Using Case-Based Reasoning to Detect Risk Scenarios of Elder People Living Alone at Home

Eduardo Lupiani[1], Jose M. Juarez[1], Jose Palma[1],
Christian Sauer[2], and Thomas Roth-Berghofer[2]

[1] University of Murcia, Spain
{elupiani,jmjuarez,jtpalma}@um.es
[2] The University of West London, UK
{thomas.roth-berghofer,christian.sauer}@uwl.ac.uk

Abstract. In today's ageing societies, the proportion of elder people living alone in their own homes is dramatically increasing. Smart homes provide the appropriate environment for keeping them independent and, therefore, enhancing their quality of life. One of the most important requirements of these systems is that they have to provide a pervasive environment without disrupting elder people's daily activities. The present paper introduces a CBR agent used within a commercial Smart Home system, designed for detecting domestic accidents that may lead to serious complications if the elderly resident is not attended quickly. The approach is based on cases composed of event sequences. Each event sequence represents the different locations visited by the resident during his/her daily activities. Using this approach, the system can decide whether the current sequence represent an unsafe scenario or not. It does so by comparing the current sequence with previously stored sequences. Several experiments have been conducted with different CBR agent configurations in order to test this approach. Results from these experiments show that the proposed approach is able to detect unsafe scenarios.

1 Introduction

According to the World Health Organization and the US National Institute of Ageing/Health [24], industrialized countries are facing the problem that the population's average age is drastically increasing. One of the side effects of an increased aged population is the rising number of elder people living alone at home. Information Technology and Artificial Intelligence (AI) may play a relevant role in looking after people living alone at home, as well as in providing care assistance. Examples of this key role are some policies such as the Ambient Assisted Living initiative promoted by the European Union. In all of these initiatives, Smart Homes are encouraged as a tool to detect unsafe scenarios at home [5], as for instance falls, which are one of the major causes of serious accidents for the elder people living alone.

Smart Home systems are usually based on agent architectures [8], where each agent is responsible of one particular task, such as the control of the home environment [18], the assistance of home inhabitants [12,13,23] or monitoring of residents' health status [6]. In order to fulfill their purpose, Smart Homes require a set of sensors to gather home data and deliver them to agents. To this end, different types of sensors have been considered, ranging from intrusive devices such as cameras or wearable sensors [7,23], to pervasive approaches such as movement or pressures devices connected by wireless sensor networks [2,20]. Once data is collected and processed, AI techniques can carry out some inference processes in order to interpret the scenario based on the processed data.

The use of Case-Based Reasoning (CBR) as a reasoning agent in Smart Home systems has several advantages [8,14]: First, the learning process is implicit in the CBR cycle so CBR agents can learn from concrete situations as time goes by, making it possible for the CBR agents to adapt themselves to the resident's specific needs. Second, the system response time can be reduced because CBR avoids resolving already solved problems, which may involve a great amount of information and computation in a Smart Home environment. Third, an expert can define personalised cases to represent a customised problem and its solutions. Finally, as CBR systems use similar past solved cases to solve a current problem, these cases can be used to provide explanations on why a concrete solution is proposed. [9,21].

Some authors have already proposed CBR agents to solve problems in Smart Homes [3,14]. However, to our understanding, little attention has been paid to the temporal dimension in the development of these CBR systems, since the use of the time dimension is limited to determine the context of the case [15].

In this work, we propose a CBR agent able to detect potential unsafe scenarios in a Smart Home, as for instance falls, using a spatial-temporal approach. The agent is based on the retrieval of previous cases which represent the different locations visited by the elder resident during one of his/her daily activities. These cases are represented by event sequences where each event consists of a location and a time-stamp. The proposed agent has been integrated in the $proDIA$ monitoring system [4]. This system consists of a wireless sensor network which uses pervasive sensors, such as motion detection infra-red sensors, pressure sensors (located in bed, chairs, sofas, etc) and magnetic sensors to detect door opening and closing. A prototype of this system has been placed in 100 houses in the province of Murcia, Spain. Furthermore, the CBR agent has been implemented using *myCBR* software development kit [22].

The remainder of this paper is organized as follows. In section 2 we review the background of this work. In section 3 we describe in detail the proposed system to detect unsafe scenarios with a CBR agent. Section 4 describes our experiments which we performed using a synthetic case-base with cases that represent daily activities at home. Finally, in section 5 we present our conclusions and describe planned future work.

2 Related Work

CBR has been used in Smart Home systems in different approaches and with different purposes. In [25], a CBR system is proposed as a decision support system to place the sensors in a Smart Home. The decision is done according to the resident's physical disabilities, such as their cognitive abilities, mobility, dexterity and other personal details. However, the system does not make use of the time dimension to reflect the change in a resident's physical state.

