A New Platform to Easily Experiment Activity Recognition Systems based on Passive RFID Tags: Experimentation with Data Mining Algorithms

P-O. Rocher, B. Bouchard and A. Bouzouane

LIARA Laboratory, Université du Québec à Chicoutimi (UQAC)
555 boul. Université, Saguenay (QC), Canada, G7H 2B1
{Pierre-Olivier.Rocher, Bruno.Bouchard, Abdenour.Bouzouane}@uqac.ca

Abstract

Healthcare, due to the aging of western populations, requires new technologies to help assisting the needs of elders. The smart home paradigm is one of the promising new trends of research aiming to bring socially and economically viable solutions to this challenge. One of the most crucial problems in developing smart environment is activity recognition. It can be defined as the process of inferring, with various sensors, what the patient is doing and then, being able to predict what he might do in the future. We can find in the literature a lot of works on this theme, however the majority remain essentially theoretical. More specifically, they often work only on a particular component of the activity recognition process, for example by focusing only on the hardware (sensors) or solely on the high level recognition part, assuming that low level recognition already works. Furthermore, we noticed that most available recognition test platforms with an infrastructure, such as MavHome, are static and involve a complex set of sensors, which inevitably has a heavy cost. The work presented in this paper aims of providing solutions to these problems by proposing a way to implement from A to Z a complete recognition platform that works, is simple to use, inexpensive, sturdy and portable. This platform is based only on RFID tags and can be reuse everywhere to test various recognition algorithms, even directly at the patients' home. We also present a first experimentation conducted with this platform using data mining recognition algorithms.

Keywords*: Smart homes, RFID, experimental test platform, recognition of activities of daily living (ADL), data mining recognition algorithm.*

1. Introduction

The aging of western populations is becoming a reality from which new challenges arise [3]. The main one is certainly related to the healthcare issue and to the quality of life that we want to offer to people. Smart home technologies have become an interesting and very active research trend, bringing hope in the effort to postpone the institutionalization of the elders [4]. A smart home can be seen as a technologically enhanced environment: an environment making decisions, remaining non-intrusive, and most importantly, helping in the completion of the Activities of Daily Living (ADL) [2]. In the last few years, the aging problem urged scientists to significantly increase the research [4, 10, 15, 20] in developing new smart home technologies. Activity recognition takes an important part in such development. Its first objective is to determine which is the patient's ongoing activity based on the data received from various set of sensors. Its second objective is to infer what the patient intends to do in the future in order to monitor his progression and to identify the right moment to provide assistance.

One of actual works major issue in this particular field is that the proposed approaches often focus only on a specific aspect of the recognition process, such as the low-level sensors part or the logical high-level component. For example, we can cite the work of [11] on activity recognition, which specifically address the low-level physical issue of merging sensors inputs, without going into details of how to use these inputs for inferring the ongoing activity. Other works propose models for a higher level inferential process, assuming that the collected data is consistent and usable, and focus exclusively on the activity recognition part using this particular data [21]. A lot of works are essentially theoretical and remains untested or are only experimented in a non-realistic context that could not determine their actual effectiveness and / or put in evidence the problems that had not been taken into account in the design. Nevertheless, experimentations and testing remains the best way to check the robustness of a recognition system.

There are very few works exploring this activity recognition related issue in a smart home and offering a complete solution for implementing a working recognition system and experimenting it from A to Z. Most existing models and platforms for experimenting recognition systems with synthetic data or with patients use fixed or non-easily transportable infrastructures [9, 13, 7, 8] that usually exploit a complex combination of sensors (binary or pressure sensors, RFID tags, video capture, motion detector, etc.). This kind of infrastructure is difficult to deploy, the corresponding software needed to exploit it is complex to implement, it is certainly expensive and complicated to maintain. Moreover, moving patients with a cognitive impairment for testing purpose in this kind of unfamiliar environment poses many problems. It would clearly be wiser to bring the system to them. Therefore, our goal is to develop a platform for experimenting activity recognition systems that can be deployed at home or in any environment and which will be able to effectively test technologies and algorithms developed using a simple and inexpensive set of sensors.

In this paper we propose the present complete design and implementation of a working recognition system based only on RFID tags, which can be used as a test platform by researchers. This platform is meant to be easily deployed, inexpensive and easily transportable, allowing the possibility of taking it for in-house experimentations or to bring it at a conference for a demonstration. Hence, we propose an answer to the following questions:

- How to model a relevant activity for testing a smart home recognition system?
- How to deploy a simple hardware sensors set up based on the use of RFID tags attached to everyday objects?
- How to interpret the low-level events coming from the RFID tags using the received signal strength to extract useful information?
- How to implement a simple working activity recognition software with this platform and how to validate it?
- How to conduct set of experimentations with this platform?

This paper is organized as follows: section 2 presents the model of the selected activities and justify its choice based on a scientific and experimental point of view. Section 3 presents a choice of hardware and sensors setup. It also discusses how the tags are attached to the objects, how many should we use and which type to select (five different types are possible). Section 4 specifies the implementation of the recognition approach and show how we proceeded to accurately recognize an ongoing activity. It concerns, in particular, the physical set-up, the software part, and the various stages involved in the data processing to finally reach the two used methods to infer activity, using data mining techniques. Section 5 presents our testing phase, details the experimental protocol, shows the obtained results, and discusses the various problems we have been confronted to. The final section concludes the paper, and opens different perspectives it could be interesting to study in future works.

2. Selecting and Modeling the Right Activities

Selecting an activity in order to perform tests with persons suffering from dementia is not an easy task. Finding one that is also consistent with scientific and experimental needs makes the task even more complex. As a starting point, we noticed that a lot of examples in the literature imply common kitchen or bathroom activities (washing hands [10], preparing tea or coffee [18], cooking [4], etc.). Thus, this constitutes a good first clue. Furthermore, we know that scientists need to model activities that are representative, but are also easily experimented with in a research context. Therefore, the chosen model must be simple and must include a sufficient but limited number of steps. More importantly, the chosen activities should also be achievable in a reasonable amount of time. To meet all these needs, we selected the activity *prepare coffee*. This activity constitutes a really good choice because it is composed of only several steps, it is well known by patients, and it can be done in less than 10 minutes. Moreover the objects implicated in this activity are light and easy to move. This activity is also similar in nature and size to a lot of smart home literature examples such as preparing tea [18], cooking pasta [4] or washing hands [10].

To determine the necessary steps to achieve the activity *prepare coffee,* we relied on the recommendations of a well-known cognitive test: the Naturalistic Action Test (NAT) [22]. This test was made to assess the cognitive performance of an individual on the completion of a set of household activities of everyday life. This test was specifically designed to monitor the activities of patients suffering from neurological disorder and uses the selected activity.

