows for online recognition: when an event *e* occurs, the active states of the map change, leading to a set of habits that might be currently ongoing. The accuracy of the recognition remains out of the scope of this work.

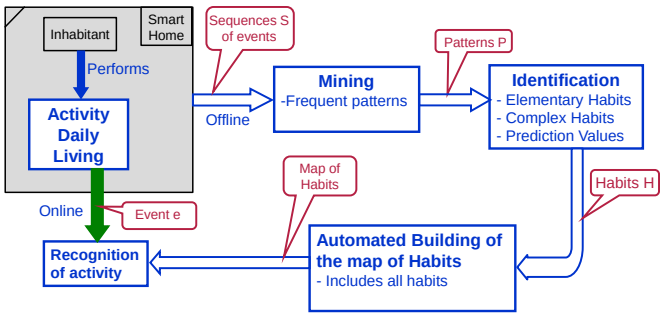


Fig. 2. Building a model from sequences of events issued of a Smart-Home

It is worth noting that until the map is built, no expert knowledge has been injected in the model. The goal is to recognize frequent habits (that have already been observed and not expertly designed), instead of trying to split and classify every sequence observed into activities. Nevertheless, once the map is obtained, an expert can study it and determine which parts and which habits correspond to which activities. That part could also be helped by coupling the habit model with a location tracking model, thus getting information on both the location and the activity of the monitored inhabitant.

A. Discovering patterns

Sequence mining is a specific field of data mining that deals with the search for relevant patterns in sequences and strings. Most of the methods used in sequence mining look for frequent itemsets in databases, in order to discover association rules between items frequently found together. Such techniques would be the *Apriori* algorithm found in [33], or the use of FPTrees found in [44]. Since the ordering of the items is irrelevant, those methods consider only databases that have been previously lexicographically ordered. However, in the case of a sequence of events, since we want to discover succession of events which would be images of habits, the order is of great importance and an adaptation is required. Based on the Apriori algorithm, the proposed algorithm is thus designed to find continuous ordered patterns.

1) Definitions:

Definition 1. Sequence and events

Let $D = \{S_i\}_{i=1..n}$ be a database containing n sequences. A *sequence* S_i is an ordered list of events such as S_i=$[e^i_1, e^i_2, e^i_3 \ldots, e^i_{l(S_i)}]$, where $l(S_i) \geq 2$ is the *length* of the sequence, and e^i_k is the k-th event of the sequence.

If Σ is the set of all the events that can be generated by the sensor network, then $\forall i \in [\![1, n]\!], \forall k \in [\![1, l(S_i)]\!], e^i_k \in \Sigma$.

Definition 2. Pattern

A *pattern* P is a sequence of events $P = [e^P_1, e^P_2, \ldots, e^P_{l(P)}]$, where $l(P) \geq 2$ is the length of the pattern. An event is not enough to define a pattern. Hence, a pattern is said *of elementary length* when $l(P) = 2$, which is the minimum length.

A pattern P is contained in a sequence S_i if, and only if $\exists s \in [\![1, l(S_i) - l(P) + 1]\!]$, $[e^i_s, e^i_{s+1}, \ldots, e^i_{s+l(P)-1}] = [e^P_1, e^P_2, \ldots, e^P_{l(P)}]$, *i.e.* P is a continuous subsequence of S_i , written $P \subset S_i$. A pattern can also be contained in another one.

Definition 3. Support

The *support* $Supp(P)$ of a pattern P is the number of sequences $S_i \in D$ in which P is contained, *i.e.* it mirrors the frequency of the pattern in the database. Thus, $\forall P, Supp(P) \in [\![0, |D|]\!]$, with $Supp(P) = 0$ meaning that the pattern is not present in the database, and $Supp(P) = |D|$ that the pattern is present in each sequence.

Let $Supp_{min} \in [\![0, |D|]\!]$ be a *minimum support*. Then, a pattern P is said to *satisfy* the minimum support iff $Supp(P) \geq Supp_{min}$. The minimum support defines the minimum presence for a pattern to be considered relevant.

2) Algorithm of sequence mining:

The global goal of the algorithm of sequence mining is to discover every pattern contained in the database, and evaluate their support. Given a minimum support, it returns every pattern whose support is equal or higher. Running it multiple times for various values of minimum support is required in order to obtain a complete overview of the dataset.

Algorithm 1 consists of three steps, the last two being repeated until no new pattern is discovered:

Init An initialization step, to discover the patterns of elementary length, described in algorithm 2
- All patterns of elementary length that satisfy the minimum support are stocked in a list of elementary patterns

CandGen A candidate generation step, described in algorithm 3. The objective of this step is to make the patterns grow in length, in order to find the patterns of maximum length.

- If two already discovered patterns partially recover themselves, then a bigger pattern including both patterns is a potential candidate for being a pattern satisfying the minimum support itself.

Count A counting step, to evaluate the support of the candidates (function *EvalSupp* in the algorithms).

- Only the patterns satisfying the minimum support are kept and added to the list of discovered patterns.
- The shorter patterns that helped generate the newly discovered pattern are deleted, because no longer relevant.

Algorithm 1 Continuous Pattern Discovery

Require: Database D of n sequences, minimum support Supp_{min}, set of events Σ
  List_{disc}=Init(D, Supp_{min}, Σ)
  List_{cand}=CandGen(List_{disc})
  while $\text{List}_{cand} \neq \emptyset$ **do**
    for pattern_{cand} in List_{cand} **do**
      EvalSupp(pattern_{cand})
      if Supp(pattern_{cand})$\geq$ Supp_{min} **then**
        Add pattern_{cand} to List_{disc}
        for pattern_{disc} in List_{disc} **do**
          if $\text{pattern}_{disc} \subset \text{pattern}_{cand}$ **then**
            Delete pattern_{disc}
          end if
        end for
      end if
    end for
    List_{cand}=CandGen(List_{disc})
  end while
  return List_{disc}, List of discovered patterns satisfying minimum support

Algorithm 2 *Init*-Initialization function

Require: Database D of n sequences, minimum support Supp_{min}, set of events Σ
  List_{elem}=[]
  for Event1 in Σ **do**
    for Event2 in Σ **do**
      pattern_{test} = [Event1,Event2]
      EvalSupp(pattern_{test})
      if Supp(pattern_{test})$\geq$ Supp_{min} **then**
        Add pattern_{test} to List_{elem}
      end if
    end for
  end for
  return List_{elem}, List of patterns of elementary length

Remark 1. Let $|\Sigma|$ be the size of the set of events, $|D|$ the size of the database, and $|S|_{max}$ the length of the longest sequence in the database. *Evalsupp* handles the problem of finding subsequences in the whole database, thus is of complexity $O(|D||S|^2_{max})$.
There are at most $|List_{disc}| = |\Sigma|(|\Sigma|-1)/2$ patterns of length 2, so Algorithm 2 is of complexity $O(|\Sigma|^2|D||S|^2_{max})$. Then, Algorithm 3 being of complexity $O(|List_{disc}|^2|S|_{max})$, and $List_{cand}$ being necessary empty after $|S|_{max}$ candidate generations, Algorithm 1 is of complexity $O(|\Sigma|^4|D||S|^3_{max})$. Since it is of polynomial complexity, it can be used to handle large databases, the main limiting parameter being the maximum length of the sequences $|S|_{max}$. To improve the understanding, an example is given in Appendix A.

