Author(s): Naeem, Usman; Bigham, John.
Article title: A Hierarchal Approach to Activity Recognition in the Home Environment based on Object Usage
Year of publication: 2008
Citation: Naeem, U. Bigham, J. (2008) "A Hierarchal Approach to Activity Recognition in the Home Environment based on Object Usage" in Proceedings of the 2008 Networking and Electronic Commerce Research Conference (NAEC 2008), Lake Garda, Italy, 2008, pp. 48-54.

A Hierarchal Approach to Activity Recognition in the Home Environment based on Object Usage

Usman Naeem, John Bigham
School of Electronic Engineering and Computer Science
Queen Mary University of London
Mile End Road
London, E1 4NS
{usman.naeem, john.bigham}@elec.qmul.ac.uk

Abstract

Being able to monitor everyday activities of daily life is seen as a key approach for mitigating functional decline among elderly people as it allows context sensitive support to be offered. This paper describes a hierarchal approach for modelling activities of daily life using task sequences generated by object usage data and a mechanism for recognising these activities from sensor data. The underlying motivation of this work is to allow people with early Alzheimer's disease to have additional years of independent living before the disease reaches the moderate and severe stages. To ameliorate intrusion into personal privacy the monitoring of activities is via simple non-visual sensors with a greater emphasis placed on intelligent reasoning that exploits structures of typical behaviours.

1. Introduction

Over the last century life expectancy in the UK has gradually increased, which in turn has lead to more elderly people in the society. A common impairment among elderly people is Alzheimer's disease, which currently costs the UK an estimated £17 billion a year. It is also predicted that by the year 2025 there will be over one million people in the UK who will suffer from dementia [1]. The structure and demands of society make it difficult for children to look after their parents when they require care or assistance, this is can be due to geographical mobility with children working and living remotely from their parents, as well as lifestyle preferences and commitments.

One of the ways to find out whether an elderly person is safe in their home is to find out what Activity of Daily Life (ADL) they are carrying out and offer assistance should problems arise. Elderly people with Alzheimer's disease are often prescribed a set of daily activities by carers and health visitors in order to deal with forgetfulness as well as giving the elderly stimulation and a framework for an independent life [2]. However, there can be many instances where the elderly person can forget what activity they are conducting, which can lead to anxiety [3] because of the awareness that they are slowly losing their independence. Therefore not only does the recognition of activities provide useful information about what activity is being conducted, it is also possible to provide information about what activity the elderly person is meant to be doing next, or even provide alternative options. It is intended that this support will also assist when activities are interweaved.

In the area of activity recognition within the home, there has been a significant amount of work that has been conducted. The recognition of ADLs can be split into three subcomponents: feature detection, feature extraction and models for recognition. 'Dense sensing' [4] is currently a favoured technique for detecting features and is based on tagging numerous objects around the home (e.g. Kettle) with wireless transponders and sensors that transmit information whenever the object is used or touched via an Radio Frequency Identification (RFID) reader. This form of feature detection non-intrusively collects a wide range of sensor data, which departs from other approaches for feature detection that rely on a visual based system. Wearable sensors are also key elements of feature detection, which have been used in the form of accelerometers and audio sensors that provide data about particular body motions and the surroundings where the activity is being conducted. Wang et al [5] have shown that a range of fine grained arm actions like 'drink with glass', 'chop with knife' may be determined by using feature detection technique based on wearable sensors. The identification of these actions is then combined with object data (e.g. Kettle) in order to achieve accurate activity recognition.

In terms of models for activity recognition, Markov models have been popular choice for the construction of probabilistic models, one such approach was by Wilson et al [6] where task recognition experiments were conducted and analysed by Hidden Markov Models (HMM) and the Viterbi algorithm. This determines which task is currently active from a sequence of sensor events. This and similar approaches are not as efficient when the tasks can be carried out in a random order. This is a problem as human beings often vary the order of task execution when achieving a goal. The use of ontologies [7] and data mining techniques have also been exploited in order to solve the problem of incomplete sensor readings or missing data. The ontologies are used to construct reliable activity models that are able to match between an unknown sensor reading with a word in an ontology which is related to the sensor event. For example, a Cup sensor event could be substituted by a Mug event in the task identification model ‘Make Tea’ as it uses Mug.

