Inference Engine Based on a Hierarchical Structure for Detecting Everyday Activities Within the Home

Usman Naeem[1]([✉]), Abdel-Rahman Tawil[1], Ivans Semelis[1],
Muhammad Awais Azam[2], and Mustansar Ali Ghazanfar[3]

[1] School of Architecture, Computing and Engineering,
University of East London, London, UK
{u.naeem,a.r.tawil,U0909617}@uel.ac.uk
[2] Department of Computer Engineering,
University of Engineering and Technology, Taxila, Taxila, Pakistan
awais.azam@uettaxila.edu.pk
[3] Department of Software Engineering,
University of Engineering and Technology, Taxila, Taxila, Pakistan
mustansar.ali@uettaxila.edu.pk

Abstract. One of the key objectives of an ambient assisted living environment is to enable elderly people to lead a healthy and independent life. These assisted environments have the capability to capture and infer activities performed by individuals, which can be useful for providing assistance and tracking functional decline among the elderly community. This paper presents an activity recognition engine based on a hierarchical structure, which allows modelling, representation and recognition of Activities of Daily Life (ADLs), their associated tasks, objects, relationships and dependencies. The structure of this contextual information plays a vital role in conducting accurate ADL recognition. The recognition performance of the inference engine has been validated with a series of experiments based on object usage data collected within the home environment.

Keywords: Activity recognition · Alzheimer's disease · Assisted living · Elderly monitoring · Activities of daily life · Activity modelling · Inference engine

1 Introduction

We live in an age where we are witnessing a significant demographic change, especially in the western world where it has been projected that the number of people aged 65 and over will increase by 23% from 10.3 million to 12.7 million in 2018. By 2035, this figure will be expected to reach 16.9 million [1]. This scenario leads to a range of challenges for the government, as it can lead to a financial burden on the National Health Services (NHS), welfare and pension schemes. As the population ages there are increasing numbers of elderly people in society, less carers' available and less money to pay for care. Therefore, we look to technological solutions to reduce the need for human carers. One specific way to reduce the burden on the health system is to create

an environment that promotes independent and healthy living for the aging population. Having additional years of independence will not only help the elderly lead an independent life, but it will also lessen the financial burden on the local authorities, NHS and families. The ability to lead an independent life is dependent on how well an elderly person can conduct everyday activities such as personal dressing, cooking, bathing and cleaning [2]. These are known as Activities of Daily Living (ADL), whose recognition plays a crucial role in observing and tracking any functional decline [3]. Useful *information* about the safety and healthy wellbeing of an elderly person cannot only help them lead an independent life but can also allow the possibility of instituting safeguards given a potential harmful scenario. The work in this paper aims to establish a reliable inference engine for unobtrusively monitoring and identifying activities of individuals within a home environment.

This paper makes the following contributions. Firstly, we introduce a novel concept of modelling and recognising ADLs as a hierarchical encapsulated entity, where each ADL has attributes that enable the inference engine to reason the internal structure and relationships of an ADL when carrying out recognition. The remainder of the paper is organised as follows. Section 2 provides an overview of the related literature, while Sect. 3 describes the structure and the key characteristics of a hierarchical structured ADL. Section 4 describes the inner workings of the inference engine and how it manages and recognises the hierarchically structured ADLs. Section 5 describes the experimental set up of home environment followed by the results that validate findings about the inference system.

2   Related Work

The ability to recognise an individual's activities within an ambient assisted living environment is very much dependent on reliable feature detection techniques and the construction of human activity models.

Feature detection can be carried out using visual based systems, which can be computationally expensive when analysing video footage and can be seen as intrusive. However, the contribution of vision-based systems should not be ignored, as there is a large body of work within this area. In addition, the activity recognition domain can be complex; hence, solutions based on the fusion of multiple sensors (including vision sensors) should not be overlooked.

An alternative to visual based systems is the use of anonymous binary sensors such as: motion detectors, break-beam sensors, pressure mats, and contact switches. These can aid the process of tracking an individual around the home and complement the whole activity recognition process [4]. However, these types of systems do not have capability of remote monitoring of data. Additionally, it is not possible to have knowledge of the context or the sequence of activities being monitored.

Wearing different types of sensors around the body is another technique for capturing features related to activities or posture [5]. These types of sensors are known as

wearable sensors and they can provide contextual information, e.g. accelerometer, gyroscope and proximity sensor etc. Such wearable sensors can work either as individual devices or as part of other devices [6], which have the ability to determine physical activities such as walking, running, climbing stairs and sitting [7–9]. It is possible that the data collected through such sensors may be useful for particular application domain such as social relationship or health care scenarios, it is not very useful while in isolation and complex activities are being detected.

