Author(s): Naeem, Usman; Bigham, John; Wang, Jinfu.
Article title: Recognising Activities of Daily Life Using Hierarchical Plans
Year of publication: 2007
Citation: Naeem, U., Bigham, J., Wang, J. (2007) "Recognising Activities of Daily Life Using Hierarchical Plans" in Proceedings of the 2nd European Conference on Smart Sensing and Context, LNCS 4793, Lake District, UK, 2007, pp. 175-189.
Link to published version: http://dx.doi.org/10.1007/978-3-540-75696-5_11
DOI: 10.1007/978-3-540-75696-5_11

Recognising Activities of Daily Life using Hierarchical Plans

Usman Naeem, John Bigham, Jinfu Wang

Department of Electronic Engineering, Queen Mary University of London,
Mile End Road, London, United Kingdom, E1 4NS
{usman.naeem, john.bigham, jinfu.wang}@elec.qmul.ac.uk

Abstract. The introduction of the smart home has been seen as a way of allowing elderly people to lead an independent life for longer, making sure they remain safe and in touch with their social and care communities. The assistance could be in the form of helping with everyday tasks, e.g. notifying them when the milk in the fridge will be finished or institute safeguards to mitigate risks. In order to achieve this effectively we must know what the elderly person is doing at any given time. This paper describes a tiered approach to deal with recognition of activities that addresses the problem of missing sensor events that can occur while a task is being carried out.

Keywords: Smart Homes, Elderly Care, Hierarchal Activities of Daily Life, Task Segmentation, Task Associated Sensor Events

1. Introduction

From the turn of the last century in 1901 the life expectancy for both men and women has continued to rise in the UK, which has lead to more elderly people in society. It has become difficult for children to look after their aged parents due to increased geographical mobility with children working and living remotely from their parents, lifestyle preferences and commitments, which leads to more elderly people depending on care homes. The introduction of smart homes has been seen as a suitable mechanism to allow people the opportunity to extend safely their independent lives and so defer entry to care homes. One of the ways to establish whether an elderly person is safe or to provide relevant help is to monitor the Activities of Daily Life (ADL) that they are carrying out and provide assistance or institute safeguards in a timely manner. In this paper we illustrate our approach through simple domestic examples. In order to understand the intentions of the elderly people, they need to be monitored. However since privacy is an issue as extensive monitoring, e.g. with cameras, can be intrusive, the approach chosen depends on the introduction of more automation in the form of algorithms that are able to discriminate between different ADLs using a few sensor events as is practicable, while achieving tolerable false positives and false negatives. In the experiments described the recognition of ADLs is based on data that is collected from RFID sensors.

There has been significant amount of research focused on efficient and reliable ADL identification. A popular technique in detecting ADLs is 'dense sensing' [1], which collects sensor data from many objects rather than relying on visual based engines. Numerous individual objects such as a kettle are tagged with wireless sensors or transponders that transmit information to a server via an RFID reader when the object is being used or touched. The interpretation of such sensor data is relatively easy for activities that are represented by sequential models that follow a standard path of execution. However, when a task can be carried out in more than one way or if a particular sensor event is missing due to data transfer problem. For example, if a person decides not to take milk or sugar in his or her tea when they usually do, this can sometimes been seen as a missing sensor event. Hidden Markov models have been used to carry out task identification. One such approach was by Wilson [2], where episode recovery experiments where carried out and analysed by a Viterbi algorithm which was responsible determining which task is active from the sequence of sensor events. The approach was successful in carrying out unsupervised task identification; however it was not as efficient when the tasks were carried out in any order. Multiple Behavioural Hidden Markov Models [3] have also been used to carry task identification. This approach was based on the idea of creating multiple hidden Markov models for each variation of a task, in order to accommodate each variation that could be carried out for a task. The latter approach was able to carry out task identification even if a sensor event was missing as a missing sensor event was treated as an insertion, which was very much like the approach used for substituting any unexpected sequence data in DNA motifs [4]. Other approaches that have been developed in order to carry out reliable task identification and mitigate the missing sensor problem have used techniques that involve ontologies [5] and data mining techniques [6]. Ontologies have been utilised to construct reliable activity models that are able match an unknown sensor reading with a word in an ontology which is related to the sensor event. For example the sensor reading 'mug' (which is an unknown sensor event in a model that is being interpreted) could be matched to a 'cup' sensor reading in a model for making tea that uses the term 'cup'.