Algorithm 3 *CandGen*-Candidate Generation Function

Require: List of previously discovered patterns List_{disc}
  List_{cand}=[]
  for Pattern_1=$[e^1_1, e^1_2, \ldots, e^1_n]$ in List_{disc} **do**
    for Pattern_2=$[e^2_1, e^2_2 \ldots, e^2_m]$ in List_{disc} **do**
      if $[e^1_2, \ldots, e^1_n]$=$[e^2_1, \ldots, e^2_{m-1}]$ **then**
        Add $[e^1_1, e^1_2{=}e^2_1, \ldots, e^1_n{=}e^2_{m-1}, e^2_m]$ to List_{cand}
      end if
    end for
  end for
  return List_{cand}, list of possible patterns

Now that all patterns and their supports have been found, each pattern could be treated individually, and independent models for recognition could be built. However, they are strongly dependent, since the shortest patterns are often included in longer pattern of lowest support. In order to build model of activities, or study the links between the habits, those inclusions have to be considered.

3) Structuring discovered patterns and building habits: Generally speaking, the discovered patterns follow a tendency: the longer the pattern, the lower the support. Hence, short patterns with high support depict a fundamental *habit* of the observed inhabitant. Those habits are often contained in longer patterns of lower support. Hence, these latter patterns are an image of *activities*, built out of habits. Multiple fundamental habits form an activity, and there are multiple long patterns that can represent the same activity.

As a pattern can be contained in a sequence, it can also be contained in another discovered pattern. Should a pattern not contain any other discovered pattern, it will be called *elementary*. Patterns of elementary length cannot contain another one, and are thus natively elementary. A pattern can only be contained in a pattern of lower support. A pattern containing at least another one will be called *complex*. A complex pattern might contain more than one other pattern, and thus the decomposition into smaller patterns is not unique. Nevertheless, Algorithm 4 can be used in order to find one of those possible decompositions for each complex pattern.

Algorithm 4 handles every pattern one after another, following a decreasing order of support. The first patterns handled are those of highest support. For the following patterns, the algorithm checks whether one of the already structured patterns, hence of a higher support, is contained within. When no more already structured patterns can be found in the handled pattern, it is added in its structured form to the list of structured

Algorithm 4 Building the hierarchy of patterns

Require: List of patterns and their supports $\text{List}_{patterns}$
  List_{str}=[ ]
  for support=Support_{max} **to** 2 (decreasing) **do**
    for pattern $\in$ {$\text{List}_{patterns}$ } such that
    Supp(pattern)=support **do**
      while $\exists$ $\text{pattern}_{str} \in \text{List}_{str}$ such that
      $\text{pattern}_{str} \subset$ pattern **do**
        Substitute pattern_{str} in pattern
      end while
      Add pattern to List_{str}
    end for
  end for
  return List_{str}, List of structured patterns

patterns. An example is presented in Appendix B.

Remark 2. Let n be the number of discovered patterns, and L the length of the longest discovered pattern. Algorithm 4 is of complexity $O(n^2L^2)$, hence polynomial as well as Algorithm 1.

B. Automated building of a model

The objective of this section is to propose a model that will be able to identify when a pattern has been reproduced in an online real-time flow of sensors events. Hence, when a new event is generated, the model must react accordingly, by identifying which patterns could currently be at stake, which ones might have just begun, and which ones have just ended. For that purpose, each pattern could, in its elementary form, be modeled by a Finite-State Automaton.

Furthermore, the structure discovered between the patterns needs therefore to be exploited, leading to the choice of an extended class of automata in order to keep that structure. Moreover, the model should also provide events representing the recognized activity to be able to detect deviation for second part of the proposition. This leads to the choice of an extended class of automata in order to keep that structure, which is presented in the following. The formalism of the Extended Finite-state Automata (EFA), as defined in [45] is recalled here:

Definition 4. A (Deterministic) Extended Finite-state Automaton is a 7-tuple: $((Q \times V), \Sigma, G, A, \delta, (q_0 \times v_0), (Q_m \times V_m))$, with:

- Q x V is an extended set of states, where Q is a finite set of locations and V the finite domain of definition of the variables
- Σ the set of sensors events
- G a set of guard predicates over 2^V
- A a set of actions over V
- $\delta : Q \times \Sigma \times G \times A \to Q$ a transition function
- (q_0, v_0) the initial state
- $(Q_m, V_m) \subset (Q \times V)$ a set of marked states

Σ is the alphabet that contains every event that can be generated by the binary sensor network. Let $q \in Q$ be the current state, $q' \in Q$ be another state such that $\exists (e, g, a) \in \Sigma \times G \times A, \delta(q, e, g, a) = q'$. Then, if event e occurs, the transition will be fired if and only if the guard g is satisfied. The current state of the automaton will then be q', and the action a will be executed, changing the values of the variables.

1) Modeling elementary patterns:
Let P=[$e_1, e_2, , \ldots, e_n$] be an elementary pattern of length n. In order for the automaton to identify in real-time if that pattern has been played, it should satisfy a few properties:

- As long as the pattern has not started (e_1 has not occurred yet), the automaton should stay in its initial state
- If the pattern has been completed, the current state of the automaton should be marked
- If the pattern is interrupted by an unexpected event, the automaton should react accordingly: if e_1 occurs, the pattern might have started again before having been completed ; if another unexpected event occurred, the pattern has stopped, and the automaton should be in its initial state again.

Figure 3 proposes such an automaton for $n = 4$. Let us say that the current state is $P_1.2$, *i.e.* the last events that occurred have been [1,2]. The transition function states: $\delta(P_1.2, 3, ., .) = P_1.3$, $\delta(P_1.2, 1, ., .) = P_1.1$ and, $\forall e \in \Sigma - \{1, 3\}, \delta(P_1.2, e, ., .) = P_1.0$. If 3 occurs, the current state of the automaton becomes $P_1.3$ and the pattern keeps being identified. If 1 occurs, the pattern starts again, and if any other event happens, the pattern halts, and the automaton reaches its initial state.

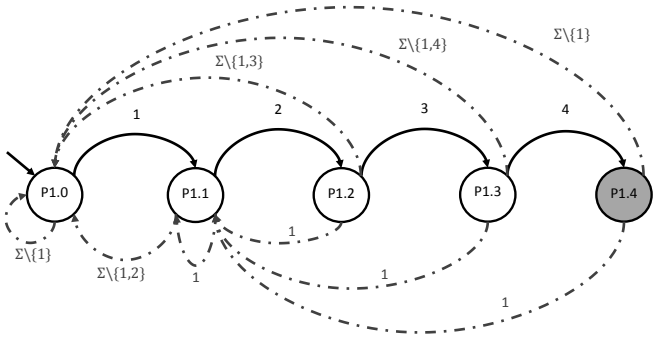


Fig. 3. EFA modelling the elementary pattern [1,2,3,4], with Σ the set of all events that can be generated by the sensor network

Elementary patterns can be modeled by simple Finite State Automata, because no variables are required. However, when complex patterns are concerned, the need for variables arises.

2) Modeling complex patterns:
The construction process of an automaton in order to model a complex pattern is also more complex. The patterns must be distinguishable, and there must be no false detection. Three cases have to be considered:

Case 1. The contained pattern shares the first event with the complex pattern. It is impossible to distinguish the two patterns at the beginning, hence they share the first states and

transitions. The complex pattern becomes a prolongation of the contained one. No variables are required in this specific case. See Figure 4 for an example with P_1=[1,2] being an elementary pattern contained in P_2=[P_1,3,4].

This complex model is no longer deterministic, because more than one transition can be activated on the occurrence of one single event. It is however easily transformable into a deterministic automaton by classical methods. For the sake of visibility, the automata will nevertheless be shown in their non-deterministic form. One must consider that a set of states are current at any time instead of a single one.

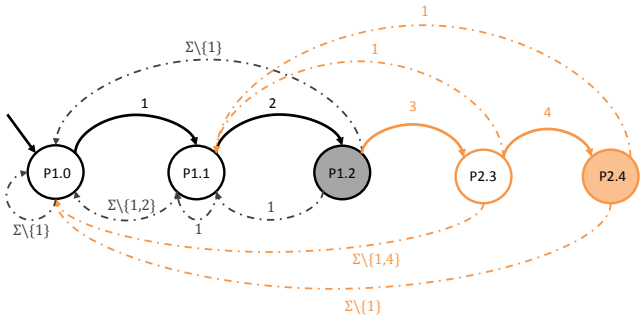


Fig. 4. EFA modelling the complex pattern P_2=[P_1,3,4], with Σ the set of all events that can be generated by the sensor network.

