People Monitoring in a Domestic Environment

Ivan Filipe Guerreiro Freire
ivan.freire@ist.utl.pt

Instituto Superior Técnico, Lisboa, Portugal

May 2018

Abstract

A few decades ago it was unthinkable that it would be possible to install systems in a home that would allow the controlling and monitoring of devices contained therein, however, there have been several advances in recent years in the area of Smart Environments and Domotics reason why nowadays it is possible to install cheap systems in our homes that to exactly that. At the same time, several solutions have also emerged that allow the localization and identification of people indoors, as is the case of their homes. A sub-area of Domotics in which several studies have recently been carried out is that of Ambient Assisted Living which aims to enable people with special needs to maintain some autonomy and to be able to live alone in their homes with the necessary security and monitoring without having to place a permanent supervisor to live with them.

This paper describes the work carried out in the scope of the Master's Dissertation in Information and Software Engineering, namely the survey of the state of the art related to the subject of domestic monitoring, location/identification of people and analysis/detection of their activity patterns, and also the presentation of an approach that joins the mentioned areas for the development of a system that allows monitoring a person in their home, knowing the actions performed during the day and detecting anomalous situations related to the existing devices in the house and to the activities performed by the monitored person.

As a result, the system successfully identified all situations related to preconfigured alert states but demonstrated some problems with the automatic alerts due to activities that were difficult to predict or that were never observed during the construction phase of the normal activities patterns. However, these automatic alerts have successfully detected quite a few situations that actually made sense.

Keywords: People Monitoring, People Detection, Alerts Management, Activity Patterns, Domotics, Ambient Assisted Living, Smart Environments

1. Introduction

Population aging is one of the most prominent global trends of the 21st century, especially in richer countries. This demographic change is a direct consequence of two situations: reduction of the birth rate and reduction of the mortality in advanced ages. It is expected that in 2030 the ratio between children and elderly people (number of people over 65 for every 100 children under 15) will already exceed 100 in all developed countries [14]. In this sense, we are faced with more and more elderly people who are left alone in their homes, without any or very little supervision, posing the problem of how to allow people to grow old alone in their homes while having the maximum possible autonomy and that they have the appropriate support to do so with all the necessary security without it being needed to place a permanent supervisor to live with them.

In recent years there have been several advances in areas that can help with the described situation and that allow the development of systems to perform a total monitoring of a house, both the devices contained therein and the people who live there. Of these areas we must highlight the following:

- Intelligent Environments - systems in which the devices of a home are continuously connected and that allows their management remotely or automatically. An intelligent environment is able to acquire and apply knowledge about it and can adapt to its inhabitants in order to improve their experience in said environment. The type of experience desired can range from ensuring the safety of its inhabitants, to reduce the cost of maintaining and using existing devices or automating typical tasks performed daily [10];

- Location/Identification of people indoors - techniques to know in which part of a closed environment are the people located and even identify them. This can be done by means of devices that the people you want to

find/identify must carry with them [16, 8, 7], by using various types of sensors [15, 22] or even through video analysis [11];

- Activity Patterns - evaluation and creation of patterns through the analysis of activities performed by a person in a given context that can be classified and to identify sequences of activities that deviate from the defined standard and which may be considered anomalous [12].

2. Goals

This work aims to simulate a domestic environment in which it is assumed that it is possible to identify the people in a home and knowing its location within it and what actions are performed during the day. It is also possible to know the current state of any existing equipment and changes made to their states. Making use of this information, it is intended to develop a system for monitoring the domestic environment, which allows triggering alerts that, after assessing the seriousness of the situation, could lead to the displacement of emergency teams to the house or alert some supervisor who assess the situation and decide what measures to take. The main focus will be the detection of anomalous situations related to people's normal activity (such as falls, faints, etc.) and the generation of alerts whenever they occur or other dangerous situations (such as gas leaks, open doors, forgotten keys on the door, etc.). To do this, two ways will be implemented to detect these situations:

- Pre-configuration of combinations of critical states of the devices in the home, which should trigger an alert if they remain for a predetermined period of time in that state;
- Detection of anomalies in the patterns of activity of the people being monitored, through pattern detection algorithms.

For the first type mentioned, it is intended to develop a way to configure possible alert situations, the time that the application should wait since the identification of the situation until the launch of the alert and the message to be shown. For the second type, it is intended to develop algorithms that identify patterns of activities taking into account the data received from the actions taken by each person and to identify deviations from those patterns, which may be due to critical situations.

3. State of the Art

There are already some solutions related to the work that is intended to be developed such as systems that allow to identify people and to determine their location within their homes, existing algorithms that allow analyzing patterns and detect deviations and existing home monitoring solutions.