2. Hierarchal Activities of Daily Life

For the work in this paper the ADLs have been modelled in a hierarchical structure, which allows us to decompose the ADLs into different models. With this type of modelling ADLs can correspond to simple tasks, such as "switch on kettle", or more a complex activities such as "make breakfast". The lowest tier of the Hierarchy of Activities of Daily Life (HADL) consists of the components responsible for gathering the sensor events within the home. The second level is task identification. A task is defined as the lowest level of abstraction in the higher tier(s). It can be associated with a simple goal of the monitored individual. The process of task identification maps each sensor event to the possible tasks which are associated with the sensor event, e.g. the sugar bowl sensor can be associated with following tasks: 'Make Coffee' or 'Make Tea'. This identification can be performed by a range of processes, such as hidden Markov models, though a simpler approach is used in the experiments described. At the higher levels there are further sub-goals and goals of the person being monitored, and these are modelled using a knowledge representation language that can represent plans. Each (sub) goal corresponds to an ADL. A task can be thought of as a lowest level goal (of the monitored individual) modelled using the planning knowledge representation language. It can however be modelled by some other modelling tool, such as a hidden Markov model. The number of levels above the task identification level depends on the complexity of the task. In this way the ADLs are nested within other ADLs. Additionally the ADLs may occur in parallel with other ADLs or have other temporal constraints. These are represented in the planning language used.

For each task (a) and sensor event (b), we can assigned a probability $P[a \mid b]$. This is required when carrying out task segmentation in the task identification. The entire sensor event stream is segmented into appropriate task segments. The segmented tasks are then used to determine which ADL is currently active. When performed, a task generates sensor events, and so task association mapping and recognition is based on analysing the sensor data, while ADL recognition is based on recognising constituent tasks.

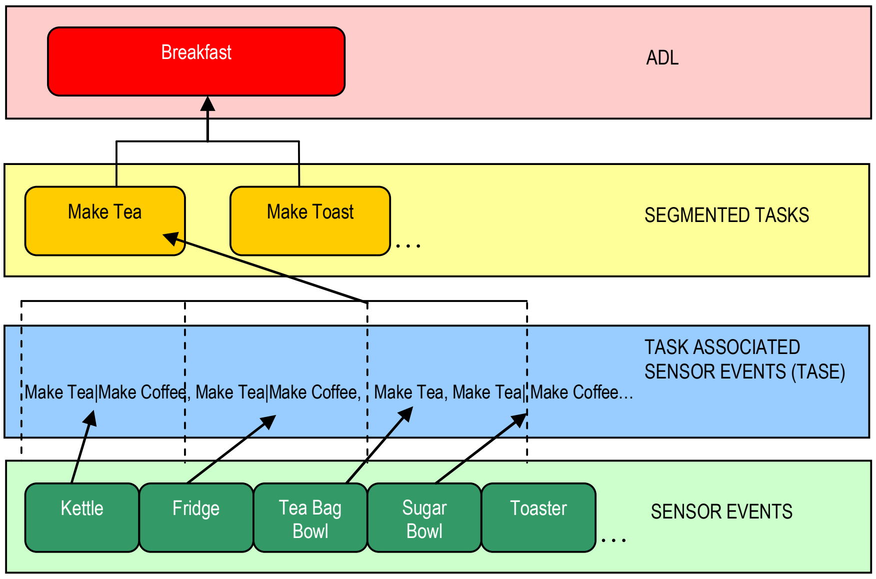
Breakfast
ADL
Make Tea
Make Toast
...
SEGMENTED TASKS
TASK ASSOCIATED SENSOR EVENTS (TASE)
Make Tea|Make Coffee, Make Tea|Make Coffee, Make Tea, Make Tea|Make Coffee...
Kettle
Fridge
Tea Bag Bowl
Sugar Bowl
Toaster
...
SENSOR EVENTS

Fig. 1. The ADL “Make Breakfast” consists of a simple sequence of tasks, Make Tea, Make Toast..., but these may be in any order, or indeed be performed in parallel. The lowest tier of the HADL is the sequence of sensor events that have been detected, these sensor events are then associated with all the tasks that correspond to the sensor event, for example kettle sensor event can be associated with make tea or make coffee. These Task Associated Sensor Events (TASE) are then segmented into tasks using a statistical model which will be explained further in the paper.

3. Task Segmentation

Segmenting tasks can be carried out by simply segmenting sensor events into segments that correspond to a particular task. However this approach can sometimes generate sensor event segments that are incorrect and bear no resemblance to the task that is actually being carried out. In order to refine this problem we have manipulated the sensor events in to Task Associated Sensor Events (TASE) and developed a segmentation algorithm that is able to segments tasks efficiently. This algorithm was based on a statistical model which was created for text segmentation by Utiyama et al [2].

This method was used to find the maximum-probability segmentation of text, and does not need any training data, as it estimates probabilities from the stream of text. In the context of segmenting tasks and using the task segmentation algorithm the TASE are converted into letters so that we get a stream of letters, for example; Task(Make Tea)= letter(A), Task(Make Coffee)=letter(B), Task=(Make Toast)=letter(C)…Task(n) =letter(n).

