



HAL
open science

Set Theory-based FSM for Managing Home Emergencies Concept, Properties and Process Algebra

Allal Tiberkak, Abdelfetah Hentout, Abdelkader Belkhir, Tayeb Lemlouma,
Ahmed Bouabdallah

► **To cite this version:**

Allal Tiberkak, Abdelfetah Hentout, Abdelkader Belkhir, Tayeb Lemlouma, Ahmed Bouabdallah. Set Theory-based FSM for Managing Home Emergencies Concept, Properties and Process Algebra. 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS), Oct 2018, Tebessa, Algeria. 10.1109/PAIS.2018.8598482 . hal-01977701

HAL Id: hal-01977701

<https://hal.science/hal-01977701>

Submitted on 30 Aug 2022

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

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

Set Theory-based FSM for Managing Home Emergencies Concept, Properties and Process Algebra

Allal Tiberkak

*Department of Mathematics and Computer Science
Faculty of Sciences, University Dr. Yahia Fares
Boulevard de l'ALN, Ain D'heb
Medea 26000, Algeria
tiberkak.allal@univ-medea.dz, allal.tiberkak@gmail.com*

Abdelfetah Hentout

*Centre de Développement des Technologies Avancées (CDTA)
Division Productique et Robotique (DPR)
BP 17, Baba Hassen
Algiers 16303, Algeria
ahentout@cdta.dz, ahentout@outlook.com*

Abdelkader Belkhir

*Computer Systems Laboratory
Faculty of Electronics and Computer Science
University of Sciences and Technology
Houari Boumediene, Algiers, Algeria
belkhir@lsi-usthb.dz*

Tayeb Lemlouma

*D2-Networks
Department of Telecommunication
and Services, IRISA Laboratory
Lannion, France
Tayeb.Lemlouma@irisa.fr*

Ahmed Bouabdallah

*IMT Atlantique
IRISA
UBL F-35576 Cesson Sévigné
France
ahmed.bouabdallah@imt-atlantique.fr*

Abstract—Home emergencies affect greatly the health of inhabitants (mostly children, elderly and dependent persons). Nowadays, the deployment of smart home doesn't require purchasing expensive devices due to advances in communication and computing technologies. However, implementing a customized home emergencies management system is not an easy task. Indeed, it involves two disciplines: (i) *Medicine* and (ii) *Information and Communication Technologies* as mentioned by the *World Health Organization*.

In this paper, we propose an approach that hybridizes the two concepts: FSM theory and set theory. The aim is being to facilitate (i) the design of the system behavior for the health professionals, (ii) the translation of the human readable format (emergency management rules) to a machine readable format for the computer specialist, and (iii) mainly the verification of the FSM properties for reducing human errors for both of them. The paper gives the formal description of the set-based FSM, properties that have to be taken into account and the representation of process algebra expressions. Finally, we describe a case study to illustrate the benefits of the proposed FSM.

Index Terms—Home emergency Management, Finite State Machine, Set Theory, Process Algebra.

I. INTRODUCTION

Nowadays, advances in ICTs (*Information and Communication Technologies*) facilitate equipping household appliances with intelligent and communicating components. As a result, it is possible to make homes smart by installing networks, adding sensors, actuators and a gateway (to run software) to control the home environment.

Home emergencies, which may require urgent reactions, are particularly important because they may represent immediate risks on the residents health. In addition, it is difficult to ensure a continuous human presence with the necessary help for each assisted person, especially with the growing number of such people (dependent persons) [1]. One of the main ways to

achieve this goal is to use an e-health system [2] for automatic detection and management of domestic emergencies.

The existing home-care systems to offer help to dependent persons suffer from many disadvantages. Some systems only consider specific situations (e.g., fall detection [3]), others only deal with abnormal situations by sending notifications to caregivers [4]. A third category of solutions uses intelligent software [5] or specific technologies [6]. Therefore, all of the above solutions are neither easily extensible nor adaptable to new possible situations. Consequently, it is highly recommended to develop a generic solution that is, at same time, independent of any specified situation, adaptable, and easily human readable.

There are generic and extensible solutions based on the use of rules. A rule is of the form $IF \langle Condition \rangle THEN \langle Action \rangle$ [7]. These techniques generate actions to manage the situation. Unfortunately, in some circumstances, the actions suggested by the home automation systems could not change the situation; subsequently, the system would indefinitely invoke the same actions. In such a case, the system may become unstable and enter an endless loop.

According to our previous study [8], the most suitable methodology to remedy the main drawback of rules-based systems (i.e., infinite loops) would be to use a FSM (Finite States Machine). FSMs are used in several works to handle issues related to smart home. In addition, several solutions have been developed for managing energy [9], managing the home network and its equipment [10], facilitating the control of household appliances [11] and managing security [12].

Other works based on FSM and directly related to enhance home healthcare services have also been proposed in the literature. In [13], a user interface based on virtual agent is proposed to permit inhabitants to operate home appliances and services. In [14] [15], a system is developed to classify the activities of

the daily life. In [16], a system is proposed to detect sleep-related activities (bedridden, awake, etc.). Nguyen et al. [17] built a system that deploys domestic robot behaviors such as opening a refrigerator door, toggling a switch and unlocking a door. Finally, in [18], the authors developed a prototype of a smart home system for the purpose of simulating human behaviors.

The *HEMS (Home Emergency Management System)* provides dependent residents with health services, especially in well-known and mastered emergency situations. The targeted populations are dependent people who need continuous help and may be alone. The environment of the HEMS is thus enriched with a set of health devices [19].