A CBR architecture for Smart Home in order to enhance the inhabitants comfort at home is proposed in [15]. The cases in this approach are representations of what actions are occurring at home and how the Smart Home should react to them to enhance the comfort at home, as for instance lowering the AC temperature or adjusting the light brightness in a room. The case structure is a frame with slots for representing the user information, data gathered from the sensors, and a time-stamp to represent when the observation of the house was done.

In [14], the authors introduced how a CBR system may be used in order to detect problems at a home, so as to propose actions to amend them, however, most of the description are not related with the spatial-temporal representation of the actions taken at home. According to this work, the success of the system relies on the quality of the retained cases in the case-base. Thus, the use of good case engineering practices are recommended not only to create the first set of cases, but also to create cases personalised to a particular user. Furthermore, since the case learning task is included in the CBR cycle, the case-base size increases with time, making the inclusion of case-base maintenance policies necessary to maintain the quality and performance of the case-base and thus the CBR system.

The core of the Smart Home proposed in [8] is a CBR system. This system is placed in a residential ward where nursery staff takes care of patients. The purpose of the CBR system is to plan the future tasks to be done by the staff. The case representation keeps a record of one task already done by a staff member, along with information related to the time in which the task started and ended as well as its priority and providing the next task to do as the solution of the case. Furthermore information related to the patients' health status is retained in the cases as well, since this data is relevant for the planning of future tasks.

AmICREEK uses a CBR system to detect the situation taking place within the system's environment [10,11]. AmICREEK is based upon a three layer architecture [4]: the *perception* layer as the middle-ware gathering the information from the sensor network, the *awareness* layer as a CBR system that detects the context in which the action is taking place and the goal of the action. Lastly, there is the *sensitivity* layer, in which a sequence of tasks is built in order to satisfy the goal given by the CBR system according to the system's context. The authors tested this approach in a hospital ward but the cases only contained one time-stamp to represent the time in which they were created.

3 A Smart Home for Alarm Detection

The architecture of the *proDIA* system is built upon three levels: *sensor level*, the *communication level* and the *data processing level*. Figure 1 depicts the system levels, as well as an example of the distribution of sensors in a house, located in room, kitchen, bathroom, bedroom and the corridor.

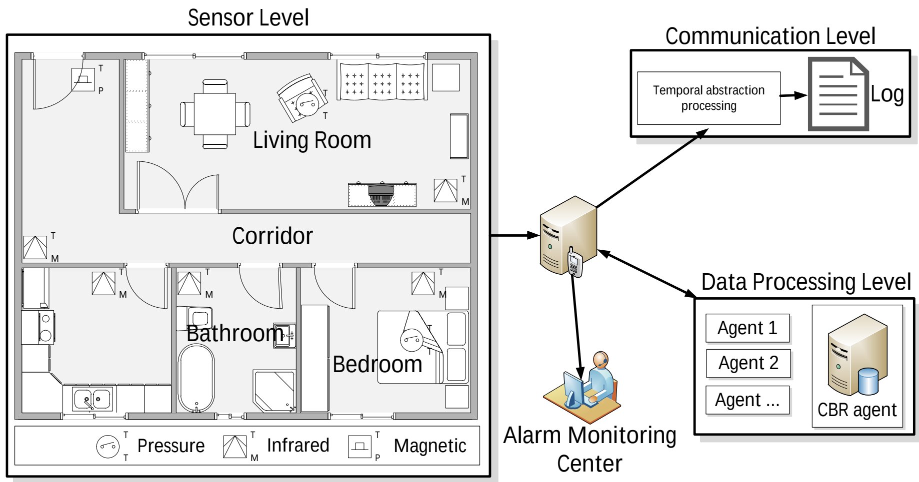

Fig. 1. The three levels of the system, as well as an example of the distribution of sensors in the house in order to monitor the person at home.

The *sensor level* is the first level of the system's architecture, it manages the sensor-data acquisition from the wireless sensor network. This network uses three types of sensors: infrared motion detection sensors, pressure sensors and magnetic sensors to detect whether the main door is opened or closed. The basic configuration of the system implies the placement of motion sensors in every location, and one single magnetic sensor to detect whether the main door is opened or closed. Furthermore, pressure sensors may be located in places such as sofas and beds to detect whether the person is resting or lying on one of them. With these pressure sensors it is possible to detect the location of the person, even if the person is not moving.