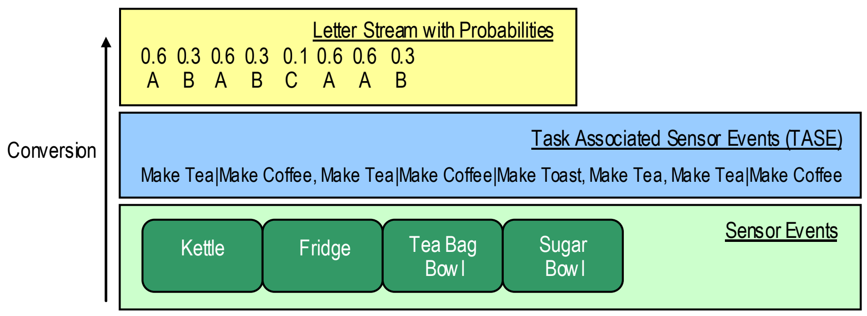
Letter Stream with Probabilities
0.6 0.3 0.6 0.3 0.1 0.6 0.6 0.3
A B A B C A A B
Task Associated Sensor Events (TASE)
Make Tea|Make Coffee, Make Tea|Make Coffee|Make Toast, Make Tea, Make Tea|Make Coffee
Conversion
Sensor Events
Kettle
Fridge
Tea Bag Bowl
Sugar Bowl

Fig. 2. This shows the different levels of conversion from sensor event to task associated sensor event to stream of letters. The probability values for the letters in the letter stream is based on the number of associations each task has with the total the number of sensor events.

After the TASEs have been converted into a stream of letters we then used our developed Sensor Event Segmentation Engine (SESE), which is responsible for working out the most likely combinations of segments that occur in the stream of letters. For example, a stream of letters consisting of ABC will have the following combination of segments: A|B|C, A|BC, AB|C, and ABC, which leads to four streams of letters with different segmentation points.

$$\sum_{j=1}^{n_i} \log \frac{n_i + k}{p+1} + \log n_i * 0.2 \qquad (1)$$

Equation 1 is applied to each segment within each stream of letters, which outputs an overall cost for each stream. The stream of letters which has the lowest cost is generally close to correct segmentation or has been correctly segmented. Therefore the SESE analyses at sample of the 10 lowest cost segmented streams, which gives a good idea of which task is currently active. It is evident that on many occasions the results provided by the SESE may not be perfect in terms of accuracy, but this is where the higher tier of the HADL is used to refine the interpretation. The higher tier will be discussed in the next section.

Let AB|C be the stream of letters that algorithm 1 is going to be applied to. In relation to this stream of letters, n_i represents the length of the segment within the stream of letters, (AB)=2, (C)=1.
k represents the frequency of each letter in the stream of letters, (A)=1, (B)=2, (C)=1.
n represents the length of the text stream, which is 3, and p is the prior probability assigned to each letter.

Table 1. Task Segmentation for stream AABCA with probabilities A=0.9, B=0.5, C=0.2.

Cost of Stream	1st Segment	2nd Segment	3rd Segment	4th Segment	5th Segment
2.592679	A	AB	CA		
2.592679	AA	B	CA		
2.592679	AA	BC	A		
2.594235	A	A	B	CA	
2.594235	A	A	BC	A	
2.594235	A	AB	C	A	
2.594235	AA	B	C	A	
2.617737	A	A	B	C	A
2.733313	A	A	BCA		
2.733313	A	ABC	A		
2.733313	AAB	C	A		
2.761272	AA	BCA			
2.761272	AAB	CA			
3.011331	A	ABCA			
3.011331	AABC	A			
3.423917	AABCA				

The task segmentation in Table 1 shows the cost of each stream with different segments, with the lowest cost shaded in orange, while the other shaded sections form the sample of the 10 lowest cost streams. From the table it is clearly evident that the segmentation carried out gives a clear indication of what task might be currently active. For example Tasks like A have been segmented correctly, as well as that this technique provides the high level with more alternatives when mapping these tasks with the ADL plans.

Whatever the method used for task identification, the next step is to use the modelling of the possible goals and sub-goals of the individual to assist the interpretation. This is now described.

4. High Level Activity Recognition

The aim is to support recognition of tasks through feedback from beliefs held about ADLs. Initial task recognition is as has been described in previous sections, while the next steps in recognition through the hierarchy of constituent ADLs. The ADLs are represented in a hierarchical plan representation language. While a common way of representing and modelling high level behaviour is workflows, which are typically modelled using an augmented Petri Net [3]. Workflows are often used to model business processes. However, workflows are too prescriptive in their ordering and in their way of representing combinations of activities when trying to model the legitimate variations in human activity associated with a set of goals, so a richer knowledge representation language has been chosen. The language that has been used for the recognition of the activities as well as the elderly person's intentions is Asbru [4]. The Asbru language is a process representation language, which has similarities to workflow modelling. The roots of Asbru are in the modelling of medical protocols, which can be complex. It is hoped that this language will prove to be a suitable representation language to model behaviours of the monitored subjects.

Asbru is a task-specific and intention-oriented plan representation language initially designed to model clinical guidelines. Asbru was developed as a part of the Asgaard project to represent clinical guidelines and protocols in XML. Asbru has the capability to represent the clinical protocols as skeletal plans, which can be instantiated for each patient that requires a specific treatment. These skeletal plans are a useful guide for physicians when monitoring patients on a treatment protocol [5]. Asbru has many features which allow each skeletal plan to be flexible and to work with multiple skeletal plans. These plans in Asbru have been used to represent ADL and sub-activities within an ADL, e.g. Prepare Breakfast is an ADL, and a sub-activity of this ADL is to enter the kitchen.