Unique identification of individuals and recognition of activities performed by them can be achieved through sensors and passive transponders on objects within the home environment. Radio Frequency Identification (RFID) technologies have become a common source of capturing object usage data non-intrusively [10] for activity recognition. This type of feature detection is known as "Dense Sensing" [11, 12]. This name comes from the concept that every individual object that can be used during different activities, get tagged with passive wireless battery-free transponders that transmit information to a computer via a RFID reader [13, 14] when the object is used or touched. Unobtrusiveness and easy installation are few major advantages of using passive transponders. In addition, these passive transponders are not reliant on battery power; hence, they can be deployed within the home environment for a very long time. However, "Dense Sensing" does have its share of flaws. Firstly, the capturing of object usage data from the transponders is dependent on the end user (participants) to wear RFID reader on their hand or finger, which is bulky and requires regular charging. Secondly, the presence of metal or water can interfere with the signals, which can have detrimental effect on the recognition. In addition, trying to capture object usage data for small objects can be problematic, as the end user is likely to hold the object with their handing covering the passive transponder, which leads to a situation where no signal is received in order to confirm that the object has been touched [15].

Capturing noise-free reliable data only solves half of the activity recognition problem, as a vital component is based on the construction of human activity models, which make it possible to detect and predict activities from the captured stream of data. The most popular models within this area of work include Hidden Markov Models [16], Naïve Bayes classifiers [10] and Bayesian Networks [17]. Unfortunately, such approaches are not very reliable when trying to recognise activities carried out in a random order, which is a typical situation in typical daily life activities [16]. Another criticism is that these approaches can sometimes suffer from a 'cold start', as large datasets are required to carry out robust recognition [18].

Ontology-based approaches [19, 20] are a viable option for building robust activity models as they exploit the semantics of an ADL, which is based on the observation of a user's current context such as current location, current time, and objects used to perform the activity.

Other challenges associated with ADL recognition approaches is scalability. One such top-down, goal driven approach [21] addressed this by structuring activities in hierarchical manner, which was made up of abstract sensor mappings and series of

execution conditions. The work proposed in this paper carries out a similar function, as it also structures ADLs as a hierarchical entity.

Two core classes in which different proposed solutions for activity recognition lays are inductive and deductive. Potential of inductive class is to learn and generalize by example [22, 23] whereas, deductive methods provides powerful means to encode semantic process knowledge [24]. Both frameworks have their benefits and limitations and the ultimate solution would be the one bringing the best of both worlds. In relation to this, the proposed hierarchical approach aims to achieve this, as the lower task recognition tier is based on an inductive framework, while the higher tier ADL recognition is based on a deductive framework.

Existing approaches for ADL inference have been focused on classification techniques that have been based on pattern recognition. The primary objective of these approaches are based on designing models that are capable of recognising activities given sequences of observable [25, 26], which can be then used to deduce behavioural patterns.

The work proposed in this paper differs from traditional classification techniques, as it has the ability to accommodate multi-layered contextual scenarios by proposing a hierarchical structure for the modelling, representation and recognition of the ADLs, its associated tasks, objects, dependencies and their relationships. The organisation of this information in a contextual structure plays a key role in carrying out robust ADL recognition.

3   ADL Model Structure

ADLs have been modelled in a hierarchical structure, where the lowest tier is responsible for feature detection. Features are captured as data streams, which are known as sensor events. Each sensor event represents the movement of an object (e.g. Tap motion has occurred) or the presence of a person entering a zone within an environment (e.g. John has entered the sink zone within the kitchen).

Hence, a sensor event is used to represent a person within a zone or the movement of an object (Fig. 2).

These sensor events are then associated with actions, while zones are associated with objects. For example, in Fig. 1, a kettle motion sensor event can be associated with the action Kettle used.

3.1   Knowledge Base of ADL Characteristics

3.1.1   Sub-ADL and Action Attributes

Before discussing the recognition framework, it is important to highlight the key attributes and characteristics that form the information stored in the knowledge base (see Fig. 1). The attributes in the knowledge base are associated with the Sub-ADLs and actions within each ADL, as these are utilised for recognising ADLs (see Table 1) based on their characteristics.
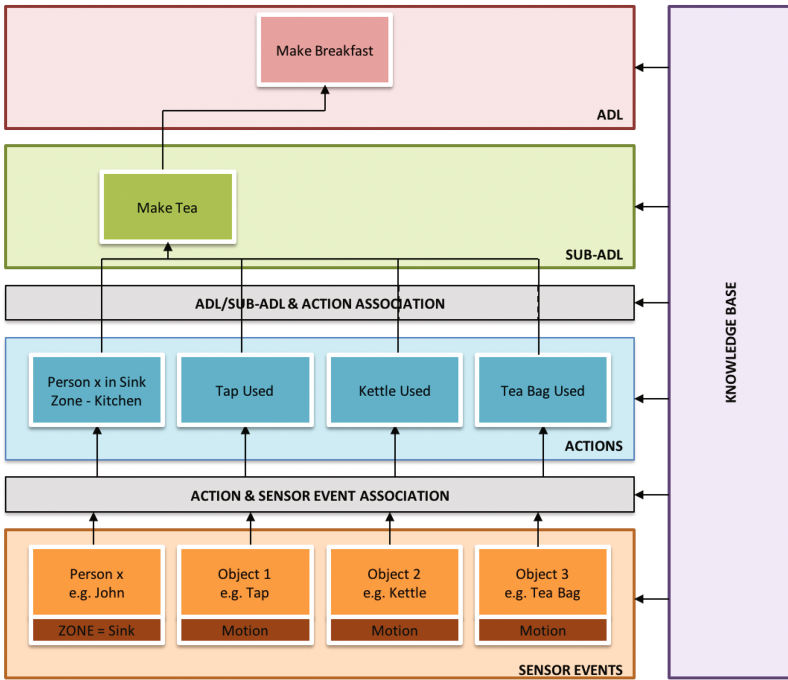
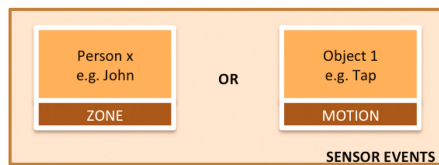**Fig. 1.** Hierarchical structure of make breakfast ADL and make tea Sub-ADL

