A hierarchal framework for recognising activities of daily life

Naeem, Usman

A Hierarchal Framework for Recognising Activities of Daily Life

Usman Naeem

Submitted for the degree of Doctor of Philosophy

School of Electronic Engineering and Computer Science
Queen Mary, University of London

2009

To my beloved family

Abstract

In today's working world the elderly who are dependent can sometimes be neglected by society. Statistically, after toddlers it is the elderly who are observed to have higher accident rates while performing everyday activities. Alzheimer's disease is one of the major impairments that elderly people suffer from, and leads to the elderly person not being able to live an independent life due to forgetfulness. One way to support elderly people who aspire to live an independent life and remain safe in their home is to find out what activities the elderly person is carrying out at a given time and provide appropriate assistance or institute safeguards.

The aim of this research is to create improved methods to identify tasks related to activities of daily life and determine a person's current intentions and so reason about that person's future intentions. A novel hierarchal framework has been developed, which recognises sensor events and maps them to significant activities and intentions. As privacy is becoming a growing concern, the monitoring of an individual's behaviour can be seen as intrusive. Hence, the monitoring is based around using simple non intrusive sensors and tags on everyday objects that are used to perform daily activities around the home. Specifically there is no use of any cameras or visual surveillance equipment, though the techniques developed are still relevant in such a situation.

Models for task recognition and plan recognition have been developed and tested on scenarios where the plans can be interwoven. Potential targets are people in the first stages of Alzheimer's disease and in the structuring of the library of kernel plan sequences, typical routines used to sustain meaningful activity have been used. Evaluations have been carried out using volunteers conducting activities of daily life in an experimental home environment. The results generated from the sensors have been interpreted and analysis of developed algorithms has been made. The outcomes and findings of these experiments demonstrate that the developed hierarchal framework is capable of carrying activity recognition as well as being able to carry out intention analysis, e.g. predicting what activity they are most likely to carry out next.

Acknowledgement

I would like to express my gratitude to my supervisor, Dr. John Bigham for his helpful advice and direction throughout the research that has been presented in this thesis. He has been an inspiration, as his logical way of thinking and guidance has been of great value to me.

Also I would like to dedicate my thesis to my family for their continuous support and love.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the beginning and closing stages of a person's life they have a high level of dependency on others. In today's working world with its emphasis on the nuclear family there are fewer problems with looking after babies, but there are significant problems in looking after the elderly. In Britain, in common with most western societies and now Asian societies (particularly China with its single child policy), there has been an increase in the proportion of elderly people and many people find it hard to look after their parents because of commitments and distance, or just do not want to look after their parents when they need care. Obversely many old people want to remain independent for as long as they can. The existence of Alzheimer's disease [1] among the elderly is also seen as a concern, as this disease gradually destroys the elderly person's memory and their capability to learn, communicate and carry out everyday activities. These elderly people are usually sent to care homes where they are looked after by other people. However this approach is not completely successful due to issues concerning isolation and even abuse. Therefore the introduction of smart homes has become of a promising solution for the elderly, as it aims to provide the elderly the ability to lead an independent life until, e.g. Alzheimer's disease reaches a severe stage. It has been suggested that smart homes are the future for elderly people who are in the early stages of the Alzheimer's disease. It is important for the wellbeing of these elderly people that they perform day-to-day tasks such as dressing, cooking, and toileting.

This has been recognised by gerontologists, who developed a detailed list of activities in 1963 [2] which should be carried out by the elderly person, known as Activities of Daily Life (ADL). They are also referred to as Activities of Daily Living [3]. Being able to monitor these ADLs is seen as a key approach for tracking functional decline among elderly people [4]. Caregivers in the US prescribe these ADLs to the elderly in order for them to carry them out. Information regarding the ADL would then be collected on each visit from the caregiver, via interaction with elderly person. This collected information is important as decisions on medicine allocation depend on it. However the way this information is collected can often lead to inaccurate data, as elderly people can misinterpret facts and forget. Additionally, the size of the window used for collecting data is narrow in comparison to period being evaluated. This illustrates that manual data collection regarding ADLs can be long and tedious, imposing further workload and burden on caregivers. This is an example of why it is important to develop algorithms that can discriminate between different ADLs and determine the intentions of old people as they carry out everyday tasks. The interpretation of ADLs in this thesis is broader than those for Alzheimer's patients. We include every day activities, such as washing, cooking, preparing a sandwich. There has been considerable amount of research on smart homes and ADLs, however the research conducted to date has focused on trying to find out what activity the elderly person is currently carrying out [5], whereas this research will take the concept of determining ADLs further by analysing and reasoning about the intentions of the elderly person, which will allow the determination of the next ADL they are going to carry out.

With privacy also being a growing concern, the algorithms developed are based solely around object usage data, which is data collected from everyday objects that are used to perform the everyday activities. Such as an object is a kettle, which is used for an activity like making tea and others.

1.1 Research contributions and novelty

The work in this thesis is aimed at more reliable identification, e.g. reducing the number of false positives and false negatives, by developing new techniques for finding ADLs from object usage data, and so interpreting the intentions of an elderly person.

The major contributions of this thesis are summarised as follows:

- A hierarchal framework for identification of ADLs is proposed. Knowledge at different levels of abstraction are used together to determine what ADL is currently active. Unlike existing activity recognition approaches, the proposed approach divides activities into two levels of recognition. The low tier is concerned with recognising constituent tasks from sensor event data, which is based on the collected object usage data, while the higher tier carries out recognition of activities from the tasks recognised in the lower tier. The higher tier is in itself hierarchical. Hierarchically structured plans represent nested ADLs where knowledge at different goals and sub-goals are used together to represent the activity as well as encapsulate the overall intentions of the elderly person. In the case of Alzheimer's patients living in a smart environment, this type of analysis allows the possibility of providing assistance and services while the person conducts ADLs, even if an ADL is interrupted by another ADL. It can also institute safeguards.
- At each tier novel components have also been developed. For example, adaptations of HMMs, namely Multiple Behavioural Hidden Markov Models (MBHMMs) and techniques based on text segmentation have been used for the low level modelling. In addition, a simple minded approach called Generating Alternative Task Sequences (GATS), has been developed which takes into account the association of each task and the objects that are used to perform that task in a direct manner. This association then leads to alternative tasks sequences being generated from which the appropriate task sequence is chosen to be the correctly classified set of tasks.
- A plan representation language Asbru has been exploited for the high level modelling. Asbru was first developed to model clinical guidelines and protocols and is ideal for the modelling of ADL hierarchies. Based on this, new techniques for plan recognition have been developed and integrated into the approach. Complexities associated with suspension and interweaving of ADLs have also been considered and how they can be handled in the overall activity recognition process. The plan representation

capability of the higher tier also allows the capability of allowing future intention analysis, by being able to determine the next activity that will most probably be conducted.

- A simple decision tree learning algorithm is used to support the prediction of following ADLs, even if there is no match to a plan in the plan kernel. The feature space used by the decision tree algorithm is extended by using sub-goals and goals associated with plans in the known plan kernel. The novelty factor here is how the decision trees are not just for classification, but as way of supporting the current hierarchal approach, when ADLs outside the framework of the core ADLs constructed need to be recognised. Like all decision trees this approach is reliant on data collected during a training period.

1.2 Organisation of this thesis

Chapter two, Background and Motivation, looks at the potential application context where the proposed research will be applied in the future. This is care of the elderly with early stage Alzheimer's disease in smart homes. Current smart home projects and related state of the art research will also be described. There has been substantial amount of research relating to interpreting ADLs and on Smart Homes. The research has been revolved around the idea of trying to determine what type of ADL the elderly person is carrying out. Note that there has been no attempt to validate the research here on actual patients as that would be premature. The thesis tries to evaluate the tools developed in related experimental settings using healthy volunteers.

In chapter three, Method, the algorithms that have been generated to distinguish between different ADLs and analyse intentions and their implementations are described in detail. The implementation of these algorithms is split into two categories, namely algorithms for models based on sensor data (also called task recognition in this thesis) and those built around models of behaviours of individuals (also known as ADL recognition in this thesis). The combination of the models and algorithms for the low level and high level models are used to determine which ADL is currently active. In addition the algorithms developed for the high level models will also be able to predict the future intentions of the subjects. For the low level models, a variety of frameworks has been developed and tested and are presented with the experiment results. For the high level models a plan representation language called Asbru [6] is used to represent and analyse the core behaviours of the subjects. These represent partial order plans.

Chapter four, Interaction between levels, proposes the approaches to the interpretation of intentions based on exploiting interactions between the two levels of modelling and describes how information is fed back between the two levels. In addition, this chapter also describes how decision trees are applied to the developed hierarchal approach in order to enhance the overall activity recognition process.

The final chapter concludes the thesis, and discusses issues that could be investigated as an extension of this research.

Chapter 2

Background and Motivation

This chapter consists of two main sections. The first section focuses on elderly people and looks at their vulnerability in terms of accident rates and at the limitations of care homes. Then the group of elderly who are in the early stages of Alzheimer's is considered. The second section looks at existing smart home projects for the elderly and describes the state of the art in this area of research, primarily concerned with the recognition of Activities of Daily Life (ADL). In addition, this section also looks at work related to the techniques that have been applied to the developed recognition approaches in this thesis.

2.1 The Vulnerability of Elderly People

2.1.1 Elderly People and Accidents

Elderly people spend most of their time in their homes, which is usually considered to be a safe option. However approximately ten people a day are killed in accidents in the home. A further 7,000 people get injured, requiring hospital or GP attention. Table 1 [7] shows the number of deaths that were caused by accidents at homes in the year 2000.

		Number of deaths	Percentage of deaths
Males	Accidents in home and communal establishments	1,945	18.7
Females	Accidents in home and communal establishments	1,705	27.9

Table 1 - Deaths caused by accidents at home

The statistics in Table 1 shows totals for males and females over all age groups, including the elderly. This is a clear indication that homes can be hazardous for all age groups. After toddlers, it is the elderly who are more likely to fall prey to accidents in the home. The number of incidents involving falls and fall-induced injuries concerning elderly people is on the increase and is seen as one of the leading cause of deaths among elderly people [8].

It is believed by analysts that the proportion of serious accidents involving over 75 year olds will increase. Table 2 [9] below highlights and estimates accident rates for serious cases within the UK from 1996 to 2010.

Age Band	Annual rate per 10 million population		Estimated change
	1996	2010	%
65-74	31	39	+26
75 and over	149	231	+55

Table 2 - Accident rates for serious cases, 1996 to 2010

The table shows a predicted increase in serious accidents for both 65-74 and 75+ age bands. The increase for people aged over 75 is double that for those in the 65-74 age bracket and demonstrates our vulnerability as we grow older. One of the most common accidents involving elderly people is a fall. Nearly three-quarters of falls that occur among the 65 and over age group result in arm, leg or shoulder injuries. As well as falls, the elderly are more likely to be involved in accidents with fire. This is due to them having poor sense of smell and also having slow mobility and less resilience to the effects of smoke and burns. Another reason why fire accidents are more frequent is because the elderly often lack a good memory. This can be fatal as the elderly person can leave the gas on if they are interrupted or distracted. This scenario can lead to a fire or even accidental poisoning, which is the third most common fatal accident that elderly people have. The causes of accidental poisoning are carbon monoxide, leaving the cooker gas on and medicine overdose. Burns and scalds can be particularly fatal to the elderly. Burns are usually caused by radiators, electric fires and cookers, while scalds are usually caused by kettles.

The most severe accidents involving the elderly either take place on the stairs or in the kitchen. The bedroom and living room are usually the most common places for minor accidents that take place.

Due to this increase of serious accidents involving elderly people, it is important to understand why the elderly are victims of such events. This could be seen as the cumulative effect of physical, mental and social disuse that can result in frailties that expose the elderly to accidents in the home environment [10].

2.1.2 Alzheimer's Disease

A factor that has a big influence on the well being and the rate of accidents among elderly people is Alzheimer's disease (AD). This is a progressive disease of the brain and is a fatal neurodegenerative disorder that is visible via cognitive and memory deterioration of the elderly person as they try to carry out activities of daily living [1]. AD progressively destroys the person's memory and their capability to learn, communicate, make judgements and carry out everyday activities by themselves. Alzheimer's disease is known to be the most common form of dementia. The term dementia is defined as a "*global impairment of Intelligence, Memory and Personality, in clear consciousness*" [11]. This can usually occur at any age; however it is more frequent within the elderly age group occurring in 5%-10% of those who are 65+ and in 20% for those who are 80+ [12].

Studies in the past have indicated that in year 2000 the number of people with AD in USA was an estimated 4.5 million, and it has been estimated that if there were no advances in therapy this figure of 4.5 million could rise to 13.2 million [13]. A study led by Knapp on the social and economic impact of dementia in the UK has discovered that currently (2009) there are 700,000 people [14] who suffer from dementia in the UK. This figure is expected to increase and it is predicted that by year 2025 (Figure 1) there will be one million people in the UK with dementia. Cost is something that needs to be considered. Currently Alzheimer's disease alone costs the UK £17 billion, which is equal to £539 per second.

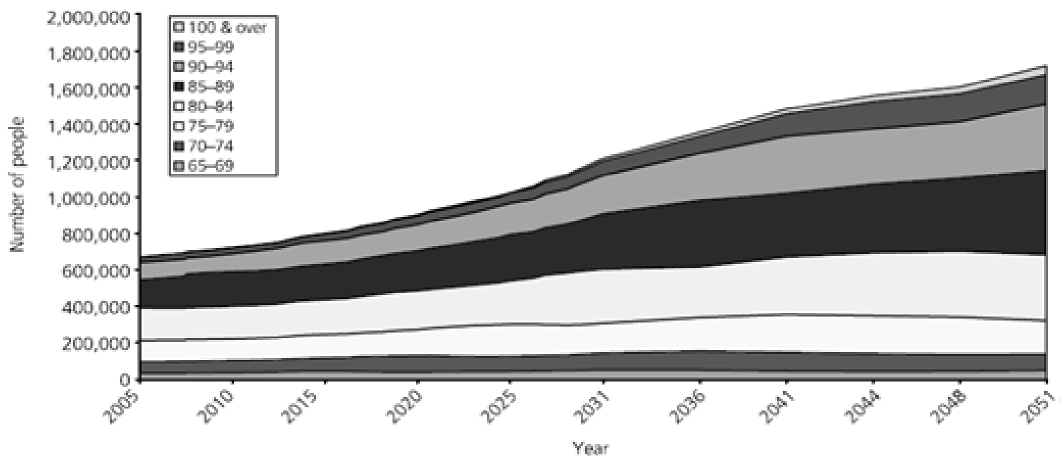


Figure 1 – Volume of different age groups estimated to suffer from dementia

The graph in Figure 1 [14] indicates that the age group of 75-79 and 80-84 is where a majority of elderly people are diagnosed with the disease.