The work described in this paper is performing much the same function, which is activity recognition using object usage data. However since privacy is a concern as extensive monitoring can be intrusive our approach utilises more knowledge about the structure of ADLs. The automation element of our approach is based on hierarchically structured plans (representing ADLs) where knowledge at different levels of abstraction are used together to determine the activity as well as analyse the intentions of the elderly person, by predicting what ADL they might conduct next. Being able to analyse intentions of the elderly person with Alzheimer’s disease allows the possibility of providing assistance while the person conducts ADLs, if the ADL is interrupted by another ADL, as well as instituting safeguards. The work in this paper targets the group of elderly people who are between the mild and moderate stages of Alzheimer’s disease.

2. Hierarchal Activities of Daily Life

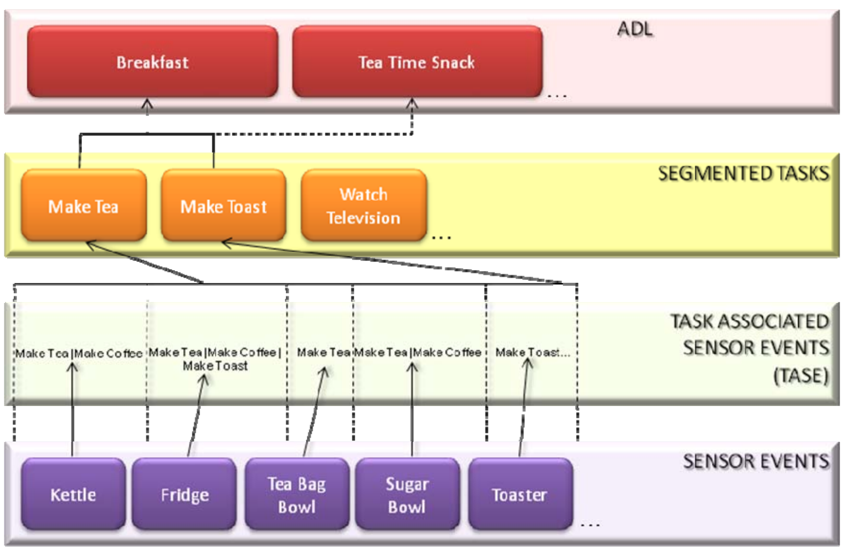

Figure 1 - Hierarchal Structure for ADL Recognition

In this paper, ADLs are modelled in a hierarchical structure of plans, which allows us to decompose the ADLs into different subcomponents. With this type of modelling, ADLs can correspond to simple tasks, such as “Switch on Kettle”, or more complex activities such as “Make Breakfast”. The hierarchal structure allows ADLs to be nested within other ADLs. In addition ADLs may occur in parallel with other ADLs.

Figure 1 is an example of a hierarchal structure of an ADL, where “Make Breakfast” ADL consists of a simple sequence of tasks, Make Tea, Make Toast..., but these may be in any order, or be performed in parallel. The lowest tier of the hierarchy of ADLs is the sequence of objects that have been detected and mapped as sensor events. These 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. The Task Associated Sensor Events (TASE) are partitioned into segments where each segment is mapped to a task, from which the activity recognition and intention analysis is carried out.

Within the hierarchal structure, the ADLs are represented in a hierarchal plan representation language called Asbru. This 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 [8]. Asbru has many features which allow each skeletal plan to be flexible and to work with multiple skeletal plans [12]. Representing ADLs in this language allows ADLs to be modelled as plan, where plans contain sub-plans. A plan that cannot be decomposed any further is called a task. Therefore the ADL recognition is based on recognising plans from constituent tasks, while tasks are recognised from sensor events sequences based on object usage data.