Case 2. The contained pattern shares the last event with the complex pattern. In order to distinguish the patterns, the last state, which is a marked state, is duplicated. However, should only the contained pattern be played, the containing pattern must not be recognized, thus requiring the creation of a variable and a guard on this specific transition. The variable should be proper to the complex pattern, and indicate whether it has actually started and is currently being recognized. It is set to 1 when the patterns starts, and reset to 0 when the pattern is interrupted or ended.

See Figure 5 for an example with P_1=[3,4] being an elementary pattern contained in P_2=[1,2,P_1].

1) Suppose that the sequence [1,2,3] has already been played. The set of current states of the automaton is $\{P_1.1\}$, and H2=1. Since $\delta(P_1.1, 4, ., .) = P_1.2$ and $\delta(P_1.1, 4, H2 == 1, .) = P_2.4$, if 4 occurs, the set of current states becomes $\{P_2.4, P_1.2\}$ and both patterns are recognized, which is the expected outcome.
2) Suppose now that the sequence [2,3] has been played. The set of current states of the automaton is $\{P_1.1, P_2.0\}$, and H2=0. If 4 occurs, the new set of current states becomes $\{P_2.0, P_1.2\}$, and only P_1 is recognized, which is the expected outcome.

Case 3. The complex pattern contains two patterns. The previous cases apply their rules if the first and/or the last event is shared. Supplementary transitions have to be created between the two patterns. If exactly one event is expected between these patterns, a transition labeled with this event should be created from the last state of the automaton modeling the first pattern

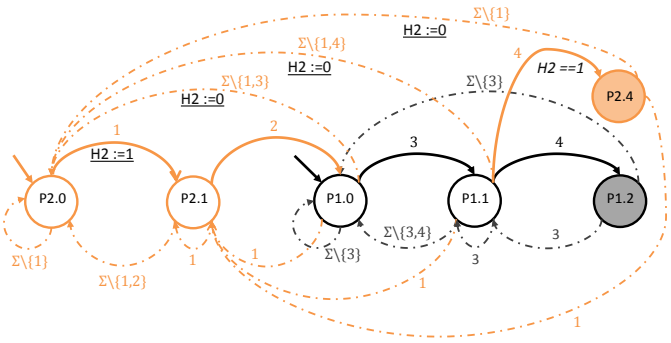


Fig. 5. EFA modelling the complex pattern P_2=[1,2,P_1], with Σ the set of all events that can be generated by the sensor network.

to the first state of the automaton modeling the second pattern. If more than one events are expected, additional states have to be added. If no events are expected, a transition labeled with the first event of the second pattern should be created from the last state of the first automaton to the second state of the second automaton.

See Figure 6 for an example of that latter case, with P_1=[1,2], P_2=[3,4] and P_3=[P_1,P_2]. A transition is created such that $\delta(P_1.2, 3, ., .) = P_2.1$.

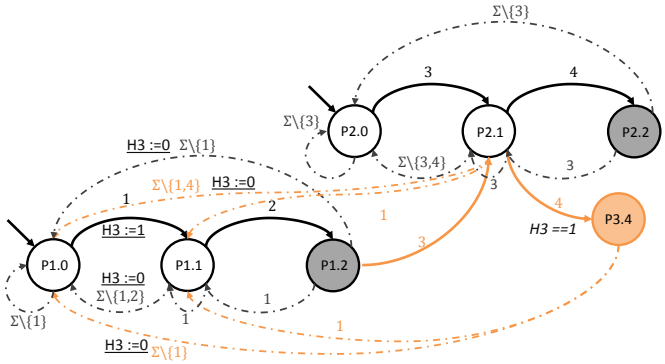


Fig. 6. EFA modelling the complex pattern P_3=[P_1,P_2], with Σ the set of all events that can be generated by the sensor network.

3) Building a map of habits:
A complex pattern, whatever its form, contains shorter patterns, and can be modeled by a complex EFA containing the EFAs of the shorter patterns. Since the operations presented in this section can be repeated, it is possible to build one single automaton representing all the links between the patterns discovered by the data mining step. Such a map can be used by the expert to characterize activities performed by the inhabitant *a posteriori*, instead of *a priori*.

4) Activity Recognition:
When using the map of habits while observing the events generated by the inhabitant at home, we are able to detect activities. Indeed, every time a marked state is reached, an event representing the activity is generated. Thus, a set of activities Σ_{act} is defined by the expert; the events of this

alphabet are generated by the recognition module of this approach. These activities events are used to detect deviations, as presented in the next section.

VI. Detection of behaviour deviations

This part, based on the work presented in previous section, aims at detecting behaviour deviations, according to requirements defined by the medical staff. The framework is illustrated on Fig. 7.

The data used are the map of habits and the list of admissible or undesired behaviours defined by the medical staff. Furthermore, the recognition function of the previous section generates events according to the detected activities that the inhabitant is currently performing.

A deviation of behaviour could be the consequence of a particular pathology. For instance, an inhabitant suffering from Alzheimer may forget his lunch, or take multiple lunches instead of only one. Then, the expertise of the medical staff is required to determine which is an admissible behaviour, and which is not. From a list of activities and the map of habits, it should provide requirements for a behaviour to be considered admissible. With these rules, a single EFA is proposed, called requirement flower, that represents all these rules. The resulting model does not depend on the inhabitant and is applicable as long as the recommendations do not change, it can therefore be applied to multiple smart homes. It is conceived to be as permissive as possible, while providing a way of detecting deviations.

Indeed, in the last step of the framework, the observation of current activity performed by the inhabitant leads to the generation of events of Σ_{act}. If events accepted by the requirement flower violate a given condition, information on the activity that does not respect the recommendations is provided. The approach is based on the residual method used in manufacturing system to diagnose this kind of fault [46].

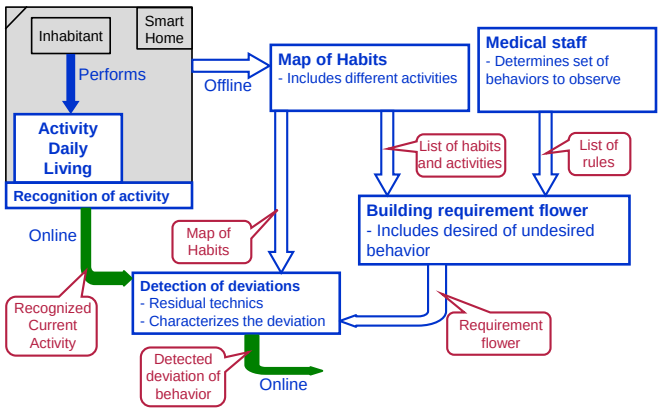


Fig. 7. Framework of behaviour deviations detection

A. Generation of the requirement flower

The model must be as permissive as possible, represent the desired or undesired behaviours and not be dependent on the inhabitant, but on medical recommendations only. These recommendations are often expressed textually, but can be translated into mathematical expressions, therefore the requirements can be modeled by conditions on variables. Furthermore, these requirements should not lead the model to a blocking state, but only allow the detection of deviations. The model must therefore accept any activity any number of times, even if the undesired behaviour is a maximal number of instances for a given activity.

The Extended Finite Automata (EFA) (see Definition 4) are kept to express these requirements with guards and variables. However, two considerations are made to be as permissive as possible:

- the EFA has a single state [47], and there is one transition by event, leading to a flower model.
- the guard is not considered in the transition function δ, *i.e* $\delta : Q \times \Sigma \times A \to Q$ to ensure that any event is accepted by the model in any condition.