3.1. Location and Identification of People

An important aspect of a home monitoring system is that it is possible to identify at each moment the location of each person within that home and also to identify each person in the event that more than one person coexists in the same space. Over the last few years, several solutions and studies have emerged in this direction and with different approaches. From the existing solutions for locating people indoors, approaches range from the use of triangulation, trilateration and multilateration, to the use of light, ultrasound or radio signals [15] and the use of various types of sensors and video analysis.

The Active Badge [7] system was the first indoor location system, developed in 1990 by the Olivetti Research Laboratory. It is a localization system that makes use of small badges that weigh about 40 grams and that must be carried by the people that it is intended to locate. These devices emit single infrared signals every 15 seconds, that serve as identifiers [15], that are received by device sensors placed on the ceilings of each room. These devices decode the incoming infrared signals and sends them to a central server. Since each division has its sensor and since the signals emitted by the badges rarely propagate to other divisions, the server is able to infer with great probability in which location the signal was received and, consequently, to know the location of the person who owns the badge that issued the signal [16, 24].

The Active Bats [8] system is a tracking system developed by AT&T Cambridge in 1999. This system makes use of small devices with about 35 grams called *bats* that are carried by the people you want to find and that emits ultrasound and radio signals [15]. In this system, several sensors are placed on the ceiling of each room of the house that capture the signals emitted by the devices. There is a central server which has the ID of each existing bat and which periodically sends a radio signal to request that the bat with a given ID sends its ultrasound signal. In this way, only one bat at a time sends its signal, so there are no conflicts between devices. The sensors are placed on the ceiling with a density sufficient enough for at least 3 of these sensors to capture the signal sent and as the location of each sensor is known, multilateration is used to pinpoint the location of the *bat* that sent the signal, knowing which division the person is in and where in that division [16].

IndorAtlas is a locating system that makes use of magnetic fields within enclosed spaces. Modern buildings constructed of reinforced concrete and steal structures have unique, spatially varying magnetic fields that can be used for positioning in the same way that certain animals are guided through

the magnetic field of the earth [18]. In this technology, anomalies in magnetic fields are used for indoor location **??**. This was facilitated by the compass technology existing in modern smartphones and the rapid development of their sensors. This system allows the creation of maps of interior spaces and then make it possible to detect the position of people with a precision between 10 centimeters and 2 meters [17]. Systems such as these may not be the best choice to monitor elderly people in their homes since it is necessary that they never forget to carry the devices.

The Smart Floors project was a system developed by R. J. Orr and G. D. Abowd under the Georgia Tech [22] Aware Homes project. They created and validated a biometric identification system based on the footstep profile of a person. The reaction of a measuring device in response to the weight and inertia applied by a body is called GRF (ground reaction force) and it is this measurement that allows the definition of the profile of a person's footstep. To do this measurement, tiles were used on the floor that have a load cell underneath. Once the tiles have been installed and all users' profiles have been saved, the data captured by the sensors are analyzed by the system and compared to the saved footstep profiles, allowing the identification of the person in question and their location. It is necessary to fill all the floors of a house with these tiles so that there are no "dead angles", which can be very expensive and require a remodeling of the house where it is intended to install this system. The objective was to show that for a small group of people (up to about 15) the profile of the footstep is different enough that the system can verify the identity of the user with a good precision. The project demonstrated a precision of 93% in identifying people based on their footstep profile, regardless of the type of footwear used.

Another alternative that is less invasive to a house and does not require people to carry any equipment is based on image processing. A study was carried out by E. Corvee, S. Bak and F. Bremond from INRIA [11] using texture descriptors based on Local Binary Patterns to detect people in video, locate their bodies and heads and identify them taking into account clothing used to create a body signature or by facial recognition. The clothing identification did not have much success if people did not wear similar clothing. On the other hand, the facial identification layer obtained a correct identification of people above 98%, for a database with 40 people, using 10 images per person.

3.2. Activity Patterns

There are already some works aimed at recognizing patterns of activities and detecting changes to the patterns created. The successful research so far has focused on recognizing simple human activities, and more complex activities remain the most challenging and the new focus of this area of research.

Several algorithms based on probability have been used to create these models beeing the most popular the Hidden Markov Model [12]. Simple activities are modulated as Markov Chains. But the more complex activities are more complicated to model in this way but by observing the signals generated by these complex tasks it is possible to construct models indirectly, called Hidden Markov Models. The main objective of this model is to construct hidden state sequences based on the information received by sensors, and it is possible to associate the most probable hidden sequences.

