Proceedings of the 2019 Winter Simulation Conference
N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, eds.

DATA-DRIVEN MODELING AND SIMULATION OF DAILY ACTIVITY PATTERNS OF ELDERLIES LIVING IN A SMART HOME

Cyriac Azefack
Raksmey Phan
Vincent Augusto
Xiaolan Xie

Center for Biomedical and Healthcare Engineering
CNRS, UMR 6158 LIMOS
Mines Saint-Étienne
Univ. Clermont Auvergne
Saint-Étienne, 42023, FRANCE

Guillaume Gardin
Claude Montuy Coquard
Remi Bouvier

EOVI MCD Santé & Services
EOVI MCD
60 rue Robespierre - BP 10172
Saint-Étienne, 42012, FRANCE

ABSTRACT

Considering the globally aging population, one of the main challenges the healthcare system would have to face is to help elderly people stay at home in good health conditions and as long as possible. Recent advances in technologies answer this need with Smart Homes and Ambient Assisted Living programs. Data collected by the sensors are labeled and used to monitor the inhabitant Activities of Daily Living (ADLs). This paper presents a new modeling framework of the smart home resident's behavior that mimics his behavior on multiple aspects and is able to simulate the resident daily behavior. Our approach is illustrated by a real-life case study application. Results show that the presented framework enables the modeling of human behavior living alone in a smart home without prior knowledge on the inhabitant. Such results enable further research on frailty prediction through simulation.

1 INTRODUCTION

1.1 Context

The global population of elderly people aged 60 years or more was 600 million in 2000; it is expected to rise to around 2 billion by 2050 according to the United Nations Population Fund (UNFPA) (Guzman 2012). A significant proportion of the elderly population suffers from multiple age-related diseases such as Alzheimer, dementia and mild cognitive impairment. When these diseases are coupled with the natural decline of functional abilities of elderly people, it can prevent them to live independently. This evolution of their global health requires a unique care approach due to their complexity. Considering the increasing shortage of health practitioners, this could have a huge impact on the healthcare economy in the years to come.

The concept of frailty is used to characterize elderly decline, allowing the identification of elder adults at risk of death, disability, and institutionalization. If such identification is done early enough, frail elderly persons may receive adequate preventive care to recover their general health state and well-being. Recent advances in Information and Communication Technologies (ICT) have opened up new opportunities in healthcare. In this paper we focus on smart home technologies, referring to a residence equipped with sensors technologies that monitor the daily life of its inhabitant. Indeed, elderly people tend to follow a certain routine, either daily, weekly or monthly and those habits give them a sense of control of their environment. Losing this sense of control can be a sign of functional health decline among elderly, changes in the habits may indicate a need for more effort to perform everyday tasks or reveal the onset of disorders like Alzheimer or dementia. Such changes can be very subtle and caused by various reasons from the elderly life, which makes them quite complex to detect and the use of smart homes can help to identify these changes. Furthermore, such ICT solutions for aging at home are cheaper than nursing home stays or hospitalization for acute care.

1.2 Related Work

Research in modeling human behavior is a complex topic and have been intensively studied. The modeling was usually knowledge-based, based on experts knowledge on the domain (Jirgl et al. 2015).

However, recent advances in ICT gave us the opportunity to gather more information on human behavior and thus increase the accuracy of the modeling. Two different types of approaches can be used to collect this kind of data: obtrusive and unobtrusive approach (Novák et al. 2012). Obtrusive approaches used either sensors attached to the user body (accelerometers or tags) (Majumder et al. 2017) or video and ultrasonics sensors (Stack et al. 2018).

Although the systems based on obtrusive sensors gave outstanding results for real-time monitoring of patient's health and behavior monitoring, they come with some drawbacks. They highly interfere with the user's daily activities since he either have to attach them on his body each day or faces a loss of privacy. The other type of approach is unobtrusive sensors. They are used to reduce the disturbance caused by the sensors on the user and to get him accustomed to the sensors: this is the case of Heath Smart Homes (HSH).

A Health Smart Home (HSH) refers to a residence equipped with sensors technologies that monitor the daily activities of its resident. The sensors are added to the infrastructure of the residence without interfering with the resident's lifestyle. Their goal is to gather data to passively monitor and assess the functional abilities of the resident. Currently, there are quite numbers of HSH project deployed around the world. Some examples are CASAS (The Center for Advanced Studies in Adaptive Systems) at Washington State University (Crandall et al. 2013) and PlaceLab at MIT (Intille et al. 2006). Once the data is gathered through smart homes, the modeling can be done using low-level or high-level data analysis (Novák et al. 2012). Low level data analysis approaches use data directly from sensors like the work done on activity recognition or anomaly detection (Bouchard et al. 2018; Arifoglu and Bouchachia 2019; Dahmen et al. 2017). Bouchard et al. (2018) used sensors data to model activities of elderly residents using behavior trees which allowed for simulation tasks. In contrast, Tax et al. (2018) showed in their work how abstraction from sensor-level to Activities of Daily Living (ADLs) like *sleeping, eating or personal hygiene*, can help build more accurate behavior models of the smart home resident. This leads us to the other type of modeling approach which is high-level data analysis.

This type of approach uses as input the ADLs performed by the resident to model its behavior (Noury and Hadidi 2012; Suryadevara et al. 2013). Suryadevara et al. (2013) proposed a method based on a probabilistic approach to forecast the wellness of an elderly living alone in a smart home. The probabilistic approach is used to annotate the data gathered by the sensors and then predict their wellness based on the ADLs detected. Another example is the work done by Noury and Hadidi (2012) who aimed to build a simulation model that mimics the behavior of the subject in order to raise an alarm when there is a significant difference between real and simulated data. He compared two probabilistic method using ambulatograms of the simulated data: the Polya distribution and Hidden Markov Model (HMM). One the common limitation of these works is the fact that the dependency or pattern between activities is not taken into account. Along the same lines, Virone and Sixsmith (2008) showed the importance of the patterns between ADLs. For them, looking closely at those relationships might help uncover complex behavioral changes and help identify the onsets of cognitive decline.