Furthermore, a time window is always associated to an observation of the behaviour of an inhabitants. For instance, it could be decided to check for deviations each day, or each week, or even more. Even if time is not taken into account in this paper - though it remains an important perspective - an event representing the end of the observation window is nevertheless required. So, in each requirement model, a specific event is created, called End_obs. When this event is observed, the observation period is over, the respect of requirements is verified, and the variables are initialized.

In the following, the steps that lead to the requirement flower are detailed. This procedure is generic and only needs the set of activities and requirements desired by the medical team.

Procedure 1. From Σ_{act} representing the list of activities that may be performed by the inhabitant:

1) Generate $\Sigma^S_{act} \subseteq \Sigma_{act}$, the set of activities considered by the requirement
2) Generate the set of variables V. Each variable is a counter for each activity (*i.e.* there exists a bijection $\Phi : V \to \Sigma_{act}$)
3) Generate the set of actions A (over V) which increments the counters. Given an activity $a \in \Sigma^S_{act}$, if a is performed, then $A(\Phi^{-1}(a)) = \Phi^{-1}(a) + 1$
4) Generate the requirements G with medical expertise, for every activity $a \in \Sigma^S_{act}$.
5) Build the requirement flower $G_c = ((\{q\} \times V), \Sigma^S_{act}, G, A, \delta, (\{q\} \times v_0), (\{q\} \times V))$, with:
   - q: the only state of this automaton. It is both initial and marked.
   - $\delta : \{q\} \times \Sigma^S_{act} \times A \to \{q\}$ the transition function. Given any $a \in \Sigma^S_{act}$, $\delta(q, a, A) = q$ and $A(\Phi^{-1}(a)) = \Phi^{-1}(a) + 1$
   - v_0 the initial values of the variables, all set to 0.
6) Add the transition $\delta(q, End_{obs}, A_{end}) = q$ with:
   - End_{obs} the event representing the end of the observation window.

- A_{end} an action that resets all variables to 0, in order to start a new observation.

A short example is given to illustrate the generation of the requirements. For simplification, the transitions will be called by their associated activities from now on:

Example 1.
Let $\Sigma_{act} = \{Eat, Wake\_up, Shower, Toilet, Pills, Sport, Smoke\}$ be a set of activities to be monitored, according to the following requirements:

- At least one meal a day: a guard $eat > 0$ is associated to End_{obs}
- Use the toilet at least once a day but less than 6 times: a guard $toilet > 0$ is associated to End_{obs}; another guard $toilet \leq 5$ is associated to $Toilet$
- Take three pills (one in the morning, one during lunch and one in the evening): a guard $pills > 2$ is associated to End_{obs}; another guard $pills \leq 3$ is associated to $Pills$
- Don't smoke: a guard $smoke! = 0$ is associated to $Smoke$
- Shower at least once a day: a guard $shower > 0$ is associated to End_{obs})

With these informations, Procedure 1 builds the requirement flower illustrated in Fig.8

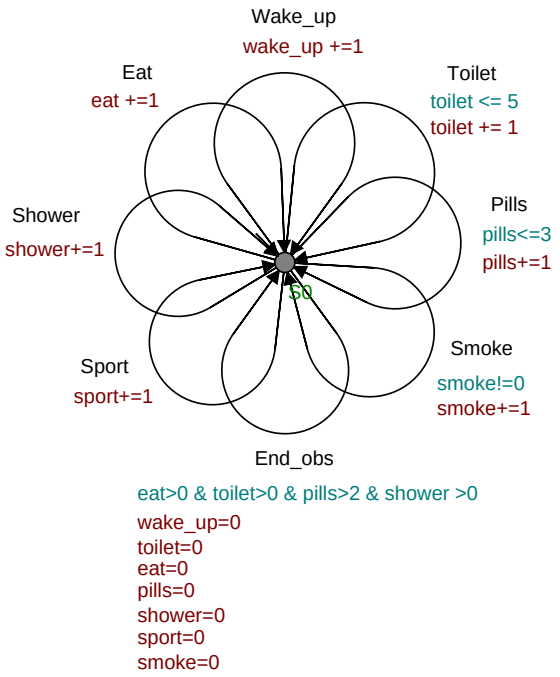


Fig. 8. Requirement flower for Example 3

B. Detection of deviations

Based on disease evolutions, two kinds of behaviour deviations are to be supervised. On one hand, sudden deviations represent a change of behaviour in a short span of time and might be critical. On the other hand, smooth evolutions of the behaviour in a long period of time could represent a new pathology or an evolution of disease. Two strategies are proposed to detect them.

1) Local deviations:
Local deviations are deviations detectable in only one sequence of events (one observation period). The method used to detect deviations is inspired by the residual technique [46]. A residual is a fault indicator based on the gap between a fault-free model and the reality. In this context, it represents the difference between the desired behaviour of the inhabitant (the requirement flower is the fault-free model expertly designed) and its actual behaviour observed (the reality). Then, two kinds of residuals are defined:

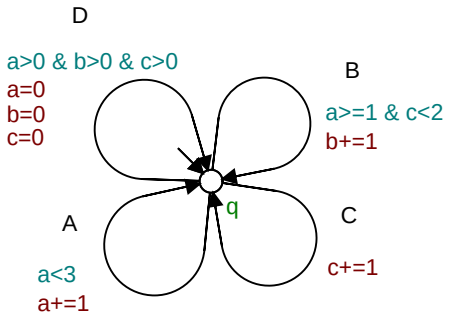
- *Res1: a behaviour is observed but was not expected*
- *Res2: an expected behaviour is not observed*

In this work, all events of Σ_{act}^{S} are always expected, because there exists a transition for each event, hence the residuals defined as above cannot be used. However, given the requirement flower, it is assumed that the guards are satisfied when a transition is fired. Therefore, an unexpected behaviour would be the firing of a transition without satisfaction of the guard. A new residual is therefore designed on guards as being the requirement(s) unsatisfied when firing the transition labeled by the observed event:

Definition 5. Let e_{obs} be the observed event. The residual on guards is defined by:
$Res = \{g_v \in G \mid \delta(q, e_{obs}, a_v) = q \wedge g_v \text{ is false}\}$
where a_v and g_v are the actions and guards associated to the transition labelled by e_{obs}

The following example is given to illustrate the purpose of residuals:

Example 2. Fig.9 represents the extended automaton with a single state; Let $\Sigma = \{A, B, C, D\}$ be the set of activities, where D is the event representing the end of observation.



evt	A	B	A	B	D
v	(0000)	(1000)	(1100)	(2100)	(2200)
g_v	$a < 3$	$a \geq 1$ $c < 2$	$a < 3$	$a \geq 1$ $c < 2$	$a > 0$ $b > 0$ $c > 0$
Res	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\{c > 0\}$
v'	(1000)	(1100)	(2100)	(2200)	(0000)

Fig. 9. Requirement flower for Example 4 and study of the residuals for the sequence 'ABABD'

Let 'ABABD' be the observed sequence. Let $V = \{a, b, c, d\}$ be the set of variables, and v be the vector of the values of the variable. In the initial state, $v = (0000)$. The table of Fig.9 shows the evolution of Res after each event of the sequence observed. In this table, g_v represents the requirement(s) in relation to the occurred event and v' the vector of variables after the occurrence of the event.

In this example, no residual is computed during the activities. However, at the end of the observation, when D occurs, the guard is not satisfied, hence a local deviation is found. Moreover, the residuals provide a set of activities which are potentially deviant. In this example, the deviation is located in activity C: it should have been performed at least once during the observation window.

This example illustrates that a local deviation can easily be detected with the residual method; it is detected as soon as possible and provides a list of activities that can be causes of the deviation.