A variant of this method was tested under the Ph.D. of A. R. M. Forkan [13] in order to detect anomalous activities in a domestic environment. Observations were recorded regarding activities carried out by an inhabitant of a house. The high-level descriptive of each action was saved along with the start and end date of the actions. Examples of high-level activities are "sleeping", "eating", "using the toilet", etc. Activities were divided by date to isolate sequences of daily activities. From these data, 1st order Markov chains are created that represent sequences of activities. With the activity sequences observed for several days, the transition probabilities between activities are calculated. This model allows teh calculation of the probability of a new sequence of activities but does not allow the validation of weather the new sequence is anomalous or not. To solve this problem, a hidden Markov model was used. In this case, the states of the model are not activities but hidden states and observed activities are called emissions. The probabilities of emission are calculated, that is, the probability of each activity being associated with each hidden state. The approach used was to use the hidden states "Abnormal Activity", "Normal Activity" and "Critical Activity" and to train the model using anomalous and normal sequences. In this way, it is possible to identify sequences of activities that deviate from the normal pattern and the respective activities that have made the sequence anomalous. The model was tested with 35 observation sequences from 10 different users and obtained an accuracy rate between 87.5% and 97.5%, with a false positive number between 0 and 4.

An alternative is the Emerging Patterns method [12]. An emerging pattern is a vector of features that describe significant differences between two data classes. Having a multi-instance dataset, each of which has a set of attributes and corresponding values, and within that set of attributes there are some that can be associated with one class of activ-

ity more correctly than others. For example, a vector {location@kitchen, object@stove} is an emerging pattern for the "Cooking" activity.

Another system to detect anomalous activities based on sensors was presented by J. Yin, Q. Yang and J. Pan [25]. In order to present their proposal, they were based on the intuition that activity-based anomaly detection techniques compare the profile of all normal activities and that any deviation is marked as a potential alarm and that, in this sense, these positive models of activities identify normal activities and assume that everything else is an anomaly, and the detection of abnormal activities has the potential to identify activities that occur rarely or are not known. The presented proposal concerns a two-stage process that combines a one class support vector machine (SVM) to filter normal activities and to reduce the false negative ratio with a collection of abnormal activity detection classifiers to reduce the false positive ratio. Through a collection of sequences of actions, the proposed algorithm acts in two phases. In the first phase of the process, all sequences are sent to a module to extract their characteristics, passing through a class support vector machine to eliminate sequences that are most likely to be normal. The remaining sequences proceed to the second step of the process in which it determines whether the sequence is in fact anomalous. To test the system, 3 sensors were placed in different parts of a person's body, each sensor using light, temperature, sound, acceleration and magnetism, and captured 431 sequences of normal activity inside a home. To capture anomalous sequences, the person was asked to simulate various activities of this type such as a fall or a faint. The proposal demonstrated a detection accuracy of 90 % and a false alarm rate of only 7%.

3.3. People Monitoring Solutions

There are already a large number of studies and proposals related to the monitoring of people in their homes, with the main focus being on the elderly people.

A study carried out in this context was done by the University of rebro, where a system called GiraffPlus was developed [23], where several components are integrated to enable monitoring. Several sensors are placed in the house which are connected to the application through a wireless connection. Motion sensors, pressure sensors, electrical sensors, switching sensors, smoke sensors, flood sensors and others are used. With the data generated by these sensors it is possible to recognize the context of the activities. One of the characteristics of the system is that it allows setting alarm conditions and notify supervisors whenever these conditions are detected. To alert the supervisors, a mobile application was created where notifications are received, which can have different levels of urgency. Since this system is connected to the Internet, a component has been developed that verifies if it has not sent information for a long time so that the technical team is alerted that there may be some problem with the application. To evaluate this system, sensors were installed in a Swedish home of an 82 years old man that spends most of his time at home. After collecting the data, it is possible to make queries to the system on specific activities to understand his usage patterns. In conjunction with the supervisor, an system may be constructed using the Allen's Interval Algebra language to make specific inquiries about activities or sequences of activities. In order to validate the obtained data, the results were discussed with the person monitored to verify if they corresponded to the truth.

Another proposal was submitted by M. Manca, P. Parvin, F. Patern and C. Santoro from the HIIS laboratory in Italy [19] to detect irregular behavior due to significant changes in their behavior in a given scenario. The solution exploits task template specifications in which the expected behavior of a user is represented. By making associations between elementary tasks defined and the events that are received, it is possible to detect deviations and trigger the necessary actions. To create the task template, the supervisor helps describe planned activities for the monitored person. These models only contemplate activities relevant for automatic analysis and only those that can be detected by sensors. After the model is built, the tasks contained in it are associated with captured events. Next, the events occurring in the context in which the elderly person lives on are recorded, keeping the timestamp in which they occur for future analysis by the deviation analysis module, that compares them to the expected behavior. This comparison consists of verifying if the sequence of elementary tasks satisfies what was defined in the task model, being considered a deviation otherwise. Example of types of diversion that this module identifies are: too long activities, too short activities, too much frequency of the same activity or missing tasks. The actions that should be triggered when a deviation is detected are also defined. The deviations have associated actions, which may be alarms, sending messages, changing the state of a device, etc. A web application was developed so that supervisor can monitor the daily life of the elderly and receive notifications when deviations are detected.