We propose the use of an extended FSM to simplify the design and the exploitation of the HEMS. We note here that the formal analysis of the HEMS behavior is not the scope of this paper.

The choice to use FSM is motivated by the fact that it allows developing generic, scalable and smart HEMS. Since FSM uses implicit memory, the proposed HEMS can invoke different services for the same observed situation. This means that the doctor can design a FSM in such a way that if a service is not able to solve the current issue, the HEMS will invoke other services having the same objective.

Henzinger presented in [20] a hybrid automata for the specification and verification of the behavior of hybrid systems (digital systems that interact with analog one). Hybrid automata is not adequate for automatic emergency management purposes; indeed, it is not designed for human behavior, and it takes as inputs only real-numbered variables. However, emergency management system is human-centric and it deals with a large kind of values that may be contents, discrete, orders, etc.

Human behaviors can be expressed in natural language and abstracted into the categories: “to be”, “to have”, and “to do” that are abstracted and described by the “mathematical logic”, the “set theory”, and the “process algebra”, respectively [21]. Therefore, we will give a formalism for the *Set Theory-based FSM* that couples the formalism of FSM with mathematical logic and set theory and a methodology to translate process algebra operations to Set Theory-Based FSM. In order to make the design of emergency management process easier, we adopt a very reduced set of operations: $\{\wedge\}$ for mathematical logic, $\{\in\}$ for set theory, and BPEL (*Business Process Execution Language*) [22] operations for the process algebra.

The rest of this paper is organized into six sections. Section II presents an overview on home emergencies management; the section also describes the proposed extended FSM which is based on sets theories. Section III enumerates the most important properties of this FSM. Section IV shows that our FSM preserves the operations of process algebra. Section V presents a case study. Finally, section VI concludes this work and gives some perspectives of future works.

II. EMERGENCIES MANAGEMENT

Automating home emergency management requires real-time recognition of the home context. This latter consists of the current values of many attributes of the domestic environment which mainly include the state of health of the hospitalized patients, the vital signs of the inhabitants, their situations and preferences, objects into the home, etc. Context values are automatically retrieved by using appropriate sensors (thermometers, etc.), by the intervention of the inhabitants for subjective values, or from the state registers of home objects. Other sources of data are needed for the healthcare system, such as databases, EMR (Electronic Medical Record), etc.

A. Overview

Fig. 1 illustrates the deployment of the HEMS proposed in [8]. On the one hand, as shown in Fig. 1a, the physician begins by designing the FSM in a human-readable format. After that, a technician transforms it into an XML file taking into account the devices description. Finally, the technician uploads this file to the home gateway. On the other hand, as shown in Fig. 1b, the messages are exchanged over the home network according to the changes in the FSM state.

This paper only focuses on the description of the proposed FSM (formalism, properties and algebra process).

B. Context

Most context values are represented into sets or intervals. In this work, we assume that the possible values of each attribute can be represented in a set of values (either ordered or unordered):

- *Environmental information (temperature and humidity):* For these parameters, World Health Organization (WHO) imposes sensors with an accuracy of $0.5^{\circ}C$ for temperature and 5% for humidity [23]. In addition, the typical range of house temperature is between $4^{\circ}C$ and $40^{\circ}C$ (with 80 possible values).
- *Vital signs of the inhabitant:* there are many vital signs that can be permanently measured by adequate devices, in particular for sick inhabitants. Normalized vital signs are body temperature, heart rate, respiratory rate, blood pressure, pain, blood glucose level, etc. [24]. All values of vital signs are expressed in scalar values. For example, the degree of pain is a subjective parameter; its value does not only depend on the pain itself but also on the person. It can take a value between 1 and 11 points [25].
- *Patient profile (age, weight, height, etc):* For example, the age generally varies between 0 and 150 years and divided into intervals. Each range has some special properties, for example, the age between 0 and 2 years is measured in months.
- *State of health:* Mainly diseases and symptoms for a resident. For example, a person may have fever, flu, pregnancy, etc. There are also some subjective syndromes such as fatigue or pain in a part of the body. The possible