2) Global deviations:

Global deviations are deviations that cannot be detected with one observed sequence only. Some pathologies like Alzheimer imply indeed a smooth deviation of behaviour of the inhabitant (forgetting more and more often to shower for instance). If a requirement is not fulfilled just once, it might be irrelevant; missing a meal once is bad behaviour but not necessary dangerous. However, missing four meals in a week becomes an issue; the repetition of the unfulfillment surely is relevant. In this case, the frequency of unsatisfied guards on multiple observation windows are studied.

For that purpose, the previous set of guards G is split into two subsets: G_l represents the requirements for local deviations while G_g represents requirements for global deviations. The latter, extended requirements are defined as following:

Definition 6. A requirement for a global deviation is defined as a couple (g, τ) such that:

- $g \in G_g$ represents predicates on variables V,
- τ represents a requirement on the frequency f of local deviations.

Given n observations, the frequency f of a local deviation is defined by:

$$f = \frac{1}{n}\sum_{i=1}^{n} x_i, \text{ where } \begin{cases} x_i = 1 \text{ if } g \in Res_i \\ x_i = 0 \text{ otherwise} \end{cases}$$

Based on the previous definition, we can define the residual in order to detect global deviations:

Definition 7. Let n be the number of observed sequences, C_g be a set of requirements for global deviations. Then, the residual in order to detect global deviations is defined by:
$Res_{global} = \{(g, \tau) \in C_g | \tau \text{ is false}\}$

Example 3. Let $n = 10$ be the number of observed sequences, and let the requirement flower of Example 4 be extended with two global requirements. In order to satisfy for instance for instance $(g_1, \tau_1) = (a < 1, f_1 > \frac{8}{10})$, the residual $\{a < 1\}$ must not appear more than 7 times in the 10 observed sequences. This is illustrated by Fig.10

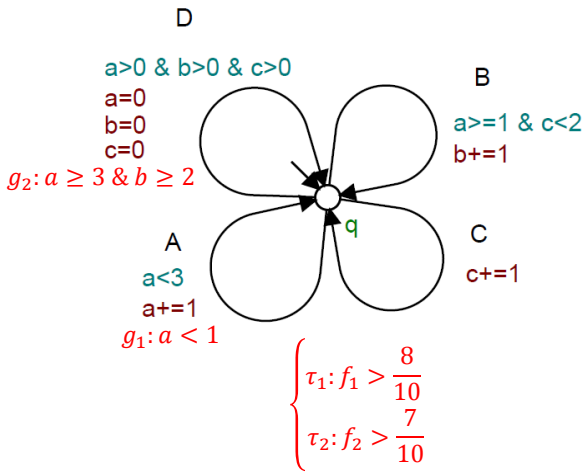


Fig. 10. Requirement flower extended with global requirements

VII. Example of Application

The framework presented in this paper has been applied to an actual example. The results are the following

- A model representing the habits of the inhabitant (the map)
- The ability to detect when an activity occurs
- A model of the requirements (the requirement flower)
- The ability to detect any local or global deviation of the inhabitant

A. Domus database

The Domus Smarthome ([48], [49]) is a living lab of the University of Sherbrooke (domus.usherbrooke.ca). This apartment, illustrated on Fig.11, is equipped by 36 binary sensors (IR, Pressure detector, Lamps, door contacts, switch contacts, flow meter). Different users have been asked to perform the morning routine in the apartment (ie from waking up to leaving the apartment), while registering the evolution of the sensors values. The Trace generated by sensors is under this form: *Sensor ID / Sensor Name / Sensor Location / Value*

Although the algorithms presented in this paper have been tested on all users, only the results obtained for the first one are presented in this paper, for the sake of understandability. Different maps are constructed for each user, according to their own habits, whereas the requirement flower remains the same. Each user performed the routine ten times, thus ten sequences of events can be compared in order to find frequent patterns. The events generated by the infra-red sensors have not been taken into account, because of their intrusiveness (a lot of irrelevant events generated), and they are mainly concerned by the localization of the inhabitant.

B. Data mining result

Using only *SensorID/Value* of the trace generated by the Domus Smarthome, the application of the method of section V.B led to the discovery of 36 patterns, summarized in Table I

It can be noted that more and more complex patterns are discovered when the support gets lower. A few elementary

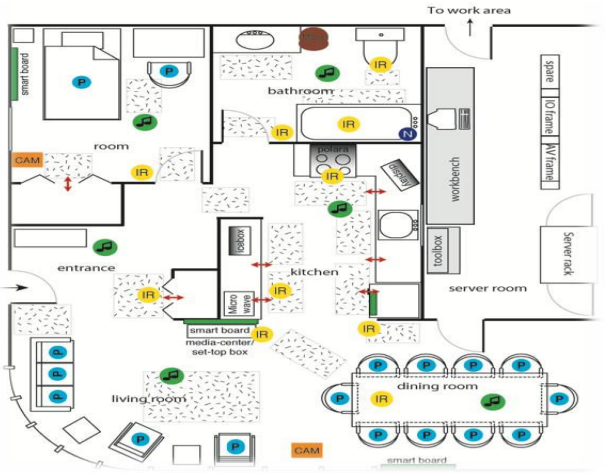


Fig. 11. Domus smart home plan

TABLE I
PATTERNS DISCOVERED FOR USER 1

Support	Elementary	Complex	Min Length	Max Length
10	5	-	2	2
9	1	3	2	6
8	1	1	3	11
7	-	2	3	6
6	-	1	7	7
5	1	3	2	12
4	1	2	2	4
3	2	2	2	9
2	3	8	2	15

patterns can still be found, but might be of little relevance in the behaviour of the inhabitant (for instance, an elementary pattern of length 2 and support 2 depicts a succession of two events that happened only twice, and can merely be considered as an habit).

C. Mapping the habits of a person

The application of the method presented in section III.C led to the construction of the automaton presented in Figure 12. For a visibility purpose, only the main transitions, *i.e.* not the interruptions, have been represented. The automata representing low support elementary patterns not contained in complex patterns have as well not been represented.

1) Identifying activities: In order to identify activities, it is possible to expertly analyse the map, and delimit areas. Five activities are expected in the Domus dataset: Waking up, Use Toilet, Preparing Breakfast, Having Breakfast, Washing Dishes. The first two can be associated to areas of the map. The third and the fifth are associated to the kitchen activities area. The fourth would need the information on the location to be distinguished, thus requiring a location tracking model to be coupled.

2) Real-time identification: When the real-time observation of the inhabitant begins, the set of active states is the set containing all the initial states. When an event occurs, the set of active sets is updated according to the transition function. When one or more marked states become active, one or more habits have been recognized, and an event of Σ_{act} is generated accordingly.

3) Adaptation of the map: After the observation is over, the observed sequence of events can be used to recompute the map. The patterns that have been recognized will see their support strengthened, confirming which are the fundamental habits of the inhabitant.

D. Detecting deviations

From the set of activities discovery in previous step, we have $\Sigma_{act} = \{$Waking-up, Use Toilet, Preparing Breakfast, Having Breakfast, Washing-Dishes$\}$.

1) Requirements from medical team: The observation window is set to one day (the morning routine is done only once a day), and the following requirements are considered:

- g_1: have exactly one breakfast (divided into two subguards: g_{1a} represents at least one breakfast, and g_{1b} represents no more than two.)
- g_2: do not use the toilet more than 4 times.
- g_3: wash dishes each day.

The following global requirements are also considered for each week (n=7):

- (g_2, τ_4): do not use the toilet more than 4 times a day, 5 days out of 7
- (g_{1a}, τ_5): do not forget to eat breakfast more than twice a week
- (g_3, τ_6): do not forget to clean the dishes more than once in 4 days

These requirements may seem simple, but actually represent the kind of requirements the medical staff may have difficulty to supervise. Indeed, elderly people may wake-up many times in the night to use the toilet, go back to sleep and do not remember. Autonomous people suffering from Alzheimer may forget whether they took a meal or not, and if not, how frequently ?