The second architecture level is the *communication level*, where the data provided by the sensor level is recorded. Using the IEEE-802.15.4 communication standard, the data gathered by the sensors is kept in a home-station (mini-PC), which synchronises the data sent by the sensors according to the timestamps in which they are received. The communication level creates a log of the data with a given frequency, which is used by the *data processing level*. The *log* is a comma separated values (CSV) file, where each line contains information related to the readings sent by the sensors, such as the identifier of the sensor, the time-stamp in which the sensor is activated, the location of the sensor and the content of the sensor's reading. Every newly created log starts empty, so the frequency in which the log is created will determine the amount of data stored in it. Therefore, if long observation periods of the home are required, the log must be created with a low frequency. On the contrary, log files created with a high frequency contain less data since they represent short observation periods.

For instance, the sensor level may produce a log file as shown in table 1. In this example, the log records the data sent by the sensors when the user arrives at home.

Time-stamp	sensor id.	Location	Message	Rationale/Justification
7932	sensor197680	Corridor	MOVE: true	getting house
7932	sensor197683	Door	isOPENDOOR: true	
7940	sensor197683	Corridor	isOPENDOOR: false	
7959	sensor347050	Bedroom	MOVE: true	going to the bedroom
7972	sensor530111	Bedroom	PRESS: true	sitting down on the bed
7980	sensor197680	Corridor	MOVE: false	
8054	sensor530111	Bedroom	PRESS: off	standing up
8113	sensor197680	Corridor	MOVE: true	going to the toilet
8121	sensor197680	Bedroom	MOVE: false	
8122	sensor536770	Bathroom	MOVE: true	getting in the bathroom

Table 1. Example of activity log.

Starting from the information provided by the communication level, the *data processing level* attempts to infer the state in which the elderly is, that is, the complete situation in accordance to the situation context is described. The system relies on the assumption that the location of the resident, the activity or absence of it, and the moment of the day in which these facts are registered are enough to detect possible emergency situations. For example, if the attendee has fallen and lost conscience or broke a bone in such a way that prevents him or her from moving, detection of this situation is based on an excessive time of inactivity being measured in a context in which this is abnormal (i.e. the attendee is in the house and she is not supposed to be resting or sleeping). To this end a behavioural model was developed, based on a finite state automaton. Once an abnormal situation is detected, the system sents an alarm to the Alarm Monitoring Centre using UMTS telecommunication technology, where a specific predefined protocol is fired.

3.1 Case-based Reasoning Agent

In this work we propose to include a CBR agent, which tries to check whether the daily activity at home is normal or abnormal, indicating an unsafe scenario is taking place. To this end, the approach followed is to keep a record of the movement of the resident at home within given time-frame. The CBR agent checks whether a current activity, or event sequence, is similar to previously recorded activities/event sequences.

According to the definition of case given in [1], a case consists of a problem and a solution. In order to classify and detect unsafe situations at home, the cases represent a daily activity and its type. Thus, whereas the problem represents the visited locations during one daily activity, the solution is a label describing the type activity or scenario. The set of valid values for the solution is not limited, being possible the inclusion of new *solution* labels on demand when the CBR is running.

The event sequences consist of ordered heterogeneous events in time, where each event is composed of an event type and a time-stamp that represents when the event occurs [16,17]. Thus, each event in the sequence is a tuple of the location visited by the person and a time-stamp.

The following expressions are shown to detail the case representation (see expression 1) and the event sequence (see expression 2).

$$\begin{aligned} \text{case } c &= (sequence, solution) \\ solution &\in \{normal, scenario_1, scenario_2, \ldots\} \end{aligned} \tag{1}$$

$$\begin{aligned} sequence = \; & \langle (loc_1, t_1), \ldots, (loc_2, t_2), (loc_n, t_n) \rangle \mid \\ & \mid \forall loc \in \{Corridor, Bedroom, \ldots\} \wedge \\ & \wedge \; t \in \mathbb{N}^+ \wedge \forall_{i=1}^{n-1} t_i \leq t_{i+1} \end{aligned} \tag{2}$$

Given a log, generated by the communication level, the operation of the CBR agent is the following:

1) The CBR agent reads the log when a new one is created by the communication level. This log contains the data from sensors chronologically ordered according to their time-stamps.
2) An event sequence is built from the collected sensor data. Later, this event sequence is used as a *input* query to the retrieval step.
3) The CBR agent retrieves from its case-base those cases with the most similar event sequence to the *input*.
4) Based on the retrieved cases, the system infers the type of the activity best matched to the *input*.
5) When the activity is classified as abnormal (according to the defined solution labels), the system sends a message to the Alarm Monitoring Centre, where the expert decides which is the most suitable action for the detected scenario.