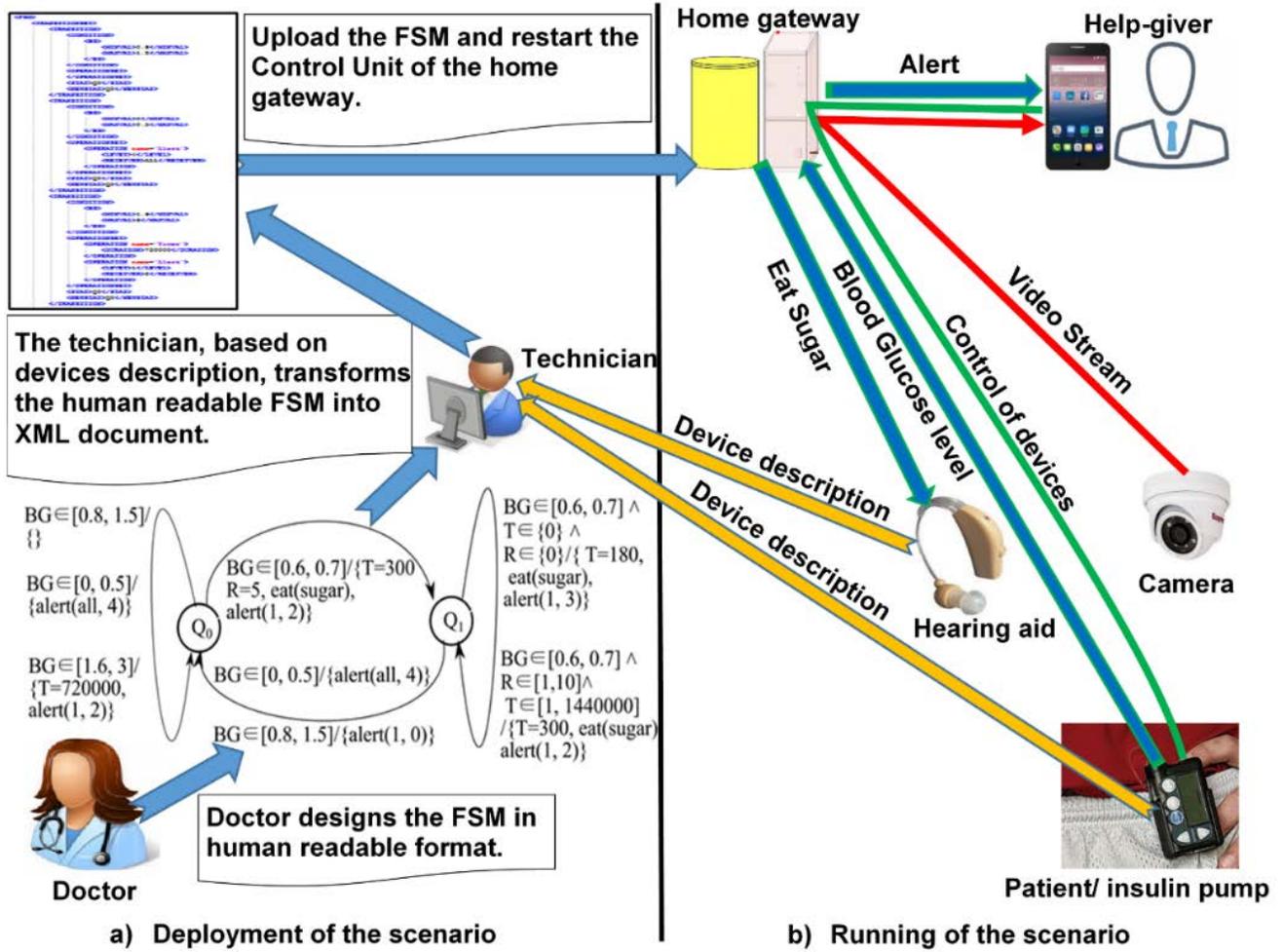


Fig. 1. The Home Emergency Management System proposed in [8].

locations of the pain are very important; they can be represented in a set of named parts of the body.

- *Preferences of the inhabitants*: generally, each inhabitant has preferences on the domestic environment and the setting of home devices. For example, environmental preferences can cover the preferred temperature and humidity. However, the device preferences may cover the broadcasted multimedia content on television, the state of the windows (open or closed), etc.
- *States of the inhabitants*: this state changes in time. The inhabitant can be asleep, stand, sitting, etc.
- *Device status*: Each smart device has a list of status registers; each has a list of possible values. These latter can be discrete such as a door state (open, closed, undefined) or continuous such as degree of opening of windows and doors, battery level, power consumption, etc.

As illustrated above, the most important domestic context parameters can be represented by a set of values (discrete

sets or intervals). In general, semantic values are given for these subsets. For example, temperature level is considered as follows [26]:

- *Cold*: indoor temperature is in $[14^\circ C, 22^\circ C]$; outdoor is in $[5^\circ C, 20^\circ C]$.
- *Comfortable*: indoor temperature is in $[22^\circ C, 26.5^\circ C]$; outside is $20^\circ C$.
- *Hot*: indoor temperature is in $[26.5^\circ C, 32^\circ C]$; outside ranges in $[20^\circ C, 35^\circ C]$.

In order to propose our FSM which is based on set theories, we take advantages of the fact that the values of the context parameters may be represented in subsets.

C. FSM based on set theories

In this work, FSM is used to handle emergencies. The proposed FSM uses implicit memory (i.e., ability to consider previous states) to call different equivalent services to deal with the same observed situation. In other words, if a given service does not solve a situation, the FSM will invoke other

services that can deal with this emergency, but differently. For example, if a situation of hypoglycemia is detected, the proposed system will first warn the patient to eat; otherwise, the system will trigger a high-level alert with (as shown in Fig. 1).

The objective of our FSM modeling is to select the set of actions to be performed according to the observed context. The formalization of the FSM is inspired by the Mealy machine [27] and UML state machine [28]. The main difference is that our FSM generates sets of actions instead of generating binary values. The proposed FSM is a 6-tuple, $FSM = (\hat{Q}, \hat{T}, \hat{A}, \hat{E}, \hat{O}, \hat{C})$, where [8]:

- $\hat{Q} = \{q_1 \dots q_i \dots q_Q\}$ is the set of all states defined in the FSM; $Q = |\hat{Q}|$.
- $\hat{T} = \{t_{ijk} : i, j, k \in N, 1 \leq i, j \leq Q, 1 \leq k \leq K_{ij}\}$ is the set of all the possible transitions that allow the FSM to pass from a state q_i to a state q_j . K_{ij} is the number of all possible transitions between q_i and q_j .
- $\hat{A} = \{a_1 \dots a_i \dots a_A\}$ is the set of all context attributes such as *temperature*, *blood glucose*, etc. For each context attribute a_i , we define a set of possible values \check{A}_i . This can include discrete values or intervals.
- $\hat{E} = \{e_1 \dots e_i \dots e_E\}$ is the set of all extended state variables. Extended state variables are internal variables used in the algorithm applied by the FSM (for example, *repetition counters*, *waiting time*, etc.). For each extended state variable e_i , we define the set of all possible values \check{E}_i .
- $\hat{O} = \{o_{ijk} : i, j, k \in N, 1 \leq i, j \leq Q, 1 \leq k \leq K_{ij}\}$ is the set of all subsets of actions that could be invoked when the FSM leaves a state. When the FSM goes to the state q_j from q_i following the transition t_{ijk} , it must invoke all the actions listed in the subset o_{ijk} . K_{ij} is the number of all possible transitions between q_i and q_j .
- $\hat{C} = \{c_{ijk} : i, j, k \in N, 1 \leq i, j \leq Q, 1 \leq k \leq K_{ij}\}$ is the set of conditions used to select the correct transitions to go from one state to another. c_{ijk} is the k^{th} condition that allows to leave the state q_i to reach q_j following the transition t_{ijk} . Each condition is a conjunction of atomic propositions expressing whether a context attribute a_i (or an extended variable e_i) belongs to a set of possible values. Each set of possible values \check{A}_{vijk} (or \check{E}_{vijk}) depends on the attributes used in the condition c_{ijk} of the transition t_{ijk} (k^{th} transition from q_i to q_j). Of course, as previously defined: $\forall i, j, k : \check{A}_{vijk} \subseteq \check{A}_v$; formally, the condition c_{ijk} is written in the following form:

$$(\bigwedge_{v=1}^A (a_v \in \check{A}_{vijk})) (\bigwedge_{v=1}^E (e_v \in \check{E}_{vijk})) \quad (1)$$

III. FEATURES OF THE PROPOSED FSM

In this section, we identify the different properties and advantages that must be respected while designing the FSM.

A. Properties

The most important include completeness, determinism and connectivity of states [29]. For each property, a verification algorithm could be proposed. For example, to verify the properties of completeness and determinism, we use set theory. Furthermore, to check the state connectivity, we use graph theory.

- *Paths*: a path is the set of successive elements (states, conditions, transitions or sets of operations) that make it possible to go from one state to another. If a path that leads from q_i to q_j is composed of the elements $y_0, y_1 \dots y_Y$, we denote $q_i \xrightarrow{y_0, y_1 \dots y_Y} q_j$:
 - *Conditions path*: the successive conditions must be checked to move from one state to another. We denote $q_i \xrightarrow{(a_0, e_0), (a_1, e_1) \dots (a_Y, e_Y)} q_j$ where (a_y, e_y) are values of context attributes and extended variables used to evaluate conditions c_i .
 - *Operations path*: this is the set of subsets of operations that are applied successively when going from one state to another.
 - *States path*: it is the set of states traversed successively which makes it possible to go from one state to another.
 - *Transitions path*: it is the set of successive transitions, which enables the FSM to pass from one state to another.

The number of conditions path is the same as that of transitions path, because for each transition there is one and only one condition to be checked. However, the number of operations paths can be smaller than that of states paths; indeed, it is possible to follow several different states paths and to perform the same sequence of operations. As shown in the example of Fig. 2, to go from q_1 to q_5 , there exist two possible conditions paths $C_{11} \rightarrow C_{21}$ and $C_{12} \rightarrow C_{31}$. In addition, there are two possible states paths: $q_1 \rightarrow q_2 \rightarrow q_5$ and $q_1 \rightarrow q_3 \rightarrow q_5$. However, there is only one operations path $o_1 \rightarrow o_2$.

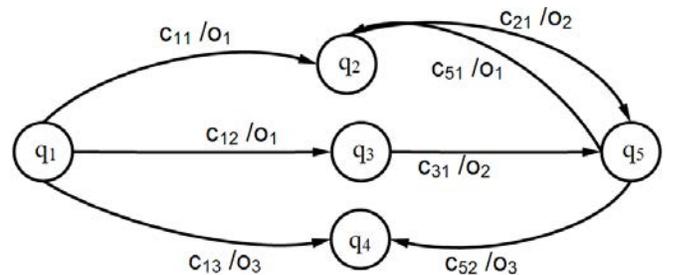


Fig. 2. The different paths in a FSM.

- *Cycles*: a cycle is a path that goes from one state and returns back to the same state. An optimal cycle is one for which the cost of performed operations is minimal. The cost can be measured in terms of energy consumption, time, number of performed operations, etc.

- *Completeness*: a complete FSM means that for any context value; there is at least one decision to make. That is, an FSM is complete if and only if all the states are complete. A state is complete if for all context values and extended variables there is at least one transition which leads out of that state; formally:

$$\forall q_i \in \widehat{Q}, \exists q_j \in \widehat{Q}, \exists k \in N : c_{ijk} = true \quad (2)$$

- *Determinism*: an FSM is deterministic means that for any context value, there is only one decision to make. An FSM is deterministic if and only if all their states are deterministic. A deterministic state means for all values of context attributes and extended variables, there is only one transition that leaves this state; formally:

$$\forall (c_{ijk}, c_{ijl}) \in \widehat{C}^2 : (c_{ijk} = true) \wedge (c_{ijl} = true) \Rightarrow k = l \quad (3)$$

- *Connectivity*: an FSM is connected means that there is always a possibility to switch from any state to any other state. In other words, FSM is connected if and only if each state is accessible from any other state ; formally:

$$\forall (q_i, q_j) \in \widehat{Q}^2, \exists (t_0, t_1 \dots t_n) \in \widehat{T}^n; q_i \xrightarrow{t_0, t_1 \dots t_n} q_j \quad (4)$$

- *Passability*: a condition is passable if there is at least one value for each state variable and a value for each context attribute that makes the value of the condition *true*; formally:

$$\forall c \in \widehat{C}, \exists v \in (\prod_{i=1}^A \check{A}_i) \cdot (\prod_{i=1}^E \check{E}_i) : c = true \quad (5)$$

A transition with a condition that is not passable is a redundant transition.