1.3 Scientific Contribution

In light of these considerations, the main contribution of this paper is to propose a modeling framework of human behavior based on high-level activities data (ADLs) and taking into account a higher level of abstraction which is build from the patterns and relationships between activities. Our goal is to build a simulation model that will mimic the behavior of the smart home resident on multiple aspects using available data. The model relies on pattern recognition techniques to identify the inhabitant behavioral patterns, graph model to represents the complex and non-deterministic aspect of human behavior and finally Petri nets for a runnable simulation model.

This paper is organized as follows. In Section 2, we describe the proposed framework methodology. Section 3 describes the methods used to build the simulation model from a sequence of events collected through the smart home. The validation of the framework and its application on a real-life case study are presented in Section 4. Conclusions and perspectives for future work are presented in Section 5.

2 BEHAVIOR MODELING

Our work methodology is presented in Figure 1. After the data are collected through sensors around the house and clustered into ADLs 1, we use data mining techniques to have a better understanding of the data by unveiling relevant relationships between events 2. Those relevant relationships are then encapsulated into stochastic graphs 3. Those graphs are converted into Petri nets and linked according to a specific conversion algorithm 4. Once the "big Petri net" (the simulation model) is built, we simulate the virtual inhabitant activities 5. The model is then validated by comparing the real inhabitant activity to its virtual counterpart 6 and is ready to use 8 or updated if needed 7. Note that the preliminary recognition of activities according to sequences of sensors activations and deactivations included in step 1 is out of the scope of this paper.

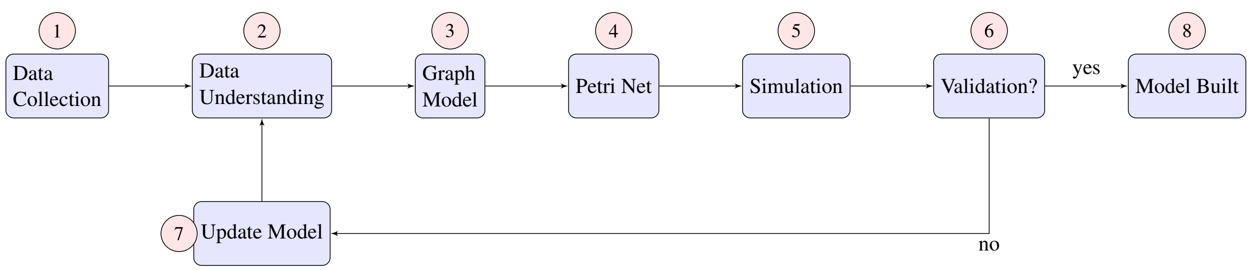


Figure 1: Global methodology.

This Section presents the mathematical formalism used for data understanding approach (step 2 of Figure 1).

2.1 Definitions

An *event sequence* is generated by the inhabitant activity around the house. For a better understanding of the data, the structure is formally defined as follow:

Definition 1 (Label) A label s represents an activity executed by the inhabitant (e.g: `sleeping`, `lunch`, `work`, `making a phone call`, `watching TV`, `...`).

Definition 2 (Domain Alphabet) The domain alphabet Π is the set of all possible labels that could appear in the input data stream.

Definition 3 (Event) An event δ is denoted as the tuple $\delta = (s,t,d)$ where s is a label, t the starting timestamp of the event and d its duration.

Definition 4 (Event Sequence) An event sequence Δ is an *ordered* sequence of events $\Delta = \langle \delta_1, \delta_2, ..., \delta_n \rangle$ with t_i the starting timestamp of δ_i such that $t_i \leq t_{i+1},\ \forall\ i \in [\![1\ ;\ n-1]\!]$. For instance, an event sequence could be $\Delta = \langle \delta_1, \delta_2 \rangle$ with $\delta_1 =$ `(eat,7:19am,15mn)` and $\delta_2 =$ `(relax,8:41am,10mn)`.

2.2 Patterns Discovery

This Section presents the formalism needed for the data mining algorithms used to extract behavior patterns from the collected data. The input event log is processed to retrieve frequent periodical *episodes* and their *occurrences*, formally defined as follows:

Definition 5 (Episode) An episode $E = \{s_1, ..., s_n\}$ is an unordered set of distinct labels. An example of episode is $E =$ `{cook, eat, relax}`. Episodes would be used to characterize sets of activities that have a strong temporal dependency.

Definition 6 (Episode Occurrence) ω is an occurrence of $E = \{s_1, ..., s_n\}$ if $\omega = \langle \delta_1, ..., \delta_n \rangle$ is an event sequence such that each label of E has one unique event associated in ω. The starting time of the occurrence is t_1. The duration of the occurrence is $t_n - t_1$ and it can be constrained using an episode length parameter T_{ep} as upper bound value.

This definition of an episode occurrence allows a *robust* recognition of an episode even if it is mixed with other events (e.g., the morning routine activities are occurring and the inhabitant go answer the phone right in the middle). The information both on episodes and their occurrences are needed because the episode by itself is not precise enough; it lacks of temporal information and also on the execution order between the different labels. Thus, the data mining algorithm used should output the episodes and their occurrences. To encapsulate information on both episodes and their occurrences we use an entity called *macro-activity* (Definition 7).