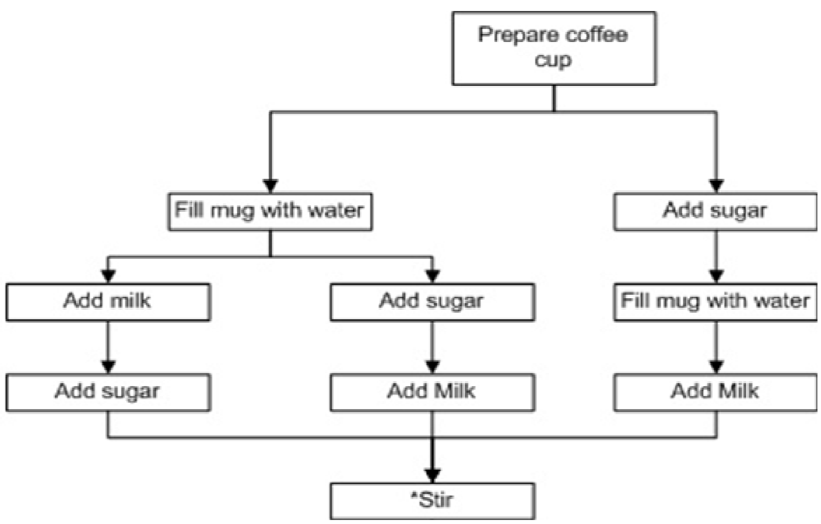

Figure 1. Diagram Detailing the Steps of the Activity *prepare coffee*

The model of the chosen activity is presented in Figure 1. As we can see, several step's combinations are possible to carry out the activity *prepare coffee*. According to the NAT test, the objects that the patient will have at his disposal are: a cup, a (hot) water jar, a sugar box and a milk jug, all placed on the top of a rectangular table. In our system, sensors will be attached to these objects in order to make inferences about the current activity. We will use this activity model and these objects as a basis for building the recognition test platform.

3. Choosing the Right Set of Sensors

Many kinds of sensors can be used to recognize an activity in a smart environment. These include GPS [17], Wi-Fi [24], video [10], accelerometers [25], RFID tags [5], etc. These different types of sensors differ in their cost, effectiveness, precision, weight, size, range or by their deployment and maintenance efficiency. As an example, the work of [17] uses GPS data to track a person's movements to assess his actual mean of transportation and to learn the person's habits. However, although such GPS based system is interesting to monitor the daily movements, it is too inaccurate for indoor activities involving close proximity between objects. Other works, such as the one of [10], considered the use of video by using a set of cameras. It showed that it can be an excellent mean of detecting activities, even with the use of nearby objects. Nevertheless, the input produced by a camera is complex to interpret and it is difficult to adapt a camera based system to a new environment (new object shapes and colors, different brightness or luminosity, etc.) [21]. Moreover, using cameras can be ethically challenging. As we said before, our goal is to build a recognition platform that is easily implemented, simple to use and portable in order to allow in-home experimentations as well as demonstrations in various locations. The cost is also an important factor and therefore the system must remain affordable for scientists.

For our system, we decided to use RFID tags as hardware basis. In fact, their employment is simple and relatively low cost compared to other considered solutions. The information that the RFID tags can provide is more accurate than the information provided by a GPS but less accurate than those obtained with a video system. The cost of RFID tags and antennas is relatively low and their use is much simpler than processing camera's images. RFID technology as also already proven its robustness. It is largely used in commercial and industrial environment. A system based on RFID tags therefore correctly answered our needs: a good compromise in terms of obtained accuracy, price, and simplicity. We can point out another advantage in using RFID tags: the impact on the individual's behaviour from tagged object is very low (less intrusive), which would, for example, not be the case with the uses of GPS devices. The RFID solution that we propose is also less intrusive than the one of Patterson [16], who use a glove that the person must wear including a battery and an antenna. Of course the accuracy is improved, but being forced to wear a glove can be disturbing, especially for seniors with dementia.

We can found on the market a large number of RFID solutions, often in kit, including a reader, one or more antennas, and an assortment of tags. However, a lot of factors are to be taken in consideration: performance, cost, recoverable information, etc. After some research, we opted for the Alien Technology RFID Kit [1]. This is a kit comprising a reader (you can also use it to write on tags), two antennas, and an assortment of passive RFID tags. The advantage of this kit lies in the fact that it embeds a proprietary technology called ITR[1] [1], which offers the possibility of obtaining signal strength, direction and speed of tags. This type of functionality is usually not offered by manufacturers and appears to be a very interesting element that we exploit in our recognition system.

3.1. RFID Technology: Using Passive or Active Tags?

RFID standard means Radio Frequency Identification. It is a method to store and retrieve data in a remote way. RFID tags are small objects (just like code bars) that can be incorporated on products or implanted, whether in animals or humans. They include a silicon chip containing the data and an antenna to facilitate communications [26, 14]. Since their

[1] Intelligent Tag Radar™

invention some years ago, several versions have been developed. The operation and the offered capacity are a bit different. We can currently distinguish five different classes. Classes 0, 1 and 2 do not contain batteries. These are called *passive* RFID tags. They are only powered through the energy of the waves emitted by the reader. Figure 2 shows the operational mechanism. The reader uses the antenna to search for tags, then sends an electromagnetic wave that will awake the passive tags by providing them energy. They will then respond using the same ephemeral energy. Obviously, tags near the antenna receive more energy and can therefore make a stronger wave in return (green wave) than those more distant (red waves).

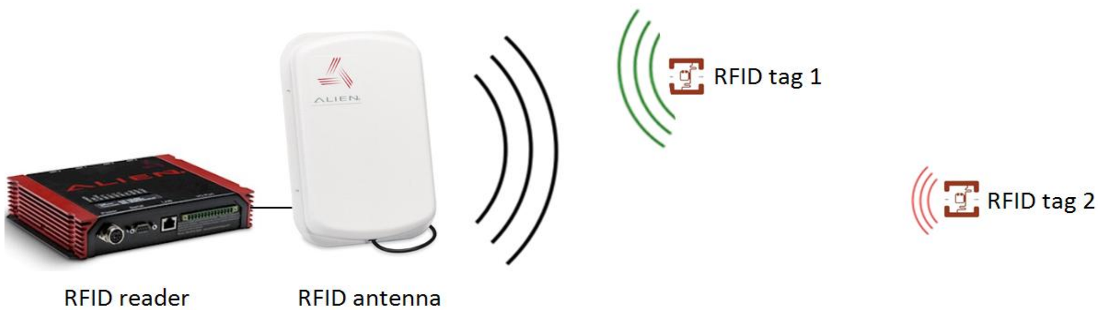

Figure 2. Read/Write Principle Pperation on RFID Tags

The other two classes, Class 3 and Class 4, also follow the same principle, but their operation is improved with the use of batteries. The Class 3 consists of so-called semi-active tags and they have a battery support. They fill the gap between shorter range passive tags and high cost active RFID systems. The class 4 is made of active tags, they answer to the reader with a built-in battery. It is sometimes considered a fifth class, which is actually a designation referring to RFID tags readers. It is obvious that the capabilities of a passive tag are lower than those offered by a semi-active or active tag. Range varies from a few tens of centimetres (passive tags) to a hundred meters (active tags). However, for the kind of close proximity recognition that we want to do, the precision offered by passive tags is sufficient. Furthermore, as we mentioned earlier in this paper, it is necessary that the impact of sensors on the patient environment would be the lower possible. Therefore, passive tags are the best choice since it is the smallest, thinnest and lightest tag type. Not to mention that they are much more affordable than the active tags.