Alzheimer's disease attacks the nerves and cells in the brain which causes plaques and tangles of certain proteins to develop around region of the brain cells [15] [16], which steadily destroys connection between the brain cells and leads to shrinkage of the brain region that is used for learning and memory [17]. Unfortunately this disease cannot be diagnosed accurately while the person is still living, as it is not possible to see the tangles and plaques until the person has died.

The three most common forms of dementia are vascular dementia, dementia with lewy bodies and Frontotemporal dementia which have different affects.

Vascular dementia is caused when there is not enough supply of oxygen to the brain after a person has just suffered a stroke [18]. In addition, a condition like hypertension can also cause vascular dementia, as hypertension is something which affects the heart and circulation of blood to the brain, therefore this can lead to progressive symptoms such as communication and concentration problems [19].

Dementia with lewy bodies is caused by protein deposits that develop over time within the nerve cells inside the brain, this leads to a malfunction in the brain as the person's memory and concentration is then affected [20]. Parkinson's disease is very similar to this type of dementia, as they both share similar symptoms, such as slow movements, tremors and on some occasions many people suffer from hallucinations [21].

Frontotemporal dementia is not as common as the previous types of dementia mentioned in this chapter. This type of dementia affects the front of the brain, however in the initial stages it does not affect the memory as much as the person's behavior and personality is affected, which can lead to dramatic changes in behavior [22] [23].

As Alzheimer's disease is a progressive disease it gets more severe over time, which means there is a certain point that the elderly person will reach where he or she will be completely reliant on another person, as they will not have the mental ability to carry out everyday activities. The symptoms of this disease are different for each person who suffers from it; however the main stages of this disease can be recognised, these are mild, moderate and severe.

The first stage of the Alzheimer's disease is known as the mild stage, this is where the person starts to get minor brain problems such as forgetting daily activities and not being able to carry out straightforward arithmetic. This on many occasions can be overcome with the help of a diary and daily activity lists. However as the person suffers from memory loss this causes anxiety [24] [25], as the person feels worried and nervous because they feel they may lose their independent way of life. In some cases the level of anxiety caused by the memory loss can also lead to night time awakening [26], which can be disturbing for the person and lead to depression.

As the disease progresses it then reaches the moderate stage. This is where brain problems develop further as the disease affects the memory of recent events. On many occasions this can also lead to confabulation [27] [28], which is when the person starts to get confused and conceives fictional events in order to occupy the gaps in their memory [29]. Very rarely this can also be in the form of a fantasy that can occur as a factual account in the memory, e.g. people claiming to be abducted by aliens.

Figure 2 - The different stages of the Alzheimer's disease

Dysphasia is a common symptom in the moderate stage of the disease, as the person with the disease can find it difficult to find the right words and therefore more than often they are lost for words [30]. As well as dysphasia, disorientation is also symptom at this stage of the disease. Disorientation [31] [32] is when a person tends to gets lost in familiar surroundings, due to the loss of awareness of the place or current time.

Within in this moderate stage the mood of the person can be unpredictable and change frequently due to frustration. This frustration is caused by the difficulties that occur in this stage of the disease, which de-motivates and alarms the person.

The final stage of the Alzheimer's disease is known as the severe stage due to the level of disorientation and confusion. Studies have also suggested that the occurrence of hallucinations is associated with the cognitive decline in the disease [33]. Paranoid delusions are also common at this stage of the disease [34]. The level of unpredictability in the person's behavior develops further as it frequently leads to aggressive behavior [35], as the person can become demanding and violent.

As well as not being able to carry out activities by themselves in this stage of the disease, many people are not able to pay attention to personal hygiene, and lose control of their bladder and bowels. In relation to the 3 stages of the disease, Table

3 shows the measure of global and cognitive dysfunction that is associated with each stage of Alzheimer's disease [36].

Stage	Duration, Year	Global Deterioration Scale, * score	Mini Mental State Exam, † score	Global autonomy
Mild	2-3	3-4	26-18	Independent Living
Moderate	2	5	17-10	Supervision Required
Severe	2-3	6-7	9-0	Total Dependence

*Scale measures progressive need for assistance in daily activities (e.g., choosing clothes, dressing); scores range from 1–2 (normal) through 6–7 (severe dysfunction).
†This 22-item scale measures cognitive function; scores range from 30 (excellent function) to 0 (severe dysfunction).

Table 3 - Measures of global and cognitive dysfunction

The measures for the mild stage shown in Table 3 indicate that independent living is possible in this stage, therefore it is important that the elderly Alzheimer's sufferer remains in this stage as long as possible. Conducting *general* ADLs such as bathing, grooming and eating; *instrumental* ADLs (IADLs) [37] [38] such as maintaining the household and preparing meals; and *enhanced* ADLs (EADLs) [39] such as using the Internet to shop online, can be used to assess the levels of functional decline among patients with Alzheimer's. It is hoped that the work conducted in this thesis will be of potential for elderly people who are in the transitional stage between mild and moderate of this disease.

2.1.3 Care Homes

Care homes have been around for many years. These include old-age care homes run by the state. There are many charities like Help the Aged that provide accommodation for the elderly. However, care provision has had its share of problems and controversies in the past and face new ones. Some are described below.

Cases of abuse in elderly care homes have been highlighted by many media campaigns. For example, in 1999, a nursing home worker, was jailed for four years, for constantly abusing eight elderly women who were in his care [40]. Such cases have tarnished the reputation of old-age care homes and put doubts into the minds of people as to whether old-age care homes are a safe place for elderly people.

After the reports of such cases the government decided to incorporate national standards intended to make elderly abuse a thing of the past. One scheme that the government introduced is called the Protection of Vulnerable Adults (POVA) [41]. This scheme required checks on individuals who decide to become home carers. This scheme has been successful to an extent and has resulted in more than 700 people banned from working with vulnerable elderly adults [42]. However, another problem is providing a good quality of life for the elderly. A study was conducted by Hancock et al to identify the unmet needs of elderly people with dementia in care homes. In order to carry out the study two hundred and thirty eight elderly people with dementia were selected from care homes across the country to take part in the study. The needs of the elderly people were identified via the Camberwell Assessment of Needs for the Elderly (CANE) [43]. It was discovered that environmental and physical needs were usually met, however, disability care, mental health care and social needs, e.g. company and daytime activities were not met. In addition to this a study was conducted by Hoe et al [44], which was concerned with the Quality of Life (QoL) that the elderly people with Alzheimer's disease receive in residential care homes. The assessment in this study was based on 119 QoL scales which were completed by both the elderly and the care home staff. The results from the elderly and the care home staff were contrasting as the elderly people's scores strongly correlated with scores for anxiety and depression, while the QoL scales rated by staff correlated with the scores for increased dependency and behaviour problems. This suggested that the elderly people's assessment of QoL was based on their mood, intentions and how they felt, while the staff members assessment was based on dependency factors. This suggests that staff members in care homes need to be aware of elderly people's intentions and moods, as this is far more important to them than the level of dependency.

Finally, the rising cost that families have to pay has also raised doubts about care homes as a solution. For example, there have been cases where the increase in prices (from £585 to £763 within a three year period) of care home accommodation has led to people feeling morally blackmailed, as they have no choice due to their work commitments but to continue paying for their parents care [45].

The rate of accidents at home and the failure of care homes suggest that the elderly are vulnerable and they find it difficult to lead an independent life. In general, with increasing age the elderly increasingly lack the following capabilities:

- Vision and sharpness
- General awareness of potential hazards
- The capability to carry out multiple tasks
- Speed and nimbleness while carrying out tasks (e.g. such as turning the cooker off late)
- Lack of awareness and forgetfulness is vulnerability which can be fatal, e.g. Leaving the cooker gas on.

In all cases, elderly people are more likely than younger adults to meet with accidents in the home, this is due to their sensory and cognitive impairment as well as medical conditions, which can lead to increased use of drugs and can present problems [46]. This can lead to slower reaction times in the event of a fire, as victims are unable to escape easily and quickly. For an old person, once an injury has been sustained, the recovery process takes longer and therefore leads to slow healing, secondary infections and complications.

This thesis does not address a way to ameliorate these losses in the person but investigates ways of monitoring, in general, that has the potential to mitigate the effects of the decline in capabilities.

2.2 Smart Homes

A Smart Home is a type of house that has a communication infrastructure installed within it. This communication infrastructure allows various systems and devices in the home to communicate with each other [47]. Smart homes are also referred to as intelligent homes, automated homes and networked homes.

A simple description that sums up the smart home concept is in a report on smart homes for the Joseph Rowntree Foundation and the Chartered Institute of Housing:

"Cars have central locking, electric windows, remote controlled mirrors, CD auto changers – and the rest! And factories, offices and shops are often highly automated, giving staff control over their environments, and making buildings more efficient. Automatic doors, blinds that close when the sun comes out, infra-red lighting controls – they are all becoming commonplace.

But you don't find that sort of thing in people's homes much … or do you?

We do have remote controls for our TVs, we do have smoke detectors and passive infra-red burglar alarms, we do have timers on our central heating. But all these devices are separate entities. Each affects only one activity or aspect of the home.

Smart Homes are about something much more exciting. They are about using the latest information and communications technology to link all the mechanical and digital devices available today – and so create a truly interactive house." [48]

One of the main aims of Smart Homes is to improve the standard and quality of living of people within the homes. This is typically done with electrical devices (sensors, monitors, aids of different kinds) that are placed around the home. These devices then work in conjunction with a manual control unit or they work according to the specifications encoded by the device software programmer, e.g. to determine whether a person is actually sleeping or awake while they are lying down on a bed.

Also the use of devices already in the home, such as a television, can be used to enhance the everyday living of elderly people. Ghorbel et al made use of a television to interact with the elderly to offer services such as medicine reminders [49]. The idea of providing services via a television is supported by the fact that a television is something that is used largely by the elderly community. However services being provided in this way will require some level of training for some groups of elderly people in order for them to take advantage of these services.

The way Smart Homes can improve the standard of living is by meeting the needs of the person whose home it is. One way this can be done is by making a system that assists the person (i.e. elderly person) in their home by enhancing their view of the environment and their memory and providing context sensitive support to safeguard the elderly person while the elderly person is carrying out daily activities.

2.2.1 Elderly People, Smart Homes and Independent Living

Elderly people spend most of their time at home. While they are at home they carry out a variety of activities such as brushing teeth, taking a shower, preparing breakfast. The home is also place where they can rest and relax, as well as socialise with friends and family.

The quality of life of an elderly person can be enhanced significantly by them living in a smart home environment because of the extra support received from the intelligent environment [50]. This view is also shared by the county council social services in South Norfolk, who aim to help the elderly carry on living in their homes independently. In April 2005, the county council social services in South Norfolk launched a smart home, which was designed for elderly people who have memory and metal health problems. The smart home itself was equipped with features such as flood alerts for sink and bath overflows, memory clocks and voice prompts. This smart home scheme has grown over the year, as the county council have recently launched a third smart home which is situated in the Thorpe area of Norwich [51]. These types of smart home are reliant upon simple prompts or surveillance carried out by external people. This is an expensive solution. The current smart homes do not have the ability to monitor and determine the state of the monitored person automatically. This feature would be a very useful feature as it would allow the smart home systems effectively to understand the daily actions of the elderly person. However it is important that smart homes that are created for the elderly are not just reliant on types of systems which will be centrally controlled by human service centres. An example of this is the types of smart home systems created by Lusora Limited. The aim of Lusora is to provide a monitoring system for providing an independent lifestyle for the elderly. The monitoring system works in conjunction with a 24 hour monitoring centres which are located in UK and US. However this is a very expensive solution, as the cost of running a 24-hour monitoring centre is high. Therefore either the elderly or the government will have to spend a lot of money to work with such monitoring systems. This monitoring system only tracks information that is provided by sensors such as pendants and wireless tags. It seems that there is no linkage of knowledge gathered by the sensors, which means the monitoring system can only detect simple problems, such as a fall. It cannot distinguish what Activity of Daily Life (ADL) the old person is doing. Indeed it can be argued that such a manual system should not. However, an automated system, where the data is retained private to the individual, is in a position to make more extensive and so be more supportive.

Smart Homes have a variety of features and goals. The principal goal of a Smart Home is to improve the quality of life by increasing self control that will allow the person to live an independent life which in turn will enable self-fulfillment [52]. Supporting independent living is also another related goal, as the smart home will make everyday life easier for the elderly. Health and fitness is important for the elderly, so another goal is to monitor the elderly person's health to prevent any illnesses. The delivery of care and medical services are provided to the elderly people through the use of technology within smart homes, an example of this is the emerging Telecare homes for the elderly and disabled [53].

It is evident that smart homes are one potential solution to helping the elderly live an independent life.

In relation to being monitored it is important to understand how the elderly will feel about a system monitoring or watching them, and observe any relevant legislation while creating system which will monitor the elderly.

One of the ways in which it was discovered how the elderly felt about monitoring was through Hampshire County Council. This Council is in charge of a smart home project for over 65s, which is currently active in the area of Hampshire. After conducting an interview with the project coordinator Julie Eden, many of the issues regarding privacy and feelings about smart homes were answered. When Julie was asked the question about how the people felt about being monitored, she said "*The people being monitored have a very positive attitude towards the idea of being monitored in a smart home. Some of the people have said that it provides reassurance, as they feel safer*". When asked whether there was any negative feedback towards being monitored, Judie replied "*there was no negative feedback; the people were very* *supportive towards the smart homes*". The people have also claimed that the introduction of the smart homes has increased their confidence, which has benefited their independent living.

An ethical problem that is likely to arise within smart homes is the data protection issues [54]. When Judie was asked about how Hampshire County Council deals with legal legislations such as data protection act concerning smart homes, she replied "*all the elderly people who have taken part in the smart home project must sign a consent form, which clearly states that they do not have any apprehensions about being monitored within the smart homes*". It is believed that all smart homes require the elderly participants of the homes to fill in an informed consent form.

A questionnaire based study conducted by Giuliani et al [55] discovered that elderly peoples' attitudes towards new assistive technologies within the home were positive. However, it is vital that the deeper needs of the elderly are understood as ignoring these needs would make it difficult for the elderly community to adopt and accept the devices.

2.2.2 Smart Home Technologies

The design of a smart home for the elderly needs take into account the emerging technologies that will respond to the elderly people's needs. In addition, it is also important to choose a suitable technology that is capable of managing a range of different equipment, infrastructures and protocols that co-exist together [56].

In order to make a successful smart home it is important to understand the types of technologies that exist and are being used in existing smart homes. Figure 3 below shows the three main types smart home technology.