We have previously developed three different approaches to task recognition. One is based on Multiple Behavioural Hidden Markov Models (MBHMM) [9] and the other using a technique inspired from an approach for text segmentation [10]. The third approach is based on Generating Alterative Task Sequences (GATS) from a stream of object usage data based on the product of each task associated sensor event. The validity of these approaches was tested by carrying out episode recovery experiments within a kitchen.

The remainder of this paper will describe our current work, which is based on merging the high level activity recognition approach with the GATS approach for task recognition. The merging between both of these approaches is formed by the generation of task sequence costs based on the low level task recognition process. These costs are then used to generate ADL set utilities, which are then used to determine which activity is currently being conducted in the high level activity recognition process.

As well as being able to carry out activity/task recognition, this merged approach allows us to:

- Take timing intervals into consideration of when an activity is being conducted, which enhances the pruning process of when trying to distinguish the correct activity that is being conducted.
- Recognise a task from a set of sensor event data which has different variations of how a task can be performed. Also recognise tasks from data where there is noise in the data (e.g. missing or unrelated sensor events).

3. Task Associated Sensor Events

Task recognition can be carried out by simply segmenting sensor events into segments that correspond to a particular task. However the problem with this approach is that there is always a possibility of sensor event segments being generated that are incorrect and sometimes may bear no resemblance to that task that is actually being conducted. Our approach assigns a probability $P[b \mid a]$, where a is a task and b a sensor event. These are assigned as prior probabilities or established during a training data phase. Using identification from the higher tier of our approach it is possible determine the probability proportions of $P[\ a \mid b]$ given the activity that has been recognised in the higher tier.

In the GATS approach each sensor event is associated with all the tasks that correspond to the sensor event. For example, kettle sensor event can be associated with the tasks make tea or make coffee. If the tasks are denoted by letters so

- Task "Make Tea" is denoted by letter A
- Task "Make Coffee" is denoted by letter B

Then the sensor event "kettle sensed" is replaced by Make Tea| Make Coffee = A+B, where + is used to represent the disjunction.

4. Generating Alternative Task Sequences and Associated Costs

The objective of the GATS approach is to output ordered lists of alternative tasks sequences given an input set of events. Each task sequence has an associated cost. The cheapest task sequence is taken as the most likely task sequence as the cost function is intended to reflect the compliance of the task sequence with the event sequence and the relative frequencies of ADLs.
The function of the low level task recogniser can be represented as:

$$e1, e2, \ldots en \quad \rightarrow \quad \{<TS1, c1> + <TS2, c2> + \ <TSm, cm>\} \qquad (1)$$

where the e_i represent the sensor events in order of observation, *TS* stands for a task sequence consistent with the event sequence. *m* is a parameter chosen when the task recogniser is asked for its set of task sequences that match the events, to limit the number of possibilities generated. *m* is an upper limit, in the sense that if there are fewer than m possibilities, then only actual possibilities are generated. As an example of the inputs and outputs, after the events *e1*, *e2* and *e3* are observed, a list of two possible task sequences, *ABC* and *ABD*, might be generated, where *A*, *B*, *C* and *D* are tasks. *ABC* will have a cost and so will *ABD*. The set of alternative (and mutually exclusive) task sequences and their costs will be represented as $\{<ABC, c1> \ + <ABD, c2>\}$.

The list of possible task sequences will have a different length, depending on the number of events to consider, the discriminatory power of the events, and the algorithm used to create the list. For computational reasons, when the events are processed the number of task sequences generated as hypotheses will have a prescribed upper-bound , but the task recogniser has the capability of generating more tasks sequences (if there are any) should the coordination layer request more sequences.
If, for the same event set, the task recogniser is asked to provide further n task sequences, it will generate an additional $\{<TS_{m+1}, c_{m+1}> + \ldots\ <TS_{m+n}, c_{m+n}>\}$ task sequence.
As a new event arrives the task recogniser is invoked, computing a new set of task sequences. Making this the output of this new invocation relate to the previous is a function of the task recognizer. It recognises when computing cost function associated with a task sequence that the more recent tasks are more important. Typically this is handled by some exponential weighting of costs, where the match of the tasks to the more recent events is given more weight.