3.2. Which Sub-type of Passive Tags Should-we Use?

The kit from Alien Technology provides five different sub-types of passive tags [1]: ALN-9629, ALN-9634, ALN-964X, ALN-9654 and ALN-9662. Although they are all built around a single integrated circuit (Higgs™-3), these tags, however, differ in shape and size. The performances are not identical for the entire range. Remember that all the available tags we have are passive ones. To determine the performance of each type of tag, we measured the RSSI[2] for each tag, from the edge of the antenna until the tag is no longer seen by the antenna. The measurements were made using a small Java program, which for each distance value has done an average of RSSI samples measured over 10 seconds. A standard deviation was also calculated. Furthermore, all measurements were made in the maximum propagation

[2] Received Signal Strength Indication

axis of the antenna. Thus, we were able to determine the maximum range of a tag and assess the accuracy of the measurement using a standard deviation. The goal here was to establish for each tag (if the precision allows it) a trend law which allows us to better assess the distance of the tag from one or more antennas (necessary for locating the objects).

The orientation of the tags about the reader is also a very important issue. Therefore, we recorded test samples for two different orientations: tag and reader arranged in a parallel and a perpendicular way. For each orientation and each tag, we drew, with experimental data, a curve dictating the evolution of the distance, function of RSSI. With these graphs, we were able to estimate the behaviour of these different curves and to produce a representative model of it. It appears from this study that an approximated location can be deduced from the RSSI values. The differences between the two orientations may be important (RSSI can vary from simple to double), but for one tag (ALN-9634) they are very low, negligible given the precision of the whole system. Finally, to maximise RSSI values, measurements were made with a single antenna connected to the RFID reader. Subsequently, we used the two available antennas simultaneously. Considering that the reader shares the power equally between the antennas, in our case, the estimated distance values needed to be divided by two.

3.3. How to Attach Tags with Objects and How Many are Needed?

The first answer we can give to this question is that the solution will depend on the shape of each object. So, on a fork, knife or spoon it is difficult to put more than one tag. These objects can be in many positions, their location will be relative but it will be possible to locate them anyway. For bigger objects, like cup, a jar or a jug, putting a single tag is not a good solution. The problem is that, given a propagation's direction, the mere fact of turning the object would have the effect of giving us a wrong estimation of the distance. Thus, when it is possible, it is more efficient to put at least two tags on a relatively big object, positioned one in front of the other, the copper side pointing outwards. We can now wonder whether it is beneficial to use different types of tags, or a different orientation, in order to better locate an object. For the orientation, the answer is simple: if we use two different directions, since the trend curves are different (see 4.1 for details), it will be difficult to know which formula to apply to a particular tag if we use various orientations. It this therefore better to fix the tag in such a way that its "normal" position will be vertical, then apply the right formula. But if a cup is knocked over, or is reversed, it is in a non-provided position, and the calculation will be wrong. We could put different type of tags on the same object, but we still have the same problem: we don't know if the tag is in the expected position or not. The only way to overcome this problem is to use only one type of tags, so the program will be able to use a single formula in all the cases. Also, we must apply the formula using the most probable position of the object for the calculation. Further attention should be paid to the maximum number of tags that may be present in the field, because it will affect the reception capacity of the reader (accuracy, speed, discernment ability, etc.). In reviewing the results provided by the statistical section, and in light of remarks that were made in this section, it seems logical to use the tag type ALN-9634. This type of tags has a good range and return signal values relatively close, whatever its orientation. Only its size does not allow to use it on very small items like cutlery. We don't use very small objects in our actual platform version but from what we know, for this kind of objects, we recommend using the ALN-964X tag type.

4. Activity Recognition based on RFID Tags

The previous sections detailed the chosen activity model and the hardware configuration, in particular, the choice of sensors and the way to use it. A brief analysis of the individual performance of the five available tags was done. We proposed, based on our conclusions, where to place the tags on the objects, the type, and the number of tags to use for each kind of object. With these basic considerations established, we must now consider which technique to use for activity recognition itself. The following sections will explain in detail the followed path. We first discuss the physical part of the proposed platform and then, the software part. Several subsections will be dedicated to how the data is organized. A section will discuss specifically the temporal aspect, then another will show how to use data mining algorithms on the data to effectively learn and recognize an activity.

4.1. Hardware Setup

In order to identify an activity based on our RFID tags inputs, we have three different types of available information: the location of the tag (actual RSSI value received by the antenna), the direction (a binary information: toward or away from an antenna) and the speed (a numerical estimation). This information is provided by the Alien ITR Technology. After some basic tests, we realized that it was possible to base the recognition process on one simple information, which is the signal strength. The velocity data are not precise enough to detect an “interesting” movement in a given area. For example, to detect that the water jar is moving in a particular way suggesting some water will be dropped. As for the notion of direction, it is not really interesting in our case since it can be inferred directly from the value of the signal strength. As we mentioned earlier, we estimated the rules (statistically) dictating the changes in the distance depending on the signal strength of the tag, relative to a single antenna. To assess the accuracy of measurements, we calculated, for each position a standard deviation value. This value represent between 10 and 30% of the value of a given RSSI sample (depends on the type of tag and its distance from the antenna). Although the precision obtained is limited, it is sufficient if we establish designated zones as a basis for activity recognition. For example, at 20 cm from the antenna, for the ALN-9629 tag, we obtained a standard deviation of 13.7 cm. For the same distance, with the ALN-9662 tag, we obtained a value of 6.6 cm. In order to have more precision in locating the object, we used two antennas. Thus, an object on the table has a good chance of being detected by both antennas. From there, with two values of RSSI, we can estimate the two distances and conclude that the object is present in a particular area. In addition to the areas, “a buffer area” was designed to try to avoid errors in zoning. However, the location of an object is far from simple, because the antennas are not isotropic (an isotropic antenna is an ideal antenna, so the emitted power at a given distance from the antenna is the same in all directions of the space). Moving away from the antenna axis therefore means a certain error on the value of RSSI, error carried over into the position estimation. The real problem is that it is impossible to know whether the object was removed by staying in the axis or if it is at the same distance from the antenna, but not in its axis. To be as fair as possible, the axis of the two antennas is aligned with the location where activity is conducted.