Finally, a new case is retained in the case-base when an *abnormal* scenario is classified correctly

3.2 Computing the Similarity between Cases

The CBR agent uses the edit distance between event sequences proposed in [16,17]. This distance measure computes the *cost* of transforming an event sequence into another. This cost is represented as the number of operations needed to perform the transformation. The operations are applied on the query event sequence, until it matches with the retrieved event sequence, which is known as pattern. Therefore, a high number of transformation operations stands for two not very similar event sequences. On the contrary, two more similar event sequences need fewer transformation operations. The set of available operations are *insertion*, *deletion* and *displacement*. While the insertion is used when the query does not contain an event that is present in the pattern, the deletion is used if the query contains an event not appearing in the pattern. The displacement operation is used when two events in both the query and pattern match with the same location. Whereas the operations insertion and deletion have a cost for their application, the operation displacement has a cost based on the difference between the timestamps of the two events. In order to ensure the correct functioning of the edit distance, the displacement costs has to be lower than the insertion and deletion operations. Thus, the difference between the timestamps is normalized between 0 and 1, and the cost of the insertion and deletion operations is set to values higher or equal to 1. Algorithm 1 presents a dynamic programming approach for searching the minimum number of operations to transform a query into a pattern.

Algorithm 1 Edit distance between two event sequences x, y [16,17]

Input: Two event sequences $x = \langle (loc_1^x, t_1^x), \ldots, (loc_n^x, t_n^x) \rangle$ and $y = \langle (loc_1^y, t_1^y), \ldots, (loc_m^y, t_m^y) \rangle$, with $loc \in \{Bedroom, Corridor, \ldots\}$, the costs $w(loc^x), w(loc^y)$ of the *insertion* and *deletion* operations.

Output: Edit distance between the two given sequences.

$r \leftarrow$ matrix of $n \times m$ dimensions
$r(0,0) \leftarrow 0$
for $i \leftarrow 0$ **to** m **do**
  $r(i,0) \leftarrow r(i-1,0) + w(loc^x)$
end for
for $j \leftarrow 0$ **to** n **do**
  $r(0,j) \leftarrow r(0,j-1) + w(loc^y)$
end for
for $i \leftarrow 1$ **to** m **do**
  for $j \leftarrow 1$ **to** n **do**
    $update_x \leftarrow r(i-1,j) + w(loc^x)$
    $update_y \leftarrow r(i,j-1) + w(loc^y)$
    $align \leftarrow r(i-1,j-1)$
    if $loc^x = loc^y$ **then**
      $align \leftarrow align + (\frac{|t_i^x - t_j^y|}{max(t) - min(t)})$
    else
      $align \leftarrow align + w(loc^x) + w(loc^y)$
    end if
    $r(i,j) \leftarrow min\,(update_x, update_y, align)$
  end for
end for
return $r(n,m)$

3.3 Study of Alarm Scenarios

Four different scenarios are considered based on different common scenarios that usually occur at home: a *normal* daily activity, a *bad night* due to sickness, a fall resulting in a conscious status and a fall with an unconsciousness status. Next, an example of a case is given for each type of scenario, where each location used by the event sequences is one of the following

$$loc = \{\ Corridor = 0, Kitchen = 1, LivingRoom = 2,\\ Toilet = 3, Bedroom = 4, Out = 5\}$$

The normal daily behaviour represents the locations visited by the resident during one of his/her daily activity taken at home, such as having a shower in the bathroom, watching the TV in the living room or sleeping on the bed in the bedroom.

$$c = (\ \langle(4,539),(3,29303),(1,29439),(4,30737),(2,31420),\\ (0,35352),(1,49882),(3,53750),(2,54011),(0,62753),\\ (1,74758),(2,76114),(1,82977),(3,85593)\rangle, \text{'normal'})$$

The *bad night* template represents the locations visited by the resident when he or she has not been able to sleep due to a sickness status. In this scenario, the event sequence represents regular visits to the bathroom during the night.

$$c = (\ \langle(4,447),(3,7685),(3,28669),(1,29196),(4,30618),\\ (2,31049),(1,49726),(3,53542),(2,54109),(0,61434),\\ (1,73228),(2,77069),(1,83260),(3,85484)\rangle, \text{'bad night'})$$

The two fall scenarios represent two types of falls: a fall where the person stays motionless after losing consciousness and another where the person stays conscious and may crawl on the floor. Both scenarios usually occur in the bathroom as a consequence of a fall in the bathtub and the difference between them is the activity of the location visited. Whereas an unconscious person after a fall does not activate any movement sensor, a conscious person try to move or crawl to other different location to call for help. The following are examples of cases of both fall scenarios:

$$c = (\ \langle(3,29040),(1,29985),(4,30871),(2,31343),(1,49764),\\ (3,53915),(3,53960),(3,54393),(3,54482),(0,54628),\\ (3,54663),(0,54892),(0,54968),(2,55115)\rangle,\\ \text{'fallen with consciousness'})$$

$$c = (\ \langle(3,29229),(3,29997),(3,30055),(3,30119),\\ (3,30178),(3,30235),(3,30290),(3,30350)\rangle,\\ \text{'fallen with unconsciousness'})$$