The GATS approach mitigates the chances of not being able to recognise tasks that have been carried out via different variations, as it takes in to consideration all the possible types of task sequences given the task associated sensor events.

5. Generate ADL Set Utility

The high level ADL recogniser takes a task sequence for input, and creates as output a list of alternative ADLs sets, each with an associated utility. Each ADL set consists of a set of incomplete ADLs and for each such ADL, its complete predecessors. If there are no interwoven tasks then the ADL set will consist of a sequence of abutting ADLs, an incomplete ADL and its complete predecessors. We do not use the term sequence for ADLs as some of the ADLs can be concurrent. Events and tasks, however, are considered atomic and so the term event sequence and task sequence is valid. So each task sequence *t1*, *t2*,…*tm* generates the alternatives *ADLS1* , *ADLS2*… where *ADLSi* denotes a set of ADLs consistent with the task sequence. ρ_i is the utility of the ADL set.

$$t1, t2, \ldots tn \quad \rightarrow \{<ADLS1, \ \rho_1> + <ADLS2, \ \rho_2> + \ <ADLSm, \ \rho_m>\} \qquad (2)$$

Again the utility function should give a higher weight to discrepancies and surprise levels in the more recent ADLs.

There is a question about which events are used. In our implementation all events are used. In many cases this could be very inefficient as only the most recent events are of interest. The next option is to keep a sliding window of events. However, the key question about a sliding window is where it should start. It could be sensible to ensure either that a sliding window starts at an event that corresponds to the beginning of a task, or starts at an event that corresponds to the beginning of an ADL. In such cases the cost functions for the task sequences and the ADL sets are likely to reflect the true degree of match. However, if there are interwoven ADLs then the number of events could be large. A task sequence may have different interpretations in terms of ADLs, and these may mean that for the same task sequence a different length of tasks need to be remembered, and hence a different number of events. The most general option is to define a window of tasks and hence events for each ADL set under consideration.

The high level ADL recognition tool takes as input a list of task sequences with associated costs, as generated by the task recogniser. The output of the ADL recognition tool is, for each task sequence considered, a list of active ADLs and the ADL that is most likely to be carried out next. This tool is also able to utilise temporal information related to tasks, specifically the times associated with each ADL, and temporal information across ADLs from the location of the ADL within the schedule of ADLs, as well as retrieving the discrepancies and surprise indices for ADLs that are associated with each hypothesised task sequence. An overall measure of recognition for an ADL (separate from any other ADL) and for a sequence of ADLs with respect to a schedule has been devised and this will be used to select the ADL sequence that is the best match.

The utility of each ADL set is based on the cost of each segmented task sequence. To achieve accurate ADL recognition it is important to recognise as many tasks as possible within a window of events. It can be difficult process to generate the utility of every possible ADL in the library at the time of an activity taking place. Therefore our utility is based on ADL schedules within a certain time frame (example 9.00am to 9.15am), as this more manageable and provides accurate recognition and eliminating some of the unlikely possibilities at the very outset of the recognition process. However there are certain ADLs that can occur at any time, which are called interruption ADLs and these are modelled with in each ADL schedule in the ADL library. This is made possible by the representation language ‘Asbru’, as it can represent and model timing intervals between ADLs.