In Asbru when a goal is achieved the plan is labelled as executed. When the pre-conditions of an ADL have been met then the ADL is classified as being executed. For example, for the goal 'eat egg' to start execution a pre-condition could be that the goal 'make egg' should be labelled as executed. Additionally an ADL can be classified as mandatory or optional. If an ADL has sub-goals that are classified as mandatory then these sub-goals must be executed before the ADL is labelled as executed. If optional then the sub-goal need not be executed. Sub-goals can be ordered in many ways. Common ones are sequential (in strict order), parallel (executed simultaneously), in any order (activated in any order but only one sub-goal can be executed at a time) and unordered (executed without synchronisation). The monitoring system being developed allows multiple activities to be tracked including tasks that may occur at the same time. Asbru also allows temporal intervals to be associated with goals, but that has not yet been incorporated into our monitoring system.

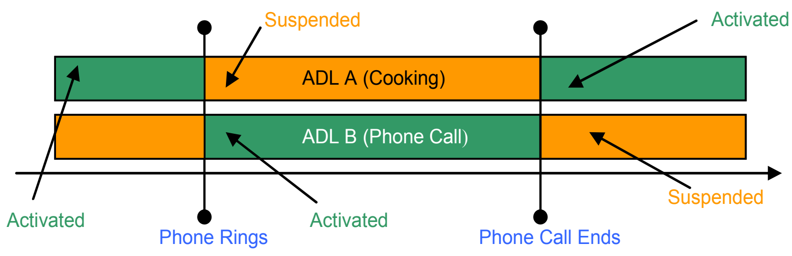

Fig. 3. If an elderly person is cooking dinner (ADL A) and the phone rings (ADL B) then the elderly picks the phone up, then with the aid of the conditions Asbru can suspend A and start ADL B. Once the elderly person is off the phone then ADL A will reactivated and ADL B will be suspended as more phone calls will come during the course of the day.

4.1 Modelling with Asbru

We briefly describe an example of how an ADL is modelled with Asbru in the higher tier of the HADL.

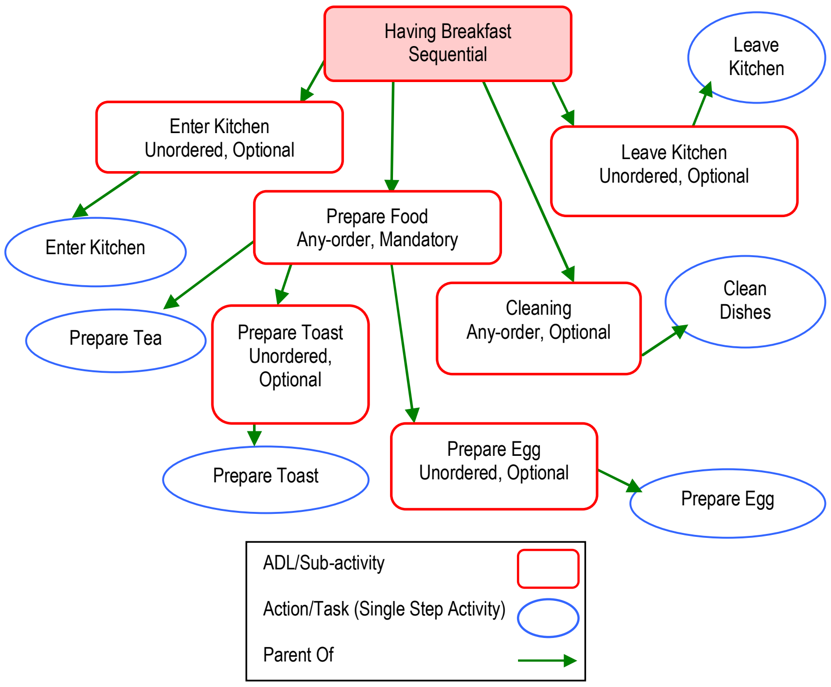

Fig. 4. Modeled example in Asbru, which is known as an ADL plan

We suppose that the following actions/tasks are detected in the lower tiers of HADL – Enter Kitchen, Prepare Toast and Clean Dishes - in this order.

At the detection of each task, the following processes will take place:
The main root ADL plan Having Breakfast is sequential, which means that the child activities within Having Breakfast will be executed in a sequential order, working its way from enter kitchen sub-activity to exit kitchen sub-activity.
When **Enter Kitchen** is detected then the sub-activity plan Enter Kitchen is set to complete, the **Enter Kitchen** is also a single step activity, which then allows the system to moves onto the sub-activity of the root ADL plan. A single step activity is a plan that cannot be decomposed any further and is called a task, which is what it is called when it is detected in the lower tiers of HADL.
The next task that may be detected is **Prepare Toast**. As this is also a single step activity then this is also set to complete, however the system could not continue to the next sub-activity of the root ADL plan as the sub activity for Prepare Food is mandatory, which means that all the child activity plans and tasks within this plan must be detected before it can proceed to the next sub-activity. As well as being mandatory, a plan may be optional, which means that a root parent activity does not need its child activities to be set to complete in order for it to move to other sub-activities.
The next task that is detected is **Clean Dishes**, this indicates the plan in Figure 4 is not the ADL that the person is carrying out, as the mandatory tasks have not been fulfilled in the previous sub activity. This therefore means that the person in question might be having a snack rather than having breakfast. Figure 4 is just one of many plans which are used to model the Activities of Daily Life, and therefore they enable us to follow all the plans concurrently which allows us to accommodate any dynamic changes, for example something which may look like having breakfast could actually be the elderly person having a snack.