Algorithm 1 Algorithm to describe the creation of macro-activities.

Require: input event sequence Δ_{input}, episode maximum length parameter T_{ep}
$MAs \leftarrow$ empty set ▷ A set of Macro-Activities
$\Delta \leftarrow \Delta_{input}$
while Δ is not empty **do** ▷ Finish when Δ empty
  $episodes \leftarrow$ findAllPeriodicalEpisodes(Δ) ▷ List of all the periodical episodes found in Δ
  $E_{best} \leftarrow$ mostFrequentEpisode($\Delta, episodes$)
  $\Omega_{best} \leftarrow$ computeEpisodeOccurrences(Δ, E_{best}, T_{ep})
  $\mathscr{A} \leftarrow (E_{best}, \Omega_{best})$ ▷ Creation of the macro-activity $\mathscr{A}$
  append $\mathscr{A}$ to MAs
  $\Delta \leftarrow \Delta - \Omega_{best}$
return Set of Macro-Activities MAs ▷ All the Macro-Activities occurrences Ω are disjoints

Definition 7 (Macro-activity) A macro-activity describes the behavior of an episode. It is written as a tuple $\mathscr{A} = (E, \Omega)$ where E is a non empty episode. $E = \{s_1, ..., s_n\}$ and Ω is the sequence of occurrences of the macro-activity. $\Omega = \langle \omega_1, \omega_2, ..., \omega_m \rangle$ such that each ω_i is an occurrence of E and $\omega_i \cap \omega_j = \emptyset$ with $i \neq j$.

The procedure to create macro-activities is presented in Algorithm 1. The pattern mining algorithm used is based on the work of (Soulas et al. 2015). The input event log is parsed to find all the periodical episodes. Among frequent episodes, the most periodical one is picked E_{best} (line 6), its occurrences are computed Ω_{best} (line 7) and we create the corresponding macro-activity $\mathscr{A}$ (line 8). Once those occurrences are included in a macro-activity, they are subtracted from the input sequence as you can see in line 10. We do the same operation all over again while the input sequence is not empty. This enables to look for the most interesting pattern at each step in the input event log.

Example 1 We consider two occurrences $\omega_1 = \langle(\texttt{cook,7am,20min}),(\texttt{eat,7:30am,10min})\rangle$ and $\omega_2 = \langle(\texttt{cook,6:45pm,25min}),(\texttt{eat,7:10pm,15min})\rangle$ of the episode $E = \{\texttt{cook, eat}\}$. These two occurrences represent respectively breakfast and dinner. The macro-activity created on E represents here the eating behavior of the inhabitant and would have as sequence of occurrences $\Omega = \langle\omega_1, \omega_2\rangle$.

3 THE BEHAVIOR SIMULATION MODEL

This Section describes the construction of the resident's behavior simulation model (steps 3 to 5 of Figure 1).

3.1 Graph Model

In this Section, we propose an extension of the pattern discovery process presented above. Markov chains are often used to represent Bayesian processes with several states and associated conditional probabilities. For our problem, we use a similar approach based on stochastic graphs. We build a workflow model for each macro-activity to understand when each task (event) is executed, in what order and for how long. For that, we use a customized graph model represented as a directed tree where each branch is a potential execution order of the labels in the macro-activity.

For the macro-activity $\mathscr{A} = (E, \Omega)$, the algorithm 2 describes how to create the structure of the tree. The construction takes the macro-activity occurrences Ω as input and for each new label order found in the occurrences $\omega \in \Omega$, a new branch is added to the tree.

Algorithm 2 Tree construction.

Require: sequence of occurrences Ω
$N_0 \leftarrow createNode(\mathscr{O})$ ▷ Source node of the graph
for ω in Ω **do**
    $N_{cur} \leftarrow N_0$ ▷ Current node
    for δ in ω **do**
        $s \leftarrow label(\delta)$
        if N_{cur} has a child node with the label s **then**
            $N_{cur} \leftarrow N_{cur}.findChildWithLabel(s)$
        else
            $node = createNode(s)$
            $N_{cur}.addChild(node)$
            $N_{cur} \leftarrow node$
return N_0 ▷ the tree structure

Example 2 We consider episode $E = \{\texttt{relax, TV, eat}\}$ and the label execution order of the episode occurrences: $\omega_1 = \langle\texttt{relax, TV, eat}\rangle$, $\omega_2 = \langle\texttt{TV, relax, eat}\rangle$ and $\omega_3 = \langle\texttt{relax, eat, TV}\rangle$. Let $\mathscr{A}$ be the macro-activity on E with $\Omega = \langle\omega_1, \omega_2, \omega_3\rangle$ as occurrences. Figure 2 shows the tree structure built by applying the tree construction algorithm (Algorithm 2). Each branch describes a potential order of label execution and therefore they have the same length as the episode E.

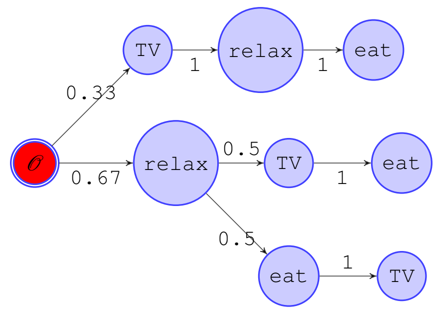


Figure 2: Example of tree construction.