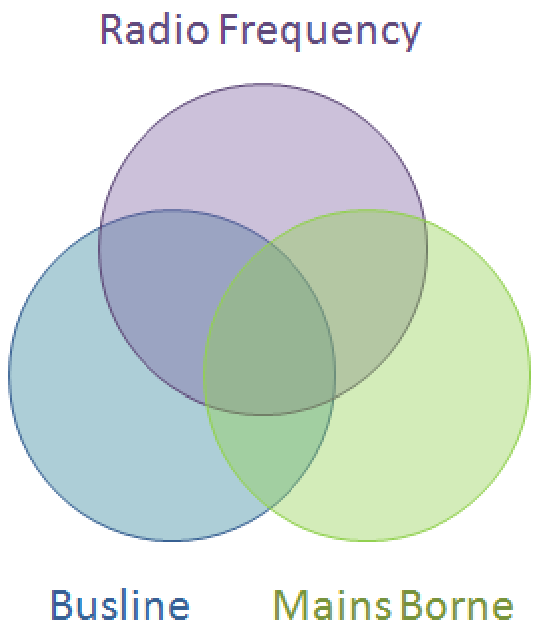


Figure 3 – Smart home technologies

2.2.2.1 Main Borne Systems

Mains borne systems use the X10 protocol or Powerline [57]. These systems consist of devices that are connected into a main power supply. This system is a very economical way of setting up a smart home as normal wiring round the house and is used to send signals to devices. Initially this type of system can be easy to setup, however over time the installation can become tedious when adding any new sensors to the system due to rewiring and repositioning of installed devices. One of the advantages of the system is the ability to fix a problem if the system fails, as the system can be repaired on the spot or through local devices around the house. However, one of the disadvantages of this system is caused by intrusion and power cuts, as the system starts malfunctioning. Due to this the system has to be reset which can be inconvenient for an elderly person in their smart home.

2.2.2.2 Busline Systems

Busline systems [58], which are also known as Konnex Association, require wiring like Main borne systems. The feature that differentiates the two is that Bus systems use a separate 12-volt cable (twisted pair) to transmit data and signals to devices. Often the twisted pair cable runs parallel to the traditional mains cable. The reason for using this type of cable is that it enables the devices to be independent of conventional mains borne supplies. In contrast to mains borne systems, Busline systems are more effective and reliable as they have the capability of being configured to prevent malfunctioning during intrusion or power cuts. This can be crucial when dealing with information that is closely related to safety and well being of an elderly person, as the data will reach its destination. One of the disadvantages of the Busline systems is that they require additional wiring around the home, which can make an installation a difficult and tedious process [59]. However, this situation of additional wiring does not occur regularly for new smart homes and so they can take advantage of the reliability offered by Busline systems.

2.2.2.3 Radio Frequency Infrared

A smart home technology that is increasing in popularity is Radio frequency and Pyroelectric Infrared [60] [61]. One reason is the ease of installation, as no wiring is required. Such types of system also have the option of being battery operated, which means the system will not be affected if there is a power cut or intrusion. This type of technology is also commonly used in security surveillance systems and alarm systems for cars. Radio frequency and infrared components can also be integrated with existing mains borne and busline systems with ease. However, one of the draw backs of this technology is that the frequencies of the sensors may sometimes conflict with each other and therefore it is important to know the frequency of the sensors before installing them together.

It has been argued that this type of system is unreliable, as they can be manipulated by an intruder with an IR code, which may give the intruder access to the home as well as being able to modify device settings [62].

2.2.3 Service Oriented Smart Home Architecture

A smart home is capable of meeting the needs of an elderly person by providing assistance or services, such as an application in a smart home might alert a hospital if the elderly person's glucose sugar level exceeds a certain threshold. Managing and creating applications which provide these services can be a complex and time-consuming process. The OSGi (Open Source Gateway Initiative) is framework which has been seen as an emergent solution for the creation and management of context aware applications for smart homes. The OSGi framework is able to routine between different service providers and devices over any network, which in this case are networks within smart homes [63]. OSGi is capable of managing the lifecycle of the software components (applications) within devices around the home. This enables software components to be installed and updated on the devices without interrupting the operation of the devices. This is made possible by the packaging format for the software components, which is known as a bundle. Bundles are simply applications that are packaged in a format (JAR file), which makes it compatible with ZIP files [64].

Zhang et al [65] made use of the OSGi framework by constructing the OSGi framework as three tier control system, as well as combining it with the Universal Plug and Play (UPnP) technology in order to achieve automation in the management and discovery of devices within the smart home. UPnP is an architecture that enables compatibility between different networking devices, equipment and software, which are part of UPnP forum that consist of 400+ vendors. One of the advantages of this architecture is that it is independent from any drivers. This architecture can work with both wired and wireless networks and is also compatible with any operating system [66].

Gu et al [67] proposed a context aware infrastructure that was based on OSGi. The infrastructure was capable of managing services securely and reliably. In addition it could also support the discovery and reasoning of different contexts in the smart home. This OSGi based infrastructure was supported by an ontology-based context model for semantic representation and a service-oriented context aware application (SOCAM) which supported rapid prototyping of context aware applications. One of the applications that were developed was a simple dining room application. The objective of this application was to play music and to adjust the lighting of the room whenever a person is having dinner in the dining room.

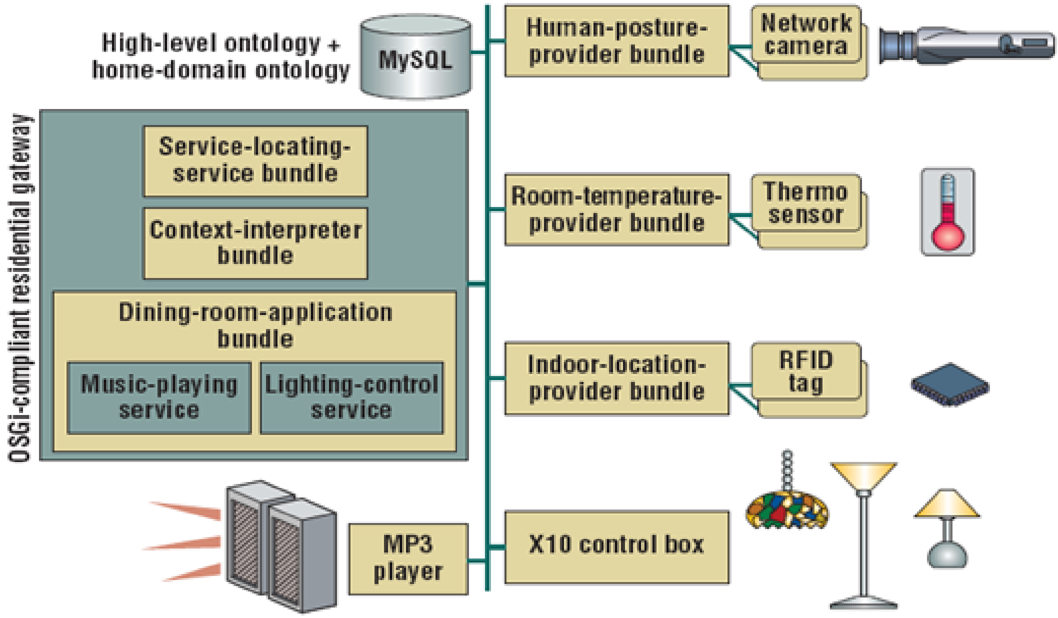


Figure 4 - Dining room application within a smart home

Figure 4 [67] illustrates the infrastructure of how different devices around the home (context providers) are managed by an OSGi-compliant residential gateway. Within the infrastructure the high level ontology and home domain ontology information are stored in a MySQL database. The information within these ontologies consists of profile-like information for each person in the house, such as preferences and birthday. The indoor location provider bundle is used to determine if there is anyone in the dining room, once this has been determined the context-interpreter bundle distinguishes which activity is taking place in the dining room based on the number of people that have been discovered by the indoor location provider bundle. Once the service-location and context-interpreter bundles have distinguished that there are x people in room y then the dining-room-application-bundle is ready to be executed.

In order to deal with dynamic situations in a smart environment, Wu et al [68] proposed a service oriented architecture (SOA) which was a peer-to-peer (P2P) model based on multiple OSGi platforms. Within this P2P model the service oriented mechanisms (such as OSGi bundles) were used to form the interaction between different service components. Mobile-agent (MA) technology was also part of this SOA. The role of the MA was to augment the interaction mechanisms for the service components.

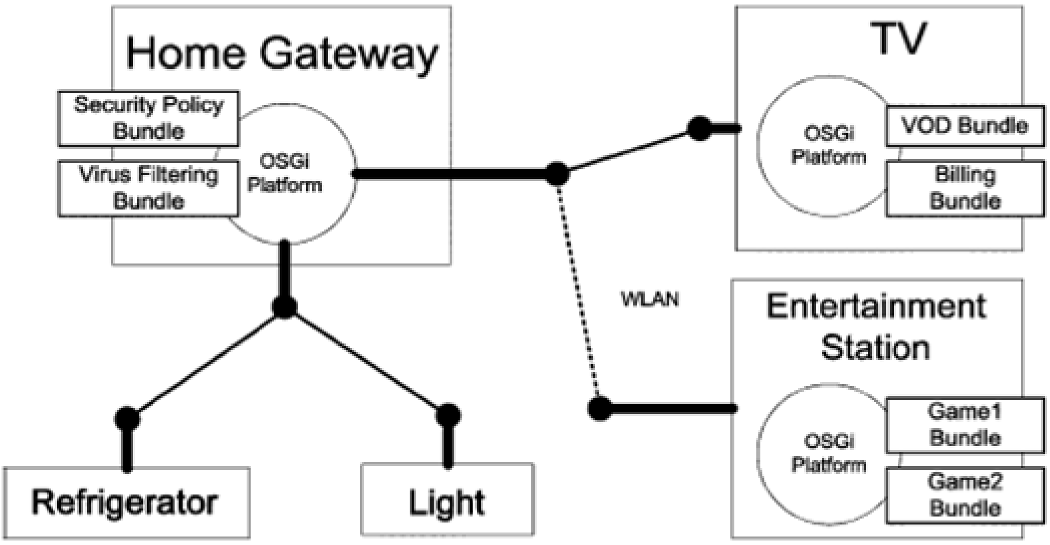


Figure 5 - Distributed P2P model

Figure 5 [68] shows the distributed P2P model that was adopted within a smart home environment. The motive behind using a distributed P2P model instead of client server architecture is that OSGi platform can be associated with or installed in devices, which distributes the device-dependent services to the relevant devices. The advantage of this is that it reduces load on the gateway as the service bundles are distributed to its respected devices. Another advantage is if the gateway crashes, the devices will not be affected. In order to form the interaction between the different components and services within the architecture mobile agent technology has been used, where a variety of agents with different motives carry out the coordination among the components. There are a range of agents such as Interface Agent, Device Agent, Service Agent, which when called upon have specific tasks they need to carry out.

Helal et al [69] developed an indoor precision tracking system for the elderly people in a smart home environment. This system used an OSGi-based framework to create a service interface to the location positioning system for the elderly person. The location positioning system was packaged into a bundle, which was then able to provide the location service to other applications and services in the OSGi framework within the smart home.

Cheng et al [70] proposed a gateway approach for smart homes that was based around the Session Initiation Protocol (SIP). SIP is an Internet protocol that is used to establish, modify and terminate a session of an application service. The aim of this approach was to be able to adapt the smart home to the user’s dynamic situation. This was achieved by allowing the person in the home (elderly person) to be able to use their preferred devices for communication, rather than the one the smart home provides. The proposed idea was to develop an adaptive SIP Context Aware Gateway (SCAG) for the ubiquitous SIP based services.

2.3 Smart Home Projects

This section will look at existing smart home projects for the elderly that have been conducted by local governing bodies and organisations in the UK.

The Gloucester Smart Home: Duke of Beaufort Court [71] was a project that was used to demonstrate that technology can be used to support people with dementia. In order to showcase this demonstration a three-bedroom house was used for the project. Some of the main features for this smart home project were applications like bath water level, sleep monitoring and automatic control of night lights. To provide these features the house was fitted with the following devices: electric bath taps for the assessing the water level in the bath, a radio-based object locator for locating the subject and a bed sensor for detecting when the person leaves the bed. Within this type of smart home project the emphasis was placed more on stand-alone devices that did not work in conjunction with each other. For example the bath taps are turned off automatically when the water is about to go over the tap level.

The Millennium Homes project was another pilot scheme project used to support elderly people. It was carried out by Brunel University. This project was first developed in 2000 and is now being used as a test bed for development work. The main emphasis of the project is based around the idea of a system which monitors combinations of events, such as lack of activity which then raises an alarm. In order to monitor activities the system uses devices like water leak sensors, cooker sensors, door sensors and lock activation sensors. However, this project has had a lot of limitations, because well-established but limited technology produced by external companies like Tunstall is being preferred rather than trying to develop innovative smart home concepts. By going for this type of approach the Artificial Intelligence (AI) techniques are being neglected and therefore the project is totally reliant on devices and programmes created by a third party. Another difficulty that this project encountered was the lack of volunteers to carry out tests and experiments.

The Joseph Rowntree Foundation is a social policy research and development charity. This charity has a keen interest in smart homes as they have a demonstration house which consists of many devices to support independent living. However, the smart homes within this project differ from the ones mentioned in the other projects. This is because the smart homes developed by the Joseph Rowntree Foundation are more concerned with devices such as vertical blind openers, sink lifters, cupboard lifters and internal/external door motors [72]. These devices are used to provide a smooth and easy daily operation for the elderly. However, there is no collaboration between the devices. This again brings us back to our argument for the need for some Artificial Intelligence support, to correlate sensor readings and reason about intentions, in order to help the elderly people to pursue an independent life. Artificial Intelligence is also important in the context of smart homes because AI has the potential to increase the range of services that smart homes can provide for their occupants.

2.4 Activity of Daily Life Recognition

There has been a significant amount of research carried out focused on efficient and reliable ADL identification. This section of the thesis will provide a detailed overview of the existing work being carried out in this area of work. In addition this section will also look at the significance of semi-supervised learning in ADL recognition.

Reliable ADL recognition relies on three main subcomponents [73]:

i. Feature Detection: is usually a sensing level that collects appropriate information about activities that are being executed. The gathering of information can be carried out with non intrusive ubiquitous sensors [74] such as RIFD [75] [76] technologies to collect activity information rather than using any visual equipment. Also the use of anonymous binary sensors such as: motion detectors, break-beam sensors, pressure mats, and contact switches can aid the process of tracking an individual around the home and complement the whole activity recognition process [77].

ii. Feature Selection: is when raw sensor data from the sensing level component is manipulated into features that can help differentiate between activities. These features can correspond to high level or low level information. The high level information could range from information related to specific objects detected to the number of people detected in a room at the time of an activity. Low level information could be as simple as frequency content or correlation coefficients between activities [73].

iii. Models for recognition: This component can be in the form a computational model (e.g. Hidden Markov Models, Bayesian Models), which makes use of the features from the feature-selection component for a more informed decision about which activity the person is engaged in.

In addition to these three sub components, semi-supervised classification is an interesting approach to ADL recognition. Semi-supervised classification is based around the idea of making use of labelled and unlabelled data for training and learning, where the volume of unlabelled data is greater than labelled data. In the context of ADL recognition, the employment of semi-supervised learning can be used for dealing with unlabelled data generated by the feature detection component. As the work in this thesis is not based on any semi-supervision techniques, this chapter will only give a brief overview of this type of classification.