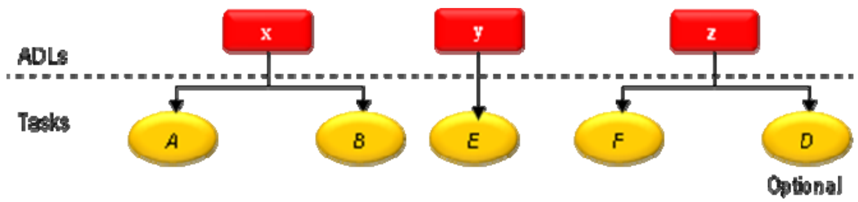

Figure 2 - Example of an ADL Schedule (9.00am to 9.15am)

Note that there is only one level in the ADLs in Figure 2, but in general there can be any number.

6. ADL High Level Recognition

In contrast to the low level task recognition, the high level ADL recognition gives an overview of all the possible ADLs that could occur within a given time. In addition, the approach is able to take into consideration any overlapping ADLs and also be able to distinguish which ADL is currently active by the tasks which are discovered in the lower level task recognition.

The next objective is to support recognition of tasks through feedback from beliefs held about ADLs. Consider a simple ADL: Prepare Breakfast, and suppose that a sub-activity of this ADL is to enter the kitchen. In Asbru whenever an ADL has been completed then it is labeled as executed. Also when the preconditions of an ADL have been met then the ADL is classified as being executed. For example, in order for the ADL ‘eat egg’ to be considered for execution a precondition may need to be fulfilled such as the ADL ‘make egg’ should already have been labeled as executed, in other words the egg should be cooked. Additionally an ADL can be classified as mandatory or optional.

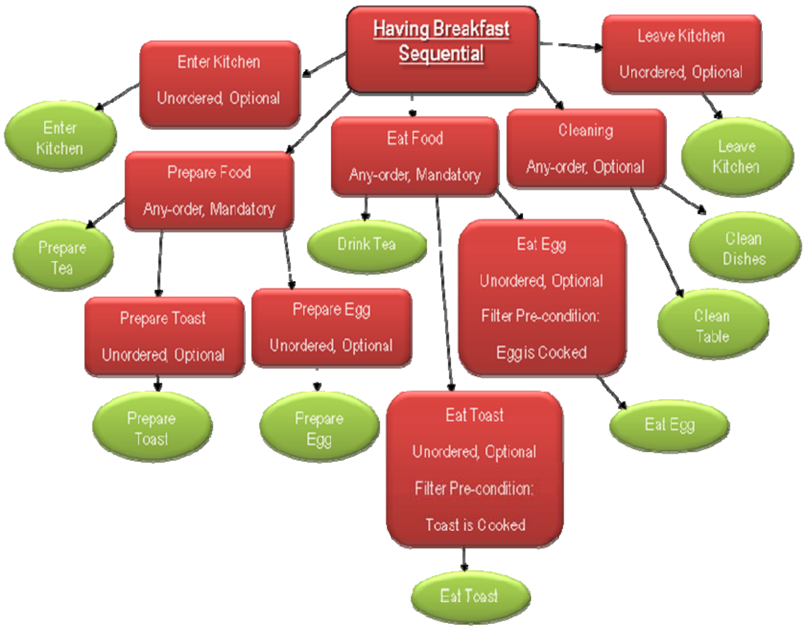

Figure 3 - ADL modelled in Asbru

Figure 3 shows an example of ADL "Having Breakfast" being modelled in Asbru, which includes the features of Asbru that have been mentioned. Our recognition system allows multiple activities to be tracked including tasks that may occur at the same time.

As Asbru is a plan representation language based in XML, the ADLs are constructed in XML. Therefore when constructing an ADL it is possible to construct one ADL per XML file, or it is even possible to construct a series of ADLs into one larger XML file, (e.g. ADL schedules). Both of these alternative ways can lead to a situation where the XML file will contain the same tasks that belong to different ADLs. Once an ADL has been detected by the high level recognition system this is then represented by the path of the XML that has been detected. If there is an instance where there are two possibilities then this is represented by two paths specifying the location of the task that has been detected. In order to distinguish between the different possibilities and correctly determine which activity is currently being conducted, this is done by calculating discrepancies and surprise indexes.