4.2. Platform Test Layout

As we mentioned before, we want to conduct tests as representative as possible on our platform. Therefore, our experimental design is based on the recommendations of the NAT cognitive test [22]. We place all the objects on a 1 by 1.5 meter table. So, to introduce the activity recognition, we placed antennas on the table at locations X and Y (see Figure 3), oriented in the shown direction (to the user). We created areas on the table such as it can be seen in the figure. Initially, all the objects necessary for the activity *prepare coffee* are disposed in the area A0. The area A1 can then be used to dispose items needed for a second activity, for instance making toasts (another activity proposed by the NAT). The area A2 is where the patient will perform the activities. The area A3 is the “buffer” zone it is made up of areas not used by other areas and aims to reduce zoning errors. To complete an activity, the patient uses the objects by taking them between the different areas.

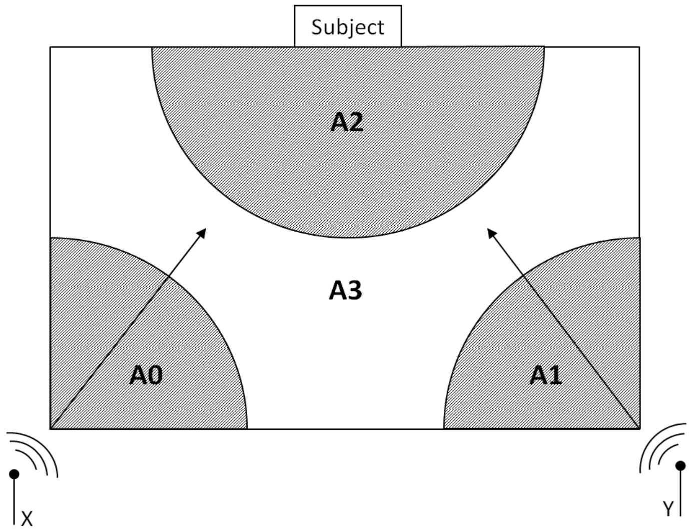

Figure 3. Diagram Showing the Organization of the Different Areas

4.3. Software Setup

For the software part, the work has been facilitated by the use of the API provided with the RFID kit [1]. For various reasons, including portability, we decided to code the software component of the recognition platform in Java, using the Eclipse IDE. To use data mining’s techniques, we chose the open source software Weka [12]. This library provides a Java implementation of a lot of useful data mining tools: many filters, algorithms, graphical representations, etc. Using it requires specific files or databases containing instances that will be used for learning patterns which here corresponds to activities performances. The software architecture presented in Figure 4 is explained below.

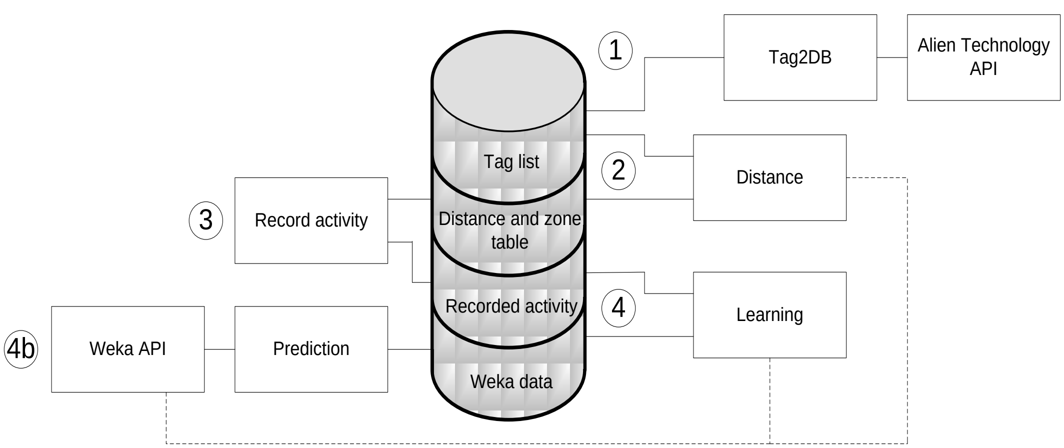

Figure 4. Software Architecture of the Recognition Platform

The first part (1) of the architecture is truly a bridge between the hardware and the software part. Its aims is to maintain, in real time, a database (MySQL) that contains a list of the different present tags with information about them, which is later used for localization (tag id, RSSI value, etc.). The second part (2) of the program calculates an estimation of the distance (works with one or two antennas) from objects that are detected. This estimation is done using regression rules established in a statistical way. Each type of used tag has its own estimation formula. As an example, for the ALN-9634 tag, it is $d = 270{,}96e^{(-1.10^{-4}x)}$ with *x* representing the RSSI value and *d* the estimated distance in centimeters. From these, the position of the object is assigned to a zone. This assignment to an area can be done in two ways: either from fixed rules or by using a data mining approach. In the last case, the area estimation is based on a database in which records were from the area in terms of distances estimated by the antennas. Now, in the database, we can found the detected objects with their respective parameters, the estimated distances relative to each antenna, and the area in which they are. The third step (3) record the activity done by the patient. The used format can be seen in the section 4.5. The last steps (4 and 4b) aim to analyse the recorded activity data, either to learn an activity or to make a prediction. The step (4) transforms raw data into a data representation usable by Weka to learn patterns. As we will further explain later, the step (4b) uses Weka recorded data to recognize the on-going activity.

4.4. Activity Recognition Using a Data Mining Approach

A lot of works on activity recognition exploit hidden Markov models (HMM) to be able to infer a specific activity [10, 16, 23]. In our case, we decided to exploit data mining techniques. The goal of this process is to uncover statistically “hidden” facts, rules or patterns in a large amount of data. One of the advantages of data mining compared to the use of HMM is that no handcrafted estimate is necessary. The learning data fully reflects the activities that have been done on the test platform. First, we had to address the issue of choosing which data to use. Our approach was as follows: try to save as much information as possible on an activity and then apply the activity recognition itself. Once the learning is achieved with the chosen data mining technique, it only remains to compare a new instance, and infer the desired parameter. In our case we use a decision tree: the C4.5 algorithm [19], called J48 in the Weka library.

4.5. Activity Record

To operate, Weka must have a file or a database containing instances for a class of choice, *i.e.* a number of parameters (numerical values, time stamps, another class) to conclude on a discrete class of choice, in our case actions/activities that are done. In addition, these instances must have the same size, omissions are possible, but the quality of learning will be affected. For simplicity, we based our system on two basic actions: *take* and *return*. An object can be taken from or returned to a particular zone. But the fact of taking or returning an object will produce several measurements (we will follow the evolution of data), the exacted number is unknown and depends on the environment (type of present materials) or the speed with which the action is performed. To use Weka, we therefore had to find a way to represent an unknown quantity of data in a single instance of fixed and known size.