2.4.1 Semi-Supervised Classification

One of the deficiencies of traditional classifiers is that they rely greatly on labelled data in order to train models. This is seen as a deficiency because labelled data can be sometimes difficult or even expensive to acquire. In contrast, unlabelled data can easily be gathered. However, the actual use of this type of data is seen as major challenge when training models. Semi-supervised classification challenges this problem by making use of more unlabelled data as oppose to labelled data, in order to build classifier models.

As semi-supervised learning provides the benefit of not having to put effort in to the labelling of data, it is therefore imperative that this effort is put towards building and designing models that are capable of carrying out semi-supervised learning for activity recognition [78]. Semi-supervised learning can be conducted in many forms of model, features, similarity functions and kernels [79].

2.4.1.1 Generative Models

The most common method of how semi-supervised learning is carried out is by using generative models. These types of models are used to randomly generate observed data given some hidden parameters. These models generally learn the joint probability model $P(X,Y)$, from which a prediction is made from the feature vector X and the label Y of the data [80]. An example is learning of similar XML data structures. Here conditional models predicting the number and type of nested elements can be constructed using known examples. Another example of the joint probability model [81] can be found in Gaussian mixture models where the assumption of the models is that $P(X \mid Y)$ is the identifiable mixture distribution within the model $P(X,Y) = P(Y)P(X \mid Y)$. Inoue et al [82] represented a joint probability model by incorporating unlabeled sequential data with a mixture of hidden Markov models, which gave positive results for the classification.

2.4.1.2 Discriminative Models

In contrast to generative models, discriminative models are used to model the dependency of an unobserved or response variable Y given an observed variable X, this is computed as a conditional probability distribution $P(Y \mid X)$.

Discriminative models directly estimate the posterior probabilities [83] as opposed to the generative models, which model prior probabilities for classification. An analogy that sums up the differences of these two types of models is as follows [84]:

The task is to determine what language person X is speaking. A generative approach would be to learn all the languages and then try and determine which language the spoken speech belongs to. On the other hand, the discriminative approach would be to learn the linguistic difference rather than learn all the languages.

Many researchers have put the point forward that discriminative models perform better as they achieve higher test accuracy than generative models [85] [86]. However by using uncomplicated Expectation Maximisation (EM) methods, the generative models are more reliable for handling missing data than the discriminative models and they tend to perform better when the size of data for training is small. Ng et al [87] proved this by using a naive Bayes approach as a simple generative classifier, which outperformed a logistic regression approach as a discriminative classifier where the amount of training data was relatively small.

Examples of discriminative models used for semi-supervised classification include:

- Boosting
- Conditional Random Field (CRF)
- Support Vector Machine
- Linear Discriminant Analysis (LDA)

2.4.2 Feature Detection

The first step to reliable ADL recognition is being able to gather information that makes it possible to recognise an activity that a person is conducting. A simple solution to this would be to make use of visual equipment and microphones, which record every movement of the person conducting the activity. This approach has a considerable amount of overhead that needs to be considered as providing a sufficient amount of labelled video footage to learn models for recognition can be very difficult. In addition the use of visual systems can be seen as intrusive, as it interferes with the monitored person's private life. Such processes for recognising activities usually works in a laboratory environment, however tend to fail in an actual home environment due to variable lighting, unexpected clutter and the different variety of activities that are carried out.

Therefore the emergence of systems using simple sensors to recognise activities by detecting changes in the state of objects and devices is seen as an alternative.

Simple sensors have the capability of providing important clues about which activity may currently being conducted by the person. Examples include pressure mat sensors that are used for tracking position and movement of a person [88] or switch sensors [62] within a bed or chairs in the home to discover if the person is sleeping or sitting on the chair. One such approach was by Ogawa et al [4] [89], who had installed and evaluated a monitoring system in an ordinary house. The monitoring system itself consisted of different sensors, such as infrared sensors, magnetic switches, carbon dioxide sensors to carry out monitoring of daily activities in the chosen home. This research conducted by Ogawa et al discovered that ADLs could be identified simply by the patterns that are generated by the sensors. However, this approach is very reliant on the length of the sensor readings and an activity cannot be identified until all the sensor readings have been retrieved.

Another approach to monitoring ADLs has been developed by Noury et al [90]. This is similar to the approach of Ogawa et al in that it makes use of a variety of sensor devices, but the emphasis of this research was on development of a smart fall sensor to detect when an elderly person falls. The research also developed approaches for interpreting the data from the sensors of the monitoring system. Firstly the system outputs the immediate position of the person that is determined by the sensor data. Secondly it presented a chronological display of the successive activities the person had carried out.

2.4.2.1 Dense Sensing

Currently a popular approach for feature detection is 'Dense Sensing' [91]. This is when numerous individual objects such as a toasters and kettles are tagged with wireless battery-free transponders that transmit information to a computer via an Radio Frequency Identification (RFID) reader [92] when the object is used or touched. The use of 'Dense Sensing' is seen as a less obtrusive approach for feature detection in comparison to existing techniques like accelerometers, visual equipment and sensor devices located around the home. In addition these types of sensors are able to detect features very well in a range of environmental conditions, being reusable, dealing with different reading ranges and ease of data transfer between tags and reader [93]. Philipose et al used dense sensing within a system called Probabilistic Activity Tool Kit (PROACT) [91], which was used for ADL recognition. PROACT was able to receive sensor data and used a probability engine to determine activities from observations generated by the sensor data. The results from the system had an efficiency rate of 88% when detecting an ADL.

Wu et al [94] showed that combining visual object recognition with information collected by RFID sensor generally performed better than dense sensing approach alone. This activity recognition approach was based on detecting object use that is not reliant on any human labelling of sensor data. However the use of video surveillance in a home environment for elderly people can be sensed as intruding on a person's privacy, even if the data is analysed automatically and discarded.

Capturing object usage data with an RFID reader can enable fine-grained activity recognition, as it not only tells us that the person is cooking, but can also determine what the person is cooking [95]. One of the reasons why dense sensing has become popular choice for feature extraction is because it offers the flexibility of being able to operate in a wireless manner and allows tags to be placed out of sight, which makes them well suited for ADL monitoring as they are not a distraction when an elderly person is carrying out an ADL. As well as that dense sensing has a lower overhead than other feature detection approaches for home activity recognition due to the ease of moving and removing transponders from active and inactive zones in the home. These transponders are also relatively cheap and easy to install, which makes them attractive to researchers and developers. This is a result of many retailers and manufacturers embedding RFID sensors in their products in order to increase efficiency in the supply chain management [96]. The integration of RFID in everyday products for the homes suits the 'dense sensing' approach, as it is cheap and feasible to set up in a smart or even standard home environment.

On the other hand, the approach of dense sensing does have its share of flaws. For example, as the approach suggests all objects associated with an activity are tagged with transponders and sensors, this may lead to a situation where multiple activities may share the same sensor object. Hence sensor data must be interpreted with this in mind.

Other flaws are described by Logan et al [97]:

- Some transponders/sensors were actually bigger than the object that needs to be tagged.
- Some activities are difficult to recognise as they do not involve interaction with objects, e.g. sleeping.
- Many activities have metal objects (e.g. dishwasher), therefore these objects cannot be tagged as the RFID transponders do not work when stuck onto metal objects [76].

A way to overcome this situation is to make further enhancements to the feature selection level and to the models for recognition, so that they can accommodate multiple activities sharing the same sensor object. This can be in the form of "boosting" to retrieve more features about the object [98] by using a set of weak classifiers to create a single strong classifier, rather than just relying on the concept of trying to carry out classification with simple object use.

2.4.2.2 Wearable Sensors

Wearing different types of sensors around your body is another technique for feature detection [99] [100] [101]. These types of sensors are known as wearable sensors, which can range from accelerometers to audio microphones that provide data about body motion and the surroundings where the data has been collected. Wearable sensors can also be in the form intelligent gadgets, which can be reconfigurable, and scalable smart objects that can be embedded into the personal everyday goods that are used by the person to be monitored. The embedded smart object generates data that is used to log and recognise the person activities. Jeong et al [102] uses these smart objects to obtain two levels of data. The low level is concerned with body movement and hand movement, and here the wearable device is attached around the waist and one is attached around the wrist. The high level is concerned with associated predefined rules to interpret the low level information. For example a rule may be: if the body movement is fast and the wrist is being used fast then it is likely that the person being monitored is running.

Previous work [103] [104] has also shown that a variety of activities [105] like climbing stairs and working in a workshop have been determined from similar techniques. For instance, Bao et al [106] carried out feature detection based on data collected from five biaxial accelerometers which were worn by 20 subjects while they conducted activities. The motivation of this work was to conduct activity detection in a naturalistic environment as opposed to in a laboratory environment. The use of accelerometers provided data which were then labelled by the subjects themselves, this labelling was carried out without any researcher supervision. This data is then used by supervised learning classifiers for training purposes. One of the benefits of using labelled data for training is that a collection of training data can be generated by the learning classifiers, which can then allow different users to train algorithms for recognition of the activities they conduct themselves. However labelling data can be a long and tedious.

An alternative approach was by Wang et al [107], who used wearable personal sensors to detect fine-grained arm actions like 'drink with a glass', 'chop with a knife', these were then combined with object-use data to achieve accurate activity recognition. A distinct feature of this approach was the low level of labelling required in comparison to existing approaches. This was because the accurate recognition was based on a joint probability model of object-use, physical actions and activities. It is based on a combination of generative and discriminative models and referred to as 'common sense based joint training'. Similar work was conducted by Petney et al [108] [109] on the State Recognition using Common Sense (SRCS) system. This system works in conjunction with dense sensing as it provides a common sense interpretation of the world by forming a bond between the dense sensors to a model that represents the Open Mind Indoor Common Sense (OMICS) database. The OMICS [110] is a database, which provides basic facts that can be reasoned about. This information is produced by a small dedicated team of humans who add facts, which can then be accessed by users over the Internet for using and adding more facts relevant to the system. The SRCS converts information from this database and data mined with the KnowItAll system [111] (used to bootstrap knowledge) into a large dynamic graphical probabilistic model, which is used to interpret real-world activity data [112]. Conventional approximation techniques have also been applied to the SRCS system, which was used to improve the performance of the SRCS by enhancing the accuracy of SRCS's prediction state by maximizing the likelihood by using small amount of labelled data [113]. The mining aspect of the SRCS approach has similarities with mined ontology models from the Internet, which will be discussed further in the chapter.

The use of wearable sensors around the body can be seen as intrusive and sometimes get in the way of activities, therefore the need for a single sensing device seems like a suitable approach for feature detection. Hence the use of an RFID reader as single sensing device is a solution as it can focus just on the objects that the person interacts with rather than capturing irrelevant data [114]. The RFID reader can come in the form of a ring-like reader, which reads information from the transponders located around the home. As well as that an RFID reader can be integrated into everyday devices that people carry with them, i.e. mobile phone [73]. The work conducted by Lester et al [115] was in relation to providing users with integrated sensing devices within everyday tools to be used in conjunction with minimal wearable sensors.

2.4.3 Feature Selection

In the context of ADL recognition, the aim of feature selection is to identify salient features from the captured data that can be used to make activity recognition possible. For example, a movement in a particular direction can be extracted from the raw sensor data from analysis of the accelerometer data, which can then used to differentiate among the different activities. One of the benefits of feature selection is to reduce the computational overhead on resource devices. Feature selection also helps in acquiring a better understanding of the data by determining what the important features for recognition are and how they are related with each other. The problem of automatically being able to discover which features are relevant when carrying out selection is more or less unresolved. Recent work has been done on this area where approaches based on boosting have been used to select the most useful features. One such example of this which was mentioned in the earlier section of this thesis was by Lester et al [98]. This approach to feature selection was used to select the correct features when using a classification system, where a sensor board captures raw sensor data from which the features of the sensor data are computed. A sensor board is a shoulder mounted device, which is used to collect 18,000 samples of data per second. In order to make use of this data, a total of 651 features are computed in order to bring out detail in the data collected. The top fifty features per class are then selected from the feature vector and are used as inputs for a group of decision stumps classifier. A decision stump classifier is known as a weak classifier that is based on a decision tree with a depth of one. Each of these classifiers outputs a sequence of decision margins at a particular time t, which are then converted to probabilities by fitting them to a sigmoid function. These probabilities are then passed to ten Hidden Markov Model (HMM) classifiers which output likelihood of each class; hence the class with the highest likelihood is the classified activity.

Wang et al [107] paired object usage information with features from a Mobile Sensing Platform (MSP) that is used for detecting arm movement and ambient conditions. It consists of the following sensors:

- A six-degree-of-freedom accelerometer
- Microphones sampling 8-bit audio at 16kHz
- IR/visible light sensor
- Barometer
- Temperature Sensor
- Compass

651 features are extracted from the MSP data, which includes mean, variance, energy, efficiency, spectral entropy, FFT coefficients, cepstral coefficients and band-pass filter coefficients which results in a stream of 651-dimensional feature vectors. So given a stream of sensor readings $S_N = s_1, \ldots, s_N$ where each s_i becomes a pair consisting of an object name and the relevant features from the MSP vector, which are then used for recognition. This feature selection was one of the initial steps allowing inference of the current action being performed and the object on which the action is being performed. Lösch et al [78] states that a minimum of four features are sufficient enough to carry out robust activity recognition. Also combining standard features with any type of statistical features generated from

accelerometer data can also achieve the performance of the activity recognition conducted [116]. However, the number of features that are needed for activity recognition can vary, as some activities may require more features than other activities due to the nature of the activity being conducted.

2.4.4 Models for Recognition

Many models have been constructed for recognising activities conducted within the home. Typical computational models are Hidden Markov Models (HMM) and Bayesian Models.

Bayesian models (in the form of Dynamic Bayesian Networks (DBN)) have been used to capture relationships between state variables of interest [108], for example, in a common sense based joint training approach [107], the DBN is able to represent the state of a system in time slices. Within the time slices each node is used to represent a random variable that gathers the state of that particular time slice, which on this instance can be the activity and action that are currently being executed, as well as the object and features involved. Kanai et al [117] applied Bayesian Networks to model observations based on the location of the person, the time of when the sensor data was detected and the status of the person being monitored. For example if a person is hungry this may be recognised by the person's behaviour, as they keep opening and closing the fridge door. Once these observations have been modelled, the confidence levels of the person's predicted situation is calculated, and the state with the highest confidence level is considered to be the current situation of the person being monitored. An audio notification system based on sound cues is then used to assess the current situation of the person being monitored, based on the state with the highest confidence level.

In relation to object usage, whenever monitored activities are conducted they generate a stream of sensor data related to object use. This stream of data has transitions between the different objects where the transitions between these objects can help determine the activity. This can lead to many possibilities as each transition could have many alternatives. HMM is a simple tool that enables transition probabilities between activities to be modelled, as well as emission probabilities that predict the sensor events according to the activities that could be currently in progress. Many approaches for carrying activity recognition make use of HMM in one way or another, whether it is simply determining the likely sequence of an activity given the objects [118] [95] or being used as temporal smoother for specific classifiers [107], and classifying likelihoods [98]. Training HMMs separately for activity recognition tends to perform poorly whenever more than one activity shares an object or occurs at the same time. However, connecting states from different independent HMMs can improve accuracy as it is possible to train the HMM by learning the transitions between different activities given the object [95].