A laboratory (called GERHOME) equipped with several sensors was built in order to evaluate a monitoring system and to explore the daily activities performed by a person [20]. This approach consists of collecting sensor data about the activity of an el-

derly person in his home and drawing a normal profile of his daily activities. Any significant deviation is considered alarming and an alert for a supervisor is triggered. This monitoring system receives 3 types of input data: video acquired by a cam, data collected from various sensors and a priori knowledge of the environment containing a person's 3D model, event models and a 3D model of the geometric information of the monitored space. A set of visual algorithms is used to detect and track a person within the monitored space. One of the algorithms segments moving pixels into a binary image by subtracting the currently captured image with a reference image (this reference image is updated over time to reflect changes made to space). When a person is detected in the video, a posture recognition algorithm detects in real time a set of human postures through the use of a 3D model of a person.

4. Solution Architecture

The proposal presented next is for an application that allows monitoring activities performed by a person in their daily tasks within a smart house connected to a Domobus [21] system. For this work, a Web application was chosenfor the advantage of being accessible by any search engine and allowing it to be hosted on the Internet so that supervisors can access the monitoring application remotely. In addition, in this way, it can complement other applications that allow manipulation of the devices in the house, which would allow for the supervisor, uppon receiving an alert related to some state of a device, to manipulate that device without having to go to the house, avoiding possible serious accidents.

4.1. Specifications and Technologies

To implement the monitoring application the ASP.NET MVC framework 4 [1] in C# was used which uses the MVC design pattern that allows the separation of the application data models from all the logic concerning the application's functionality, and from its layout. The data is presented through tables on the screen using the JQuery DataTables [5] plug-in to create them. In order to make the application responsive so that it can run on different platforms, the Bootstrap [3] framework was also used in the *front end* component that is developed in HTML, CSS and JS. A Microsoft SQL Server [4] database was used to store all the required application data (Domobus data, history of performed actions and alerts). An application that allows the upload of files to the server was also developed in ASP.NET. In addition, this same application is responsible for handling the XML specification file data of the DomoConfig [21] system, uploading it to the database and handling files with action. For the purpose of communication between the applications, the message broker Rabbit MQ [6] was used to send messages to a queue, which can be consumed by other applications that have access to it. This consumption is done in FIFO mode so the order of the sequence of actions is not affected. It also uses an acknowledgment system (ACK) that ensures that the message is received/consumed at least once. This message broker runs in a virtual machine that is allocated on the server. The mensioned components were allocated on the Internet through the Azure [2] application service.

4.2. System Structure

The system has several components and each is responsible for a specific part of the application. The system is composed by a database, a management application, a file directory, a monitoring application and a message broker. The structure of each component and the communication between them can be observed in the figure 1.

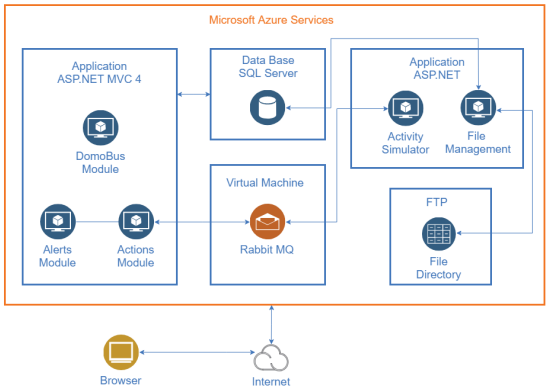

Figure 1: Application structure

To accommodate all the data that is used and presented by the monitoring application a database was used. Several tables have been created to save the structure specified in a Domobus system [21] as well as tables to store a history of the monitored people's activities and the alerts that are generated by the system. In order to speed up the access to the data, 3 data visualizations (*views*) were created, where the data resulting from the linking of various tables is presented.

4.3. Management Application

An auxiliary application has been developed that is used to load data into the system, parse and analyze that data, and load it into appropriate data structures, making it available to the other system components. File management is responsible for receiving incoming files and act differently depending on which type of file is loaded (TXT or XML).

If a XML file is imported, it is validated to check that the document has a valid XML structure and

contains the expected TAGs of a Domobus system (divisions, users, etc.), returning an error repor if it is verified that the file was not valid or that some critical specification of the system was missing. After the necessary validations have been made, the system tables are cleared and the data is processed, and the values for each specified section are stored in the respective tables in the database. To enable a pre-configuration of combinations of device states that are considered critical, an extension to the existing XML specification for the Domobus system was created. The structure of this extension is shown in the figure 2.