Based on these requirements, $\Sigma^S_{act} = \{$Use Toilet, Having Breakfast, Washing Dishes$\}$, to which the event End_{Obs} is added.

2) Building the requirement flower: Procedure 1 is applied, and the requirement model is shown on Fig.13.

3) Deviations detection with residuals: With the previously built model, behaviour detection is possible. In order to test the ability to detect some deviations, some sequences of the Domus database have been modified or added, each sequence representing one day:

- a cycle "Waking-up, Use Toilet" is reproduced at the beginning of some sequences in order to simulate an inhabitant that wakes up in the middle of the night.
- in some sequences, the events representing breakfast are removed;
- in other sequences, the events representing shower are removed;

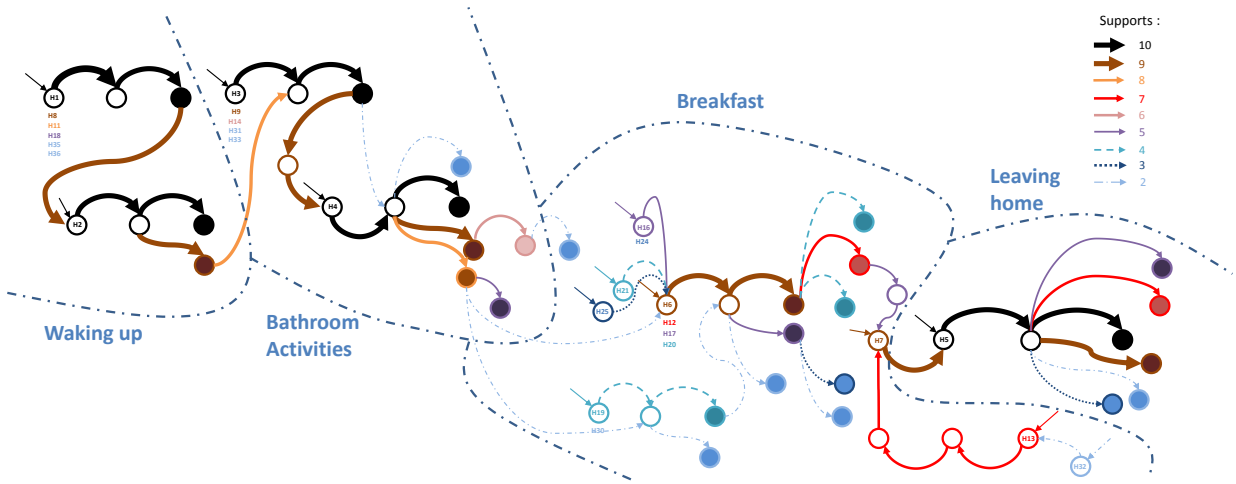


Fig. 12. Map of the habits of User 1 of the Domus Database. Selfloops and labels are not represented to improve the lisibility; the emphasis is put on the support to show sequences of habits that are very representative of activities.

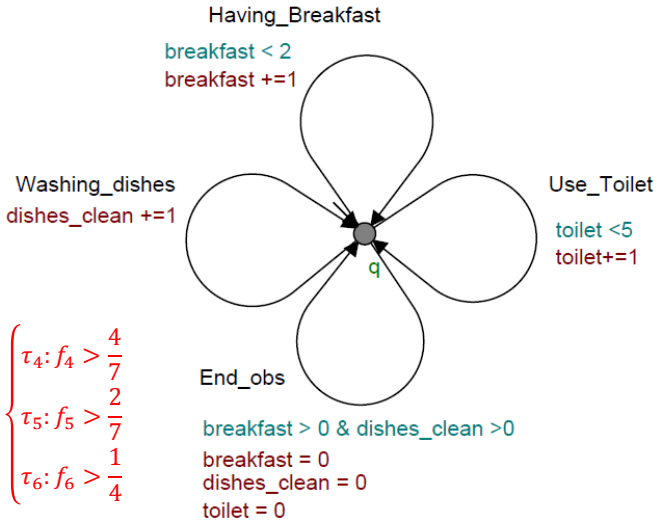


Fig. 13. Requirement flower for the Domus dataset

- in a few sequences, the events representing dish washing are removed;

Table II shows the results of deviations detection for each day, and its interpretation. Until Day 9, no global deviations are detected, whereas some local deviations occur. The medical staff can consider that these local deviations are part of the irregularity of the human behaviour (eating a bit too much, using the toilet, forgetting the dishes), or that they are more critical (forgetting to eat) and require an intervention. Then, in the last days, multiple alerts from global deviations are generated: the toilet has really been used too much compared to what the medical staff expected, the dishes are forgotten too often,... There could be a medical trouble (urinary deficiency, Ahlzeimer's disease, a fall ?), and the staff can contact the inhabitant to check for trouble, and monitor him into taking the breakfast he forgot. Local deviations become important if we consider their repetition in a larger timescale, and both kind of alerts can be relevant for the medical staff.

Conclusion and Future Work

In this paper, we propose a model-based framework, composed of two approaches, based only on binary sensors (like motion or pressure sensors), to detect behaviour deviations of a monitored inhabitant. Firstly, from a log of binary sensor events (rising and falling edges), the habits of the inhabitant are extracted by sequence mining techniques and modelled by Extended Finite Automata (EFA). Such a model is then used for online recognition of the activities of daily living. Then, we propose an approach, based on residuals techniques, to detect two kinds of deviations of the behaviour of the inhabitant. Local deviations are detected on a short timescale whereas global deviations consist in repetitions of local deviations, and are detected on a longer timescale. They represent unfulfilled medical requirements, and the medical staff monitoring the inhabitant can react accordingly.

Future work focuses on the next major improvements. The first one is the inclusion of time in the patterns, in order to detect inherent temporal deviations of the execution of the habits (slower or faster than usual). The second improvement concerns the assumption that a sequence of habits does not hold repetitions of the same event in the sequence. This assumption was holding for the morning routine presented in this paper, but should be removed to develop a more robust method, applicable to larger timescales. A third improvement would be the development of clustering techniques to automatically build activities from the map of habits. Finally, formal expressions could be used for a better use of the medical textual requirements.

Appendix A
Example of Algorithm of Sequence Mining

Let D be the following dataset, over the set of events $\Sigma = \{1,2,3,4,5,6,7,8\}$:

- S_1=[1,2,3,4,5,6]

TABLE II
DEVIATIONS DETECTED BY RESIDUALS

Days	Local deviations	Global deviations	Interpretation of local deviation(s)	Interpretation of global deviation(s)
Day 1	$\emptyset$	$\emptyset$	-	-
Day 2	$\emptyset$	$\emptyset$	-	-
Day 3	$\{g_2\}$	$\emptyset$	Toilet used more than 4 times	-
Day 4	$\{g_{1b}, g_3\}$	$\emptyset$	The inhabitant took 2 breakfasts and did not clean dishes	-
Day 5	$\{g_2\}$	$\emptyset$	Toilet used more than 4 times	-
Day 6	$\{g_2\}$	$\emptyset$	Toilet used more than 4 times	-
Day 7	$\emptyset$	$\emptyset$	-	-
Day 8	$\{g_2, g_3\}$	$\emptyset$	Toilet used more than 4 times; dishes not cleaned	-
Day 9	$\{g_{1a}, g_2\}$	$\{(g_2, \tau_4)\}$	The inhabitant did not eat this day; Toilet used more than 4 times	Toilet used more than 4 times, 5 times in the last 7 days
Day 10	$\emptyset$	$\emptyset$	-	-
Day 11	$\{g_{1a}, g_2, g_3\}$	$\{(g_2, \tau_4), (g_3, \tau_6)\}$	The inhabitant neither ate, nor cleaned the dishes; Toilet used more than 4 times	Toilet used more than 4 times, 5 times in the last 7 days; Cleaning dishes forgotten for the second time in the last four days
Day 12	$\emptyset$	$\emptyset$	-	-
Day 13	$\{g_2\}$	$\emptyset$	Toilet used more than 4 times	
Day 14	$\{g_{1a}, g_2, g_3\}$	$\{(g_{1a}, \tau_5), (g_2, \tau_4), (g_3, \tau_6)\}$	Same as Day 11	Same as Day 11, plus the inhabitant did not eat for the second time in the last seven days