Wang et al [107] have made use of HMM within their common sense based joint training approach, by learning action models in order to reduce the labelling overhead.

The problem of being able to deal with multiple activities by using HMMs has been addressed in this thesis as one of the low level modelling approaches.

In relation to the well-being of an elderly conducting everyday activities, Wilson et al [119] formalised an approach to rate how well elderly people perform day-to-day activities, to provide caregivers with information that consists of rating summaries that can be used to assess the well-being of the elderly. This approach represents activities (e.g. making soup) as a set of steps (e.g. preheat water, open can, mix ingredients, serve, and clean up), while the steps consist of actions (e.g. use can opener for step open can). When an everyday routine is conducted by a person the system collects traces of that particular routine. A trace is a set of actions that comprise of an execution of an activity. These traces are used to learn dynamic models like HMM and Hidden Semi-Markov Model (HSMM), where the hidden states in this instance are activity steps. As these first order models are unable to capture higher order correlations, a human rater adds a set of constraints on the sequence of the hidden states or any observations that specify any high order correlations. In terms of learning the rating thresholds of the activities, a human rater then rates each trace with a rating of either pass or fail, where a pass indicates that a sequence of actions closely matches a particular trace of an activity. This is then used to calculate the likelihood threshold L to separate the passes from the fails. This information is then used to generate a rating and a justification, for instance given a collection of traces $Y_1, \ldots, Y_n$ each with a rating of r_i, which is either pass or fail. If the automated rater rates the trace as fail, then the automated rater then tries to produce a repaired trace $Y_1', \ldots, Y_n'$ which has smallest possible distance between Y and Y'.

In contrast to generative models an approach by Landwehr et al [120] has been developed which primarily focuses on tagging rather than classification for trying to identify which activity is being performed. This approach is based on relational transformation-based tagging which is applied to data streams generated from sensors. This provides an expressive relational representation of the data stream, which provides a rich representation for the sequence elements. This is done by tagging the sequence of interactions with activities. Once tagging has been done then a relational transformation rule approach is applied, which helps to identify the activity. For example, in the context of natural language processing the word "move" is initially tagged a as *verb*, however if the preceding word after "move" is "article" then it would be retagged as *noun*. This approach of combining tagging with transformation-based learning is based on a rule-based learning approach, where at each iteration stacks a rule on top of each other in order to improve the performance of activity recognition.

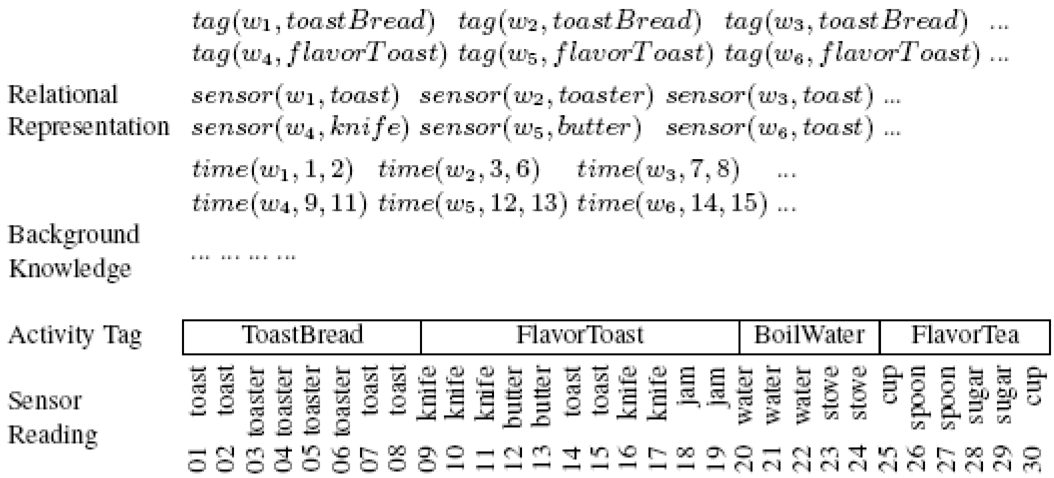


Figure 6 - Relational representation of the stream of object data

Figure 6 shows an example from this approach, which is relational representation of an ADL scenario "Make Breakfast" [120]. In this example the activity "Make Breakfast" is being conducted, and at the same time a stream of object usage data is being collected from an RFID reader. This data is then represented in structure that merges the identical sensor readings into one sequence element, labelled w_i. This relational representation allows valuable information for each observation to be encoded as predicates. For example, the relational representation for the activity tag "ToastBread" with the sensor "toast" is encoded as following way: *tag(w_1,toastBread) sensor(w_1,toast).*

Predicates can also be used to encode the starting point and duration of an observation, e.g. *time(w_1,1,2)*. In addition, further background and prior knowledge about the sensor event can also be encoded in a predicate. The use of context information makes it possible to recognise the correct activity, for example if we use the object *spoon* then this indicates that the person is either applying sugar to tea or eating cereal. However if we take into consideration the context knowledge, then we would know the following from the following observations:

- If a spoon is used and is followed closely by sugar bowl then this indicates that person is flavouring tea by applying sugar.
- If a spoon is used after the milk bottle and cereal box have been used then this indicates that the person is eating cereal.

2.4.4.1 Model Incompleteness Problem

Activity recognition based on object usage data generally uses activity models, which are models that map activities to objects that are used to carry out the activity. However, when constructing these activity models, missing information is seen as a common problem, as information from the sensing level or during feature extraction can be unexpected or even misinterpreted. Another reason for this is that it can be difficult sometimes to recognise all the relevant objects that are required to carry out the activities, this can be because of the following:

- Lack of labelled data.
- The volume of noise while collecting object data could lead to missing information.
- Objects that are not modelled are sensed. For example, if a person makes tea in a cup everyday and then decides one day to use mug, which is not

encoded in the system. Another example is if a person makes tea with sugar and milk and then decides one day to have no milk or sugar then this can also be seen as the model being incomplete.

A stream of natural languages terms can be used to represent the sequence of objects used to conduct an activity. For example, a stream of object data 100110111100012131121093232 can be mapped into: Kettle-Sugar-Bowl-Milk Carton. This representation allows the mapping from these terms (based on the object data) to activity names (e.g. make tea), which makes it possible to mine generic activity models from the web in order to segment the stream of object data into instances of activities. For example, given a stream of object data that consists of "kettle" and "tea bag bowl", we would segment out many instances of the activity making tea. This is then followed by using the labelled instances to learn custom models of the activity "make tea" from the collected data. During this learning process, it is also possible to learn the variations and the use of different object to perform the activity, e.g. having milk, but no sugar. Perkowitz et al [121] was able to carry out classification of activities with this approach to a certain extent, as it worked well with hand-segmented data. However, since these models are generic they are reliant on websites that follow a particular format so that information can be mined from them. This results in certain activities being unrecognised as the information from the web does not have any relation with the actual objects used to conduct the activity. Wyatt el [122] also developed a technique for mining from the web, and these can be applied to segments of unrecognised stream data, as well as being able to label streams of object-usage data. Given a set of activities A (e.g. Make Tea), this technique mines a set of objects O (e.g. Mug) from the web that are used for each activity a in A. In addition, the associated usage probabilities are also determined (1). Figure 7 shows the mining process employed for this approach.

$$\mathrm{P}(o \in O \mid a \in A) \tag{1}$$

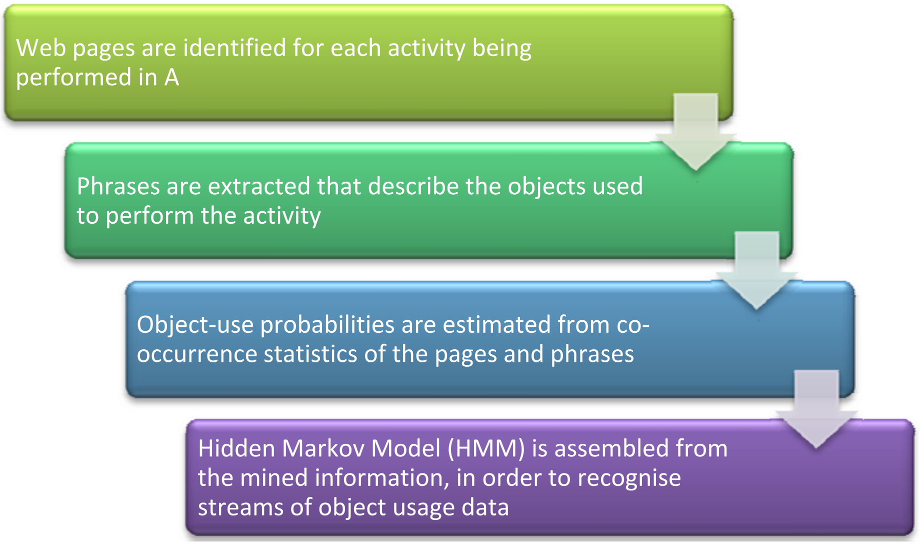


Figure 7 - Four steps for mining activity models

Figure 7 shows that the four steps of mining activity models lead to a HMM being assembled from the mined information that is capable of recognising activities given the different segments in the object usage data. For this type of HMM, the hidden states are the activities, while the observations are the objects used.

The approaches mentioned so far for mining models are mainly focused on being able to deal with unlabeled data, e.g. unsupervised. The problem of missing information or if a model omits an object while incorporating a similar one has been explored by Tapia et al [123], who have developed a unsupervised approach that uses information mined from an ontology of reference system for English languages, called WordNet. Ontologies have been utilised to construct reliable activity models that are able to match an unknown sensor readings with a word in an ontology which is related to the sensor event. For example object 'Mug' (an unknown sensor event) could be substituted for a 'Cup' object, as the model *make tea* recognises 'Cup', because 'Mug' has not been modelled in the *make tea* model. In addition a statistical smoothing technique called shrinkage has been applied to this approach. This technique has been used by many researchers for situations where it is not possible to reliably compute parameter values of a given model from training data alone, hence shrinkage is used to improve a given model's estimated parameter values. In the context of ontologies, shrinkage has been used to improve the probability estimates of leaf nodes that are generated within the ontology. Each leaf node within the ontology represents $P[o_i \mid a_j]$, which is the probability estimate of the observation o_i while activity a_j is being conducted. The maximum likelihood (ML) probability estimates at each node is computed using (2):

$$P(o_i \mid a_j) = \frac{N(o_i, a)}{\sum_{s=1}^{|o|} N(o_s, a)} \tag{2}$$

$N(o_i, a)$ represents the number of times object o_i occurs in activity a, while $|o|$ represents the cardinality of the set of all objects.

As well as activity recognition, ontologies have also been deployed for collaborative healthcare experiments to support and enhance the living of elderly people. One such example is where Wang et al [124] deployed ontologies to manage a collaborative healthcare environment, where the ontology models were based on information concerning elderly people are used to present important context aware information, which can be crucial for emergency treatment.

As described above, there are several existing approaches used to recognise ADLs, and the state of the art is such that in many cases it is possible to determine ADLs from the use of objects. The approach described in Chapter three is also able to recognise ADLs via object use data.

The approaches that have been mentioned may be able to carry out classification and learning for the activities, however they cannot handle interactions, suspension and interweaving well and are unable to predict what ADL may follow from a previous ADL, i.e. do not reason about sequences of ADLs. Such reasoning is a key element of analysing an elderly person's intentions and is one of the novel aspects of the proposed research.

2.4.5 Asbru Related Work

Asbru is the plan representation language that has been used for the modelling of ADLs in this thesis. Planning systems are now mainstream AI and such systems are used to schedule activities in a wide range of applications. Asbru is not an automatic planning system, even though it represents plans. Asbru is a framework for representing protocols. Asbru was developed as part of the Asgaard project, and is used to represent and monitor clinical guidelines. The Asbru framework has been used within the clinical research area for utilizing guidelines for newborn infants [125]. For example, in relation to premature babies, ventilation is sometimes required as they often suffer from respiratory distress syndrome. Artificial ventilation is needed to support the breathing of a patient until patient's respiratory efforts are enough for them to live. Asbru is used to help manage the application of clinical guidelines. The reason why Asbru has been applied to the monitoring in this thesis is because Asbru can capture the requirements of a dynamically changing environment and the specification (which took several years to refine) is openly available. Asbru provides a dynamic knowledge representation language with a set of temporal relations between plans and sub-plans. For example, if the breathing of a baby is getting slightly better then the Asbru execution engine (the protocol interpreter) may recommend to reduce the amount of ventilation. However the plan for the ventilation can still be idle, just in case the breathing levels were to change again. As well as monitoring ventilation for babies, Asbru has also been used to represent other medical scenarios such as the monitoring of jaundice.

Asbru has been specifically designed for monitoring people and as such it has been chosen for modeling ADLs for the work in thesis rather than a generic planning language.

Chapter 3

Method

This chapter looks at techniques and algorithms that have been developed to identify ADLs and also analyse the intentions of the elderly. A set of algorithms utilising low level and high level models have been investigated in order to determine which ADL is active and these are described.

3.1 Levels of Modelling

ADLs can correspond to simple tasks, such as "*switch on kettle*", or more complex activities such as "*make breakfast*". To encompass the range in this thesis ADLs are modelled as plans. Plans can contain sub-plans. A plan that cannot be decomposed any further is called a *task*. When performed, a task generates sensor events based on the objects used to perform the activity, and so task recognition is based on analysing sensor data. A 'dense sensing' [91] approach has been used to gather this data. ADL recognition is based on recognising constituent tasks.

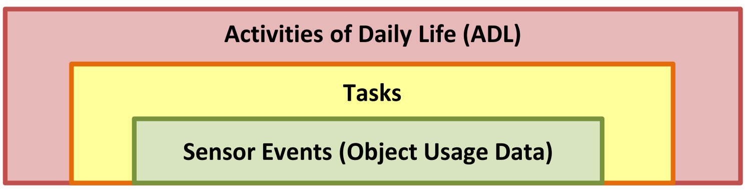


Figure 8 - Hierarchy of concepts

Figure 8 gives a schematic representation of the concepts used in the recognition process. Starting from the bottom, a (potentially variable) number of sensor readings correspond to a particular task, which could be currently active. A number of tasks determine an ADL that is active or set of ADLs that could be active. An ADL can be nested in another ADL.