In relation to the task segmentation that is carried out in the lower tiers of the HADL, the higher tier retrieves the tasks that have been segmented correctly from the stream of data and sees whether the tasks fit into the plans which are currently idle. For example, if enter kitchen has been segmented correctly then the higher tier planning tool will suspend all the current plans (activities/sub-activities) that do not take place in the kitchen. This reduces the possibilities of which activity is active at a given time. After this the higher tier looks at the number of times a task occurs within the time frame of the tasks which have already been detected. For example if enter and exit kitchen have been detected, then we will look at which task has occurred the most within that time frame of entering and exiting the kitchen. This fills the gap of the task(s) which has not been detected. This is worked out by an occurrence model, an example of this is shown in Figure 5.

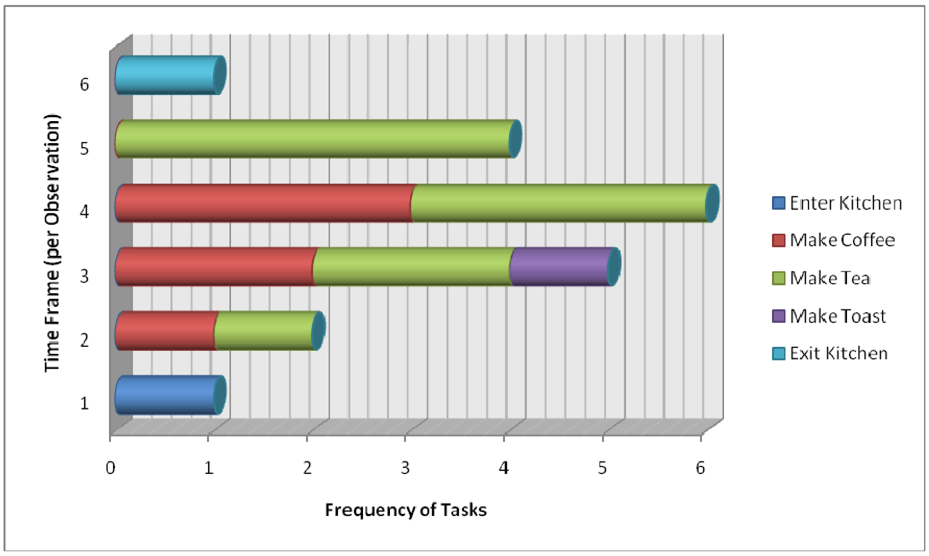

Fig. 5. The occurrence model shows the frequency of each task given the time frame, which is measured in sensor observations.

In Figure 5 if enter and exit kitchen have been detected then according to the occurrence model the most likely task that could fill one or many gaps in the higher tier ADL plans is Make Tea. This is because the frequency of the task "Make Tea" is incrementing with each time frame until time frame five, which suggest that the task may have been completed. However up until time frame four it is not evident whether "Make Tea" or "Make Coffee" is being carried out by the elderly person. In this situation both tasks can be mapped to the high level plans as the planning language Asbru is capable of managing concurrent tasks and once it is evident that make tea is the correct task as shown in time frame five then task make coffee can be suspended.

5. ADL Detection Experiments

The objective of the ADL detection experiments was to determine which ADLs are active from the collected sensor data stream. The accuracy of these experiments was determined by the percentage of detection rate of identifying an ADL. For each possible plan a discrepancy count and more importantly a surprise index is computed, whenever a new task is recognised in the lower tier of the HADL. The discrepancy count simply computes the number of sensor events that are consistent with the plan being the current ADL. The surprise index is used to account for the fact that the absence of some sensor events can be more unusual than others, and quantifies this by accruing a measure of how likely a sensor event is when a task is being executed.
A discrepancy is computed whenever there is a missing mandatory action/task, such as make tea for the ADL Make Breakfast. The surprise index is the maximum of the conditional probability of a missing sub plan/activity and actions/tasks. In order to

generate the detection rates for each ADL in these experiments, each ADL has been assigned a surprise index threshold. If the surprise index exceeds an ADL's surprise threshold when the ADL is actually being performed, then that is taken to mean that the ADL has not been detected correctly. For example, the ADL Make Breakfast has a sequential execution order and has surprise index threshold of 2. While carrying out the experiment if a surprise index that is over 2 is found then this means that Make Breakfast is not the ADL. Table 2 shows the surprise threshold for each of the ADLs and sub activities that have been used in the experiment.