Let $G_{\mathscr{A}}$ be the directed graph built from the occurrences of the macro-activity $\mathscr{A}$. It can be formally defined as $G_{\mathscr{A}} = \langle V, A, \mathscr{L}, \mathscr{D}, \mathscr{P} \rangle$ where:

- V is a set of nodes with a designated **root** node $\mathscr{O} \in V$, such that $\forall v \in V$, there is exactly one path from $\mathscr{O}$ to v. $V^* = V \backslash \{\mathscr{O}\}$
- A is a finite set of ordered pairs of nodes of V called directed arcs. The directed arcs with $\mathscr{O}$ as starting node are called *source arcs*. $A^* = A \backslash \{\textit{source arcs}\}$
- $\mathscr{L} : V \to E$ is a labeling function that associates a label to a node
- $\mathscr{D} : (V^* \cup A^*) \to \sum_{\mathscr{D}}$ is a function that associates to a node or a directed arc a *probability density function* (PDF) $\mathscr{D}$. A PDF for a continuous random variable X is a function $f_X : \mathbb{R} \to [0, 1]$ that describes the probability for X to take a given value. $\sum_{\mathscr{D}}$ is the set of PDFs for random variables representing durations ($\mathbb{R}_+$). The PDF on nodes describe events duration on the corresponding label and PDF on arcs describe idle time duration between two consecutive events.
- $\mathscr{P} : A \to [0\ ;\ 1]$ is a function that associates a transition probability to every directed arc of A.

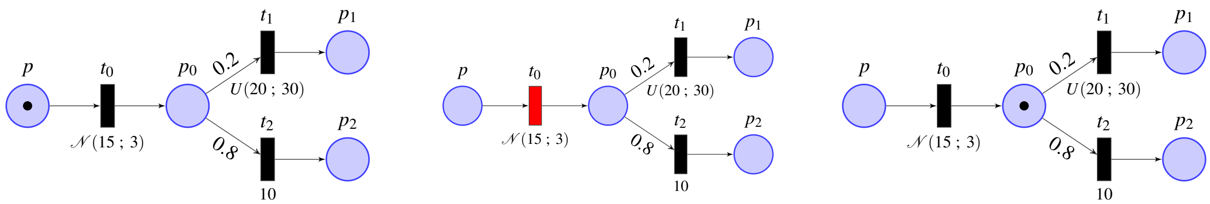
For a given node $v \in V$, let $v^\bullet \subset A$ be the set of v output arcs. if v is not a terminal node (node without exiting arcs) we have $\sum_{a \in v^\bullet} \mathscr{P}(a) = 1$, otherwise it is 0.

3.2 Petri Net Model

A graph model has been proposed in the last Section. It offers a representation of the inhabitant behavior patterns through macro-activities. For now, the different patterns are modeled separately, so there is a need to group them back in a bigger model for the overall behavior modeling. Although the graph model is easily readable and understandable, we need to run this model to assess its validity and to perform scenario simulation. Instead of describing a new simulation algorithm for the graph model, we convert it to a runnable Petri net (Heiner, Herajy, Liu, Rohr, and Schwarick 2012). This conversion will also allow the aggregation of all the macro-activities into a unique formal entity.

For that, we use a Stochastic Timed Petri net (STPN). A STPN is a timed Petri net with distributions on delays attached to transitions (seen as events here) and probabilities attached to arcs (Definition 8). As for Petri nets, tokens are processed from the preset of a transition to its postset but this is not immediate. The time that elapses is sampled according to a historical distribution attached to the transition. In this context, the token represents the house inhabitant and the transitions are the activities. After the firing of a transition (execution of an activity), the token is transferred to the output place which is followed by probabilistic arcs representing the likelihoods of next activities.

Definition 8 (Stochastic Timed Petri net (STPN)) A STPN is a tuple $\mathscr{N} = (P, T, F, \lambda, \theta, \pi)$ where:



(a) Initial state. **(b)** Firing of t_0. **(c)** Token production.

Figure 3: Example of stochastic timed Petri net.

- P is a set of places.
- T is a set of transitions.
- $F \subseteq (P \times T) \cup (T \times P)$ is a set of directed arcs
- $\lambda : T \to \Pi \cup \{\varepsilon\}$ a labeling function which assigns to each transition $t \in T$ a label, from the label alphabet Π, or an empty string ε.
- $\theta : T \to \sum_{\mathscr{D}}$ is the application of temporisation associating a probability density function for delays to a transition. Transitions with null delays are called *immediate transitions*.
- $\pi : F_{P^\bullet} \to [0;1]$ is the choice function assigning a probability to directed arcs of $F_{P^\bullet}$ (set of directed arcs exiting from a place). Places having more than one probabilistic arc are called *choice places*.

For a given place $p \in P$, let $F_{p^\bullet}$ be the set of directed arcs exiting from p. If p is not an exit place then $\sum_{a \in F_{p^\bullet}} \pi(a) = 1$ otherwise it is 0. The STPN also verify the following properties: (i) the weight of all the directed arcs in the net is 1; (ii) at any time, the sum of tokens in the net is 1; (iii) the firing delay of a temporized transition is randomly draw at the timestamp of the firing, according to the transition PDF.

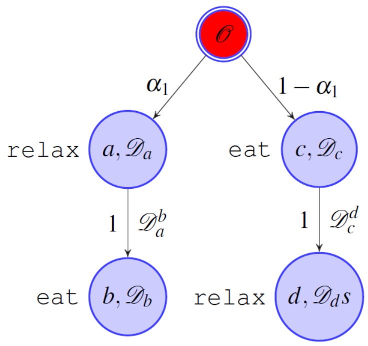
Example 3 We consider the STPN presented in Figure 3. Let x be the timestamp of the system at an initial state (Figure 3a). The initial marking of the net allows the timed transition t_0 to be enabled. The firing delay according to the net is normally distributed. After the transition is enabled, a firing delay d_0 is randomly drawn according to the distribution described (Figure 3b). Once the firing delay is elapsed the token is transferred into the choice place p_0 at the timestamp $x + d_0$ (Figure 3c). In the next state, there are two possibilities: either the transition t_1 (with the firing delay following an *uniform* distribution) is enabled with probability 0.2 or the transition t_2 (with a constant firing delay) is enabled with probability 0.8.