Fig. 2. Sensor event representation

An ADL encompasses Sub-ADLs and actions, as each of them has attributes associated with the ADL they belong to. For example, the action use of toilet roll will be observed more frequently for defecation as opposed to urination ADL.

3.1.2 ADL Attributes

Like the attributes in Table 1, ADLs have attributes that are required for the recognition process. These are based on characteristics of the relationships between all the possible ADLs that have been modelled.

The attributes described in Tables 1 and 2 collectively form the knowledge model necessary to bootstrap the system for initial ADL Recognition. The information in the knowledge model can be adjusted or modified based on the location setting in order to suit the current environment.

Table 1. Sub-ADLs and action attributes

Attributes	Description
Maximum duration	This is threshold of the maximum duration of performing an action
Minimum duration	This is threshold of the minimum duration of performing an action
Maximum occurrence	This is threshold of the maximum number of times that a certain Sub-ADL or action may occur in an activity
Minimum occurrence	This is threshold of the minimum number of times that a certain Sub-ADL or action may occur in an activity. This also determines if an action is mandatory or optional for an activity, as an occurrence which is greater than 1 makes its mandatory that this action should be observed in order for the activity to be performed
Mutually exclusive Sub-ADL or action	This states whether the Sub-ADL or action are mutually exclusive to the ADL, hence they do not occur in any other ADL. For example, Toothpaste used would only occur in *Brush Teeth* ADL
Prerequisite Sub-ADL or action	This determines if certain Sub-ADLs and actions need to occur before any of the Sub-ADLs and actions that will be expected to occur when this ADL is conducted

Table 2. ADL attributes

Attributes	Description
Maximum duration	This is threshold of the maximum duration of performing the ADL
Minimum duration	This is threshold of the minimum duration of performing the ADL
Associative actions	This determines whether the ADL has any Sub-ADLs and actions that are associated with other ADLs
Interweaved ADL	This determines if the ADL can be interweaved with another ADL. For example, a person can start brushing teeth and then start flossing, which can then be followed by brushing teeth
Shared ADL	This determines whether the ADL can be performed simultaneously with another ADL in the same zone. For example, person x and person y using the Tap in the sink zone at the same time, where person x is brushing teeth and person y is washing hands
Assistive ADL	This determines if two people are involved while the ADL is being performed. For example, person x is brushing teeth, while person y passes toothpaste to person x
Interruptible ADL	This determines if ADL can be suspended, while another ADL is conducted
Repetitive ADL	This determines if certain actions or Sub ADLs are likely to be repetitive when the ADL is being carried out

4   ADL Recognition

The recognition of the ADLs is based on recognising the patterns and occurrences of Sub-ADLs and actions that are generated by sensor event sequences. However, there is an issue as regarding the length of the sensor event stream that should be used for recognition. The first option could be to use the entire sensor event stream captured. However, this could be very inefficient as only the most recent events are of interest within a particular time frame. The other option is to assign a sliding window of events; however, this would raise an issue as to where the sliding window should start. A sensible approach is to ensure that the sliding window starts when a person enters or exits a particular zone (e.g. sink zone), as this could mark the end of one ADL and the start of another. However, what would happen if a person moves between zones whilst carrying out an ADL? The proposed approach has addressed by combining a series of windows in order to accommodate interweaving ADLs that might be carried out over a series of windows that are not structured sequentially. The proposed ADL recognition engine is divided into a series of functions (see Fig. 3), which represent the logical steps for recognising an ADL. A description of each function follows:
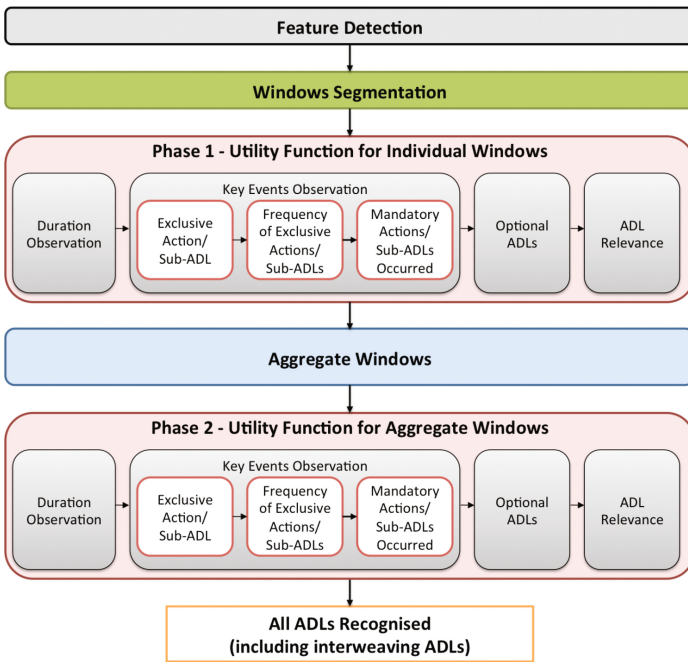