Table 2. Surprise Threshold and Execution Order of the ADL/ Sub Activities, with the ADLs in bold and the sub activities in italics.

ADL/ Sub Activities	Surprise Threshold	Execution Order
Breakfast	1	Sequential
Prepare Food	1.25	Any Order
Clean Dishes	1.5	Unordered
Laundry	1	Sequential
Wash Clothes	1	Sequential
Dry Clothes	1	Sequential
Put Shopping Away	1.25	Any Order
Unpack Shopping	1.25	Any Order
Prepare Meal	1.3	Any Order
Make Chicken Curry	1.25	Sequential
Make Fish & Chips	1.25	Sequential
Warm up Meal	1.25	Sequential
Clean up Kitchen	1.3	Any Order
Clean Dishes	1.5	Unordered
Dish wash Dishes	1.3	Sequential

The six experiments were in a kitchen (Figure 6) and the ADLs being tested for detection are all kitchen oriented. The experiments were conducted with non-intrusive RFID transponders installed around the kitchen and on its cupboards and utensils, such as on the kettle, dishwasher, and toaster. The data generated from the transponders was collected by a RFID reader that is the size of match box and was worn on the finger of the subject conducting the experiment. For these experiments 10 adult volunteers had been recruited from the community to carry out the ADLs. The ADLs ranged from making breakfast to putting shopping away. The reason why 10 subjects were chosen is because people have different ways and ordering of carrying out a particular ADL, so there will be variability in the sensor stream.

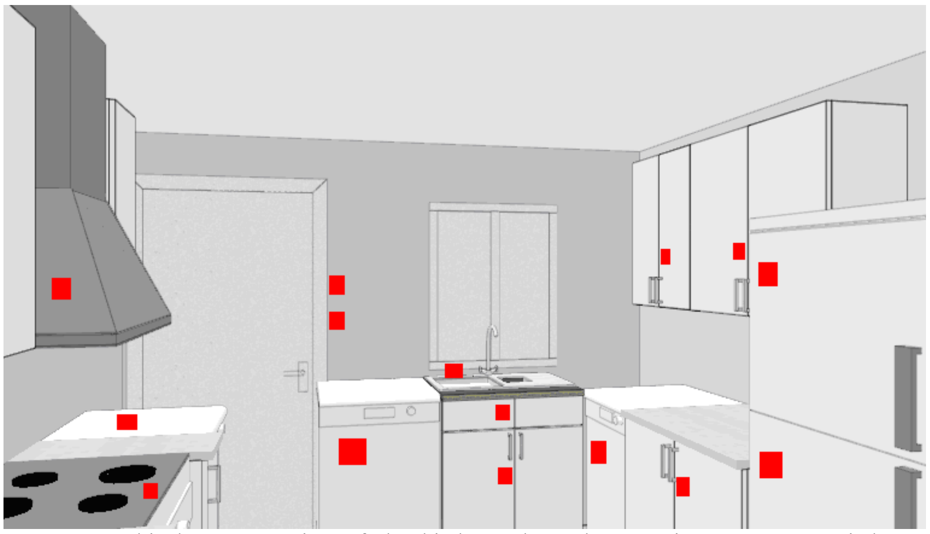

Fig. 6. Graphical representation of the kitchen where the experiments were carried out, showing the locations of some of the sensors.

The experiments are divided into two sets; one set is a ‘distinctive’ series of sensor data while the other set is the ‘non-distinctive’ series. The distinctive series makes use of sensor events where there is usually a determining sensor reading for each ADL. For example the ‘fairy bottle’ sensor is exclusive to the task ‘washing up dishes’, which makes it a distinctive sensor event which could determine if the ADL is active. On the other hand, the non-distinctive series does not make use of any sensor events which might be a distinctive when detecting an ADL. This is a harder challenge. Within the two sets of experiment there were three experiments that were conducted, which means each subject conducted six experiments in total. Table 3 shows the objective of each experiment conducted.

Table 3. Experiment Objectives

Experiment Number	Type of Experiment
1 & 2	Distinctive Series & Non- Distinctive Series – Subjects carried out 5 ADLs specified in the prescribed order provided. The tasks which were optional did not need to be carried out.
3 & 4	Distinctive Series & Non- Distinctive Series – Subjects carried out 5 ADLS in any order and were allowed to carry out the tasks within an ADL in any order. The ADLs are not interweaved.
5 & 6	Distinctive Series & Non- Distinctive Series – Subjects were allowed to carry out any 2 ADLs concurrently and in any order, e.g. make tea while putting the shopping away. Here the ADLs are interweaved.

The experiment is modelled around 5 ADLs, which consist of 25 tasks and 45 sensor events, Figure 7 shows the ADLs with their associated tasks that have been used for the experiments.