3.3 Macro-Activity Graph Model to STPN Conversion Algorithm

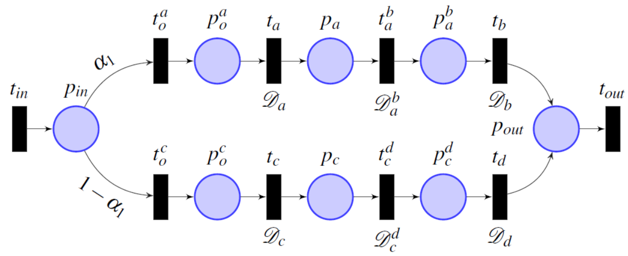
Let $G_{\mathscr{A}} = \langle V, A, \mathscr{L}, \mathscr{D}, \mathscr{P} \rangle$ be a graph built from the macro-activity $\mathscr{A}$. The goal here is to convert $G_{\mathscr{A}}$ to a STPN. Let $\mathscr{N} = (P, T, F, \lambda, \theta, \pi)$ be that STPN. The graph source node is converted into an immediate transition (with an empty string as label) followed by a place. Each of the graph nodes left is converted into a place followed by a temporized transition such that the node and the transition have the same label and PDF. Source arcs from the graph are converted into probabilistic arcs, with the same probability as the graph arc, followed by an immediate transition and a place. The rest of the graph directed arcs are converted into probabilistic arc, with the same probability as the graph arc, followed by a temporized transition, with the same PDF and label as the graph arc, followed by a place. The places corresponding to the graph sink nodes are merged into a final place followed by a final transition. Figure 4 gives an example of conversion from a macro-activity graph model to a STPN where $\mathscr{D}_i$ denotes the PDF of events duration generated by the graph node i and $\mathscr{D}_i^j$ the idle time from node i to j.

3.4 The Resident Behavior net

The Resident Behavior net (RB-STPN) is a STPN compiling all the macro-activities discovered. Let $\sum_{\mathscr{A}}$ be the set of all the macro-activities discovered in the input event log. We define $\mu_{\mathscr{A}} : \mathbb{R}_+ \to [0\ ;\ 1]$



(a) Graph model.



(b) STPN model.

Figure 4: Example of conversion from graph model to STPN.

the occurrence time probability density function of $\mathscr{A}$. It describes for each timestamp, the probability of occurrence of $\mathscr{A}$. At any given time, the inhabitant is either performing one macro-activity or doing "nothing". We define the macro-activity $\mathscr{A}_{idle}$ as the macro-activity describing the idle behavior of the inhabitant (time period without activity).

Figure 5 shows a representation of a RB-STPN where all the macro-activities are combined into a chain representing the life of the inhabitant. The blocks $\mathscr{A}_1, \ldots, \mathscr{A}_n, \mathscr{A}_{idle}$ represent the macro-activities respective STPNs. They are linked to choice places through probabilistic arcs with probability depending on the current timestamp x of the system. The RB-STPN describes the fact that the inhabitant performs a macro-activity when the time is suitable and when finished performs the next one and so on.

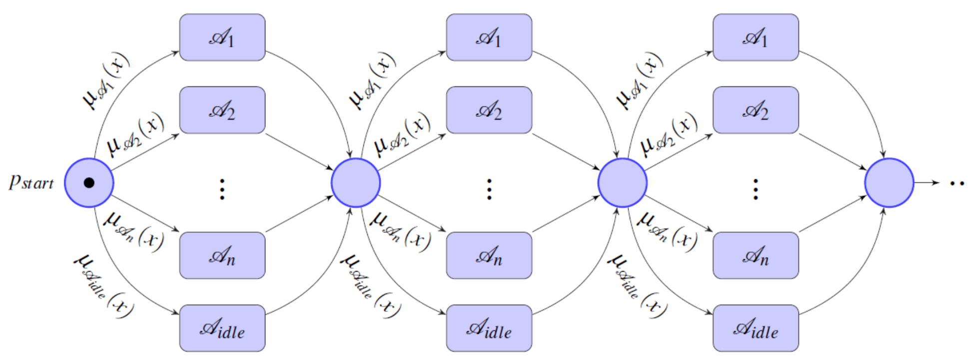


Figure 5: Resident Behavior STPN with x as the system current timestamp.

4 NUMERICAL EXPERIMENTS

The RB-STPN defined above is executable and can be directly used to simulate the lifestyle of the inhabitant. A token is generated into the starting place of the RB-STPN and a clock for the system is set for a starting timestamp. Transitions enabled by a choice place are randomly chosen according to the probability of the probabilistic arcs. All temporized transitions are fired as soon as possible. The firing of a temporized transition t with a non-empty string as label generates an event $\delta = (s, t, d)$ with $s = \lambda(t)$ the label of the transition, t the current timestamp of the system and d the firing delay of the transition. The event sequence generated describes the simulated log of activity. To validate our model, we use a ratio of the available data as training data for our model and the rest is used for validation. The goal of the validation is to assess how much the RB-STPN built describes the daily behavior of the resident.