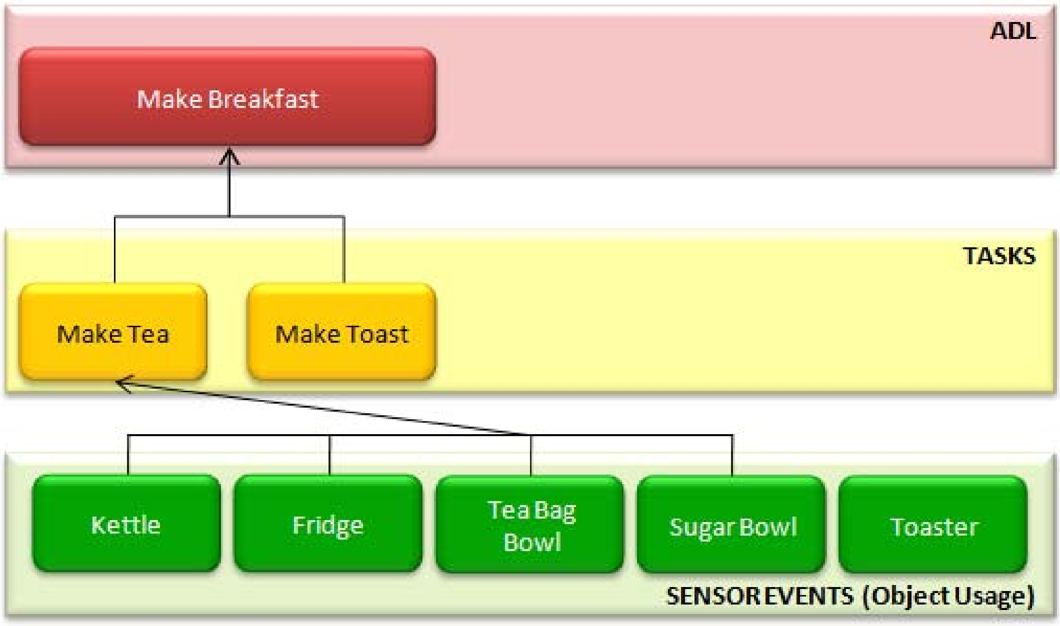


Figure 9 - Example of Hierarchal ADL (HADL)

Figure 9 illustrates a structure of a Hierarchal ADL (HADL), which shows that the ADL *"Make Breakfast"* contains a simple sequence of tasks, *Make Tea*, *Make Toast*. The sequences of the sensor events at the lowest level (Kettle Sensor, Fridge Sensor, Tea Bag Bowl Sensor, and Sugar Bowl Sensor) correspond to sensors triggered during the task *"Make Tea"*. The models developed as part of this work are shown in Figure 10. At the lower tier, three different approaches to task recognition have been developed. One is based on Multiple Behavioural Hidden Markov Models (MBHMM), which essentially accommodates different possible task orderings with different models, while the second technique is based on an approach inspired from a text segmentation technique, called Task Associated Sensor Events (TASE) segmentation. The third approach is an extension of the TASE approach, which generates a set of different task sequences from a stream of object usage data that is based on the conjunction of the disjunction of task possibilities for each sensor event. This approach is called Generating Alternative Task Sequences (GATS).

For the higher tier, the number of levels above the task identification level depends on the complexity of the task. ADLs may occur in parallel with other ADLs and have other temporal constraints. Also, not all sub-activities need to be executed. The knowledge representation language used is Asbru [126], which is a task-specific and intention-oriented plan representation language initially designed to model clinical guidelines. The 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 kitchen*. Based on the plans an ADL recogniser has been developed, which uses the tasks from the task recognition component to determine the activity that is being conducted and so determine the current and future intentions of the elderly person. Future intentions are established by predicting what ADL the subject might conduct next. In order to generalise the activity and intention recognition capability outside the framework of the core ADLs constructed to support recognition, decision trees are constructed using a well known induction algorithm during a training period. Once the tree has been developed the trees are used as a support tool for determining if a correct task or ADL has been recognised at the current iteration of the recognition process.

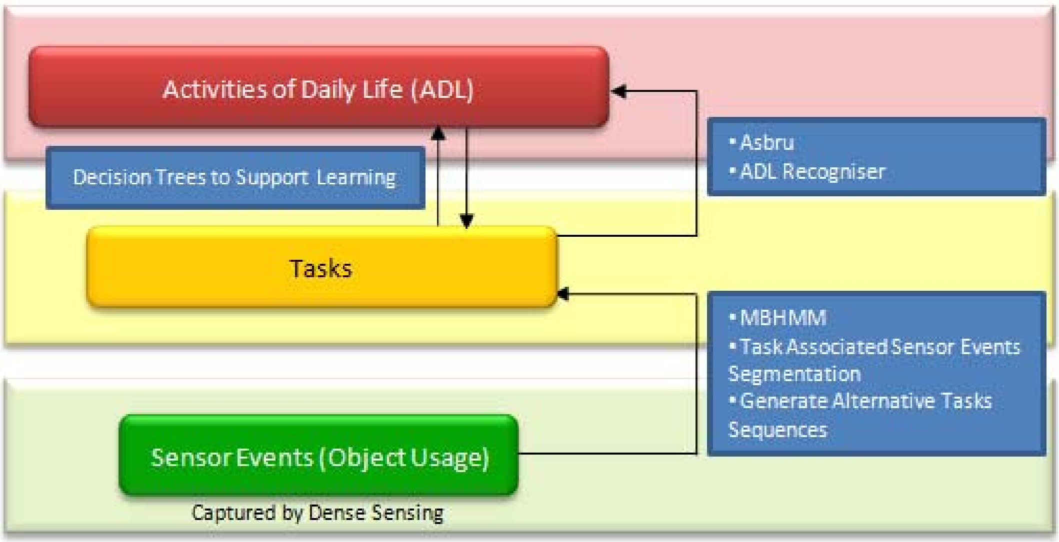


Figure 10 - Approaches developed and applied for the recognition of ADLs

3.2 Task and ADL Recognition Experiments

In order to recognise or predict an ADL it is vital that the tasks that have been determined in the low level modelling are accurate as possible, as the information that is generated in low level modelling is the source that is used to compute the discrepancies in the high level plan recognition. Sometimes a set of sensor events do not uniquely identify a task, and in such cases the contextual information encoded on the plans can be used to uniquely identify the ADL(s) being performed.

Pilot experiments have been conducted to establish the performance of the algorithms that have been developed for the low and high level modelling. This section will outline how the data is collected for the experiments. The objectives of the experiments conducted vary, therefore the structure and results of each experiment are described in the sections where the algorithms are explained.

Figure 11 - Experiment locations

All the initial experiments were conducted with non-intrusive RFID transponders installed around a kitchen (Figure 11) and on its cupboards and objects, such as kettle, dishwasher, utensils, and toaster. Some of the experiments were extended further, as other rooms such as living room, bedroom and bathroom (Figure 11) were used in order to carry out activity recognition with activities that originate from objects which are not just based in the kitchen (e.g. tooth paste).

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 all 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 to brushing teeth. 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 of the high and low level algorithms.

3.3 ADL Recognition

This section describes the method developed for the high level modelling of ADLs. The objective of the high level modelling is to determine which ADL is being conducted based on identified tasks. In addition, the algorithm used to model the high level section of the ADL will also predict future intentions of the elderly person by predicting the next ADL that will follow the previous ADL.

In contrast to the approach used in the low level modelling, the high level modelling had to develop an approach that gave an overview of all the possible ADLs that could occur within a given time. In addition, the approach had to be 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.

The elements of an ADL are made of behavioural patterns and the ADL itself can be classified as a type of behaviour. A potentially good way of representing and modelling high level behaviour could have been by using workflows, which are commonly modelled using an augmented Petri Net [127]. Within a workflow system "*a process represents a set of tasks that need to occur in a prescribed sequence to achieve an outcome*" [128]. Workflows are now used extensively in modelling business processes. The goals of a person typically require particular constituent activities (tasks or sub-activities) to be ordered sequentially or in parallel. A majority of ADLs that the elderly people carry out are process oriented and so workflow systems are potentially a good modelling tool.

Currently there is a lot of research into dynamic workflow processes. One approach to enable ad-hoc and evolutionary changes is to be found in [129]. Ad-hoc changes are usually caused by rare events occurring, while evolutionary changes often arise in order to make the workflow more efficient. An example of the latter could be removing unused nodes in the Petri net.

However, workflows are too prescriptive in their ordering. If workflows are applied in dynamically changing environments they require a large number of permutations to be explicitly enumerated. Workflows scale badly to cases where there are many possibilities, which are often the case for goals performed by people [130]. More flexibility is required on the modelling of tasks.

3.3.1 Modelling with Asbru

The Asbru language is a process representation language, which has similarities to workflow modelling, but has been designed to provide more flexibility than workflows. Its roots are in the modelling of medical protocols, which can be complex and monitoring the application of such protocols to patients. Asbru was selected as a suitable representation language as it allows a considerable flexibility in how it can represent temporal events, namely their duration and sequence.

Asbru is a task-specific and intention-oriented plan representation language for defining 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 [126]. Asbru has many features which allow each skeletal plan to be flexible and to work with multiple skeletal plans.

In relation to the high level modelling, Asbru is being used as a representation language to model ADLs. The skeletal plans in Asbru are used here 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*. Like workflows, in Asbru when a goal is reached it represents the plan as an executed plan. In the case of the high level modelling of ADL, when all the *phases* and *conditions* of an ADL have been met then the ADL can be classified as being executed. As well as that an ADL will only be classified as executed once all its mandatory sub-activities have been executed. For example, if a *Prepare Breakfast* ADL has a mandatory sub-activity called *Make Tea*, this sub-activity needs to be executed in order for *Prepare Breakfast* ADL to be classified as executed.

3.3.1.1 Phases and Conditions in ADL Execution

When modelling with Asbru, each ADL can have 7 possible phases in its execution. The plan phase model is referred as the ADL phase model throughout this thesis. The ADL phase model shows a possible sequence of ADL phases. For an *activated* ADL, the *suspended* phase, *completed* phase and *aborted* phase are optional. As shown in Figure 12, the first three phases (*considered, possible,* and *ready*) constitute the *preselection phase*, while the latter four (*activated, suspended, aborted,* and *completed*) form the *execution phase*.

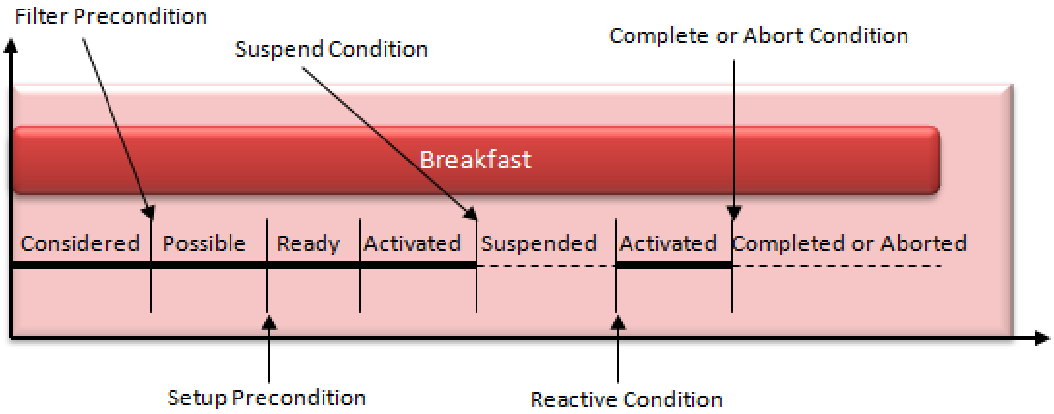


Figure 12 - ADL phase model representation in Asbru

<u>Preselection Phases</u>

1. Considered – This is the first phase of the ADL and considers any filters which have to be fulfilled before moving on. If the filter preconditions are fulfilled then the ADL moves onto the next phase, which is the possible

phase. If the filter conditions are not fulfilled then the ADL does not execute any further, meaning it is not considered for execution. For example: The ADL "*Breakfast*" would only be considered if the person has been awake for '10' minutes or more. Figure 13 shows an XML representation of this filter precondition.

```xml
<filter-precondition>
    <parameter-proposition parameter-name="Person Awake">
        <is-known-parameter/>
        <context>
            <any/>
        </context>
        <time-annotation>
            <time-range>
                <starting-shift>
                    <latest>
                        <numerical-constant unit="min" value="-10"/>
                    </latest>
                    </starting-shift>
                    <finishing-shift>
                        <earliest>
                            <numerical-constant unit="s" value="0"/>
                        </earliest>
                        </finishing-shift>
                </time-range>
                <reference-point>
                    <now/>
                    </reference-point>
            </time-annotation>
        </parameter-proposition>
</filter-precondition>
```

Figure 13 - Filter precondition in XML

2. Possible – This pre-selection phase of the ADL has to see whether all the setup preconditions of the main ADL have been fulfilled. Setup preconditions are imposed when the filter precondition cannot be achieved. These setup preconditions need to be fulfilled in order for the ADL to be in the ready phase. The difference between filter preconditions and setup preconditions is that filter preconditions consider whether it is possible for an ADL to be carried out, while setup preconditions must hold before an ADL can be executed. Setup conditions can also have a dependency on time. For example, if a setup condition is not fulfilled during a particular time frame that has been defined by an optional waiting period then the ADL is not executed and is rejected. However, if there is no time frame assigned for the ADL, then the ADL stays in the possible phase until all the preconditions have been fulfilled.

3. Ready - Once the setup conditions have been fulfilled then the ADL is ready to be moved onto the activation phase. Depending on the type of ADL or sub-activity within the ADL, the ADL may not move on to the activation stage straight away. This is because if an ADL or sub activity has to be executed in a parallel order then the ADL that is in the ready phase must wait for the ADL that is in the activated phase to be completed, aborted or suspended.

Execution Phase

4. Activated - Before an ADL is activated it takes into consideration the activate condition. This condition is a token that determines if an ADL needs to be started manually or automatically. This is specified by using the following attributes `overridable` and `confirmation`. These attributes are generally used for plans that have been modelled for clinical plans, therefore they are not used when carrying out modelling for ADLs. However once an ADL is in the activated phase it will then either move on to any one of these three phases: suspended, aborted or completed. An example of an ADL being activated is if a task has occurred that is part of an ADL such as *Make Tea*.

5. Suspended – An ADL in an activated phase will only move on to the suspended phase if the conditions for suspension have been fulfilled. The only way an ADL can move back out of the suspension phase into the activated phase again is if the reactive conditions have been fulfilled. These reactive conditions are used to determine when a suspended ADL needs to be reactivated.

6. Aborted – Likewise, an ADL in an activated phase will only move on to the aborted phase if the conditions for aborting the ADL have been fulfilled.

7. Completed – When an ADL is in the completed phase, then this means that all the sub-activities (consist of tasks from low level modelling) and actions (tasks in the low level modelling) within it have been completed, therefore this allows the next ADL in the ready phase to be activated.