Fig. 3. ADL recognition engine

4.1   Feature Detection – Sensor Event Detection

The feature detection for the work in this paper has been conducted by installing a collection of Radio Frequency Identification (RFID) transponders onto household objects (such as utensils, cups, and everyday products) around the home environment. The motion duration of a touched object is based on the proximity the RFID reader has with the transponders that are attached to the objects. For example, the first point of contact with a utensil would be the start of a motion, while the final point of contact would be end of the motion. The main components of the system and its usage within the home environment are summarised in Table 3.

Table 3. Components for feature detection system

Components	Description
RFID reader	The RFID reader is a size of match box, which is worn as a ring on the hand of the user conducting an activity. Every time a user touches an object, this information is then transmitted by the RFID reader via Bluetooth to a server. The reader also serves a positioning tag, so that the users location can be tracked within the different zones
RFID object transponders	These transponders are affixed to everyday objects within the home environment
Zone exciters	Zone within rooms (e.g. sink area within kitchen) are created by low frequency exciters, which can be set at interval ranges between 0.15 m and 15 m in diameter

The reason for using RFID transponders is due to its low cost and its ability to unobtrusively monitor behaviours of multiple individuals within a household via object usage data.

4.2   Windows Segmentation

Once the data (streams of sensor events) has been captured by the feature detection component, the next step is to determine the length of the sensor event stream that is going to be used for inferring the activities and the individual that is conducting them. Hence, the objective of this step is to segment the entire captured sensor event streams into individual windows, so that each window can be used for activity inference in the preceding step, which generates a utility for each window.

The windows segmentation function is dependent on two following parameters:

- Time intervals between observations: This is considered when the time stamps of the sensor events indicate that there has been a significant interval between the movements of two objects. For example, the last object (e.g. frying pan) within the sensor event stream was captured at 19:26:05, which is then followed by another object (e.g. Cup) at 23:12:42.

- Location of the observed person: This is based on the person moving from one zone to another zone. For example, moving from sink zone to cooker/oven zone could signify the beginning or end of an activity.

The segmentation function has two phases of segmentation. First phase is to segment the captured streams into windows given the interval length between the objects observed. The next phase then carries out further segmentation of the generated windows by segmenting based on movement of a person between zones.

Algorithm: Windows Segmentation
Input: $S= \{s_1, s_2, s_3 \ldots s_n\}$ is a stream of sensor events.
Output: $<IW_k, s_{k1}, s_{k2}, s_{k3} \ldots s_{kn}>$ Individual Windows of sensor events.

---

1. **For each** S **do**
   - i. Observe t (time interval) between s_n and s_{n+1}
   - ii. **if** $t > t*t^t$
     - **Return** $<W_k, s_{k1}, s_{k2}, s_{k3} \ldots s_{kn}>$
2. **For each** W **do**
   - i. Observe z_n (zone) between s_{kn} and s_{kn+1}
   - ii. if $z_n = z_{n+1}$
     - **Return** $<IW_k, s_{k1}, s_{k2}, s_{k3} \ldots s_{kn}>$

4.3 Utility Function Algorithm

This component is responsible for generating an initial utility for all possible ADLs being detected given the current window of sensor events. This function is computed once the sensor events have been associated with an action.

The ADL that has the highest utility is considered to be the most probable ADL that is being conducted given the sensor event stream in each window. Figure 3 shows the structure of the utility function, which is divided into four steps that will determine the initial utility of each ADL. The output of the four steps is used to compute the initial utility. A brief description for each step is described as follows:

Step 1 – Duration Observation

The ability to recognise the duration of an action plays an important role in determining the ADL being carried out. The objective of this step is to see if the duration of the observed actions are within the maximum and minimum duration thresholds that are stored in the ADL characteristics knowledge base. If the observed duration is within the threshold then the output of this function would be computed as 1. However, for cases where the observed duration is not within the threshold are computed as follows:

Case 1: Observed duration is less than the minimum duration threshold

If an observed duration for action (e.g. tap used) associated with ADL (e.g. wash hands) is less than the minimum duration threshold that is currently stored in the

knowledge base then a linear probability scale is computed which is based on the linear difference between the observed start time and the minimum accepted duration.
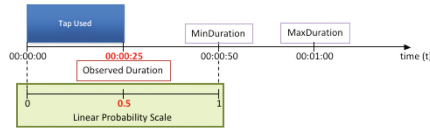
Fig. 4. Observed duration less than the minimum duration threshold

For example in Fig. 4, the minimum duration for Tap used is 00:00:50, while the observed duration is 00:00:25. In this instance the linear probability scale will be computed based on the observed action and the associated threshold data stored in knowledge base, in this case it would be 0.5.