4.1 Key Performance Indicators

4.1.1 Activities Daily Profile

We extract the occurrence time distribution of the activities along the days. The distributions are extracted from the testing set (real data) and from the generated set (simulated data) Then we compute two different metrics between simulation results and real data to validate the quality of the model:

- *Histogram Intersect*: The activity profile distribution is computed through a histogram with 288 bins, one bin per five minutes. The metric is computed as the intersection between the profile histograms in the training set and the simulated results. Higher the intersection area, means better matching between the activity time distribution in the simulated data and real data.
- *Density Intersect*: Instead of computing the intersection between the histograms, we compute the activity profile probability density using a *kernel density estimator* (KDE). The distance is computed here as the intersection area between the profiles kernel density estimates. The purpose here is to remove the time discretization bias.

For these two indicators, the closer they are from 1, the better is the activity's daily profile representation in the resident behavior simulation model.

4.1.2 Daily Sequence Alignment

The activities execution order is an important aspect of the inhabitant lifestyle so we need to validate that the sequences generated by the model are close to the one from real data. We use the Sequence Alignment Needleman-Wunsch Algorithm (Needleman and Wunsch 1970) to compare the training set and the simulation results in a sequential manner. The purpose of the algorithm is to find the optimal alignment between two sequences. The similarity between two sequences is defined as the quality of the alignment. Let us consider two sequences S_1 and S_2 the alignment between S_1 and S_2 is computed as follows $\{\forall a_1 \in S_1, \forall a_2 \in S_2\}$, *Match:* $+8$ ($a_1 = a_2$), *Mismatch:* -2 ($a_1 \neq a_2$), *and each gap symbol:* -2 ($a_1 =$ '$-$' or $a_2 =$ '$-$'). As an example, let us define two sequences; S_1 and S_2, as follows: $S_1: DBBAADB$, $S_2: DCCABDAB$. Figure 6 shows the optimal alignment between the two sequences and we compute the alignment score obtained is $8-2-2+8-2-2-2+8 = 24$.

D	B	B	A	A	D	-	B	S1
D	C	C	A	B	D	A	B	S2

Figure 6: Example of sequence alignment.

For each day in the validation period, we extract the sequence of activities in the training set and the simulation results and we compute the sequence alignment score. A random sequence of activities is also generated to compare the model results to a totally random model. Let S be the sequence to validate (from the simulation or randomly generated), $S_\varnothing$ an empty sequence of activities. The score is normalized as presented in Equation (1).

$$\text{score}_{\text{alignment}} = \frac{align(S_{real},\ S)\ -\ align(S_{real}, S_\varnothing)}{align(S_{real},\ S_{real})\ -\ align(S_{real}, S_\varnothing)} \quad (1)$$

4.2 Case Study

To validate the accuracy of our resident behavior modeling, we used the Aruba dataset provided by the CASAS team in the Washington State University (Cook 2010). The data was collected in the house of an elderly woman for 220 days. The residence was equipped with 34 sensors and 11 activities were labeled.

More than $1,7$*million* sensors recording collected. 6440 occurrences of activities are present overall the dataset: *bed to toilet*(156), *eating* (255), *enter home*(427 instances), *housekeeping* (33), *leave home*(427), *meal preparation* (1596), *relax*(2907), *others*(6), *sleeping*(398), *wash dishes*(64), *work*(171). To build the resident's behavior model, we used 80% of the dataset (176 *days*) and the 20% left (44 *days*) were used for validation. After the model is built, we generate multiple replications (10 replications) through simulation where each replication is an activity log of 44 *days* generated by the model. Then we compute the key performance indicators on each replication and average the results or compile them to obtain a 95% confidence interval.

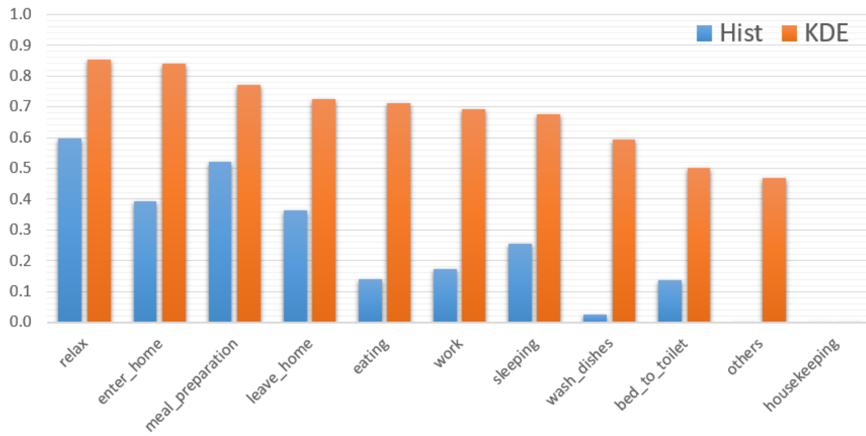


Figure 7: Activities daily profile distance between training dataset and simulation results.

Figure 7 shows the results obtained for the activities daily profiles, for each activity, we computed the *histogram Intersect* and the *density Intersect*. The results look significantly better when checked from a *density intersect* point of view. That is because instead of checking each of the histogram's bin separately, the profile is smoothed on neighboring bins. The three least frequent activities (`others` , `housekeeping` and `bed_to_toilet`), which combined represent less than 1% of the available data, have the poorest daily profile representations in the model. This is due to the lack of data. On the other side, the most frequent activities (`relax` and `meal_preparation`) which represent approximately 70% of the dataset have one of the best representation. This result makes sense because the more information we have on an activity, the more accurate will be its modeling.