3.4 *myCBR* Extension for Temporal Similarity

A key goal in the developing of the open source software *myCBR* is the aim to provide a compact and easy-to-use tool for rapidly prototyping CBR applications. The *myCBR* tool is especially intended to be used in the contexts of research and teaching as well as to allow businesses to allow for the timely implementation of CBR systems with low initial development effort. To enable easy development *myCBR Workbench* provides graphical user interfaces for modeling case structures and attribute-specific similarity measures. *myCBR Workbench* further provides a GUI based retrieval interface for evaluating the retrieval quality of an implemented CBR system. Further *myCBR Workbench* includes tools for generating the case representation automatically from existing raw data as well as importing cases from CSV files. Next to the *myCBR Workbench* *myCBR* offers a Software Development Kit (SDK) which allows for the integration of the developed CBR systems into other applications. A key factor of the SDK is to allow also for the implementation of extensions into the SDK. These extensions can, for example, be based on specific requirements such as additional similarity calculations, as it is demonstrated in this paper.

As described in this paper the need for providing a similarity measure for event sequence was met with the integration of this similarity measure into the *myCBR* SDK. This extension of the SDK was eased by the high modularity of the existing *myCBR* SDK's code and its extensive documentation. As *myCBR* is implemented in Java it follows the object oriented approach, easing the extension of the SDK. The ability to extend the SDK allows for researchers and businesses to quickly integrate their innovations or experimental features into a robust existing CBR system development tool. Thus a CBR developer can test run their own innovative feature within a robust existing CBR system without having to implement the whole CBR system from scratch [22]. The development team working on *myCBR* is currently also cooperating with the development team of the well established CBR development tool jColibri [3] to allow for the import of similarity measures developed within *myCBR* into CBR systems based on jColibri [19]. This will allow the creation and testing of *myCBR* extensions that then, in turn, could be integrated into jColibri based systems.

The extension of *myCBR* that was implemented to provide the similarity measure for event sequences was based on the creation of a new attribute. This new attribute represents event sequences. To define this new attribute we adopt the event sequence definition of expression 2.

Figure 2 shows the new classes that were implemented as an extension to the existing *myCBR* SDK. The extension allows us to represent an event sequence as an attribute of a case, which allows *myCBR* to compute the similarity between two different attribute values, e.g. event sequences.

EventSeqAtt: Contains the information of a particular event sequence.

EventSeqRange: Contains the possible values of event sequences. Also allows for the creation of new *EventSeqAtt*.

[3] http://gaia.fdi.ucm.es/research/colibri/jcolibri

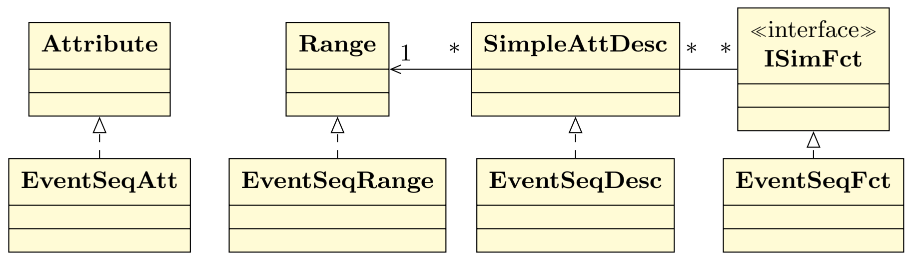

Fig. 2. Sequence of operations to compute the similarity between two cases.

EventSeqDesc: Contains the details about how the event sequences are implemented.

EventSeqFct: This corresponds to the implementation of the similarity measures that we have worked on in the present paper. When an object of this class is created it is mandatory to specify which type of similarity measure it should employ.

Next to the generation of CBR systems and the extension of these systems by experimental features, *myCBR* also allows for the generation of cases from CSV files. This form of rapid case acquisition is highly desirable for the use of sensor based systems as these systems' data output can be transformed into CSV data quite easily. Aside from the project focused on in this paper, the authors are currently taking part in research work on the effectiveness and accuracy of large numbers of cases from CSV-formatted sensor data.

4 Experiments

The goal of the experiments was to study whether the CBR agent is able to detect the type of a scenario occurring at a residents home, particularly unsafe scenarios. The CBR agent used in the experiments was configured with regard to its case-base, which was created synthetically, and the length of the home observation, that is, the length of the queries generated from the log. The CBR agent retrieved the most similar cases using an 1-NN global similarity function. As only one case was retrieved by each query, no adaptation process was performed. So the solution of the retrieved case was returned as the inferred type of scenario taking place at the residents home.