This can be simplified as,

$$\frac{x_t}{y_t} = p \tag{1}$$

where x is the observed duration, y is the minimum duration in the knowledge base and t represents the unit of time. For example:

$$\frac{25_{sec}}{50_{sec}} = 0.5$$

Case 2: Observed duration is greater than the maximum duration threshold

If an observed duration for action (e.g. using kettle) associated with ADL (e.g. make tea) is greater than the maximum duration threshold that is currently stored in the knowledge base then another linear probability scale is computed as;

$$1 - \left(\frac{x_t - y_t}{z_t + 2z_t}\right) = p \tag{2}$$

where x is the observed duration, y is the minimum duration, z is the maximum duration while tt represents the unit of time. For example:

$$1 - \left(\frac{240_{sec} - 60_{sec}}{120_{sec} + 240_{sec}}\right) = 0.5$$

In Fig. 5, the minimum duration is 00:01:00, maximum duration is 00:02:00, while the observed duration is 00:04:00. The output of this function (2) based on the observation and the data in knowledge base would 0.5.
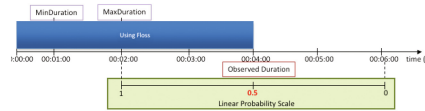

Fig. 5. Observed duration greater than the maximum duration threshold

Step 2 – Key Events Observation

2a. Exclusive Action/Sub-ADL

This step determines the proportion of actions and Sub-ADLs that are exclusive to the possible ADLs given the window of sensor events. For example, Toothpaste used would only occur in Brush Teeth ADL, hence this action would also be considered mandatory for this ADL to be recognised. This would be computed as:

$$\frac{x}{\sum y_1, y_2 \ldots y_n} = p \qquad (3)$$

where x is the observed exclusive action and $\sum y_1, y_2 \ldots y_n$ are the total number of associated exclusive actions with possible ADLs given the window of sensor events.

2b. Frequency of Exclusive Actions/Sub-ADLs

The objective of this step is to determine the frequency of observed exclusive actions and Sub-ADLs, where the frequency is above the expected mandatory threshold of actions and Sub-ADLs for the possible ADLs given the window of events. For example, the ADL characteristics knowledge would identify the frequency of the action loo roll used to be in the range of 1–5 for the ADL defecation, which would be considered mandatory. However if the captured frequency event for this action were greater than 5 then this action would be considered optional. This is computed as the in Function (3), where x is the observed optional exclusive action and $\sum y_1, y_2 \ldots y_n$ are the total number of optional exclusive actions that are associated with the all possible ADLs given the window of sensor events.

2c. Mandatory Actions/Sub-ADLs Occurred

This step determines the proportion of mandatory actions and Sub-ADLs that have been observed given all the possible ADLs that could occur within the current window of sensor events. This is computed as in Function (3), where x is the observed mandatory actions and Sub-ADLs and $\sum y_1, y_2 \ldots y_n$ are the total number of actions and Sub-ADLs that are associated with all possible ADLs within the current window of sensor events.

Optional ADLs
This step determines the proportional of optional actions and Sub-ADLs that have been observed given all the possible ADLs that could occur within the current window of sensor events.

ADL Relevance
This step determines the proportion of unrelated actions and Sub-ADLs that have been observed given all the possible ADLs that could occur within the current window of events.

The outputs of the four steps described are used to compute the utility of all the possible ADLs given the current window of events. The computation of the utility (u) is based on the average of the outputs of the 4 steps (s), which is as follows:

$$\frac{\sum_{i=1}^{n} s_i}{n} = u \tag{4}$$

Based on the recognition environment the ratio of importance for each step can be changed, however for the following example (Table 4) the ratios are considered all equal.

This utility function is applied in two phases, where the first phase is for individual windows to determine the ADLs given each window of events. While the second phase is applied to aggregate windows in order to determine ADLs that are interweaved. For example if window 1 is ADL x, window 2 is ADL y, and window 3 is ADL x, then this implies that ADL x is interweaved with ADL y.

Table 4. Initial utility for ADL X given window of events

Steps	Ratio	Output
(s_1) Duration observation	1:4	1
(s_2) Key events observation	1:4	0.8
(s_3) Optional ADLs	1:4	0.3
(s_4) ADL relevance	1:4	0.7
	Initial utility (u)	0.7

4.4   Aggregate Windows Algorithm

There can be many instances where an activity can be carried out in parallel with another activity. For example, a person could be making tea while they put bread in the toaster to make toast. The recognition of these types of interweaving instances is made possible by grouping the detected windows into aggregate of related windows, which reflect the interweaved activities.

Algorithm: Aggregate Windows
**Input**: $W= \{w_1, w_2, w_3...w_n\}$ is a stream of individual windows.
Output: $<a_k, W_{k1}, W_{k2}, W_{k3}...W_{kn}>$ Aggregate Windows of individual windows.