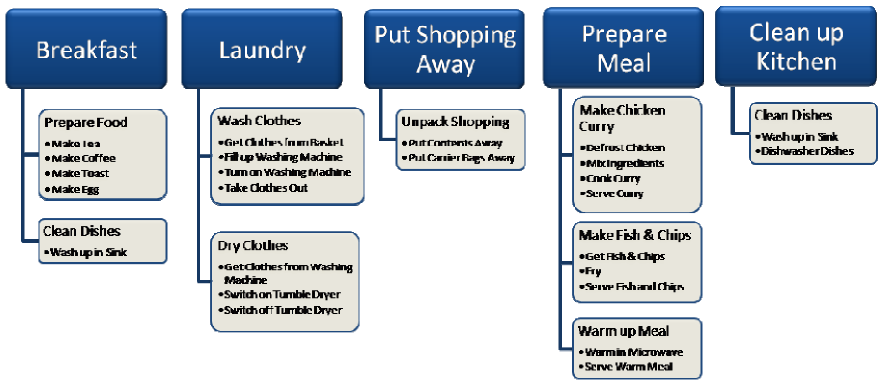

Fig. 7. The root plan is the ADL (e.g. Breakfast), the child nodes are the sub-plans/activities which are made up of tasks which are also known as single step plans.

However more ADLs have been modelled as plans in Asbru, so that there are conflicting situations where one task could be a part of more than one ADL. The reason for conducting different type of experiments is to have a sufficient amount of data to test the HADL approach, which includes TASE mapping, Task Segmentation and ADL Recognition.

6. Evaluation and Results

For all of the experiments the percentage of the detection rates for each ADL was determined using the surprise index and how many times the 5 ADLs carried out for the experiments were recognised successfully.

Table 4. Distinctive Series Results for Experiments 1, 3, and 5, with the ADLs in bold and the sub activities in italics.

ADL/ Sub Activities	Experiment 1 Prescribed Detection Rate [%]	Experiment 3 Random Detection Rate [%]	Experiment 5 Concurrent Detection Rate [%]
Breakfast	90	87	84
Prepare Food	93	91	86
Clean Dishes	86	83	80
Laundry	100	96	95
Wash Clothes	100	96	95
Dry Clothes	100	96	95
Put Shopping Away	95	92	89
Unpack Shopping	95	92	89
Prepare Meal	89	82	80

Make Chicken Curry	84	80	78
Make Fish & Chips	86	79	75
Warm up Meal	90	89	88
Clean up Kitchen	89	86	80
Clean Dishes	86	83	80
Dish wash Dishes	100	97	96

The results of the experiments carried out with the set of distinctive sensors (Table 4) show that ADLs like "Breakfast", "Laundry", "Put Shopping Away", "Warm up Meal" and "Dish Wash Dishes" were detected correctly on a regular basis. As well as that the detection rate percentage for these ADLs did not have a radical change when carrying out these ADLs in a random or concurrent with other ADLs. This does not mean to say that the other ADLs were not regularly detected correctly; we just feel it was important to outline the mentioned ADLs as they are reliant on distinctive sensor events in order for them to be recognized (e.g. microwave was a distinctive sensor event for the task warm meal). The results of these particular ADLs will be compared with the experiment results for the non-distinctive series. In summary these results show that our developed hierarchical approach is capable of managing concurrent as well as randomised sensor events and tasks and most importantly to recognize which ADL is currently active.

Table 5. Non-Distinctive Series Results for Experiments 2, 4, and 6, with the ADLs in bold and the sub activities in italics.

ADL/ Sub Activities	Experiment 2 Prescribed Detection Rate [%]	Experiment 4 Random Detection Rate [%]	Experiment 6 Concurrent Detection Rate [%]
Breakfast	82	79	77
Prepare Food	85	83	79
Clean Dishes	80	77	75
Laundry	96	92	90
Wash Clothes	96	92	90
Dry Clothes	96	92	90
Put Shopping Away	89	85	84
Unpack Shopping	89	85	84
Prepare Meal	85	81	77
Make Chicken Curry	82	77	76
Make Fish & Chips	83	75	74
Warm up Meal	85	81	80
Clean up Kitchen	81	78	74
Clean Dishes	80	77	75
Dish wash Dishes	97	95	93

The results from non-distinctive experiments (Table 5) show a slight decrease in the detection rate for each of the ADLs. A decrease was expected as the distinct sensor events were taken away from these set of experiments. However, the decrease that was witnessed was small, as the average of the detection rates for all the ADLs after all the experiments was 86.3%. Figure 8 shows the detection rates for the five ADLs mentioned and from this we see that it does not make a significant change to the detection of the ADLs if the distinct sensors have not been detected.