4.1 Generating a Synthetic Case-Base

The evaluation is based on the creation of synthetic case-bases. The main reason to do so is to keep control of the different existing scenarios at home, and being able to study if the CBR agent is able to detect them.

From the proposed scenarios in subsection 3.3, new cases are created to populate the case-base. Every created case contains an event sequence which represents the visited locations for one behavioural template. That is, the templates may be understood as workflows, and the event sequences as their executions. Furthermore, the case-base has a 90% of its cases representing normal daily activities, since abnormal scenarios should appear less frequently than normal scenarios. In order not to discriminate any abnormal scenario, the 10% of the remaining cases is equally shared by the abnormal cases. In particular, the normal behaviour is created using as template a workflow given in figure 3.

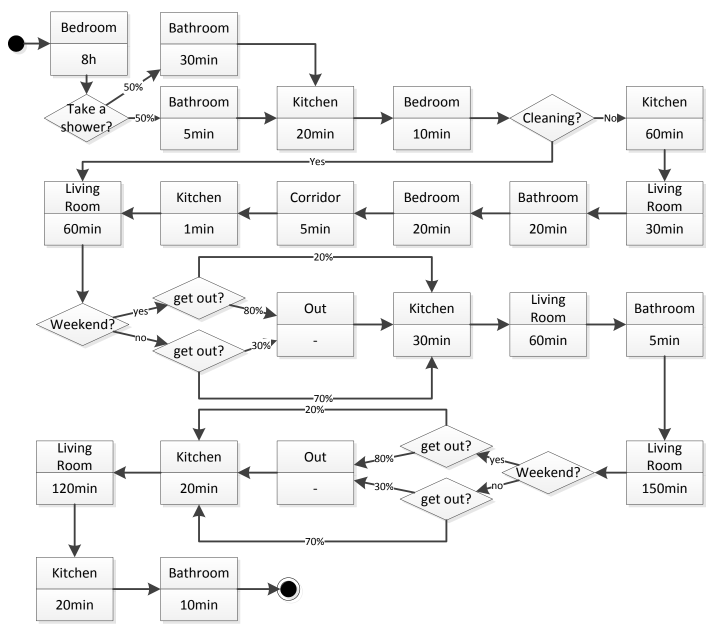

Fig. 3. Workflow of activities for a normal behaviour at home.

Regarding the similarity between the different types of scenarios, while the bad night activities are almost similar to normal activities, the fallen activities are similar between themselves but not to normal and bad night activities.

4.2 Experiments Results

Several experiments have been run in order to evaluate the effects of the case-base size and the frequency in which the log is created by the communication level. That is, the CBR agent is evaluated for different case-base sizes in different log scenarios. The proposed log creation frequency are 6, 12, 18 and 24 hours. The cases in the case-base representing normal and bad night activities cover up to 24 hours of movements at home, and the rest of the cases represent normal activities until the unsafe scenario occurs.

Figure 4 shows the evaluation results from the performed experiments. For each experiment, the accuracy of the system is observed, as well as the false positive rate regarding the normal behaviour. Additionally, the true positives rates for the different behaviours are recorded to study if the CBR agent succeeds to detect them. Note that the false positive rate means the proportion of unsafe scenarios that were classified as normal, which must be avoided because this type of misclassification is the most dangerous one for the residents safety.