- *Matching*: verification or matching is the process of choosing which transition should be taken to exit the current state. The choice is made according to the current context and the values of extended variables. The transitions are checked one by one and the selected transition is that with the condition value is *true* (each context attribute value is contained in the corresponding set of possible values within the condition expression).

The analysis of each of these properties can give an overview on the system behavior; it allows also to detect possible mistakes done by the designer. For example:

- *Path* permits identifying the followed path in solving an abnormal situation and whether it is the optimal one.
- *Cycles* allow to detect if it is possible to return back to some states in the future.
- *Completeness* permits to be sure that there is always a decision to make for any possible situation encountered by the HEMS.
- *Determinism* ensures that any time, there is one and only one decision to make.

- *Connectivity* allows to detect unreachable states that may mean a design error.
- *Passability* for assuring that there are no useless transitions to avoid the clutter in the FSM.

B. Advantages

The proposed FSM has many advantages; we cite the following ones:

- *Easy to design*: the proposed FSM is based on set theory; so it is very easy to design conditions and operations sets. In addition, it uses a small number of mathematical operators: $\{\in, \wedge\}$. Besides, the designer could easily use all process algebra operators as illustrated in Fig. 3.
- *Fast*: generally, the matching is based on the operators \in and \wedge . First, using the operator \in with an interval can reduce the number of elementary operations. For example, we assume that the temperature ranges is in between 5 and 40 with a sensitivity of 0.5 degree; thus, there is 80 possible elementary comparisons. However, if we use intervals, as suggested in [30], we make only three comparisons. Furthermore, conditions are of the form $\bigwedge_{v=1}^A (a_v \in \check{A}_v)$; as a consequence, when the FSM is doing the matching, it deduces that the condition value is *false* when it finds the first item v where $a_v \in \check{A}_v$ is *false*.
- *Observability of device behavior*: when a change is happening in the home, the FSM starts solving the problem by executing an operations sequence. This constructs operations path. Sometimes, the constructed operations path is not the best one (i.e., optimal path). By comparing the obtained path and the optimal one, we could have an idea about devices that are not operating well. Especially if there are exceptions in the observed operations.
- *Prediction*: by implementing a machine learning, we can predict the behavior of the FSM. When the FSM is in one state, we can assume that the next one and the time when it would be in that state. This criterion makes possible to implement an adaptable FSM to the inhabitant habits.
- *Deadlock*: it is very important to detect if the FSM will facing deadlock situations in the future. The proposed FSM is entirely based on mathematical concepts. Therefore, it is possible to develop algorithms that are able to detect deadlock situations and to solve them.

IV. ALGEBRA OF PROCESSES

Algebra of processes is a formal modeling of process behavior [31]. Several languages are proposed to deal with this algebra such as CSP (*Communicating Sequential Processes*) [32], CCS (*Calculus of Communicating Systems*) [33], ACP (*Algebra of Communicating Processes*) [34] and LOTOS (*Language Of Temporal Ordering Specification*) [35].

The goal of the proposed FSM is to schedule the execution of operations. Thus, the FSM indicates the operations that must be performed in a given situation and the proper execution

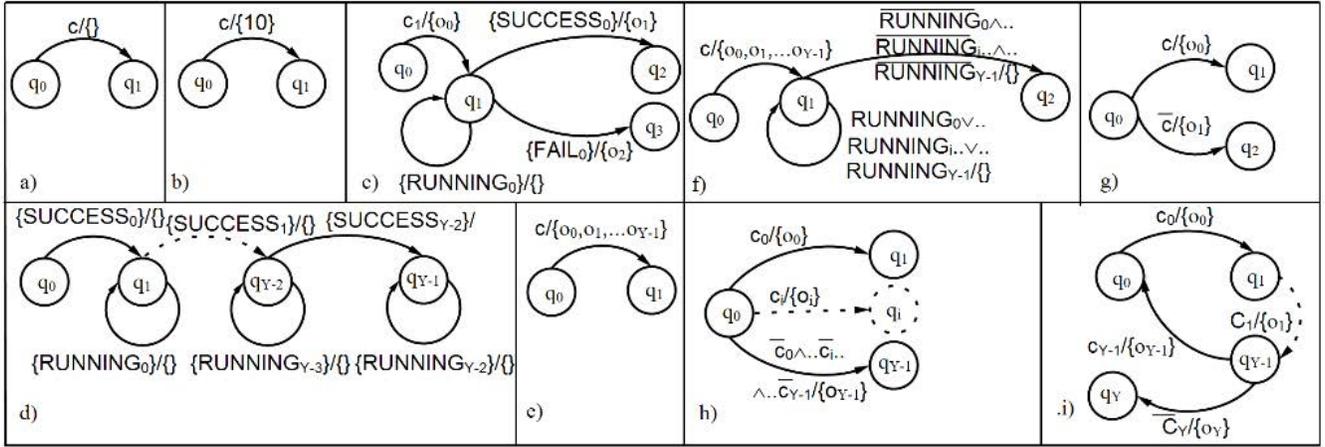


Fig. 3. Set-theory based FSM and process algebra.