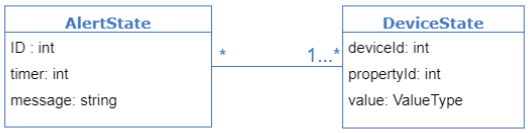

Figure 2: Extension model for the DomoBus XML to configure alerts

With this extension, it is intended to be able to configure several alert states with the time that the application should wait from the moment the alert state is detected until the alert is actually generated and the message that should be displayed. Each alert state must have at least one equipment state and which pair of property and value should trigger the alert state. An example of the alert state specification to be placed in the DomoBus XML file is shown in Figure 3.

```
<DomoBusSystem>
  ...
  <AlertStateList>
    <!-- Door closed with key on -->
    <AlertState ID="1" Timer="2" Message="Chave deixada na porta!">
      <DeviceState RefDevice="6" RefProperty="1" Value="0" />
      <DeviceState RefDevice="6" RefProperty="2" Value="1" />
    </AlertState>
    <!-- Stove hob gas on with no flame -->
    <AlertState ID="2" Timer="4" Message="Gas ligado sem chama!">
      <DeviceState RefDevice="7" RefProperty="1" Value="1" />
      <DeviceState RefDevice="7" RefProperty="2" Value="0" />
    </AlertState>
    <!-- Door Open -->
    <AlertState ID="6" Timer="5" Message="Porta aberta!">
      <DeviceState RefDevice="6" RefProperty="1" Value="1" />
    </AlertState>
  </AlertStateList>
</DomoBusSystem>
```

Figure 3: Example of XML alert specification

If a TXT file is received, it must be validated to verify that the document has the expected structure (correct lines for specifying a move action for a division or use of an equipment), returning an error report if the file was not valid. This file represents a sequence of actions and, for this file type to be valid the following structural rules must be followed:

- Action that represents a person's movement:

  GO|PERSON-1|DIVISION-1|TIMESTAMP-20/10/2016 00:19

- Action that represents the use of a device by a person:

  USE|PERSON-1|DEVICE-6|PROPERTY-1|STATUS-0|TIMESTAMP-20/10/2016 00:20

After the necessary validations are made, each line is sent (in the order it is contained in the file) to a message queue using the message broker, which can be consumed by the monitoring application's action module. To create these action sequence files, real activity data sets from people in a house was used, which are available under the CASAS [9] project. This data was treated so it has the simplified structure used by this system and which was previously referred to.

4.4. Monitoring Application

The monitoring application is where all information related to the system will be presented and has a graphical layout that is presented in the figure 4, which is divided into four distinct parts that allow monitoring different factors of the system: devices section (lists all the devices of a home and the current state), people section (lists the various people registered in the system and the place where they are located), actions section (lists all the actions performed by the residents) and an alert section (lists all the generated alerts). Three buttons were placed at the top of the screen:

- "History Mode" Button - this button enables or disables the history mode. If the mode is enabled, all actions and alerts in the database are displayed. When history is disabled only actions and alerts that have never been cleared by the buttons described below are displayed. When History mode is enabled, the " Clear Actions " and " Clear Alerts " buttons are disabled;

- "Clear Actions" Button - this button erases the actions that are displayed on the screen and marks them as viewed;

- "Clear Alerts Button" - this button erases the alerts that are displayed on the screen and marks them as viewed;

The application was divided into 3 modules, each responsible for a specific part of the application and associated with one or more sections (the modules were created using the ASP.NET MVC 4 "Area" component) and that uses the MVC design pattern: domoBus data module (devices and people sections), action module (actions section) and alerts module (alerts section).

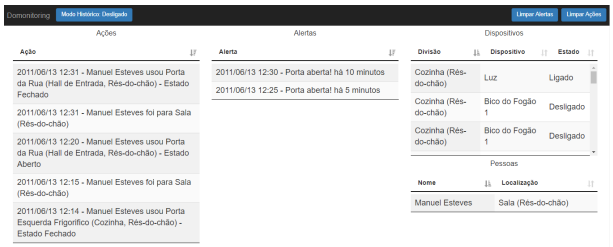

Figure 4: Graphical layout of the monitoring application

The DomoBus data module is responsible for displaying all the information regarding the devices and people in the house. Its model consists of three classes: DomobusData (class that contains all the information that is presented in the device and people sections), Device (class that contains information about each device) and Person (class containing information about each person). The part referring to what is presented on the screen is in the module's view that is implemented in .CSHTML format. The information contained in the class DomobusData is read by the view, creating a table for each section. For the devices section, a table with three columns (Division, Device and State) is created and the list of devices is traversed and the data of each element is placed in the table. There may be multiple entries for the same device if it has more than one type of state - for example, the front door may have the "Open" and "Closed" status but also the "With key" and "No key" which are isolated from one another (the door can be open with a key on, locked without a key, etc.). For the people section, a table with two columns (Name, Location) is created and the list of people is traversed and the data of each element is placed in the table. All the logic associated with the Domobus data module is found in the controller that runs whenever the application is started. When it is executed, the controller creates an object of the class DomobusData, a list of objects of class Device, a list of objects of class Person and makes two queries to the database in order to receive the necessary data to feed the created objects that will be used by the view to display all the necessary data for the two sections associated with this module. The first query is made to the view (in the database) that contains the information about the states of the existing devices, saving the resulting data of each entry in an object of class Device, adding it to the list of objects of the same type. After loading all existing data, the list of devices is stored in the DomobusData object. The second query is made to the view containing information about existing people, saving the resulting data of each entry in an object of the Person class, adding it to a list of objects of that same type. After loading all existing data, the device list is stored in the DomobusData object, which will be used by the view of this module to display the data.