- S_2=[1,2,3,4,7,8]
- S_3=[1,2,4,7,8,5]
- S_4=[1,2,3,7,8,4]

Let the minimum support be 2. Then:

Init List$_{elem}$=[1,2 ; 2,3 ; 3,4 ; 4,7 ; 7,8]
CandGen [1,2] and [2,3] overlaps, thus leading to [1,2,3] being a potential candidate, and so on.
List$_{cand}$=[1,2,3 ; 2,3,4 ; 3,4,7 ; 4,7,8]
Count Supp([3,4,7])=1, hence pattern [3,4,7] does not satisfy the minimum support, and so on.
List$_{disc}$=[1,2 ; 2,3 ; 3,4 ; 4,7 ; 7,8 ; 1,2,3 ; 2,3,4 ; 4,7,8]. [1,2] is contained in [1,2,3], can thus be cleaned out, and so on. Hence: List$_{disc}$=[1,2,3 ; 2,3,4 ; 4,7,8]
CandGen List$_{cand}$=[1,2,3,4]
Count List$_{disc}$=[1,2,3,4 ; 4,7,8]
CandGen List$_{cand}$=$\emptyset$

Finally, List$_{disc}$=[1,2,3,4 ; 4,7,8] contains the largest patterns whose minimum support is 2.

By repeating for supports 3 and 4:

- List$_{Support4}$ = [1,2]
- List$_{Support3}$ = [1,2,3 ; 7,8]
- List$_{Support2}$ = [1,2,3,4 ; 4,7,8]

APPENDIX B
EXAMPLE OF BUILDING THE HIERARCHY OF PATTERNS

Let List$_{patterns}$ be the following:

- P_1=[1,2], Support: 10
- P_2=[4,5], Support: 10
- P_3=[1,2,3], Support: 8
- P_4=[6,7,8], Support: 5
- P_5=[1,2,3,4,5], Support: 4

Then, using algorithm 4:

P_1 (natively) Elementary pattern.
P_2 (natively) Elementary pattern.
P_3 P_1 is included in $P_3 \rightarrow P_3$=[P_1,3]
P_3 No more pattern is included in P_3
P_4 Elementary pattern.
P_5 P_1 is included in $P_5 \rightarrow P_5$=[P_1,3,4,5]
P_5 P_2 is included in $P_5 \rightarrow P_5$=[P_1,3,P_2]
P_5 P_3 is included in $P_5 \rightarrow P_5$=[P_3,P_2]
P_5 No more pattern are included in P_5

Finally, and after the conversion of patterns into habits, List$_{str}$ is:

- H_1=[1,2], Support: 10
- H_2=[4,5], Support: 10
- H_3=[H_1,3], Support: 8
- H_4=[6,7,8], Support: 5
- H_5=[H_3,H_2], Support: 4

On one hand, habits H_1 and H_2 can be considered as fundamental, with high support. On the other hand, H_5, which consists of the succession of two habits, can be considered as a way to perform an activity. H_5 also provides information on the observed behaviour: it states that sometimes, H_3 is followed by H_2. Hence, should H_3 be observed, there is a chance for H_2 to start next.

REFERENCES

[1] T. Kleinberger, M. Becker, E. Ras, A. Holzinger, and P. Muller, "Ambient intelligence in assisted living: Enable elderly people to handle future interfaces," in *Universal Access in Human-Computer Interaction*. Springer, 2007, pp. 103–112.

[2] Eurostat, "Eurostat population projections 2010-based," Eurostat, Tech. Rep., 2010.

[3] W. H. Organization, "Good health adds life to years," World Health Organisation, Tech. Rep., 2012. [Online]. Available: http://www.who.int/ageing/publications/whd2012_global_brief/en/

[4] R. Poppe, "A survey on vision-based human action recognition," *Image and vision computing*, vol. 28, no. 6, pp. 976–990, 2010.

[5] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *Communications Surveys & Tutorials, IEEE*, vol. 15, no. 3, pp. 1192–1209, 2013.

[6] F. J. Ordónez, P. de Toledo, and A. Sanchis, "Activity recognition using hybrid generative/discriminative models on home environments using binary sensors," *Sensors*, vol. 13, no. 5, pp. 5460–5477, 2013.

[7] M. Berenguer, M. Giordani, F. Giraud-By, and N. Noury, "Automatic detection of activities of daily living from detecting and classifying electrical events on the residential power line," in *e-health Networking, Applications and Services, 2008. HealthCom 2008. 10th International Conference on*, July 2008, pp. 29–32.

[8] M. Danancher, J.-J. Lesage, L. Litz, and G. Faraut, "Online Location Tracking of a Single Inhabitant based on a State Estimator," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics - SMC 2013*, Manchester, United Kingdom, Oct. 2013, pp. 391–396. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00873247

[9] ——, "A Discrete Event Model for Multiple Inhabitants Location Tracking," in *9th IEEE International Conference on Automation Science and Engineering - CASE 2013*, Madison, WI, United States, Aug. 2013, pp. 922–927, 6 pages. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00855156

[10] M. Danancher, G. Faraut, J.-J. Lesage, and L. Litz, "A DES Simulator for Location Tracking of Inhabitants in Smart Home," in *Proceedings of the 8th EUROSIM Congress on Modelling and Simulation - EUROSIM 2013*, Cardiff, Wales, United Kingdom, Sep. 2013, pp. 330–335, 6 pages. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00862386

[11] D. J. Cook and N. C. Krishnan, *Activity Learning: Discovering, Recognizing, and Predicting Human Behavior from Sensor Data*. John Wiley & Sons, 2015.

[12] D. J. Cook, N. C. Krishnan, and P. Rashidi, "Activity discovery and activity recognition: A new partnership," *Cybernetics, IEEE Transactions on*, vol. 43, no. 3, pp. 820–828, 2013.

[13] T. Roenneberg, A. Wirz-Justice, and M. Merrow, "Life between clocks: daily temporal patterns of human chronotypes," *Journal of biological rhythms*, vol. 18, no. 1, pp. 80–90, 2003.

[14] M. Floeck and L. Litz, "Inactivity patterns and alarm generation in senior citizens' houses," in *Control Conference (ECC), 2009 European*. IEEE, 2009, pp. 3725–3730.

[15] J. Saives and G. Faraut, "Automated Generation of Models of Activities of Daily Living," in *12th International Workshop on Discrete Event Systems-WODES 2014*, Cachan, France, May 2014, pp. 13–20. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00999505

[16] T. Dimitrov, J. Pauli, and E. Naroska, "Unsupervised recognition of ADLs," in *Proc. of the 6th Hellenic conference on Artificial Intelligence: theories, models and applications*, ser. SETN'10, 2010, pp. 71–80.

[17] X. Yu, X. Wang, P. Kittipanya-Ngam, H. Eng, and L. Cheong, "Fall detection and alert for ageing-at-home of elderly," in *Proc. of the 7th International Conference on Smart Homes and Health Telematics (ICOST'09), Tours, France*, 2009, pp. 209–216.

[18] S. Chernbumroong, S. Cang, A. Atkins, and H. Yu, "Elderly activities recognition and classification for applications in assisted living," *Experts systems with Applications 40*, pp. 1662–1674, 2013.

[19] D. Patterson, D. Fox, H. Kautz, and M. Philipose, "Fine-grained activity recognition by aggregating abstract object usage," in *Proceedings of the Ninth IEEE International Symposium on Wearable Computers*, 2005, pp. 44–51.