order. This section illustrates how to represent the most used process algebra operations by our FSM. These operations are derived from BPEL:

- *Event*: a state variable is able to trigger the FSM only if it is defined as a trigger. In addition, all the possible values are split into subsets; the FSM is only triggered if the value changes the subset. For example, for the variable $V \{\{1,2,3\}, \{4,5\}, \{6,7\}\}$, there are three predefined subsets of possible values $\{\{1, 2, 3\}, \{4, 5\}, \{6, 7\}\}$. The FSM is triggered by V only if the new value of V does not belong to the same subset as the old one.
- *Empty*: in such a case, an empty operations set is put on the transition. As shown by Fig. 3a, when the FSM changes from q_0 to q_1 , it does nothing (the set of all the operations to be done is empty).
- *Waiting*: to allow the FSM to wait during a predefined period, the extended variable T is used as a trigger. Its value is automatically decremented (e.g., every second); it triggers the FSM when the value reaches zero. As shown in Fig. 3b, when the FSM reaches q_1 , it waits 10 seconds before going to the next state.
- *Exception handling*: the main purpose of the proposed FSM is to control domestic devices. When the FSM initiates an operation, we are not sure whether this operation is performed successfully or not. Therefore, the FSM must allow the designer to be aware of the workflow. To do this, each operation must return a state variable. Three values can therefore be distinguished:
 - *RUNNING*: it means that the operation is in progress.
 - *SUCCESS*: it means that the operation is completed successfully.
 - *FAIL*: it means that the operation is terminated unexpectedly with a failure.

As shown in Fig. 3c, in the state q_1 , tests are performed on the state variable s_0 which contains the operation state o_0 . If the value is *RUNNING*, the FSM loops locally without

doing anything (waiting for the operation end); otherwise, if the value is *FAIL*, it goes to the first state of the FSM that handles operations exceptions. Commonly, we can define other state values; for example, a state value for each possible error.

- *Sequence*: for the sequential execution of some actions, we make a states sequence and put an operation in each transition. Only the first transition contains the condition part defined by the designer; the other transitions contain the condition part $\{SUCCESS_i\}$ (this means that the previous operation o_i has been successfully completed). Each intermediate state has a transition that loops to the same state with the condition $\{RUNNING_i\}$ and with an empty operations set. As shown in Fig. 2d, if the condition is satisfied, the FSM starts operation o_0 and goes to q_1 . It loops on this state without doing anything until the state variable of o_0 takes the value *SUCCESS*; then, it starts o_1 and goes to the following state q_2 and waits for the end of operation o_1 . This is repeated for all operations that must be executed sequentially until reaching q_{Y-1} .
- *Concurrency*: to indicate that a set of operations must be carried out concurrently, they are all put in the same set within the operations part of the transition. As illustrated in Fig. 3e, all the operations set $\{q_0, q_1 \dots q_{Y-1}\}$ will be executed in parallel.
- *Synchronization*: the end of all or a subset of operations can be synchronized by adding a local loop. The condition part of the local loop in the transition is in the form $\bigvee_{i=0}^{Y-1} RUNNING_i$ and the operations part is an empty set. This means that the FSM has to wait (without doing anything) until all the operations finish their execution. The FSM leaves the local loop by a transition with the condition $\bigwedge_{i=0}^{Y-1} RUNNING_i$. As illustrated in Fig. 3f, state q_2 will not be reached before the completion of each of the operations $\{o_0, o_1 \dots o_{Y-1}\}$.
- *Conditional (if)*: the *IF-ELSE* statement can be used in

the FSM by two transitions from the same state. The condition part of a transition is the negation of the condition part of the other transition. In such a situation, we are sure that there is only one transition of both will be taken regardless of the context and values of the extended variables. Fig. 3g shows a part of an FSM where the two operations o_0 and o_1 will never be executed in the same circumstance; however, always only one of them will be executed.

- *Conditional (Choice)*: sometimes, we have to choose between several options; in such a case, a transition is defined for each option. To ensure that only one transition is taken into account at a time, the FSM must be deterministic. In addition, to ensure that a transition will always be taken into account, the FSM must be complete. If it is not, a transition is added with a condition part of the form of $\bigvee_{i=0}^{Y-1} \bar{c}_i$, where c_i is the condition part of the transition t_i . In the operations part, all the operations to be executed in the default case are put. In the example of Fig. 3h, operation o_y will be executed if none of the conditions of the set $\{c_0, c_1 \dots c_{y-1}\}$ is satisfied.
- *Loop*: to repeat the same sequence of operations, a cycle composed of one or more states is built. To avoid infinite loops, a transition that leads out of this cycle is added: (i) the transition starts from the first state to avoid entering the cycle, (ii) from the last state of the cycle to force the FSM to enter the cycle at least once, or (iii) from any state of the cycle to allow interrupting the cycle. As it is shown in Fig. 3i, the actions sequence $o_0, o_1 \dots o_{y-1}$ is executed as long as the condition \bar{c}_{y-1} is not verified.