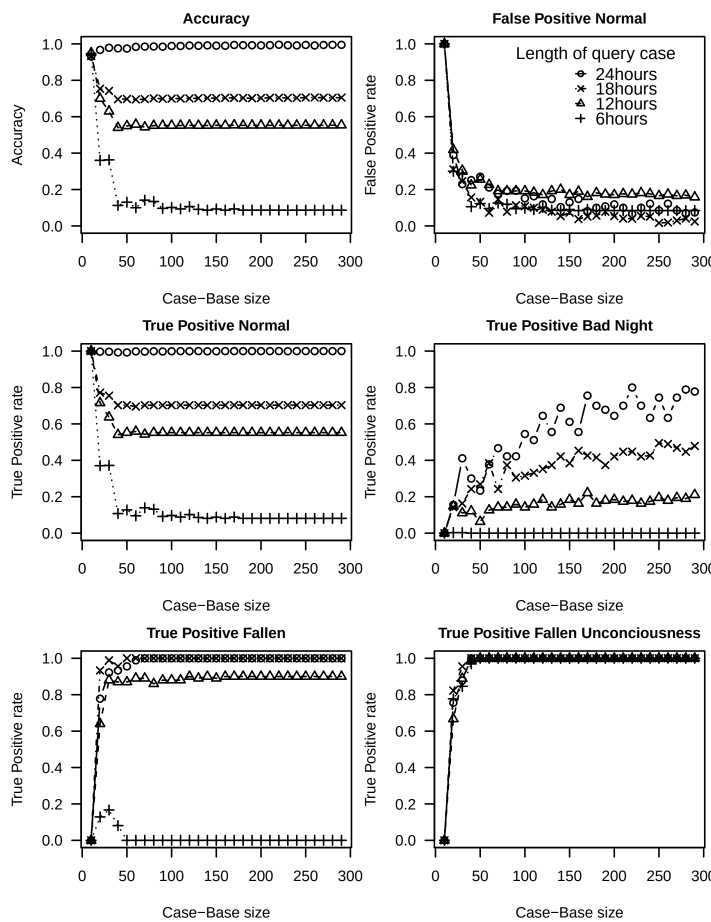

Fig. 4. Evaluating the CBR agent using varying the case-base size.

4.3 Discussion

Based on the results shown in figure 4, the following observations could be made:

- The accuracy is stable in all the scenarios when the case-base size is over 50 cases. However, a CBR agent using short observation periods has a lower accuracy than a CBR agent using longer observation periods. In fact, the experiments where the observation period spanned up to 24 hours gets an accuracy value close to 1, even with small case-bases.
- Retaining a large amount of cases decreases the ratio of false positives for the normal behaviour. That is, increasing the number of stored cases means a decrease of the number of times in which the agent miss-classifies a behaviour as normal when in fact there is an abnormal situation at the residents home.
- The true positives rate for the normal activities is directly correlated to the accuracy of the system, which is normal due to the predominance of this type in the case-base (90% of the cases). Nonetheless, when the observations are frequently acquired, then the CBR agent is increasingly unable to detect the normal activities, which increases the ratio of false alarms of the system. That is, the system is permanently classifying the abnormal situations as normal.
- Regarding the true positive rate of bad night scenarios, in all the experiments an increment of the case-base size will ease the correct classification of these scenarios, except for the shortest observations, which never classifies this activity correctly. In fact, CBR agents using longer observation periods classify the bad night scenarios correctly more frequently than those using shorter observation periods.
- The detection of falls is possible with most observation lengths, except for the shortest observation periods, which fail to detect falls without loss of consciousness. Notwithstanding, longer observation periods are not suitable for the needed quick response in the fall scenario.

5 Conclusions

In the present work, we propose a CBR agent for a Smart Home system to detect potential dangerous scenarios for elder residents. The CBR agent gathers data on the activity of the monitored resident (in the form of event sequence) and uses temporal similarity to retrieve previously stored activities. The agent is implemented using the *myCBR* SDK. We evaluate the suitability of this approach using a simulation of activities at home representing normal and different dangerous scenarios.

Unlike other work [10,11,14,15], our proposal is based on a spatial-temporal representation using event sequences. Furthermore, our system relies on a low-cost pervasive sensor network. That is, the available information is limited and the temporal dimension plays an essential role. Essentially the CBR agent is a temporal case retrieval system using a temporal edit distance.

Our experiments are based on synthetic case-bases using a simulator in order to analyse the responses provided by the CBR agent. Results show that the CBR agent is able to detect the proposed unsafe scenarios, although this detection is limited by the amount of data within each observation of the house. Thus, the accuracy is affected by the frequency in which the house status is observed and the amount of data in each observation. On the one hand, long observations of the house ease the identification and classification of all the activities. However, low frequent observations can increase the response time of the system when a fall occurs. On the other hand, high frequency observations make possible for the system to response quickly to falls, but they make very difficult to detect long activities such as having a bad night. What is more, if the observation frequency is very high then the system is not able to detect falls neither.

We can also conclude that a CBR agent can detect unsafe scenarios at home, although some considerations need to be addressed before using a CBR agent in a real deployment. The observation of the environment needs to be done with short and long intervals in order to get a quick response to dangerous situations, such as falls, and to detect long activities. Finally, since new cases are added constantly to the case-base, there is a risk of a decreasing performance of the CBR agent. This problem may be solved with the application of an appropriate Case-Base Maintenance task. Our future steps will be focused on exploring available maintenance approaches, the adoption of other temporal similarity measures, analyse other experiment parameters and the practical evaluation of the CBR agent in a real world test environment.