The Actions module is responsible for presenting all the information regarding the actions performed by the inhabitants. The model of the action module consists of two classes: ActionData (class that contains all the information that is presented in the actions section) and Action (class containing information about each action). The part referring to what is presented on the screen is in the module's view. The information contained in the ActionData class is read by the view, creating a table in the actions section. The table created is composed of a column (Action) and the list of actions is traversed and the data of each element is placed in the table with a specific formatting depending on the type of action. All logic associated with the action module is found in the controller that is executed whenever the application is started and whenever the system receives an action that has been performed by one of the inhabitants. When executed, the controller creates an ActionData class object and a list of Action class objects. The messages queue is then queried to see if there are any pending actions to evaluate. For each message in this queue, an analysis is made of the information contained in it and the data is processed in order to update the database with the new information received. A step is performed to parse each message and storing an action in teh database with the data contained in it. The second step consist in updating the database devices table (if it is an USE action) or the users table (if it is a GO action). The third step made for each action consumed is the insertion of that action into the database action table indicating that the message has not yet been reported. In the fourth step, the system executes the process from the alerts module that is responsible for checking for alerts to be triggered. This process will be detailed in the section of the alerts module. Finally, as a last step, a query is made to the view (in the database) that contains information about the actions performed. If the history mode is enabled, only the resulting actions that are identified as not reported will be handled, otherwise all the resulting actions will be handled. For each action, the respective data is stored in an Action class object, adding it to the list of objects of that type. After loading all existing data, the action list is stored in the ActionData class object, which will be used by the view of this module to display the data. The action module controller is executed whenever there are messages in the message broker queue, and the alert checks are performed taking into account the current time in which it was executed.

In order to be able to evaluate a day of activi-

ties without actions being uploaded to the system at the actual time they are executed, which would happen if the system were effectively connected to a house as expected, a one-day simulation process was created. To do this, an action file is loaded into the management application containing a complete sequence of actions for a whole day of activities, which are sent, in order, to the message broker's message queue. When the action module detects that messages exist in the queue, they are consumed and stored in an ordered list of actions. The system evaluates the 1st action contained in this list and fills the variables corresponding to the beginning and end of the activity day, taking into account its timestamp. After this initial step, a cycle is started which will cycle through all the time stamps between the start date and the end date of the activity day. At each iteration of this cycle, if the time stamp of the evaluated action is equal to the time stamp corresponding to the current iteration, the logic contained in the controller of the action module corresponding to steps 1, 2, 3 and 4 are executed. At the end of the iteration, the action is removed from the list and the next action is consumed, executing the next iteration of the cycle until the end date of the activity day is reached. When the cycle is complete, the last step of the controller of the actions module is executed, making all the data available for the view.

The alerts module is responsible for verifying and generating alerts that may exist and for presenting all the information regarding the alerts that are generated. The alert module model consists of two classes: AlertData (class that contains all the information that is displayed in the alerts section) and Alert (class containing information about each alert). The part referring to what is presented on the screen is in the module view that is implemented in .CSHTML format that has access to the model of this module. The information contained in the AlertData class is read by the view and a table si created in the alerts section. The table created is composed of a column (Alert) and the list of alerts contained in the AlertData object is traversed and the data of each element is placed in the table. All the logic related to alerts is in this controller. When executed, the controller creates an AlertData class object, a list of Action class objects, and a list of alert state identifiers. A method was created that checks for pre-defined alerts that must be generated. This method checks for devices that are on alert status and, if so, stores the identifiers of the new alert states identified. If any of the alert states contained in the list are no longer present, they are removed from the list. This method can be executed on request by the stock module controller, when analyzing actions received. As long as there are alert states in this list, the alert system will periodically check them and whenever the timer in one of the states has been reached and alert is triggered. A method was also created to check if there are other types of alerts generated from the analysis of algorithms, which will be detailed further. This method is only executed on demand by the stock module driver. The last step is to query the database for information about the generated alerts. If the history mode is enabled, only the resulting alerts that are identified as unreported are handled, otherwise all the resulting alerts are handled. For each alert handled, the respective data is stored in an object of the Alert class, adding it to the list of objects of that type. After loading all existing data, the list of alerts is stored in the AlertData class object, which will be used by the view of this module to display the data.