---

1. Initialize $a_1 = w_1$
2. **For each** W **do**
   i. Observe t (time interval) between w_n and w_{n+1}
   ii. **if** $t > t*t^t$
      Return $<a_k, W_{k1}, W_{k2}, W_{k3}...W_{kn}>$

Construction of the related aggregate windows is carried out by assigning the first recognised window w_1 as a starting point for the newly constructed aggregate window a_1. A linear search is then performed on the rest of detected windows to see if it is possible to add a related window w_n. to the current aggregate window a_1. The construction of the aggregate window is dependent on timing interval between the individual windows, because if the timing interval between two individual windows is over a certain threshold (e.g. 15 min) then the current aggregate window a_1 can be finalised (Fig. 6).
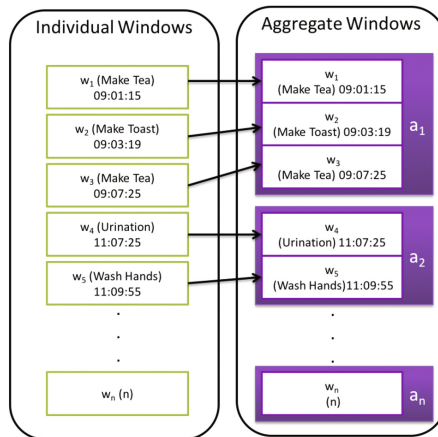
Fig. 6. Construction of aggregate windows

Once all of the aggregate windows $a_1...a_n$. have been constructed, the utility function is then applied in order to carry out the second phase of classification based on the new constructed aggregate windows.

5 Experimental Setup

The objective of the conducted experiments was to validate the performance of the inference engine given collected object usage data. The effectiveness of the proposed inference engine was measured by calculating the precision and recall rates of the ADLs recognised given the aggregate windows.

The precision (P) and recall (R) for this experiment has been calculated as follows:

$$P = \frac{True\ Positive}{True\ Positive + False\ Positive}$$
$$R = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (5)$$

The feature detection approach deployed for these experiments was based on a dense sensing [12] approach, where household objects (e.g. cup) are tagged with RFID transponders. Data based on the usage of these objects is collected by ring-link portable RFID reader, which transmits usage information to the server whenever the objects are touched or are within close proximity of the RFID reader. Intentionally there were many instance where the RFID reader captured noise and unrelated objects, which will validate the robustness of the proposed inference engine.

For this particular dataset, ten adult volunteers were recruited from the community to carry out a series of experiments. Table 5 describes the objective of each experiment.

Table 5. Experiment description

No.	Experiment
1.	Each subject was asked to conduct a sequential sequence of ADLs using a predefined order of objects
2.	Each subject was asked to perform a sequential sequence of ADLs using a non prescribed order of objects. The objective was to assess how well ADLs are recognised with different orderings of objects
3.	Each subject was asked to perform a set of Sub-ADLs in a parallel sequence, for example making tea while making toast. The objective was to assess how well the inference engine recognizes interweaved activities as an aggregate ADL

For each experiment, the subjects were asked to record the ADLs they conducted, which was used a ground truth to validate the system output. The experiments were based around 12 ADLs, which were made up of a series of Sub-ADLs that belonged to more than one ADL (see Table 6). This was done intentionally to test the robustness of the inference engine when trying to recognise similar ADLs.

Table 6. ADL, Sub ADLs carried out by subjects for experiments

ADLs	Sub-ADLs
Prepare Breakfast →	Make Tea, Make Coffee, Make Toast
Eat Breakfast →	Drink Tea, Drink Coffee, Eat Toast
Cleaning after Breakfast →	Clean Table, Clean Dishes
Prepare Lunch →	Make Sandwich, Make Wrap
Cleaning after Lunch →	Clean Table, Clean Dishes
Put Shopping Away →	Fridge Shopping Away Cupboard Shopping Away
Prepare Snack →	Make Tea, Make Coffee, Get Biscuits
Eat Snack →	Drink Tea, Drink Coffee, Eat Biscuits
Clean Kitchen Floor →	Sweep Floor
Clean Kitchen Worktop →	Wipe Countertop with Wipes
Laundry - Wash Clothes →	Wash Clothes - Washing Machine
	Dry Clothes using Tumble Dryer
Warm up Ready Meal →	Heat up food in Microwave

6 Results

The results for experiment 1 in Fig. 7 show that the precision rates ranged from 83% to 98%, which is based on the subjects conducting a sequential sequence of ADLs using a predefined order of objects.

Fig. 7. Experiment 1 ADL precision and recall rates

While the recall rates ranged from 91% to 100%, indicating that the inference engine was able to consider all possible relevant sub-ADLs and actions when carrying out ADL recognition, which demonstrates the robustness of the inference engine even where there is noise. The inference engine was able to recognise ADLs that were carried out using a prescribed order of objects. As expected, ADLs that consisted of sub-ADLs that belong to more than one ADL had a slight drop in the recognition rates.