A discrepancy is a task that has not been detected, which should have been detected, if the ADL is executed. The overall discrepancy of an ADL is computed by summing the discrepancies of the sub-activities. In order to compute the overall discrepancy, two discrepancy counts for each ADL are calculated - completed discrepancy and incomplete discrepancy. If the sub-activity is known to be complete then the completed discrepancy of the sub-activity is used when computing the sum, otherwise the incomplete discrepancy is used.

The discrepancy count is further aided by a surprise index. The surprise index is used to account for the fact that the absence of some tasks can be more unusual than others, and quantifies this by accruing a measure of how likely a task is when an ADL is being executed.

While the discrepancy is computed whenever there is any missing mandatory task, such as make tea for the ADL Make Breakfast, the surprise index is the maximum of the conditional probability $P[a \mid b]$ of a missing sub-activity and tasks (a) given the ADL (b) that is being conducted.

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.

7. Experiments

The experiments conducted were based around ADLs that are normally conducted in the kitchen, hence these experiments took place in a kitchen with RFID transponders installed on its cupboards and objects, such kettle, dishwasher, utensils, and toaster (Figure 4).

Figure 4 - The kitchen showing all the locations of the passive transponders

The object 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 cleaning. The reason why 10 subjects were chosen is because people have different ways of ordering of carrying out a particular ADL, so there will be variability in the sensor stream. The activity sequence that the subjects report after carrying out the experiment is treated as ground truth, which is later compared with the recognition results from our approach.

The objective of these experiments is to work out the accuracy rate of our approach for ADL recognition. The accuracy has been measured by calculating the precision and detection rates based on the object usage data collected in the kitchen with the RFID reader. The precision rate is determined by the following:

$$precision = \frac{|\{relevant\_tasks\} \cap \{segmented\_tasks\}|}{|\{segmented\_tasks\}|} \quad (3)$$

The *relevant_tasks* are the tasks that are relevant to the ADL that is actually being conducted, while the *segmented_tasks* are the tasks that have been segmented correctly from the gathered sensor data, regardless whether they relate to the actual ADL being conducted.

$$\det ection = \frac{|\{\det ected\_tasks\} \cap \{\exp ected\_tasks\}|}{|\exp ected\_tasks|} \quad (4)$$

The overall detection rate of the tasks on this occasion has been determined by (4), where the detected tasks are tasks that have been correctly detected and are relevant to the ADL that is being conducted. The expected tasks are the number of tasks that are expected to be conducted within the ADLs, which is based on the collected ground truth data.

Two sets of experiments were conducted, the first experiment was based around trying to recognise constituent tasks that are relevant to the ADL being conducted. This was to see if the relevant tasks were being segmented correctly, hence the number of tasks that are actually relevant were not considered when calculating the precision rate. The second experiment made use of sensor event data for multiple ADLs being conducted at the same time. So the objective was to see if correct segmentation and classification of the constituent tasks still could be achieved.

8. Results

ADLs	Precision [%]	Detection [%]
Breakfast	95	90
Laundry	100	95
Put Shopping Away	95	90
Prepare Meal	100	95
Clean up Kitchen	100	95

Table 1 – Results for Task Relevance Experiment

It can be seen from the results in Table 1 that precision rates were high for all of the ADLs, as the GATs approach takes into consideration all possible task sequences, however it only considers the stream with lowest cost. Hence, the stream with lowest cost provided segmented tasks which more than often consisted of the relevant tasks that had been conducted. Another encouraging aspect of these results is that the detection rates are high, as the expected tasks frequently matched the ground truth data collected. Overall, this led to accurate ADL recognition.