5. Automatic Alerts

In order to detect anomalous situations that are not included in the defined alert states of the Domobus system specification, methods were used to analyze the actions received by the system and decide if the action corresponds to a situation that is not in accordance with the normal activity patterns observed in the system. In order for the system to have data to carry out the necessary analysis, sequences of a person's real daily activities were retrieved from data that is available through the CASAS [9] project, corresponding to a period of 2 months, which were stored in the database to be available for consultation. Each saved action has information about the action taken before it. Whenever the time difference between two sequential actions is greater than 15 minutes, a replica of the previous action is saved for each 15 minute period between these two actions, simulating that the person remained in that location during that time.

The first method that was used was to determine the probability of an inhabitant beeing in a certain division at each hour of the day and to issue an alert when a move action is performed for the less probable division. For this method, a table was created which the probabilities of an inhabitant beeing in a certain location every hour of the day. In order to fill in the data, the analysis actions were divided into 24 groups (one group for each hour t of the day). The probability of a new action x related to one of the N divisions in the housing (i) was estimated to belong to each group. An action on a division executed at a given time is represented by x_t^i and its probability is presented in the formula 1.

$$\Pr(x_t^i) = \sum x_t^i \sum x_t, \sum_i^N \Pr(x_t^i) = 1 \qquad (1)$$

When the system receives a new action, it is for-

warded to the alerts module to be evaluated and if this is the least likely to happen at the time it was executed then an alert is issued. This solution is not effective for divisions whose actions can not be predicted because the time of execution varies a lot, as is the case of the use of the toilet. Using the room at times when it is less likely can help indicate deviations in sleep patterns.

Another method was used do determine when a person stays in teh same location for a period of time. One way to evaluate action sequences is through a Markov model. In order to construct this model. All transitions between divisions and their respective timestamp are saved. With this data it is possible to analyze all the transitions that occurred in a certain period of the day. It is intended with this method to detect sequences of displacement activities that correspond to the permanence in the same division. For this method all transitions a that occurred in a period of time t of a departure division i to an arrival division j were grouped. The transition between two of the N divisions in a given time period is represented as $a_t^i j$. The probability that in an action x there is a transition from the departure division x^i to the arrival division x^j is calculated as indicated in the formula 2 and the sum of these probabilities must be 100%.

$$\Pr(a_t^{x^i x^j}) = \frac{\sum a_t^{x^i x^j}}{\sum_j^N a_t^{x^i j}} \quad e \quad \sum_j^N \Pr(a_t^{x^i j}) = 1 \quad (2)$$

The analysis of sequences of actions is carried out periodically by the system and, if no new action has been received, the system assumes a replica of the previous movement action and evaluates the sequence that contains a transition from the division in which the inhabitant is to it self. For each sequence of actions analyzed, a Markov chain (exemplified in figure 5) is created for the perod corresponded to the actions sequence.

For this solution the system only evaluates transitions from a division to itself and calculates the probability of it. If this transition is the least probable or if there is no record of this transition already occurred in the period in which the action was observed then the system issues an alert. This method allows the identification situations such as falls or faints, not throwing alert when the permanence was expected to happen, as an example are examples: staying in the room during sleep hours or staying in the kitchen at meal times.

Again, the problem arises with actions with varied hours of execution such as the use of the toilet. Also it does not allow to detect falls or faints occurred in divisions where the person was expected to spend some time. For example, if there is a fall

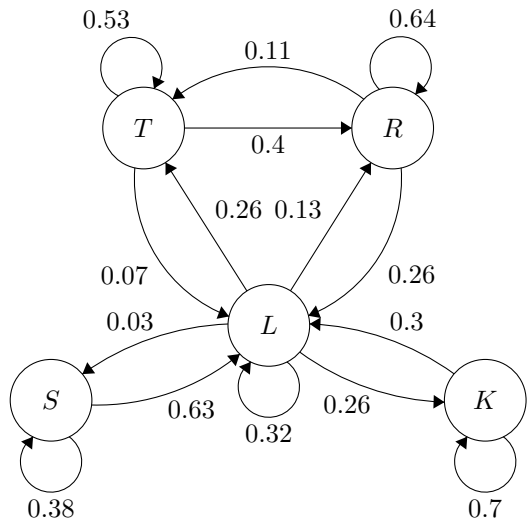

Figure 5: Markov chain for a sequence of actions at certain hour. T - Toilet, R - Room, K - Kitchen, L - Living Room, S - Street

during the preparation of a meal at lunchtime or if the person gets up during the night time to go to the toilet and fall in the room the system will not consider it as an anomalous situation.

6. Conclusions

This paper presented a proposal for a solution that allows monitoring people in domestic environments so that they can live in them with the maximum of autonomy possible and safely so that the supervisor can be at rest and can remotely monitor the activities performed inside the housing. A brief introduction was made to the current context of population aging, which served as a motivation for this work.

Some approaches have been described to automatically detect activities that are considered anomalous and that also trigger alerts when they are detected and the downpoints of such approaches were indicated.