V. CASE STUDY

This section gives an example of FSM for diabetes management in order to illustrate the usability of the proposed FSM. Many factors affect the health status of a person with diabetes. However, for simplicity purposes, only a context attribute, namely the blood glucose level (BG), is considered. To identify the risk levels, BG values are presented in the form of the following ranges (values are given in grams per liter): $[0.0, 0.5[$, $[0.5, 0.8[$, $[0.8, 1.5[$, $[1.5, 3.0[$. In addition, a risk level and a set of actions to perform are associated to each interval. Furthermore, five levels of risk have been defined:

- *Level 0*: for a normal situation; in some cases, the system should inform the caregiver that there is no risk.
- *Level 1*: is used to ask for help or to prevent an emergency.
- *Level 2*: indicates that the situation is manageable by a non specialist.
- *Level 3*: it notifies an emergency requiring the intervention of a health professional.
- *Level 4*: is defined to signal an extreme emergency.

The presented FSM (Fig. 4) uses the two extended state variables R and T . R is used to indicate how many times the home gateway calls the same set of actions under the same conditions. Whereas, T indicates how long the gateway should

wait before moving on to the next transition. Two actions $eat()$ and $alert()$ are used. $eat()$ action is used to inform the diabetic that he should eat something to raise its BG. However, $alert()$ action is used to inform caregivers that there is an abnormal situation. The first argument of the $alert()$ action is the caregiver identifier (all value means all available caregivers related to the person confronted with an abnormal situation); while the second argument is the alert level.

As illustrated by Fig. 4, the FSM is deterministic, all transitions are passable, and it is connected. However, the FSM is not complete. Fortunately, this is not a problem; the missed transition are never reachable.

VI. CONCLUSION

This paper proposed a new methodology for managing home emergencies. The approach extends classical FSM and used set theories. In addition, a set of properties that the FSM must meet has been identified and an algorithm to check each one has been described. We have also illustrated that it is possible and easy to implement the instructions of the process algebra, especially that defined by the BPEL (*Business Process Execution Language*). However, we have presented some usability cases of the FSM without giving algorithms and illustrations. The implementation of these algorithms is not the scope of this work. In future works, each case will be studied in detail. Finally, to allow managing complex situations (several types of emergencies, following many inhabitants or visitors, etc.), other operations such as merging and splitting will be defined.

REFERENCES

- [1] "World population prospects 2017," <http://esa.un.org/unpd/wpp/>, United Nations - Department of Economic and Social Affairs/Population Division, accessed: 2018-01-02.
- [2] "ehealth," <http://www.who.int/ehealth/en/>, World Health Organization, accessed: 2018-06-19.
- [3] C. Garripoli, M. Mercuri, P. Karsmakers, P. J. Soh, G. Crupi, G. A. Vandenbosch, C. Pace, P. Leroux, and D. Schreurs, "Embedded dsp-based telehealth radar system for remote in-door fall detection," *IEEE journal of biomedical and health informatics*, vol. 19, no. 1, pp. 92–101, 2015.
- [4] K. Benson, C. Fracchia, G. Wang, Q. Zhu, S. Almomen, J. Cohn, L. Darcy, D. Hoffman, M. Makai, J. Stamatakis *et al.*, "Scale: Safe community awareness and alerting leveraging the internet of things," *IEEE Communications Magazine*, vol. 53, no. 12, pp. 27–34, 2015.
- [5] A. Alahmadi and B. Soh, "A smart approach towards a mobile e-health monitoring system architecture," in *Research and Innovation in Information Systems (ICRIIS), 2011 International Conference on*. IEEE, 2011, pp. 1–5.
- [6] J. Han, J. Yun, J. Jang, and K.-R. Park, "User-friendly home automation based on 3d virtual world," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, 2010.
- [7] Z. Meng and J. Lu, "A rule-based service customization strategy for smart home context-aware automation," *IEEE Transactions on Mobile Computing*, vol. 15, no. 3, pp. 558–571, 2016.
- [8] A. Tiberkak, T. Lemlouma, A. Belkhir, A. Bouabdallah, and A. Hentout, "A novel approach for generic home emergency management and remote monitoring," *Software: Practice and Experience*, vol. 48, no. 4, pp. 761–774, 2018.
- [9] P. Rajalakshmi *et al.*, "Context aware building energy management system with heterogeneous wireless network architecture," in *Wireless and Mobile Networking Conference (WMNC), 2013 6th Joint IFIP*. IEEE, 2013, pp. 1–8.

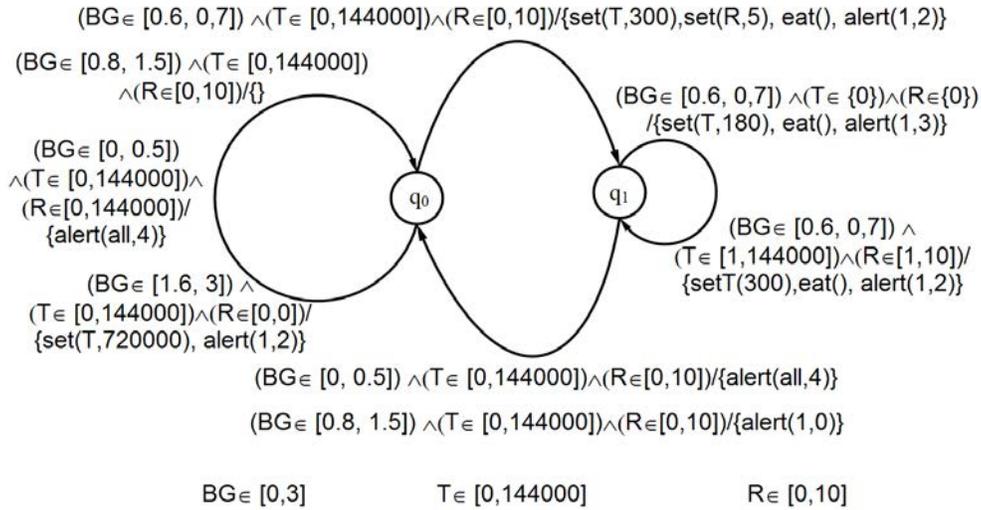


Fig. 4. Part of set theories-based FSM for diabetes management.