id	pos_ant_1	pos_ant_2	dir_ant_1	dir_ant_2	vit_ant_1	vit_ant_2	temps
tasse_verte	114,343155	82,90406	1	1	0	0	10:12:49
tasse_verte	106,86689	76,53774	1	1	0	0	10:12:52
tasse_verte	111,28272	99,657974	1	1	0	0	10:12:53
tasse_verte	111,87856	81,83819	1	1	0	0	10:12:53
tasse_verte	111,87856	96,50975	1	1	0	0	10:12:54
tasse_verte	86,370316	96,50975	-1	1	-0,577	0	10:12:54
tasse_verte	69,27776	96,50975	-1	1	-0,304	0	10:12:55
tasse_verte	19,004656	96,50975	-1	1	-0,322	0	10:12:56
tasse_verte	19,004656	117,195564	-1	1	-0,322	0	10:12:56
tasse_verte	35,000908	117,195564	1	1	0	0	10:12:57
tasse_verte	10,019707	109,00612	1	1	0	0	10:12:57
tasse_verte	36,380905	109,00612	1	1	0	0	10:12:58

Figure 5. Events Recorded for the Action *return green cup*

Figure 5 shows the variation of such parameters as perceived by the system for the action *return green cup*. In this figure, *id* represents the name of the object, like green cup (in French here "tasse_verte") or sugar bowl. *Pos_ant_x* represents the estimated distance from an antenna *x* to the object. *Dir_ant_x* is the direction of the object related to the same antenna *x* and *vit_ant_x* is its estimated speed. The last column is used to record the duration. We can see that the estimated position values are sometimes identical. The program has a system that allows it, for each object, to ignore the record *n* when values of the two antennas position record *n + 1* are considered identical, in our case when the difference is less than 5% (adjustable value). Speed and direction values are saved, even if they are not actually used.

4.6. Base Frame Elaboration

To build the basic instances, which will contain the basic actions *take* and *return*, the program uses two thresholds: a low threshold and a high threshold. When objects are in the area A1, the estimated distance values are below the lower threshold, while in the area A2, the estimated values exceed the upper threshold (see Figure 6). When the program finds no value below the low threshold or above the high threshold, it will increment the below threshold and decrement the upper threshold then perform a new search. From this, events can be decomposed into basic actions. For the moment we are limited to two consecutive basic actions (problems on stop loop conditions). Once the basic actions are known, the program determines, for each of them, a number of parameters including the object name, departure date and duration (in ms) of the action. The maximum and minimum values, and the slope values for the two antennas (calculated with a linear regression on distance data) are also added. Slots are provided for data direction and speed, but are not used for now. Finally, a basic frame consists of thirteen different fields. An example of a basic frame is given below.

Green cup	33.5	3.2	98.7	0	0	-3.9	89.1	117.5	0	0	09:29:23	2000

$\Downarrow$

Base frame *n*

Figure 6. Basic Frame Generated for the *take green cup* Base Action

As you may have noticed, the action itself is not contained in the basic frame, it is added subsequently, during the manufacture of the final frame. The following paragraph describes how these finals frames are built.

4.7. Final Frame Creation

An activity can obviously contain several basic actions, which involves several basic frames. For example, if the user takes the cup, the action *take cup* will be recognized and there will be a single base frame. Take another example: if the patient performs the activity's step *add sugar*. A possible sequence of basic actions may be as follows: *take the cup, take the sugar, add the sugar, then return the sugar,* which in this case involves three basic recognizable actions, thus three basic frames. Whether for learning or predicting, all the built basic frames are encapsulated in a final frame. A final frame may contain a number *n* of basic frames. The following Figure 7 illustrates that.

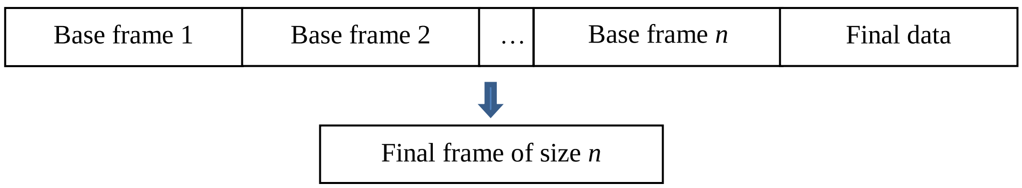

Figure 7. Establishment of a Final Frame

Once the basic frames are added, as shown in Figure 7, it also adds final data. The data consists of three fields: the starting date, duration of the total activity record, and the name action performed by the patient. Once the final frame constructed, it is either stored in the database properly, either used to make a prediction. Thus, the final frame of the activity's step *add sugar* will contain the three previously presented basic frames, plus the final data block. As we previously mentioned, the size of the final instance (relative size to the number of basic frames it contains) may vary. This means that actions having the same size will be recorded in a single table in the database. This also implies the use of not one but several decision trees, in fact one for each final frame size.

4.8. Temporal Aspect

Temporal data are important in the field of activity recognition [23], so the basic frames, and the final data block, contain time relative information. The final frame of an activity's step contains a large number of temporal information, enough to represent the actions of the user in the form of a timeline. Below, the Figure 8 shows a final frame containing three basic frames, this activity's step is so made of three basic actions.

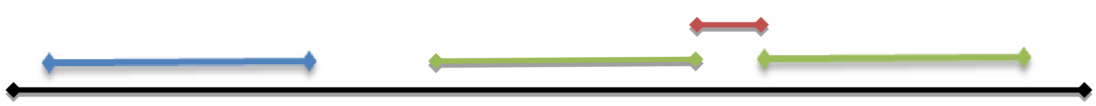

Figure 8. Example of an Obtained Timeline

In this case, the black line represents the total recorded length, blue and green lines the durations of basic actions (same color represents the same object). The information contained in the three base frames can say that the first and second actions are *take* type and that the third base action type is *return*. Considering that the blue color represents the *green cup* object and the green color represents the object *sugar*, then we can reasonably assume that the activity that took place on the table was *add sugar*. Furthermore, we can know the time taken to complete the addition of sugar (represented by the red line). Downtimes are also visible.

4.9. Use of the C4.5 Algorithm on Data

C4.5 algorithm was proposed by [19] in 1993. It generates decision trees used to make classifications (training part), then to predict a missing class attribute in a set of data. Figure 9 presents its pseudo-code. It uses the fact that each attribute of the data can be used to make a decision by splitting the data into smaller subsets.

```
Check for base cases
For each attribute a
    Find the normalized information gain from splitting on a
Let a_best be the attribute with the highest normalized information gain
Create a decision node that splits on a_best
Recur on the sub lists obtained by splitting on a_best, and add those nodes as children
of node
```

Figure 9. Pseudo-code of the C4.5 Algorithm used in our Recognition System

Before using the decision tree algorithm, we must proceed to a learning phase. The data is recorded as we have previously explained. A table is created for each size of a final frame. If the size (the number of basic frames) of the final frame is *n* and is greater than 2, then another record is added in a new table. This recording of size *n + 1* consists of *n* predictions obtained by applying the C4.5 algorithm on the size 1 data, which will then be used for learning. As you can imagine, the remaining cell host the performed activity, as specified by the operator during the recording procedure. For proper operation, learning activities must begin with the basic actions *take* and *return*, performed with all different available objects, otherwise the second method would fail. For the recognition of the ongoing activity, two different methods are available. The first choice allows constructing a decision tree on all the recorded data**,** applying the C4.5 algorithm once on all data fields (an option allows the non-use of the date type attributes). Figure 10 describes in details all the followed steps to infer activity with this first method.