The precision rates for experiment 2 (Fig. 8) ranged from 81% to 95% and the recall rates ranged from 88% to 97%, which is based on the subjects performing a sequential sequence of ADLs using a non-prescribed order of objects.
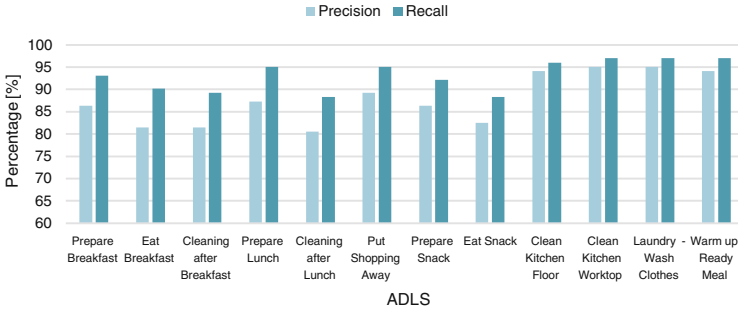
Fig. 8. Experiment 2 ADL precision and recall rates

The results for experiment 3 in Fig. 9 show that the precision rates ranged from 79% to 93%, while the recall rates ranged from 87% to 96%. This is a slight decrease from the other experiments, as each subject performed a set of sub-ADLs in parallel sequence. Taking into consideration issues related to noise and similar sub-ADLs, the inference engine was still able to recognise interweaved activities as aggregate ADLs.
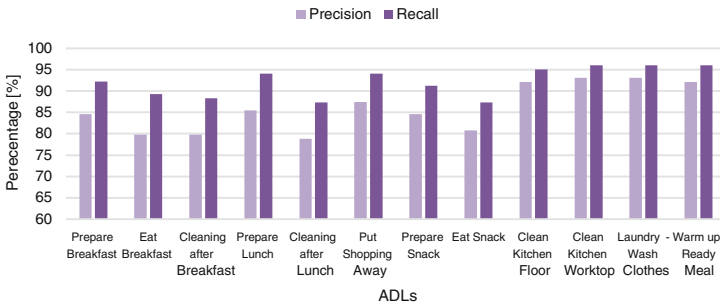

Fig. 9. Experiment 3 ADL precision and recall rates

Overall the results indicate that the proposed inference engine was able to recognise and consider all sub-ADLs and actions when inferring a range of ADLs in different experimental scenarios (e.g. sequential and parallel ADLs performed using ordered and unordered objects). The precision and recall rates suggest that the proposed inference engine was able to recognise more relevant instances of an ADL made up of sub-ADLs and actions as opposed to irrelevant instances. This is made possible by the hierarchical modelling of the ADL, which takes into consideration the actual ADL, its associated sub-ADLs, actions and objects.

The results from these three experiments is comparable in terms of the recognition rates achieved with existing ADL recognition approaches [27, 28]. Though the other approaches deployed feature detection techniques that captured richer data (e.g. ambient temperature readings, acceleration data for movement and pressure sensors). While the approach proposed in this work is based on object usage data collected by simple RFID transponders. In order to improve the recognition rates in this paper, we could deploy similar feature detection techniques that provide richer data for analysis.

7 Conclusion

The work described in this paper looked at how everyday ADLs have been modeled and recognised as a hierarchical encapsulated entity, where each ADL has attributes that enable the inference engine to reason the internal structure and relationships of an ADL when carrying out recognition. A series of experiments based on object usage data were conducted, which indicated that the hierarchical structure of the ADLs and inference engine made it possible to recognise ADLs given different recognition scenarios. The feature detection technique used was based on low cost simple RFID transponders, hence the inference engine had to be robust in terms of dealing with noise and missing data.

The work presented in this paper has the potential to be used for intention analysis for the elderly community. As the hierarchical modelling of ADLs can enable recognition to be more pre-emptive in terms predicting ADLs of the elderly person being monitored. This enables the possibility to initiate safeguards given a particular situation.

Further work will be carried out, as the current engine has the potential to be adapted for real time recognition. This will be done by deploying a learning mechanism to populate the knowledge base given the changes that take place in the individual's activity patterns associated with the attributes in the model.

References

1. Rutherford, T.: Population Ageing: Statistics. Commons Library Standard Note (2012). http://www.parliament.uk/briefing-papers/sn03228. Accessed 12 Aug 2015
2. Katz, S., Ford, A.B., Moskowitz, R.W., Jackson, B.A., Jaffe, M.W.: Studies of illness in the aged: the index of ADL: a standardized measure of biological and psychosocial function. J. Am. Med. Assoc. **185**(12), 914–919 (1963)
3. Fleury, A., Vacher, M., Noury, N.: SVM-based multimodal classification of activities of daily living in health smart homes: sensors, algorithms, and first experimental results. IEEE Trans. Inf. Technol. Biomed. **14**(2), 274–283 (2010)
4. Wilson, D.H., Atkeson, C.: Simultaneous tracking and activity recognition (STAR) using many anonymous, binary sensors. In: Gellersen, H.-W., Want, R., Schmidt, A. (eds.) Proceedings of the Third International Conference on Pervasive Computing, PERVASIVE 2005, pp. 62–79. Springer, Berlin (2005)
5. Kern, N., Schiele, B., Schmidt, A.: Multi-sensor activity context detection for wearable computing. In: Ambient Intelligence. Lecture Notes in Computer Science, vol. 2875, pp. 220–232 (2003)
6. Dernbach, S., Das, B., Krishnan, N.C., Thomas, B.L., Cook, D.J.: Simple and complex activity recognition through smart phones. In: Proceedings of the Eighth International Conference on Intelligent Environments, pp. 214–221. IEEE (2012). doi:10.1109/IE.2012.39
7. Khan, A.M., Lee, Y.-K., Lee, S.Y., Kim, T.-S.: Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis. In: Proceedings of the 5th International Conference on Future Information Technology, pp. 1–6 (2010)
8. Kwapisz, J.R., Weiss, G.M., Moore, S.A.: Activity recognition using cell phone accelerometers. SIGKDD Explor. Newsl. **12**(2), 74–82 (2011). doi:10.1145/1964897.1964918