The area of monitoring smart environments is still growing and there are improvements to do in various methods. Interesting works that can be developed are as follows:

- How to isolate activities that are performed at random hours of the day and that are therefore difficult for an anomaly detection system to distinguish them from activities that are effectively at risk, as is the case of toillet usage at times that were not observed in the learning phase of the system;

- Having a system that despite having a learning phase, continues to update itself when activities that generate alerts are detected but are in fact not critical activities but new activities

that have never been observed but can be considered normal.

References

[1] Asp.net mvc 4. https://docs.microsoft.com/en-us/ aspnet/mvc/mvc4/. Acedido: 12/01/2018.

[2] Azure. https://azure.microsoft.com/pt-pt/. Acedido: 12/01/2018.

[3] Bootstrap. https://getbootstrap.com/. Acedido: 12/01/2018.

[4] Microsoft sql server. https://www.microsoft.com/pt-pt/sql-server/. Acedido: 13/02/2018.

[5] Plu-in jquery - data tables. https://datatables.net/. Acedido: 09/04/2018.

[6] Rabbit mq. https://www.rabbitmq.com/. Acedido: 02/03/2018.

[7] the active badge. http://www.cl.cam.ac.uk/research/dtg/ attarchive/ab.html. Acedido: 12/01/2018.

[8] the active bat. http://www.cl.cam.ac.uk/research/dtg/ attarchive/bat/. Acedido: 12/01/2018.

[9] D. Cook, A. Crandall, B. Thomas, and N. Krishnan. Casas: A smart home in a box. *IEEE Computer, 46(7):62-69 - http://ailab.wsu.edu/casas/hh/*, 2013.

[10] D. Cook and S. Das. Smart environments: Technology protocols and applications, pp. 3. New York:Wiley-Interscience, 2004.

[11] E. Corvee, S. Bak, and F. Bremond. People detection and re-identification for multi surveillance cameras. *Proc. VISAPPInt. Conf. Computer Vision Theory and Applications*, Sophia Antipolis, France, Fevereiro, 2012.

[12] S. H. Eunju Kim and D. Cook. Human activity recognition and pattern discovery. *IEEE Pervasive Computing, VOl. 9, No.1, pp.48-53*, 2010.

[13] A. R. M. Forkan. A cloud-based, predictive and context-aware system for ambient assisted living. RMIT University, Melbourne, Australia, Junho, 2016.

[14] L. A. Gavrilov and P. Heuveline. Aging of population. In *P. Demeny and G. McNicoll, The Encyclopedia of Population.* Macmillan Publisher, EUA, 2003.

[15] H. Koyuncu and S. H. Yang. A survey of indoor positioning and object locating systems. *International Journal of Computer Science and Network Security, Vol. 10, No.5, pp.121-128*, Maio 2010.

[16] A. LaMarca and E. de Lara. Location systems: An introduction to the technology behind location awareness. *Synthesis Lectures on Mobile and Pervasive Computing Vol. 3, No. 1, pp.91-102*, Morgan and Claypool Publishers, 2008.

[17] J. Leber. Startup uses a smartphone compass to track people indoors. *https://www.technologyreview.com/s/428494/ startup-uses-a-smartphone-compass-to-track-people-indoors/*, Julho, 2012.

[18] I. Ltd. Ambient magnetic field-based indoor location technology. *Bringing the compass to the next level - http://web.indooratlas.com/web/WhitePaper.pdf*, Julho, 2012.

[19] M. Manca, P. Parvin, F. Patern, and C. Santoro. Detecting anomalous elderly behaviour in ambient assisted living, cnr-isti, hiis laboratory. Pisa, Itlia.

[20] Z. N, F. BREMOND, M. THONNAT, A. ANFOSSO, . PASCUAL, P. MALLEA, V. MAILLAND, and O. GUERIN. Assessing computer systems for monitoring elderly people living at home, inria, cstb, chu. Frana.

[21] R. J. C. Nunes. Modelo de especificao e programao de um sistema domtico. *Instituto Superior Tcnico / INESC-ID*, Lisboa, Portugal.

[22] R. J. Orr and G. D. Abowd. The smart floor: A mechanism for natural user identification and tracking. *Extended Abstracts on Human Factors in Computing Systems, pp.275-276*, ACM, EUA, 200.

[23] J. Ullberg., A. Loutfi, and F. Pecora. A customizable approach for monitoring activities of elderly users in their homes. *Mazzeo, P.L., Spagnolo, P., Moeslund, T.B. (eds.) AMMDS 2014. LNCS, vol. 8703, pp. 1325. Springer, Heidelberg*, Springer, Heidelberg, 2014.

[24] R. Want, A. Hopper, V. Falcao, and J. Gibbons. The active badge location system. *ACM Transactions on Information systems Vol. 40, No. 1, pp.91-102*, Janeiro 1992.

[25] J. Yin, Q. Yang, and J. Pan. Sensor-based abnormal human-activity detection, ieee trans. knowl. data eng., vol. 20, no. 8, pp. 1082-1090. 2008.