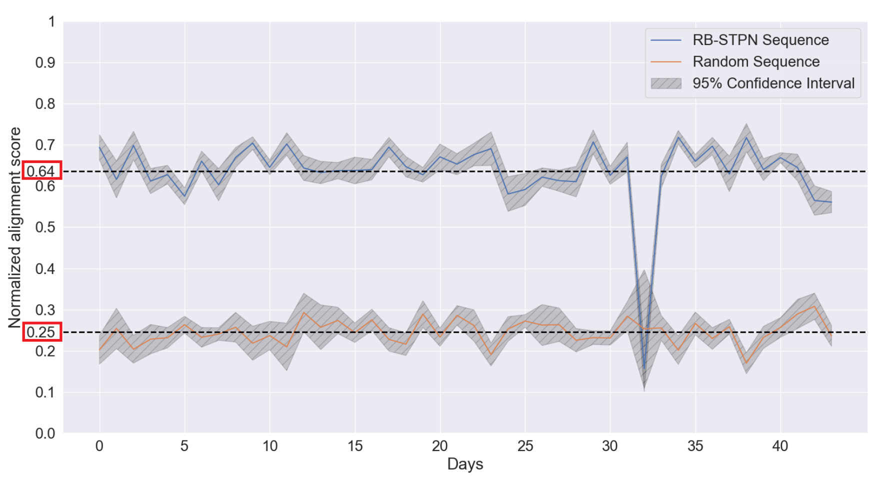


Figure 8: Daily sequence alignment score between real and simulation data.

Figure 8 on the other side shows the results obtained for the sequence alignment of the simulated data (mean value for all the replications), for the randomly generated sequence and the real data. We see that on an average, 64% of the sequence generated by the model is aligned with the real data against 25% for randomly generated sequences. The graph also shows a negative peak on the 32^{th} day of the dataset for the simulated data. After further investigation, on that day, only three activities were recorded, two early in the day and one at night. Since the abnormal behavior lasted only for one day, we can assume that maybe there was some problem with the sensors, or it was a day for infrastructure maintenance. By definition, our model focus on the daily average behavior of the resident so as soon as the resident deviates from his average behavior, the model give poor results. We can use this to raise alerts in case of sudden drift of the resident behavior which can show signs of falls or cognitive related problems.

5 CONCLUSION AND FUTURE WORK

The purpose of this work was to propose a modeling framework of human behavior in smart homes. A novel framework has been presented to model a smart home resident's behavior based on high-level activities-labeled data and taking into account the patterns and relationships between these activities.

The resident's behavior model is built from scratch by learning from high-level activities gathered in a smart home environment, without prior domain knowledge. Based on the results, it appears that activities-labeled data gathered in a smart home are sufficient to build a virtual representation of the inhabitant lifestyle. This work proves the feasibility of a simulation model which can mimic the lifestyle of a smart home inhabitant. The proposed approach has been designed for people living alone. One main downside of the model presented here is that the training data is considered static, meaning that each day is treated independently, our future work aims at considering the chronological link between days as the evolution of the model parameters with time through the training data. This model could also be easily augmented with time series analysis used as a way to describe model parameters evolution through time and time series forecasting to predict their evolution.

REFERENCES

Arifoglu, D., and A. Bouchachia. 2019. "Detection of Abnormal Behaviour for Dementia Sufferers Using Convolutional Neural Networks". *Artificial Intelligence in Medicine* 94:88–95.

Bouchard, B., S. Gaboury, K. Bouchard, and Y. Francillette. 2018. "Modeling Human Activities Using Behaviour Trees in Smart Homes". In *Proceedings of the 11th PErvasive Technologies Related to Assistive Environments Conference*. New York, NY, USA.

Cook, D. J. 2010. "Learning Setting-Generalized Activity Models for Smart Spaces". *IEEE Intelligent Systems* 27:32–38.

Crandall, A. S., N. C. Krishnan, B. L. Thomas, and D. J. Cook. 2013. "CASAS: A Smart Home in a Box". *Computer* 46(07):62–69.

Dahmen, J., B. L. Thomas, D. J. Cook, and X. Wang. 2017. "Activity Learning as a Foundation for Security Monitoring in Smart Homes". *Sensors* 17(4).

Guzman, J. 2012. *Ageing in the Twenty-First Century: A Celebration and A Challenge*.

Heiner, M., M. Herajy, F. Liu, C. Rohr, and M. Schwarick. 2012. "Snoopy – A Unifying Petri Net Tool". In *Application and Theory of Petri Nets*, edited by S. Haddad and L. Pomello, 398–407. Berlin, Heidelberg: Springer Berlin Heidelberg.

Intille, S. S., K. Larson, E. M. Tapia, J. S. Beaudin, P. Kaushik, J. Nawyn, and R. Rockinson. 2006. "Using a Live-In Laboratory for Ubiquitous Computing Research". In *Pervasive Computing*, edited by K. P. Fishkin, B. Schiele, P. Nixon, and A. Quigley, 349–365. Berlin, Heidelberg: Springer Berlin Heidelberg.

Jirgl, M., M. Havlikova, and Z. Bradac. 2015. "The Dynamic Pilot Behavioral Models". *Procedia Engineering* 100:1192–1197. 25th DAAAM International Symposium on Intelligent Manufacturing and Automation, 2014.

Majumder, S., E. Aghayi, M. Noferesti, H. Memarzadeh-Tehran, T. Mondal, Z. Pang, and M. J. Deen. 2017. "Smart Homes for Elderly HealthcareRecent Advances and Research Challenges". *Sensors* 17(11).

Needleman, S. B., and C. D. Wunsch. 1970. "A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins". *Journal of Molecular Biology* 48(3):443–453.

Noury, N., and T. Hadidi. 2012. "Computer Simulation of the Activity of the Elderly Person Living Independently in a Health Smart Home". *Computer Methods and Programs in Biomedicine* 108(3):1216–1228.