Conditions "need to hold in order for a plan to be started, suspended, reactivated, aborted, or completed" [6]. Some ADLs modeled with Asbru have preconditions that can only be started if a certain action (task) that satisfies the ADL's precondition *has been* executed. For example, a precondition for an ADL "*washing face*" may be to apply soap, which then lets the ADL begin. Another important feature of the condition element is that it allows ADLs to suspend and restart if another ADL is going to become active. For example (Figure 14) if an elderly person is *cooking dinner* (ADL A) and the *phone rings* (ADL B) then the elderly person picks the phone up, then with the aid of the conditions Asbru can suspend ADL A and start ADL B. Once an elderly person is off the phone then ADL A will be reactivated and ADL B will be suspended as more phone calls are likely to come during the course of the day.

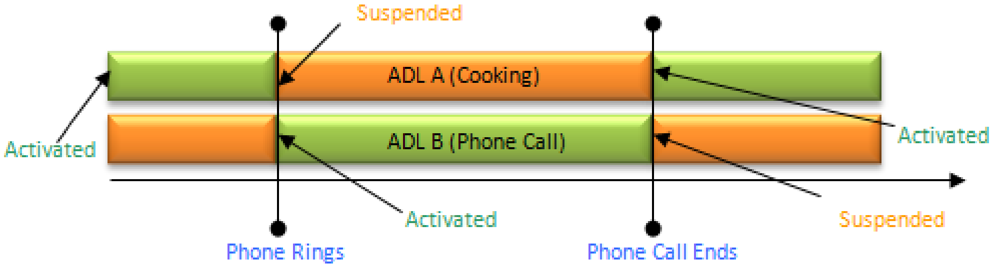


Figure 14 - Using conditions to suspend tasks

When a suspension occurs it is important that certain conditions are satisfied (like the hob is turned down) or certain monitors to check that certain conditions (such as the food on the hob is not boiling over) are setup.

The example above demonstrates the suspension and activation of two ADLs, and shows how an ADL resumes after being interrupted. However, this does not mean that another ADL could not be before the initial ADL resumes. This is important as there might be situations where the elderly person with Alzheimer's disease conducts an initial activity and after an interruption forgets to resume the activity and starts executing another ADL. In this situation the ADL that has been suspended may have a condition triggering abortion if the ADL has not been reactivated within a few hours. This is something that is of importance when constructing ADL plans for Alzheimer's patients, as they have a tendency to do something else or wander off when an interruption occurs.

The following is an example of how a simple ADL plan traverses the different plan phases. Table 4 shows the parameters that are used to instigate the conditions that are enforced on the ADL plan *Make Breakfast*, while Table 5 shows the details for each of these conditions.

Parameter	Initial Value	Initial Scenario	Change in Scenario	Value after Scenario
A	1	The person is asleep.	The person has woken up, and gone into the bathroom.	2
B	1	The person is brushing teeth/washing face.	The person has finished brushing teeth/ washing face.	2
C	1	The person is conducting activity, *Breakfast*.	The phone rings, and the person answers phone and suspends current activity.	2
D	1	Person is speaking on the phone.	Person finishes phone call and carries on *Breakfast* ADL.	2
A	2	Person is making tea and toast.	Person has finished making and eating breakfast.	3

Table 4 – Parameters for ADL plan, *Make Breakfast*

Condition	Parameter	Earliest Starting Shift	Latest Starting Shift	Earliest Finishing Shift	Latest Finishing Shift	Minimum Duration	Maximum Duration
Filter	A=2	-	-	-	-	-	-
Setup	B=2	09:00:00	09:05:00	09:10:00	09:15:00	2 Minutes	7 Minutes
Suspend	C>1	-	-	-	-	-	-
Reactive	D>2	-	-	-	-	-	-
Complete	A=>3	10:00:00	10:05:00	10:25:00	10:30:00	3 Minutes	6 Minutes

Table 5 – Conditions in ADL plan, *Make Breakfast*

Note that in Table 5, not all of the durations and timing intervals for the conditions have been included. This is because there are some activities, which may occur at any time during day (e.g. answering phone), and you cannot make an assumption on the time or duration of the phone call. Table 6 shows what happens when each condition is met during the execution of an activity.

Date	Time	A	B	C	D	Order of events
12.10.2008	08:30:00	2	1	1	1	ADL *Breakfast* becomes considered as the person has woken up and the filter precondition is 2.
12.10.2008	08:45:00	2	1	1	1	The person is washing face/ brushing teeth.
12.10.2008	09:10:00	2	2	1	1	The setup precondition has been met, as the person has finished brushing teeth and washing face.
12.10.2008	09:00:00	2	2	1	1	The ADL plan *Breakfast* is currently being conducted by the person.
12.10.2008	09:20:00	2	2	1	1	
12.10.2008	09:25:00	2	2	2	1	The suspend condition has been met, as it 2. This is because the person has started to *answer the phone*.
12.10.2008	09:35:00	2	2	2	1	The reactive condition has still not been met, therefore the ADL plan *Breakfast* remains suspended.
12.10.2008	09:40:00	2	2	2	2	The reactive condition has been met (D=2), and the *Breakfast* ADL has resumed.
12.10.2008	09:50:00	2	2	2	2	The ADL plan *Breakfast* is still being conducted by the person.
12.10.2008	10:20:00	3	2	2	2	The complete condition for *Breakfast* has now been fulfilled, as the activity has been completed.

Table 6 - Conditions and intervals for *Breakfast*

3.3.1.2 ADL Execution Synchronisation

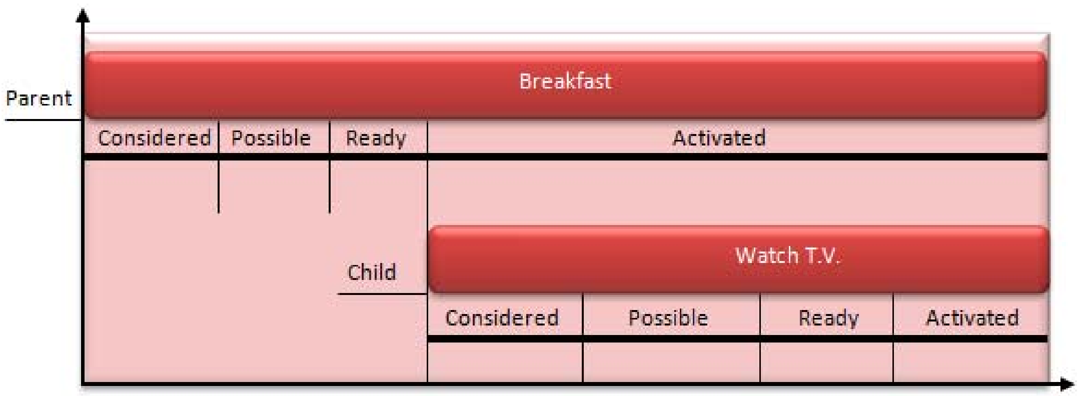


Figure 15 - Parent-Child synchronisation between ADLs in Asbru

Asbru has the capability of representing and managing the execution of more than one ADL at a given time. This is because of the parent-child like synchronization between different ADLs. In Figure 15, the child is an ADL (sub-activity "*Watch T.V.*") invoked by another parent ADL (ADL "*Breakfast*"). The child's preselection phase starts only after the parent's preselection phase terminates. In other words, the sub-activity's filter condition is not checked until the ADL is activated. Thus an ADL is executed once the complete condition of the ADL has been fulfilled and all of its mandatory sub-activities have been completed.

Another important aspect of Asbru is that it allows different ADLs to have different execution orders. The execution orders of an ADL have been represented with Asbru as Sequential, Parallel, Any-order and Unordered execution order.

Sequential Execution Order

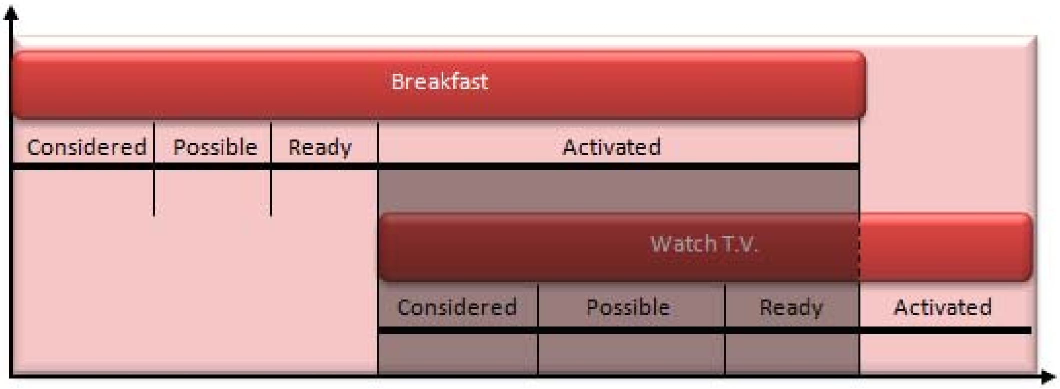


Figure 16 - Sequentially ordered ADL

For an ADL that has a sequential execution order, its children execute in the prescribed sequence. The second ADL's preselection phase cannot begin until the first ADL completes or aborts (Figure 16). This is also the same for any sub-activities that are sequential within an ADL that might not have a sequential execution order.

Parallel Execution Order

Figure 17 - Parallel ordered ADL

All the sub-activities that have a parallel execution order are executed so that they are all synchronised together. If the conditions or filters in the preselection phase of ALD 1 are not fulfilled, then ADL 2 has to wait until ADL 1 has fulfilled its conditions. If ADL 1 is aborted, then ADL 2 cannot be executed, therefore this leads to the ADLs not being executed. For instance, an ADL "*Make Breakfast*" may have two parallel sub-activities, which are "*Make Tea*" and "*Make Coffee*", as the person being monitored maybe making tea for themselves and coffee for someone else. In Figure 17, the preselection phase could be that the kettle has not reached boiling point, hence "*Make Coffee*" has not been activated. Until the kettle does reach boiling point then none of these sub-activities can be executed.

Any-Order Execution

With this type of execution the pre-selection phase is done in parallel to the other ADLs, however the execution is done one at a time. The other ADLs remain idle when the execution of an ADL is taking place (Figure 18).

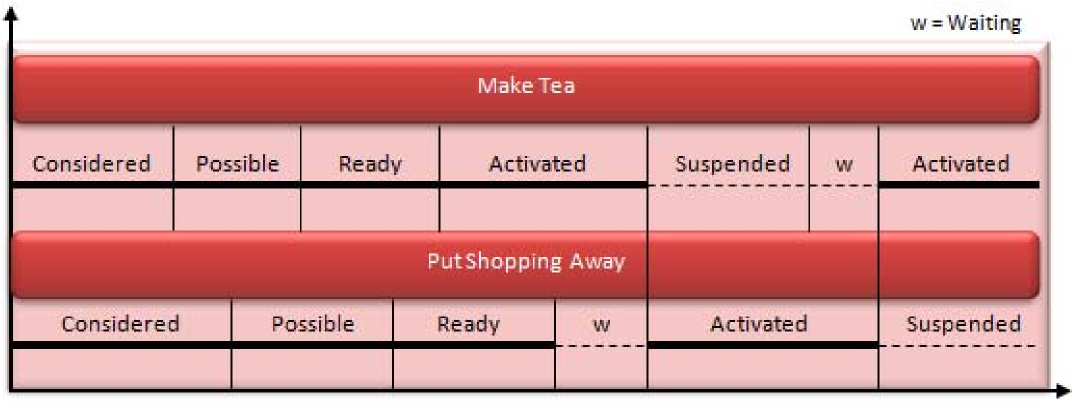


Figure 18 - Any order ADL execution

Unordered Execution Order

In contrast to any-order execution, an ADL which has an unordered execution order is able to execute all the phases of an ADL together (in parallel) or in any order, which means that ADLs can stay idle throughout the pre-selection and execution phase (Figure 19).

Figure 19 - Unordered ADL execution

3.3.1.3 Asbru Representation of Elements for ADLs

Order of execution is a crucial aspect of modelling in Asbru, however it is also important to understand what are the elements that make the different orders of execution possible.

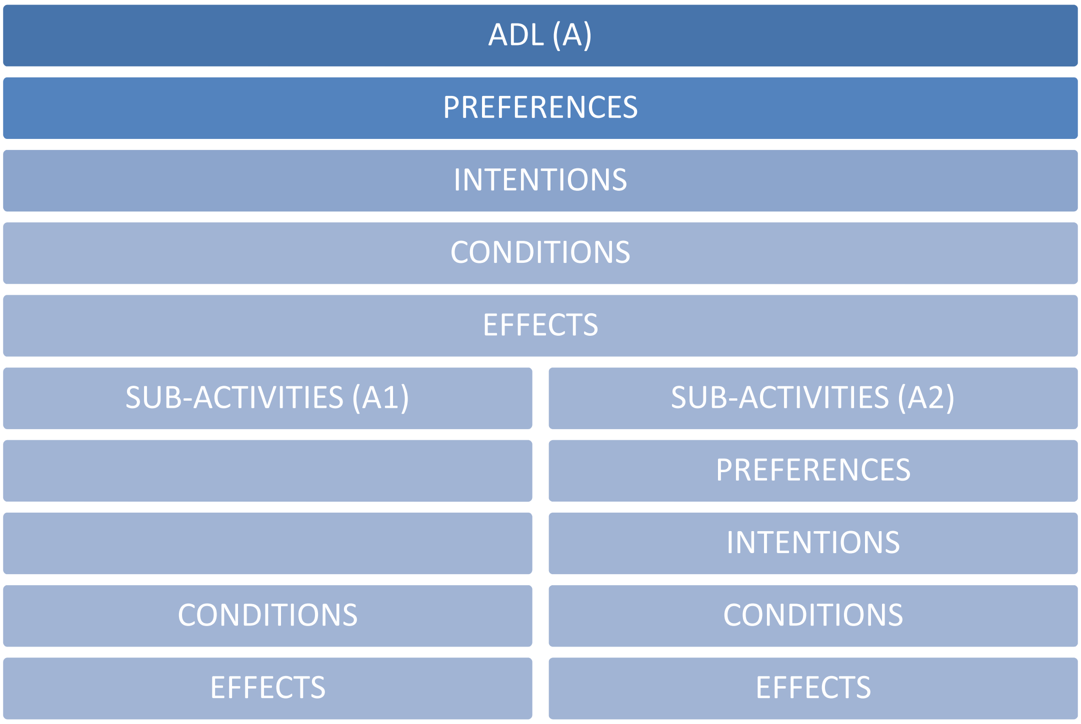


Figure 20 - Representation of the elements

Figure 20 shows an example of how the elements are structured within an ADL. ADL A is the root ADL and it consists of sub-activities A1 and A2, while both sub-activities A1 and A2 may have further sub plans nested within them. As well as that ADL A and the sub-activities will have actions nested within them. Actions are also known as the tasks that were discovered in the low level modelling.

Asbru makes it possible to associate ADLs, sub-activities and tasks with time intervals, as each of them has time assigned for executing the ADLs. An ADL modelled in Asbru consists of an ADL name, five key components and a set of arguments that include a time annotation which is used to represent the temporal scope of the ADL. The five key components (preferences, intentions, conditions, effects and plan body) and set of arguments are optional, making the ADL name a mandatory component.