Acknowledgements. This work was funded by the Seneca Research Foundation of the Region of Murcia under project 15277/PI/10.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues , methodological variations, and system approaches. AI Communications 7, 39–59 (1994)
2. Akyildiz, I., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: a survey. Computer Networks 38(4), 393 – 422 (2002)
3. Augusto, J., Nugent, C.: Smart homes can be smarter. In: Designing Smart Homes, Lecture Notes in Computer Science, vol. 4008, pp. 1–15 (2006)
4. Botía, J.A., Villa, A., Palma, J.T.: Ambient assisted living system for in-home monitoring of healthy independent elders. Expert Systems with Applications 39(9), 8136–8148 (2012)
5. Broek, G.V.D., Cavallo, F., Wehrmann, C.: AALiance Ambient Assisted Living Roadmap. IOS press (2010)
6. Copetti, A., Loques, O., Leite, J.C.B., Barbosa, T., da Nóbrega, A.: Intelligent context-aware monitoring of hypertensive patients. In: Conference on Pervasive Computing Technologies for Healthcare, 2009. pp. 1–6 (2009)
7. Corchado, J.M., Bajo, J., de Paz, Y., Tapia, D.I.: Intelligent environment for monitoring alzheimer patients, agent technology for health care. Decision Support Systems 44(2), 382 – 396 (2008)

8. Corchado, J., Bajo, J., Paz, Y.: A CBR system: The core of an ambient intelligence health care application. In: Soft Computing Applications in Industry, Studies in Fuzziness and Soft Computing, vol. 226, pp. 311–330 (2008)
9. Cunningham, P., Doyle, D., Loughrey, J.: An evaluation of the usefulness of case-based explanation. In: ICCBR'03. pp. 122–130 (2003)
10. Kofod-Petersen, A., Aamodt, A.: Contextualised ambient intelligence through case-based reasoning. pp. 211–225. ECCBR'06 (2006)
11. Kofod-Petersen, A., Aamodt, A.: Case-based reasoning for situation-aware ambient intelligence: A hospital ward evaluation study. In: Case-Based Reasoning Research and Development, pp. 450–464. Springer (2009)
12. Krishnan, N.C., Cook, D.J.: Activity recognition on streaming sensor data. Pervasive and Mobile Computing 10, Part B(0), 138 – 154 (2014)
13. Kröse, B., Kasteren, T.V., Gibson, C., Dool, T.V.D.: Care: Context awareness in residences for elderly. In: ISG'08 (2008)
14. Leake, D., Maguitman, A., Reichherzer, T.: Cases, context, and comfort: Opportunities for case-based reasoning in smart homes. In: Designing Smart Homes, Lecture Notes in Computer Science, vol. 4008, pp. 109–131. Springer Berlin Heidelberg (2006)
15. Ma, T., Kim, Y.D., Ma, Q., Tang, M., Zhou, W.: Context-aware implementation based on cbr for smart home. In: WiMob'05. vol. 4, pp. 112–115 Vol. 4 (2005)
16. Mannila, H., Moen, P.: Similarity between event types in sequences. In: DataWarehousing and Knowledge Discovery (1999)
17. Moen, P.: Attribute, Event Sequence, and Event Type Similarity Notions for Data Mining. Ph.D. thesis, University of Helsinki (2000)
18. Rawi, M., Al-Anbuky, A.: Passive house sensor networks: Human centric thermal comfort concept. In: ISSNIP 2009. pp. 255–260 (2009)
19. Roth-Berghofer, T., Garcia, J.A.R., Sauer, C.S., Bach, K., Althoff, K.D., Diaz-Agudo, B., Calero, P.A.G.: Building case-based reasoning applications with mycbr and colibri studio. In: Proceedings of the 17th UK Workshop on Case-Based Reasoning, pp. 71–82 (12 2012)
20. Safavi, A., Keshavarz-Haddad, A., Khoubani, S., Mosharraf-Dehkordi, S., Dehghani-Pilehvarani, A., Tabei, F.: A remote elderly monitoring system with localizing based on wireless sensor network. In: ICCDA'10. vol. 2, pp. 553–557 (2010)
21. Sormo, F., Cassens, J., Aamodt, A.: Explanation in case-based reasoning - perspectives and goals. Artificial Intelligence Review 24(2), 109–143 (2005)
22. Stahl, A., Roth-Berghofer, T.: Rapid prototyping of CBR applications with the open source tool myCBR. In: Advances in Case-Based Reasoning, vol. 5239, pp. 615–629 (2008)
23. Tabar, A.M., Keshavarz, A., Aghajan, H.: Smart home care network using sensor fusion and distributed vision-based reasoning. In: 4th ACM International Workshop on Video Surveillance and Sensor Networks. pp. 145–154. VSSN'06 (2006)
24. WHO, NIA, NIH: Global health and ageing. NIH Publications, WHO (2011)
25. Wiratunga, N., Craw, S., Taylor, B., Davis, G.: Case-based reasoning for matching smarthouse technology to people's needs. Knowledge-Based Systems 17, 139 – 146 (2004)