Novák, M., M. Bias, and F. Jakab. 2012. "Unobtrusive Anomaly Detection in Presence of Elderly in a Smart-Home Environment". In *2012 ELEKTRO*, 341–344.

Soulas, J., P. Lenca, and A. Thpaut. 2015. "Unsupervised Discovery of Activities of Daily Living Characterized by their Periodicity and Variability". *Engineering Applications of Artificial Intelligence* 45:90–102.

Stack, E., V. Agarwal, R. King, M. Burnett, F. Tahavori, B. Janko, W. Harwin, A. Ashburn, and D. Kunkel. 2018. "Identifying Balance Impairments in People with Parkinsons Disease Using Video and Wearable Sensors". *Gait & Posture* 62:321–326.

Suryadevara, N., S. Mukhopadhyay, R. Wang, and R. Rayudu. 2013. "Forecasting the Behavior of an Elderly Using Wireless Sensors Data in a Smart Home". *Engineering Applications of Artificial Intelligence* 26(10):2641–2652.

Tax, N., N. Sidorova, R. Haakma, and W. van der Aalst. 2018. "Mining Process Model Descriptions of Daily Life Through Event Abstraction". In *Intelligent Systems and Applications*, edited by Y. Bi, S. Kapoor, and R. Bhatia, 83–104. Cham: Springer International Publishing.

Virone, G., and A. Sixsmith. 2008. "Monitoring Activity Patterns and Trends of Older Adults". In *Proceedings of the 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, edited by K. Chon, D. Hudson, A. Lain, X. Pan, P. Bonato, P. Vicini, M. Khoo, R. Buetra, R. Jones, B. Layton, J. Patton, D. Panescu, E. Sloane, N. Lovell, and J. Monzon, 2071–2074. Vancouver, Canada: Institute of Electrical and Electronics Engineers, Inc.

AUTHOR BIOGRAPHIES

CYRIAC AZEFACK received a MSc degree in Computer Science and Machine Learning from IMT Atlantique Bretagne Pays de la Loire, France, in 2017. He is currently a Ph.D. student at the Center for Health Engineering at Ecole Nationale Superieure des Mines de Saint-Etienne (ENSM.SE), France and works with the french health organization EOVI MCD. His research interests include smart homes, behavior modeling, data mining, process mining, and machine learning applied to health-care. His email address is cyriac.azefack@emse.fr.

RAKSMEY PHAN received his Ph.D. degree from the Institut Suprieur d'Informatique, de Modlisation et de leurs Applications (ISIMA), France, in 2013 and his engineering degree, in 2010. Currently, he is a researcher-engineer in the Center for Health Engineering and in the IEOR team of CNRS UMR 6158 LIMOS, EMSE. His fields of interest include Operational Research, Databases and Internet of Things. His web address and e-mail are respectively http://emse.raksmey.fr/ and raksmey.phan@emse.fr.

VINCENT AUGUSTO received his Ph.D. in Industrial Engineering from Mines Saint-Etienne, France, in 2008 and the Habilitation Diriger des Recherches degree from the University of Saint-Etienne, France, in 2016. He was a visiting scholar at CIRRELT (Centre Interuniversitaire de Recherche sur les Rseaux dEntreprise, la Logistique et le Transport), University of Laval, Quebec, Canada in 2009 and 2015. Currently, he is a professor of industrial engineering in the Department of Health care Engineering at Mines Saint-Etienne. His research interests include modeling, simulation, optimization of health care systems and their supply chains, process mining and machine learning applied to health-care. His e-mail address is augusto@emse.fr.

XIAOLAN XIE received his Ph.D degree from the University of Nancy I, Nancy, France, in 1989, and the Habilitation à Diriger des Recherches degree from the University of Metz, France, in 1995. Currently, he is a distinguished professor of industrial engineering, the head of the department of Healthcare Engineering of the Center for Health Engineering and the head of IEOR team of CNRS UMR 6158 LIMOS, Ecole Nationale Supérieure des Mines (ENSMSE), Saint Etienne, France. He is also a chair professor and director of the Center for Healthcare Engineering at the Shanghai Jiao Tong University, China. He is author/coauthor of over 250 publications including over 90 journal articles and five books. He has rich industrial application experiences with European industries. He is PI for various national and international projects including ANR-TECSAN HOST on management of winter epidemics, NSF China key project on planning and optimization of health care resources, French Labex IMOBS3 project on home health cares, FP6-IST6 IWARD on swarm robots for health services. His email address is xie@emse.fr.

GUILLAUME GARDIN received a MSc in general management of social enterprise from ESSEC Business School, Paris, in 2013. He worked as a regional director of the company Mutualite Francaise Loire - Haute Loire SSAM for 7 years. He is now the director of the Research and Development department of the company EOVI MCD Sante et Service, a non-profit french health organization which offers a wide range of care and services. His e-mail address is guillaume.gardin@eovi-mcd.fr.

CLAUDE MONTUY COQUARD received a MSc in private law from the University Jean Monnet, Saint-Etienne, France in 1988. She works as director of the medico-social department of the company Mutualite Francaise Loire - Haute Loire SSAM since 31 years. Her e-mail address is cmontuycoquard@mutualite-loire.com.

RÉMI BOUVIER received a MSc in health and finance management from the Toulouse Business School, Toulouse, France, in 1991. He is been working in the field of health since 30 years. He is now the CEO of the companies EOVI MCD Sante et Service and Mutualite Francaise Loire - Haute Loire SSAM. Both are non-profit french health organization which offer a wide range of care and services. His e-mail address is rbouvier@mutualite-loire.com.