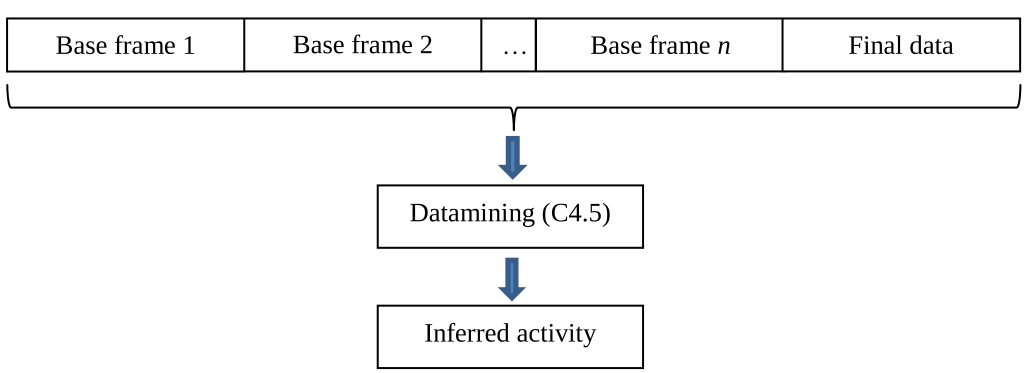

Figure 10. First Prediction Method

The second choice is only operational for activities involving more than two basic actions (size of 2). In a first-hand it uses the C4.5 algorithm to infer a basic action on the basic frames, subdued in a final frame: for a size *n*, *n* predicates are consequently obtained. Then the predicted activity can only be of the type *take objet* or *return objet*, *object* representing any available objects for testing. The second step uses another C4.5 tree, this time from the results (the *n* predicates) to predict the ongoing activity. Figure 11 summarise this explanation.

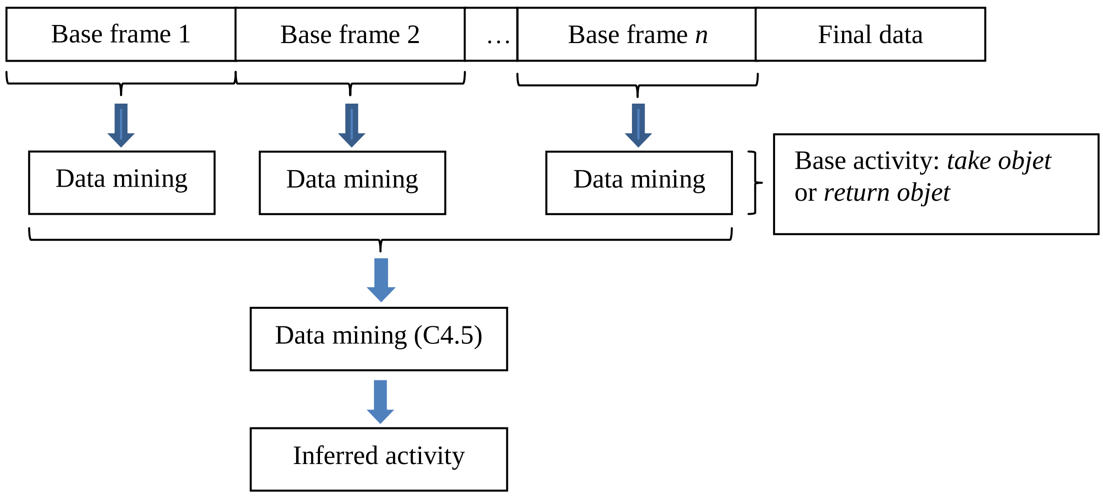

Figure 11. Second Prediction Method

The second prediction method presented here has been implemented because, during testing, it happened that the C4.5 algorithm operations was only based on objects' names in the frames, and didn't really reflects the movement's data. This method ensures that the concept of movement is taken into account and encapsulated by the two sub-actions *take* and *return*. Therefore, an object can only be taken from a zone or returned to a zone.

5. Experiments

We conduct a first experiment on a table about 1 by 1.5m, on which we placed two antennas, following the set up presented in Figure 3. These antennas are pointed towards the user. We used four items: a cup, a water jar, a milk jug and a sugar box. They allow performing the activity *prepare coffee*. Each object has been tagged using two ALN-9634 type RFID tags.

5.1. Used Learning Protocol

We started the learning stage with the four available objects. The possible actions were as follows: *take objet* or *return objet*, with *object* representing any available objects. Each basic action has been performed 10 times. The table contains only one basic action by record and has therefore 80 records. We conducted tests on each step of the activity *prepare coffee* with several basic actions, and this in an ascending order. So, we have tested each of the following steps: *add water*, *add sugar* and *add milk*. These activity's steps can be inferred when the basic actions corresponding actions are detected. For example, with the step *add sugar*, the basic actions are *take sugar* and *take cup*, or vice versa. For basic actions, 10 recordings were made. The 2 size table therefore includes 30 records. For the table of size 3, we can add other combined actions, such as: *add water and sugar*, *add sugar and milk* and *add water and milk* and more possible actions already present in the table of size 2, which will this time the result of three basic actions. For instance, for the *add sugar* action, we can imagine the following sequence: *take cup*, *take sugar* then *return sugar*. Again, each action is repeated 10 times, so there is 60 records in the table of size 3. So we used up to three tagged objects.

During the learning phase, several elements were taken into account. First, we noted when the program failed in the separation of basic actions (yes or no). Secondly, we tested the learning phase and recognition phase separately. Therefore, before adding the record in the database, we make sure that it was of good quality. If there were some non-correct values, we modified them, before the recognition process, in order to have a consistent training data set. Indeed, if learning is poor, it would be very difficult to conclude about the recognition capabilities of the system. We specified each time the amount of data that have been either deleted or modified. Learning is also done with the objects themselves, without content.

5.2. Test Protocol

For our experimentation, we followed a similar approach to that used for learning. We tested five times all of the possible activity steps. There were therefore a total of 85 tests to be performed. These tests have been conducted with the two prediction methods presented in section 4.9. The first step was to determine if the program fails or not in the separation of the basic actions (yes or no). If it was successful, then we started the activity recognition with both methods and we noted the result. We looked up the data in detail, if values were not correct, they were corrected, this allowed trying the activity recognition, and this, by overriding the hardware part of the platform, considering our sensors as perfect.

5.3. Obtained Results

After having carefully conducted this experiment, we constructed tables to conclude on several aspects of the system. The graph featured below in Figure 12 summarize these points. The following figure highlights two elements: the failure rate of basic actions separation (before any correction) depending on size (relative to the number of objects used in this case) and the percentage of corrected samples. We consider all the tests conducted, that is to say, learning and recognition together, a total of 100 tests for one used object, 105 for two used objects and 45 for 3 used objects.