9. Yang, J.: Toward physical activity diary: motion recognition using simple acceleration features with mobile phones. In Proceedings of the 1st International Workshop on Interactive Multimedia for Consumer Electronics, IMCE 2009, pp. 1–10. ACM, New York (2009). doi:10.1145/1631040.1631042

10. Tapia, E., Intille, S., Larson, K.: Activity recognition in the home using simple and ubiquitous sensors. In: Pervasive. LNCS, vol. 3001, pp. 158–175 (2004)

11. Buettner, M., Prasad, R., Philipose, M., Wetherall, D.: Recognizing daily activities with RFID-based sensors. In: Proceedings of the 11th International Conference on Ubiqitous Computing, Orlando, pp. 51–60 (2009)

12. Philipose, M., Fishkin, K.P., Perkowitz, M., Patterson, D.J., Kautz, H., Hahnel, D.: Inferring activities from interactions with objects. IEEE Pervasive Comput. Mag. **3**(4), 50–57 (2004)

13. Kalimeri, K., Matic, A., Cappelletti, A.: RFID: recognizing failures in dressing activity. In: Proceedings of the 4th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth), pp. 1–4 (2010)

14. Philipose, M., Smith, J.R., Jiang, B., Mamishev, A., Roy, S., Sundara-Rajan, K.: Battery-free wireless identification and sensing. Pervasive Comput. IEEE J. **4**(1), 33–45 (2005)

15. Buettner, M., Prasad, R., Philipose, M., Wetherall, D.: Recognizing daily activities with RFID-based sensors. In: Proceedings of the 11th International Conference on Ubiquitous Computing - Ubicomp 2009, p. 51 (2009). doi:10.1145/1620545.1620553

16. Wilson, D.H., Wyaat, D., Philipose, M.: Using context history for data collection in the home. In: Proceedings of Pervasive 2005: Workshops – ECHISE, Munich, Germany (2005)

17. van Kasteren, T., Krose, B.: Bayesian activity recognition in residence for elders. In: 2007 International Conference on Intelligent Environments, IET, pp. 209–212 (2007)

18. Sanchez, D., Tentori, M., Favela, J.: Activity recognition for the smart hospital. IEEE Intell. Syst. **23**(2), 50–57 (2008)

19. Chen, I., Nugent, C.D., Wang, H.: A knowledge-driven approach to activity recognition in smart homes. IEEE Trans. Knowl. Data Eng. **24**, 961–974 (2012)

20. Riboni, D., Bettini, C.: COSAR: hybrid reasoning for context-aware activity recognition. Pers. Ubiquit. Comput. **15**(3), 271–289 (2011)

21. Rafferty, J., Chen, L., Nugent, C.: Ontological goal modeling for proactive assistive living in smart environments. In: Proceedings of the 7th International Conference, UCAmI 2013, Carrillo, Costa Rica, 2–6 December 2013, pp. 262–269 (2013)

22. Delgado, M., Ros, M., Vila, A.: Correct behaviour identification system in a tagged world. Expert Syst. Appl. **36**(6), 9899–9906 (2009)

23. Ros, M., Delgado, M., Vila, A.: Fuzzy method to disclose behaviour patterns in a tagged world. Expert Syst. Appl. **38**(4), 3600–3612 (2011)

24. Suzuki, M., Matsushima, T., Hirasawa, S.: On a deductive reasoning model and method for uncertainty. In: Proceeding of the IEEE Conference on Tools with Artificial Intelligence, pp. 161–164 (1999)

25. Naeem, U., Bigham, J.: Activity recognition in the home using a hierarchical framework with object usage data. J. Ambient Intell. Smart Environ. **1**, 335–350 (2009)

26. Azam, M.A., Loo, J., Naeem, U., Adeel, M.: Recognising indoor/outdoor activities of low entropy people using Bluetooth proximity and object usage data. Int. J. Pervasive Comput. Commun. **9**(4), 346–366 (2013)

27. Rashidi, P., Cook, D., Holder, L., Schmitter-Edgecombe, M.: Discovering activities to recognize and track in a smart environment. IEEE Trans. Knowl. Data Eng. **23**(4), 527–539 (2011)

28. Baños, O., Damas, M., Pomares, H., Rojas, F., Delgado-Márquez, B., Valenzuela, O.: Human activity recognition based on a sensor weighting hierarchical classifier. Soft. Comput. **17**(2), 333–343 (2013)