[20] S. Intille, E. Tapia, J. Rondoni, J. Beaudin, C. Kukla, S. Agarwal, L. Bao, and K. Larson, "Tools for studying behavior and technology in natural settings," in *Proc. of UBICOMP 2003*, 2003, pp. 157–174.

[21] D. Wilson and C. Atkeson, "Simultaneous tracking and activity recognition (star) using many anonymous, binary sensors," in *Pervasive Computing*, ser. Lecture Notes in Computer Science, H.-W. Gellersen, R. Want, and A. Schmidt, Eds. Springer Berlin Heidelberg, 2005, vol. 3468, pp. 62–79. [Online]. Available: http://dx.doi.org/10.1007/11428572_5

[22] E. Kim, S. Helal, and D. Cook, "Human activity recognition and pattern discovery," *Pervasive Computing, IEEE*, vol. 9, no. 1, pp. 48–53, 2010.

[23] B. Cheng, Y. Tsai, G. Liao, and E. Byeon, "HMM machine learning and inference for activities of daily living recognition," *J. Supercomput.*, vol. 54, pp. 29–42, 2010.

[24] N. C. Krishnan and D. J. Cook, "Activity recognition on streaming sensor data," *Pervasive and Mobile Computing*, vol. 10, pp. 138–154, 2014.

[25] J. Aggarwal and L. Xia, "Human activity recognition from 3d data: A review," *Pattern Recognition Letters*, vol. 48, pp. 70–80, 2014.

[26] H. Fang and C. Hu, "Recognizing human activity in smart home using deep learning algorithm," in *Control Conference (CCC), 2014 33rd Chinese*. IEEE, 2014, pp. 4716–4720.

[27] C. Zhu and W. Sheng, "Motion-and location-based online human daily activity recognition," *Pervasive and Mobile Computing*, vol. 7, no. 2, pp. 256–269, 2011.

[28] J. Botia, A. Villa, and J. Palma, "Ambient assisted living system for in-home monitoring of healthy independent elders," *Expert Systems with Applications 39*, pp. 8136–8148, 2012.

[29] M. Ros, M. Cuellar, M. Delgado, and A. Vila, "Online recognition of human activities and adaptation to habit changes by means of learning automata and fuzzy temporal windows," *Information Sciences 220*, pp. 86–101, 2013.

[30] D. Wyatt, M. Philipose, and T. Choudhury, "Unsupervised activity recognition using automatically mined common sense," in *AAAI*, vol. 5, 2005, pp. 21–27.

[31] T. Gu, S. Chen, X. Tao, and J. Lu, "An unsupervised approach to activity recognition and segmentation based on object-use fingerprints," *Data & Knowledge Engineering*, vol. 69, no. 6, pp. 533–544, 2010.

[32] D. Trabelsi, S. Mohammed, F. Chamroukhi, L. Oukhellou, and Y. Amirat, "An unsupervised approach for automatic activity recognition based on hidden markov model regression," *Automation Science and Engineering, IEEE Transactions on*, vol. 10, no. 3, pp. 829–835, 2013.

[33] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proc. of 1994 Int. Conf. Very Large Data Bases (VLDB94), Santiago, Chile*, 1994, pp. 487–499.

[34] E. Tapia, S. Intille, and K. Larson, "Activity recognition in the home using simple and ubiquitous sensors," *IEEE Pervasive Computing*, pp. 158–175, 2004.

[35] B. Chikhaoui, S. Wang, and H. Pigot, "A frequent pattern mining approach for ADLs recognition in smart environments," in *Proc. of 2011 International Conference on Advanced Information Networking and Applications*, 2011, pp. 248–255.

[36] I. Navarrete, J. A. Rubio, J. A. Botía, J. T. Palma, and F. J. Campuzano, "Modeling a risk detection system for elderly's home-care with a network of timed automata," in *Proceedings of the 4th International Conference on Ambient Assisted Living and Home Care*, ser. IWAAL'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 82–89. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-35395-6_11

[37] R. Planinc and M. Kampel, "Detecting unusual inactivity by introducing activity histogram comparisons," in *9th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*, 2014.

[38] T. Duong, H. Bui, D. Phung, and S. Venkatesh, "Activity recognition and abnormality detection with the switching hidden semi-markov model," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, June 2005, pp. 838–845 vol. 1.

[39] J. Yin, Q. Yang, and J. Pan, "Sensor-based abnormal human-activity detection," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 20, no. 8, pp. 1082–1090, Aug 2008.

[40] A. R. M. Forkan, I. Khalil, Z. Tari, S. Foufou, and A. Bouras, "A context-aware approach for long-term behavioural change detection and

abnormality prediction in ambient assisted living," *Pattern Recognition*, vol. 48, no. 3, pp. 628–641, 2015.
[41] J. Wen, M. Zhong, and Z. Wang, "Activity recognition with weighted frequent patterns mining in smart environments," *Expert Systems with Applications*, vol. 42, no. 17-18, pp. 6423 – 6432, 2015.
[42] H. Mannila, H. Toivonen, and A. I. Verkamo, "Discovering frequent episodes in sequences," in *Proc. of KDD'95*, 1995, pp. 210–215.
[43] M. Magnusson, "Discovering hidden time patterns in behavior: T-patterns and their detection," *Behavior Research Methods, Instruments, & Computers*, pp. 93–110, 2000.
[44] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in *Proc. of 2000 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD00), Dallas, TX*, 2000, pp. 1–12.
[45] M. Sköldstam, K. Åkesson, and M. Fabian, "Modeling of discrete event systems using finite automata with variables," in *Proc. of the 46th IEEE Conference on Decision and Control, New Orleans, LA, USA*, 2007, pp. 3387–3392.
[46] M. Roth, J.-J. Lesage, and L. Litz, "The concept of residuals for fault localization in discrete event systems," *Control Engineering Practice*, vol. 19, no. 9, pp. 978–988, Sep. 2011. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00640169
[47] P. Bergagard, "Modeling and analysis of restart, transport, and resource allocation in manufacturing systems using sequences of operations," *Chalmers University of Technology*, 2012.
[48] R. Kadouche, H. Pigot, B. Abdulrazak, and S. Giroux, "Support vector machines for inhabitant identification in smart houses," *UIC 2010*, pp. 83–95, 2010.
[49] B. Chikhaoui, S. Wang, and H. Pigot, "A new algorithm based on sequential pattern mining for person identification in ubiquitous environments," *KDD Workshop on Knowledge Discovery from Sensor Data*, pp. 19–28, 2010.

Jeremie Saives recieved the M.Sc. degree in complex systems engineering from the Ecole Normale Superieure de Cachan, France, in 2013. He is currently undergoing a Ph.D. degree in the Automated Systems Engineering team of the Automated Production Research Laboratory in Cachan, France. His research interests include behavioural identification of Discrete Event Systems from sensors and activators data, and activity discovery in sensor-equiped smart environments.

Clement Pianon recieved the M.Sc. degree in complex systems engineering from the Ecole Normale Superieure de Cachan, France, in 2014. He also recieved a M.Sc. degree in robotic systems from the University Pierre et Marie Curie (Paris VI), France, in 2015. He is currently working as an engineer in a company specialized in programming and mechanical design, named "Structure Computation", and based in Orsay, France. His research interests include modeling and design of smart and complex systems.

Gregory Faraut received M.Sc. degrees in Electronic, Electrotechnic and Automatic from the University of Nice-Sophia Antipolis in 2006 and the Ph.D. degree in Automatic Control from INSA Lyon, Ampere Lab. in 2010. Since 2011, he is associate Professor of Automatic Control at LURPA, ENS Cachan, France. His research interests concerns the field of formal methods and models of Discrete Event Systens (DES). Application focus on identification and Ambient Assisted Living (AAL).