ADLs	Precision [%]	Detection [%]
Multiple ADLs	65	90

Table 2 – Results for Task Relevance Experiment

Another important aspect of our work is recognising tasks/ADLs that are conducted together, which is known as interweaving, e.g. making tea while putting shopping away. It can be seen that there is drop in the precision rate, this is due to a greater number of tasks being segmented which may not be relevant to the ADL. However, the detection rate is encouragingly high, as our high level ADL recognition approach is able to take relevant tasks and see where it fits in the ADL, which can still help it identify which activity is being conducted. One of the advantages of the high level ADL recognition is that even if the low level recognition approach misses a task, the ADL recognition approach can still distinguish which ADL is being conducted.

9. Conclusion

The work that has been described in this paper is a first half of a large scale experiment, which is currently being conducted around the home as oppose to just focusing on ADLs conducted in the kitchen. These future experiments will

also test the ability of our ADL recognition approach for analysing the intentions of the person conducting the experiment, e.g. what ADL will the person conduct after completing the one they are currently doing now.
Future enhancements to our recognition approach will also incorporate storing and learning of ADLs conducted on a regular basis.

10. References

[1] M. Knapp, "Dementia UK". London School of Economics and Institute of Psychiatry at King's College London, 27 February 2007, Published by Alzheimer's Society, UK, (2007).

[2] The Alzheimer's Association. "Activities at Home, Planning the day for a person with dementia". Alzheimer's Association National Office 225 N. Michigan Ave., Fl. 17, Chicago, IL 60601, 1 -12 (2005).

[3] L. Ferretti, S.M. McCurry, R. Logsdon, L.E. Gibbons, L. Teri, "Anxiety and Alzheimer's Disease". Journal of Geriatric Psychiatry and Neurology, vol. 14, no. 1, 52-58 (2001).

[4] M. Philipose, K.P. Fishkin, M. Perkowitz, D.J. Patterson, H. Kautz, D. Hahnel, "Inferring Activities from Interactions with Objects". IEEE Prevasive Computing Magazine, vol. 3, no. 4, 50-57 (2004).

[5] S. Wang, W. Pentney, A. Popescu, T. Choudhury, M. Philipose, "Commonsense-based Joint Training of Human Activity Recognizers". Proceedings of the 21st International Joint Conference on Artificial Intelligence (2007).

[6] D.H. Wilson, D. Wyaat, M. Philipose, "Using Context History for Data Collection in the Home". In Gellersen, H.-W., Want, R., Schmidt, A. (eds.) PERVASIVE 2005. LNCS, vol. 3468 (2005).

[7] E. Munguia-Tapia, T. Choudhury, M. Philipose, "Building Reliable Activity Models using Hierarchical Shrinkage and Mined Ontology". Proceedings of the 4th International Conference on Pervasive Computing, 17–32 (2006).

[8] Fuchsberger, C., Hunter, J., McCue, P, "Testing Asbru Guidelines and Protocols for Neonatal Intensive Care". Proceedings of the Tenth European Conference on Artificial Intelligence, Aberdeen, United Kingdom, 101–110 (2005).

[9] U. Naeem, J.Bigham, "A Comparison of Two Hidden Markov Approaches to Task Identification in the Home Environment". Proceedings of the 2nd International Conference on Pervasive Computing and Applications, 383-388 (2007).

[10] U. Naeem, J. Bigham, J. Wang, "Recognising Activities of Daily Life Using Hierarchical Plans". Proceedings of the 2nd European Conference on Smart Sensing and Context, LNCS 4793, 175-189 (2007).

[11] R. Kosara, S. Miksch, Y. Shahar, P. Johnson, "AsbruView: Capturing Complex, Timeoriented Plans - Beyond Flow-Charts". Second Workshop on Thinking with Diagrams, Aberystwyth, United Kingdom, 119–126 (1998).

Usman Naeem is a PhD candidate at the School of Electronic Engineering and Computer Science at Queen Mary University of London. His research interests include activity recognition in the home, modelling different variants of human behaviour and sensing.

John Bigham is a Reader at the School of Electronic Engineering and Computer Science at Queen Mary University of London. His interests in security relate to the protection of critical infrastructures and middleware, the discovery of anomalies and fraud detection, e.g. from within an organisation.