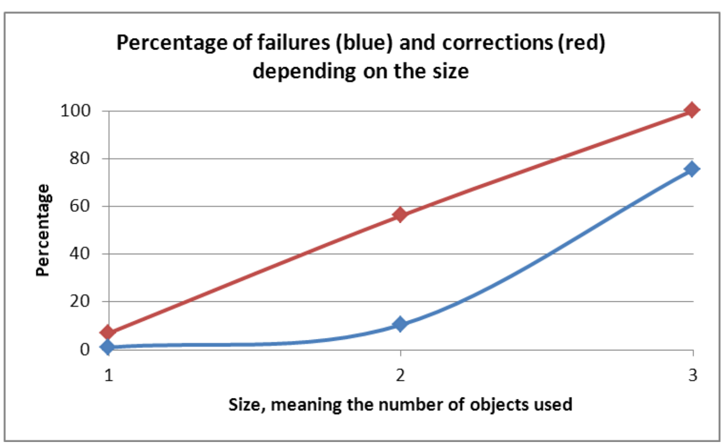

Figure 12. Percentage of Failure and Correction for All Tests

Figure 12 shows the obtained results. We can see the influence of the number of items on reader performance. In the case of a single basic performed action with one object, the software works well (only 1 failure in 100 tests). When the number of objects increases, the performance tend to degrades. Hence, with two objects, the success rate of software is about 89%. With three simultaneous objects, it decreases significantly and reach 25%. It is the same for the percentage of corrected values which also evolves in this direction. Besides the issues related to the reader itself when the number of items increases (slower, less accurate), we can point out another issue easily observable in the data. There is a mutual interference between tags leading to an inaccurate estimated distance. Because of the use of thresholds, the software cannot further separate basic actions. This phenomenon is relatively important when the tags are in their starting areas, *i.e.* the area A0 for the *prepare coffee* activity. This blurring disappears when the objects passed away in area A2. The estimated distances are then correct for all objects. You'll notice that we did not use content during testing, *i.e.* the water jar was not full of water and the milk jug was empty. We anticipated that liquid may interfere with the signal strength. Nevertheless, whatever the objects contain liquid or not, we noticed interference with the signal when they are close to each other.

Table 1 shows the results of recognition rate for the three studied sizes of actions for both possible prediction methods (for sizes 2 and 3 only).

Table 1. Percentage of Failed Activity Recognition for both Methods

	Before correction		After correction	
	method 1	method 2	method 1	method 2
size 1	7,5	NaN	0	NaN
size 2	6,7	40	0	0
size 3	70	63,3	40	26,7

Once the final frames is stored in a database, the use of data mining techniques, in particular the C4.5 decision tree algorithm, work well when the size (the number of basis frames) is low. More it increases, more the algorithm has difficulty to operate effectively. This problem may be due to a too small sample of data feeding the algorithm. So, for the recognition of basic actions, the success rate before correction is 92%, and 100% after correction of the possible irregularities in the data. For operations with two basic frames, the first method is 93% accurate, when the second has a 60% exactitude rate. This result seems logical since the second method uses more data mining operations, therefore its success rate is statistically lower. In terms of size 3, the recognition rate is relatively low before correction (30% for the first method and 43% for the second), this can be explained because of the high failure treatment rate of the software (75%). After corrections, the success rate in recognizing actions increases, respectively at 60% and 73%. It seems that the sample size for learning in the case of activities involving three basic actions is too low. It would be interesting to increase the number of records in order to improve the learning phase and to re-test the two activity recognition methods. Finally, with a consistent and sufficiently large data sets, the data mining solution seems suitable for such recognition system.

We can compare our results to the work of [16], which is quite similar in nature to ours. He exploited a RFID reader, mounted on a glove wore by the person, and a bunch of tagged objects. His recognition system use a HMM and DBN (Dynamic Bays Net). If we base our comparison on the percentage of the time where actions are correctly inferred (first metric in the work of Patterson), we can see that the success rate is quite similar to results presented in

Patterson's work. But we cannot really go further than that, because the chosen activity and recognition models are different in nature, complexity and size in both work.

6. Conclusion and Perspective

The aging of the population constitutes the challenge of the next decade. New solutions to this challenge can be found through the concept of smart home [2]. Activity recognition, as one of the most studied and complex problem, needs to be solved to enable this technology. As we shown, many researchers have addressed this problem [4; 10; 15; 20], however very few works exploring this issue offer a complete solution for implementing a working recognition system and experimenting it from A to Z. Most of the existing models and platforms for experimenting such recognition systems use heavy non-transportable infrastructures [9, 13, 7, 8] exploiting a complex assortment of heterogeneous sensors. This kind of infrastructure is very expensive, uneasy to deploy, complex to implement, complicated to maintain, and cannot be moved to the patient's home. Testing in this kind of environment, unfamiliar to the cognitively-impaired patient, poses many problems. To address these issues, we proposed in this paper a complete solution for deploying a simple activity recognition system: the model of the activity, the design, the choice of sensors, the implementation, and the recognition algorithms. This recognition platform that we offer is based on a recognized cognitive test [22], is simple, extendable, and easily transportable (a table of 1x1.5m, RFID reader, two antennas, a set of few light objects, and a computer). We also presented a first validation phase, conducted on our system. The results showed that the use of RFID tags associated with a data mining-based analysis (C4.5 decision tree algorithm [19]) works reasonably well. We initially chose to focus on the recognition of stages of a single activity: *prepare coffee*. On further works, we plan to test the recognition of steps from several activities at the same time. To enhance the platform, we also plan to add few activities based on the NAT recommendations.

This work constitutes a good first step toward the development a simple, flexible, robust and portable activity recognition platform that will allow testing recognition solutions directly at the patient's house. Nevertheless, several discussed issues, mainly related to physical sensors limitations, remains a challenge that we need to address in order to improve the efficiency and the robustness of the system. The use of RFID sensors designed for harsh environments could be solutions to consider for future work. It should be noted that many further experiments will be needed to establish more consistent database that will allow a better quality of learning, and thus a logically greater accuracy in predictions. Moreover, experiments conducted directly on the field with actual patients will be required. It is why a second experimental phase is already planned and will be conducted with Alzheimer patients using the platform enhanced to take into account several activities. We signed a formal collaboration agreement with the regional rehabilitation center Cleophas-Claveau of La Baie (QC), Canada. This center is able to provide us with an adequate group of cognitively-impaired people for our experiment, mostly Alzheimer's patients at moderate stages. We already have been approved by the ethical committee to make a clinical experiment with a group of 20 people suffering of Alzheimer disease. The multidisciplinary team that will manage this second phase is composed of a neuropsychologist researcher, two smart home researchers, and several grad and undergraduate students in psychology, computer science and engineering. We hope to be able to recruit patients and begin the experimentation in 2011.