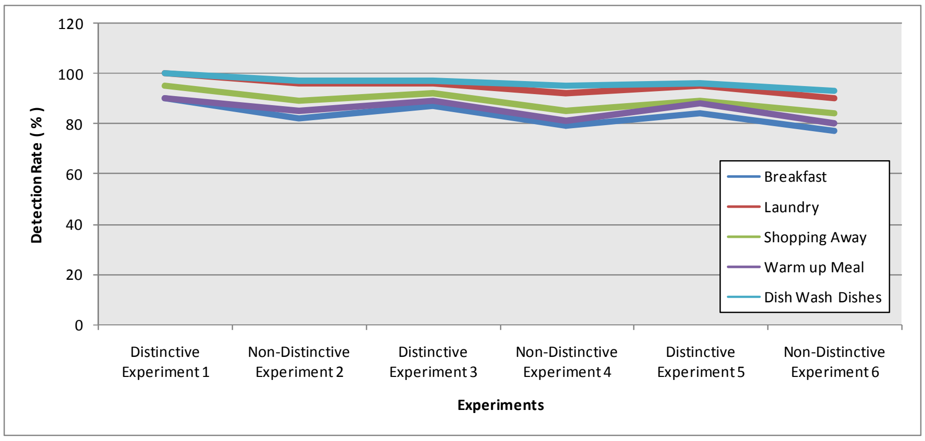

Fig. 8. Comparison of all the experiments with the ADLs that rely on distinct sensor events

The reason why our approach was able to detect ADLs without their distinct features was because of the planning capability of the higher tier. The planning capability of the representation language used was able to have all the ADLs mapped as plans which allowed our approach to be able to predict which ADL was active. The predictions were made on the basis of the events and their probabilities that had been gathered in the lower tier of our approach. Additionally, the higher tier is capable of dealing with tasks which occur in any order or are missing, as long as few tasks which are associated with ADL have occurred. Otherwise, it would be impossible to detect the ADL.

In terms of the dealing with the missing sensor events at the lower tier, the TASE was capable of dealing with this, as it provided all the possible task associations for the sensor event. Therefore if a sensor event was missing then the other sensor events which got manipulated into TASE will be able to provide some of idea of what task is active. With the aid of the task segmentation we are then able to filter out the most likely tasks that have occurred, which is then refined and mapped in plans in the higher tier of the HADL.

A limitation of this approach is that it does not take time into consideration. This is limiting as time can play a crucial part in detecting which ADL at what time of the day is active. For future enhancements the higher tier of our approach will incorporate task (and goal) durations. Also since the detection system will be aware of what time of day it then it will know the ADL plans which are usually executed around that time. In addition timing will play an important part in the lower tier of our approach, as time can be used to measure how long it takes on receiving different type of sensor events.

7. Conclusion

In this paper we described a tiered approach to interpreting sensor data when monitoring ADLs. The problem of missing sensor events and different orders of execution has been addressed. Our experiment results indicate that our approach was capable of dealing with missing distinct sensor events and still being able to detect which ADL is currently active. For the lower tier we also established Task Associated Sensor Events that are then segmented using a technique taken from research into text segmentation that has been reworked and improved for the detection of a task.

The results here can only indicate the potential of the approach. It is planned to link the ADL plans to more sophisticated approaches to tasks identification, and then to use the identification based on plan recognition to feedback to the task identification stages. It is hoped that this will result in a powerful approach to ADL recognition.

References

1. Philipose, M., Fishkin, K.P., Perkowitz, M., Patterson, D.J., Kautz, H., Hahnel, D.: Inferring activities from interactions with objects. In: IEEE Pervasive Computing Magazine, Volume 3, Issue 4, pp. 50-57. (2004)
2. Wilson, D.H., Wyaat, D., Philipose, M.: Using Context History for Data Collection in the Home. In: Proceeding of PERVASIVE 2005: ECHISE Workshop. Munich, Germany. (2005)
3. Naeem, U., Bigham, J.: A Comparison of Two Hidden Markov Approaches to Task Identification in the Home Environment. In: Proceedings of the 2nd International Conference on Pervasive Computing and Applications, Birmingham, United Kingdom (2007)
4. Eddy, S.R.: Profile hidden Markov models (Review). Bioinformatics, Volume 14, pp.755-763. Department of Genetics, Washington University School of Medicine, USA. (1998)
5. Munguia-Tapia, E., Choudhury, T., Philipose, M.: Building Reliable Activity Models using Hierarchical Shrinkage and Mined Ontology. In: Proceedings of the Fourth International Conference on Pervasive Computing, pp.17-32. Dublin, Ireland. (2006)
6. Wyatt, D., Philipose, M., Choudhury, T.: Unsupervised Activity Recognition using Automatically Mined Common Sense. In: Proceedings of the Twentieth National Conference on Artificial Intelligence, pp.21-27. Pittsburg, PA. (2005)
7. Utiyama, M., Isahara, H.: A Statistical Model for Domain-Independent Text Segmentation. In: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics, pp.499-506. Toulouse, France (2001)
8. Zurawski, R., Zhou, M.: Petri Nets and Industrial Applications: A Tutorial. In: Industrial Electronics, IEEE Journal. Volume 41, Issue 6, pp.567-583. (1994)
9. Fuchsberger, C., Hunter, J., McCue, P.: Testing Asbru Guidelines and Protocols for Neonatal Intensive Care. In: Proceedings of the Tenth European Conference on Artificial Intelligence, pp.101-110. Aberdeen, United Kingdom (2005)
10. Kosara, R., Miksch, S., Shahar, Y., Johnson, P.: AsbruView: Capturing Complex, Time-oriented Plans - Beyond Flow-Charts. In: Second Workshop on Thinking with Diagrams, pp.119-126. Aberystwyth, United Kingdom (1998)