- [10] C. Vannucchi, D. R. Cacciagrano, R. Culmone, and L. Mostarda, "Towards a uniform ontology-driven approach for modeling, checking and executing wsans," in *Advanced Information Networking and Applications Workshops (WAINA), 2016 30th International Conference on*. IEEE, 2016, pp. 319–324.
- [11] P. H. Dalsgaard, T. Le Guilly, D. Middelhedde, P. Olsen, T. Pedersen, A. P. Ravn, and A. Skou, "A toolchain for home automation controller development," in *Software Engineering and Advanced Applications (SEAA), 2013 39th EUROMICRO Conference on*. IEEE, 2013, pp. 122–129.
- [12] Z. Yang, A. O. Lim, and Y. Tan, "Event-based home safety problem detection under the cps home safety architecture," in *Consumer Electronics (GCCE), 2013 IEEE 2nd Global Conference on*. IEEE, 2013, pp. 491–495.
- [13] H. Horiuchi, S. Saiki, S. Matsumoto, and M. Nakamura, "Designing and implementing service framework for virtual agents in home network system," in *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2014 15th IEEE/ACIS International Conference on*. IEEE, 2014, pp. 1–6.
- [14] S. Okour, A. Maeder, and J. Basilakis, "An adaptive rule-based approach to classifying activities of daily living," in *Healthcare Informatics (ICHI), 2015 International Conference on*. IEEE, 2015, pp. 404–407.
- [15] T. Teixeira, D. Jung, G. Dublon, and A. Savvides, "Recognizing activities from context and arm pose using finite state machines," in *Distributed Smart Cameras, 2009. ICDSC 2009. Third ACM/IEEE International Conference on*. IEEE, 2009, pp. 1–8.
- [16] W. O. de Moraes and N. Wickstrom, "A lightweight method for detecting sleep-related activities based on load sensing," in *Serious Games and Applications for Health (SeGAH), 2014 IEEE 3rd International Conference on*. IEEE, 2014, pp. 1–7.
- [17] H. Nguyen, M. Ciocarlie, K. Hsiao, and C. C. Kemp, "Ros commander (rosco): Behavior creation for home robots," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 467–474.
- [18] L. Shi, Z. Li, and J. Zhang, "Research on personality behavior in intelligent virtual environment," in *Intelligent Control and Automation (WCICA), 2012 10th World Congress on*. IEEE, 2012, pp. 769–772.
- [19] J. Meyer and S. Boll, "Digital health devices for everyone!" *IEEE Pervasive Computing*, vol. 13, no. 2, pp. 10–13, 2014.
- [20] T. A. Henzinger, "The theory of hybrid automata," in *Verification of Digital and Hybrid Systems*. Springer, 2000, pp. 265–292.
- [21] Y. Wang, "Using process algebra to describe human and software behaviors," *Brain and Mind*, vol. 4, no. 2, pp. 199–213, 2003.
- [22] J. Rosenberg, "Web services business process execution language version 2.0," in *OASIS, Tech. Rep.* oASIS Standard, April 2007.
- [23] S. W. H. Organization, "Temperature and humidity monitoring systems for fixed storage areas." WHO, 2014.
- [24] S. N. Uwaezuoke, "Kangaroo mother care in resource-limited settings: implementation, health benefits, and cost-effectiveness," *Res Rep Neol*, vol. 7, pp. 11–8, 2017.
- [25] C. Hartrick, J. Kovan, and S. S., "The numeric rating scale for clinical pain measurement: a ratio measure?" *Pain Practice*, vol. 3, no. 4, pp. 310–316, 2003.
- [26] S. A., "Standard 55-2010, thermal environmental conditions for human occupancy." Atlanta: American society of heating, refrigerating, and air-conditioning engineers.
- [27] M. G. H., "A method for synthesizing sequential circuits," *The Bell System Technical Journal*, vol. 34, no. 5, 1995.
- [28] <http://www.omg.org/spec/UML/2.2/Superstructure/>.
- [29] K. Androustopoulos, D. Clark, M. Harman, R. M. Hierons, Z. Li, and L. Tratt, "Amorphous slicing of extended finite state machines," *IEEE Transactions on Software Engineering*, vol. 39, no. 7, 2013.
- [30] A. Tiberkak and A. Belkhir, "An architecture for policy-based home automation system (pbhas)," in *Green Technologies Conference, 2010 IEEE*. IEEE, 2010, pp. 1–5.
- [31] M. Tribastone, "Behavioral relations in a process algebra for variants," in *Proceedings of the 18th International Software Product Line Conference-Volume 1*. ACM, 2014, pp. 82–91.
- [32] H. CAR, "Communicating sequential processes," *Communications of ACM*, vol. 21, no. 8, p. 666677, 1978.
- [33] M. Hennessy, "An introduction to a calculus of communicating systems," *Dordrecht: Springer Netherlands*, p. 245262, 1982.
- [34] J. A. Bergstra and J. W. Klop, "Algebra of communicating processes," *CWI Monograph series*, vol. 3, pp. 89–138, 1986.
- [35] T. Bolognesi and E. Brinksma, "Introduction to the iso specification language lotos," *Comput. Netw. Syst.*, vol. 14, no. 1, p. 2559, 1987.