Acknowledgement

We would like to thank our main sponsors for their financial support: the Natural Sciences and Engineering Research Council of Canada (NSERC), the Quebec Research Fund on Nature and Technologies (FQRNT), The Canadian Foundation for Innovation (CFI), The UQAC Foundation and our University. We also thank our precious partners at the Cleophas-Claveau medical center of La Baie providing us access to patients through our formal collaboration accord. Finally, a special thank to our neuropsychologist colleagues Julie Bouchard, Audrey Potvin and Hugo Laprise working with us on the smart home project at the LIARA laboratory.

References

[1] Alien http://www.alientechnology.com.

[2] J. C. Augusto and C. D. Nugent, *Designing Smart Homes: The Role of Artificial Intelligence*, Vol. 4008 of Lecture Notes in Artificial Intelligence (LNAI), Springer, **(2006)**.

[3] A. Bharucha, V. Anand, J. Forlizzi, M. A. Dew, C. F. Reynolds and S. Stevens, "Intelligent assistive technology application to dementia care: current capabilities, limitations and future chalenges", *American Journal on Geriatric Psychiatry*, vol. 17(2), pp. 88-104, **(2009)**.

[4] B. Bouchard, A. Bouzouane and S. Giroux, "A Keyhole Plan Recognition Model for Alzheimer's Patients: First Results", *Journal of Applied Artificial Intelligence* (AAI), Taylor & Francis, Vol. 22 (7), pp. 623–658, **(2007)**.

[5] M. Buettner, R. Prasad, M. Philipose and D. Wetherall, "Recognizing Daily Activities with RFID-Based Sensors", *In ubicomb*, **(2009)**.

[6] T. Choudhury, S. Consolvo, B. Harrison, J. Hightower, A. LaMarca, L. LeGrand, A. Rahimi, A. Rea, G. Borriello, B. Hemingway, P. Klasnja, K. Koscher, J. A. Landay, J. Lester, D. Wyatt and D. Haehnel, "The Mobile Sensing Platform An Embedded Activity Recognition System", IEEE Pervasive computing vol. 7(2), pp. 32-41, **(2008)**.

[7] D. J. Cook, M. Youngblood and S. K. Das, "A multi-agent approach to controlling a smart environnement", In Augusto J.C., and Nugent C.D., eds., *Designing Smart Home: The Role of Artificial Intelligence*, volume 4008 of *Lecture Notes in Artificial Inteligence*, Springer, pp. 53-83, **(2006)**.

[8] S. Giroux, T. Leblanc, A. Bouzouane, B. Bouchard, H. Pigot and J. Bauchet, "The Praxis of Cognitive Assistance in Smart Homes", in *Ambient Intelligence and Smart Environments,* IOS, Volume 3, pp. 183-211, **(2009)**.

[9] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura and E. Jansen, "The Gator Tech Smart House: A programmable pervasive space", Computer, Vol. 38(3), IEEE Computer Society Press, pp. 50-60, **(2005)**.

[10] J. Hoey, A. von Bertoldi, P. Poupart and A. Mihailidis, "Assisting persons with dementia during handwashing using a partially observable Markov decision process", *International Conference on Vision Systems (ICVS'07),* Best Paper, pp. 89-99, **(2007)**.

[11] X. Hong, C. Nugent, M. Mulvenna, S. McClean, B. Scotney and S. Devlin, "Evidential fusion of sensor data for activity recognition in smart homes", *Pervasive and Mobile Computing*, Elsevier, Vol. 5(3), pp. 236-252, **(2009)**.

[12] I. H. Witten and E. Frank, "Data Mining Practical Machine Learning Tools and Techniques", Morgan Kaufmann Publishers, **(2005)**.

[13] A. Mihailidis, J. Boger and T. Craig, "The COACH prompting system to assist older adults with dementia through handwashing: Am efficacy study", *BMC Geriatrics*, 8(28), **(2008)**.

[14] A. N. Nambiar, "RFID Technology: A Review of its Applications", Proceedings of the World Congress on Engineering and Computer Science, Vol 2, **(2009)** October.

[15] R. Orpwood, C. Gibbs, T. Adlam, R. Faulkner and D. Meegahawatte, "The design of smart homes for people with dementia-user-interface aspects", *Universal Access in the Information Society*, 4(2), pp. 156-164, **(2005)**.

[16] D. J. Patterson, D. Fox, H. Kautz and M. Philipose, "Fine-Grained Activity Recognition by Aggregating Abstract Object Usage", IEEE International Symposium on Wearable Computers, pp. 44-51, **(2005)**.

[17] D. J. Patterson, L. Liao, D. Fox and H. Kautz, "Inferring High-Level Behavior from Low-Level Sensors", Ubicomp 2003: Ubiquitous Computing In Ubicomp 2003: Ubiquitous Computing, pp. 73-89, **(2003)**.

[18] H. Pigot, A. Mayers and S. Giroux, "The intelligent habitat and everyday life activity support", *5th Int. Conference on Simulations in Biomedicine,* IOS press, Ljubljana, Slovenia, pp. 507-5016, **(2003)**.

[19] J. R. Quinlan, "C4.5: Programs for Machine Learning", Morgan Kaufmann Publishers, **(1993)**.

[20] M. Rowe, S. Lane and C. Phillips, "CareWatch: a home monitoring systems for use in smart homes of persons with cognitive impairment", *Journal of Geriatric Rehabilitation*, (23), pp. 3-8, **(2007)**.
[21] P. C. Roy, B. Bouchard, A. Bouzouane and S. Giroux, "Challenging issues of ambient activity recognition for cognitive assistance", Handbook of Research on Ambient Intelligence and Smart Environments: Trends and Perspectives, Mastrogiovanni, F. and Chong, N. (Eds.), IGI global, pp.1-32, **(2010)**.
[22] M. Schwartz, M. Segal, T. Veramonti, M. Ferraro and L. J. Buxbaum, "The Naturalistic Action Test: A standardised assessment for everyday action impairment", *Neuropsychological Rehabilitation*, 12(4), pp. 311-339, **(2002)**.
[23] G. Singla, D. J. Cook and M. Schmitter-Edgecombe, "Incorporating Temporal Reasoning into Activity Recognition for Smart Home Residents", AAAI Workshop on Spatial and Temporal Reasoning, pp. 53-61, **(2008)**.
[24] S. J. Pan, J. T. Kwok, Q. Yang and J. J. Pan, "Adaptive localization in a dynamic WiFi environment through multi-view learning", In Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI'07), Vancouver, British Columbia, Canada, pp. 1108-1113, **(2007)** July.
[25] K. K. van Laerhoven and H. Gellersen, "Spine Versus Porcupine: A Study in Distributed Wearable Activity Recogintion", In Proc. IEEE Int. Symp. on Wearable Computers, Vol. 1, pp. 142-149, **(2004)** October.
[26] R. Want, "An Introduction to RFID Technology", *IEEE Pervasive Computing*, Volume 5(1), pp. 25-33, **(2006)**.