Preferences is a component that is used to describe the requirements needed for an ADL to be carried out. Below are some of the elements of the preferences component. These are optional:

- Strategy – this is the approach that is employed in order to deal with the ADL that is going to be carried out, for example: conservative, aggressive. These are generally related to treatment strategies for medicines within clinical guidelines. Therefore on some occasions the use of strategies may not be required when modelling ADLs.
- Resources – this is the set of recommended, obligatory or discouraged resources that can be taken into consideration when carrying out an ADL.
- Responsible-Actor – this is the set of actors who are needed to carry out the execution of the ADL.

Below is an example of how the preferences component can be used within an ADL.

```
<preferences responsinle-actor="elderly person" strategy="conservative">
    <resource-constraint name="kettle" type="recommended">
        <time-annotation>
            <time-range>
                <duration>
                    <minimum>
                        <numerical-constant unit="m" value="10"/>
                    </minimum>
                    <maximum>
                        <numerical-constant unit="m" value="15"/>
                    </maximum>
                </duration>
                </time-range>
                </self>
            </time-annotation>
        </resource-constraint>
</preferences>
```

Figure 21 - The use of preferences within ADLs modelled in Asbru

The example in Figure 21 shows that the execution of this ADL will require a kettle for 10-15 minutes. In addition the strategy for this ADL is labelled conservative and the elderly person in the home is responsible for carrying out the ADL.

Intentions is a component that is used to specify the high level goal for an ADL and sometimes for its sub-activity. The goal is an element that allows annotation of the ADL and can be used to give the meaning of the ADL. This element is very important as it provides information about what the objective of the ADL is at the time of execution, therefore making it easier to label a context once an ADL has been recognised.

3.3.2 ADL Recogniser

The ADL Recogniser is Java based software that has been developed for this research (See the Appendix for details of this and other software that has been developed in order to validate the algorithms developed for this thesis). There are two versions of the ADL recogniser and both have been used to conduct ADL detection experiments. One version takes in a stream of tasks and works out the possibility of each ADL being an active ADL by calculating the discrepancies with each ADL and sub-activities that could currently be active. By discrepancy it is meant the count of observed tasks that are inconsistent with a particular ADL. This software assumes that only one task is being performed at one time and there is no interweaving of tasks using suspension and resumption. The second version also calculates the discrepancy, but has been further enhanced by incorporating surprise index for each ADL, to reflect the fact that some tasks are more likely than others. In addition the second version is capable of allowing interweaving of tasks. The method behind the discrepancy and surprise index will be described further in the chapter.

The software reads in the ADLs and stores them into memory as a DOM tree. The ADLs are constructed in XML as each ADL has the relevant sub-activities and tasks nested within them. The XML files are created either by hand as a source XML document by a graphical tool called AsbruView, which is developed by researchers working on the Asgaard project (Figure 22).

Once the ADLs have been loaded into memory, the ADL recogniser then acts as a server which listens for incoming task notifications. These task notifications are the tasks that have been determined from the low level modelling. After each action is read, the estimator then outputs the names of the ADLs and sub-activities that may be currently active. Depending on how many ADLs the task belongs to, the ADL recogniser provides a list of the most probable ADLs that may be currently active. The list is in an ascending order, with the most probable being at the top and the least probable at the bottom. As each action is read into the system the output possibilities regarding the current ADL get smaller and therefore make it easier to determine which ADL is currently active. The output of the most probable ADL is determined by the discrepancy and surprise indices which are calculated by the ADL recogniser.

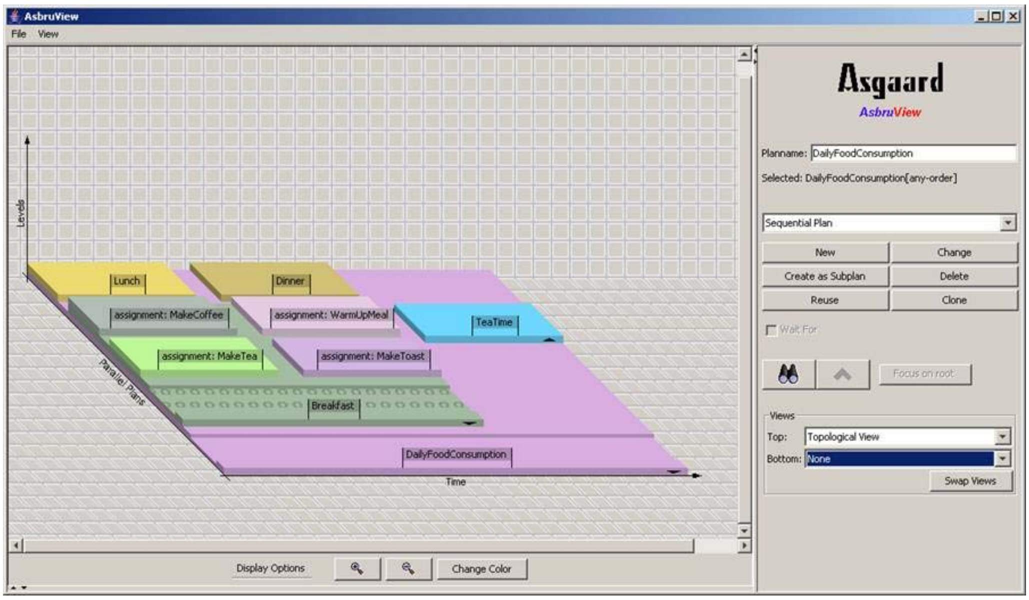


Figure 22 - AsbruView used to construct ADL XML files

3.3.2.1 Computing Discrepancies and Surprise Indexes

When constructing an ADL it is possible to construct one ADL per XML file, or several ADLs can be constructed into one larger XML file. Both of these options are likely to lead to a situation where one XML file will contain the same tasks.

When an ADL has been detected by the ADL recogniser this is represented by the path of XML file that has been detected. If two possibilities are detected then this will be represented by the path to both files.

In an XML file, a discrepancy is an action/task (i.e. single step plan), that has not been detected that should have been detected if the ADL were executed. The overall discrepancy of an ADL is computed by summing the discrepancies of its sub-activities.

To compute the overall discrepancy, two discrepancy counts for each ADL are calculated, namely the completed discrepancy count and incomplete discrepancy count. 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.

Whether an ADL has been logically completed or not it is represented by true or false of its completed label. The completed label has a default false value. All labels in one path of an XML file are set recursively to true once a new action is detected in the XML file. When the completed label is set to true, the ADL can be idle as tasks within this ADL might be detected later.

The mechanism to mark labels as complete is based on:

1. The execution order - sequential, parallel, any-order, or unordered.
2. The continuation condition – whether a sub-activity is optional or mandatory for its parent ADL's continued execution.
3. The filter pre-condition – the compulsory conditions for an ADL to be activated.

Once a new action/task is detected or otherwise known as completed, the following discrepancy counting processes occur:

Process 1: If the parent ADL has filter preconditions, then all other ADLs that are compulsory to fulfil the pre-conditions should have been completed. Hence these ADLs are set as being completed.

Process 2: If all actions and mandatory sub-activities of an ADL have been set to completed, then this ADL is set as being completed.

Process 3: An ADL is only set as completed, once it has been completed, according to the assigned order of execution. For example, if a parent ADL is sequential, then all its preceding mandatory child ADLs should have been completed in the sequential order. This is also true for ADLs that have parent plans that are either parallel, any-order, or unordered, as the child ADLs will only be set as completed once they have been executed in a particular order.

Process 4: If an ADL has been set as completed then all mandatory children should have been completed, hence these mandatory children are set to complete. This process traverses down the ADL to the sub-activities that are nested within it.

Process 5: The process continues in a depth first search like manner - traversing from the current ADL to its siblings, then parents, repeating process 1-4 until no parent ADL is available, e.g. reach the highest level ADL (also known as the root ADL). The completed discrepancy and incomplete discrepancy of each ADL are updated if any changes take place.

Working Example

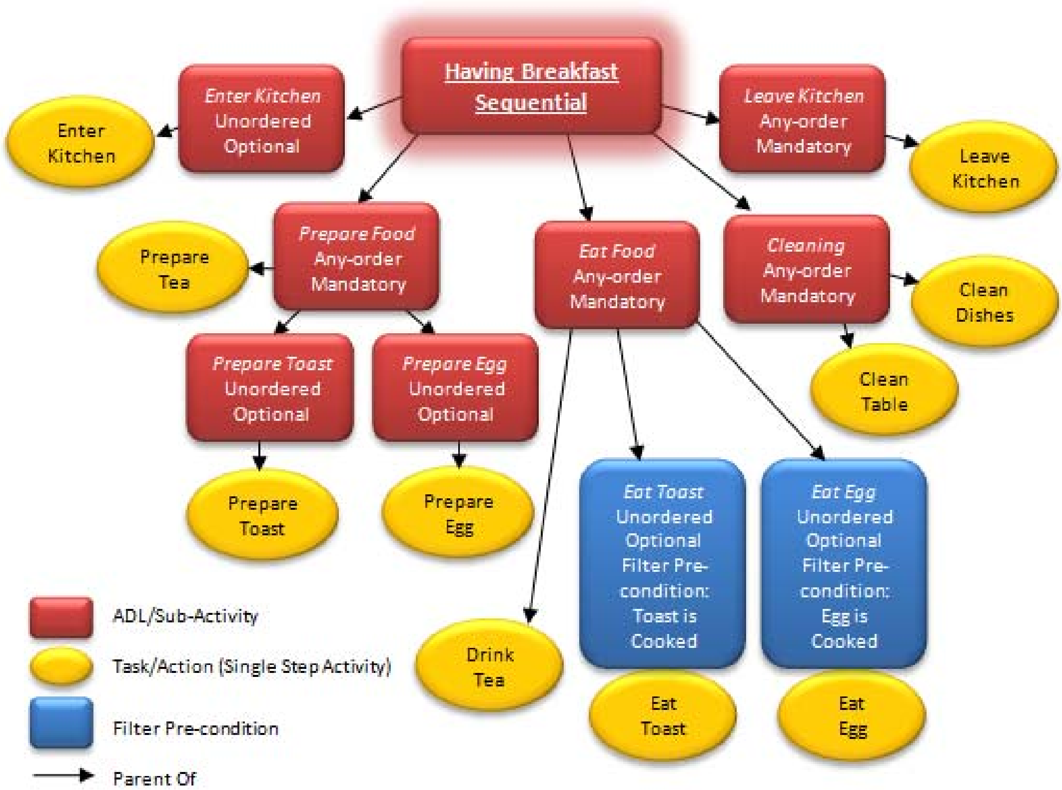


Figure 23 - Modelled ADL example of 'Having Breakfast'

All the examples that have been modelled in this thesis are examples chosen to illustrate the method and different points, but they could also have been modelled in a variety of ways with other plans. This working example has been modelled to illustrate how discrepancies are computed for a simple "*Having Breakfast*" ADL (Figure 23): It is supposed that the following actions/tasks are detected in the low level modelling – "*Enter Kitchen*", "*Prepare Toast*", "*Drink Tea*", "*Eat Egg*", "*Clean Dishes*", and "*Leave Kitchen*" - in this order. At the detection of each action, the above recognition processes (1 to 5) will take place. Note that by convention, in Asbru ALL single action plans are mandatory. If a single action needs to be optional it has to be embedded in another optional plan, which can contain the single activity.

1. Enter Kitchen is detected

Enter Kitchen is the only task in sub-activity *Enter Kitchen*, hence Process 2 will occur here and the single step plan (task) *Enter Kitchen* is set to completed. The update process continues and stops when reaching the sequential root plan *Have Breakfast*, since *Enter Kitchen* has no preceding plans.

2. Prepare Toast is detected

Similar to the case when *Enter Kitchen* was detected, Process 2 also occurs here and the single step plan (task) *Prepare Toast* is set to completed. The discrepancy counting algorithm goes to the sequential sub-activity *Prepare Food*, and Process 3 occurs because the single step plan (task) *Prepare Tea*, which as a preceding mandatory child, should have been completed. However, it has not been detected and so is calculated as a discrepancy. The update process continues until the root ADL is reached.

3. Drink Tea is detected

Process 2 occurs here as sub-activity *Eat Food* has been set as completed since the only mandatory child single step plan (task) *Drink Tea* is completed. Also the sub-activity *Prepare Food* is set to complete, as it is a preceding sub-activity to *Eat Food*. The reason why *Prepare Food* is set to complete is because there is a possibility that the task recognition component may have not discovered the task *Prepare Tea*. Even though the sub-activity *Prepare Food* has now been set to complete, the discrepancy count remains the same. The discrepancy count is important, as an ADL plan which has a high discrepancy count is less likely to be the ADL that is being conducted.

4. Eat Egg is detected

Process 1 occurs here as in order to fulfill the filter condition "*egg is cooked*", the single step plan (task) *Prepare Egg* should have been completed. Like the previous *Prepare Tea* situation, *Prepare Egg* is also set to complete, where the discrepancy count for the sub-activity *Prepare Food* remains the same and does not decrement.

5. Clean Dishes is detected

Any-order sub-activity *Cleaning* is not set to completed because only task clean dishes was detected and both of the tasks (*clean dishes* and *clean table*) were required to be carried in order for the sub-activity to be set to complete, as the sub-activities were mandatory.

6. Leave Kitchen is detected

Process 2 occurs here as the ADL *Leave Kitchen* is set to completed, also as this is the last task of the task sequence the overall discrepancy of the ADL can be calculated.

The completed discrepancy and incomplete discrepancy count of each ADL, sub-activity and single step plan (tasks) are updated if any changes take place. The overall discrepancy is calculated as the sum of the chosen completed or incomplete discrepancies of each ADL and sub-activity.

In this example the modelled ADL's final matching result is shown in Table 7. It can be seen from the result that the overall discrepancy of "*Having Breakfast*" is 3, which means if there all other ADLs have a higher overall discrepancy than 3 then "*Having Breakfast*" is the ADL that is being conducted. The recognition process does not necessarily just rely on the overall discrepancy, as at each step when a task is discovered the individual discrepancies and complete labels can be used to assist the recognition process, meaning there is no need to wait for a complete stream of task sequences before determining the activity.

ADL/ Sub Activities/ Task	Execution Order	Mandatory or Optional	Complete Label	Complete Discrepancy Count	Incomplete Discrepancy Count
Having Breakfast	Sequential	Root Plan	False	0	0
Enter Kitchen	Unordered	Optional	True	0	0
Leave Kitchen	Unordered	Optional	True	0	0
Prepare Food	Any-order,	Mandatory	True	1	0
Prepare Toast	Unordered,	Optional	True	0	0
Prepare Egg	Unordered,	Optional	True	1	0
Eat Food	Any-order,	Mandatory	True	0	0
Eat Egg	Unordered	Optional	True	0	0
Eat Toast	Unordered	Optional	False	0	0
Cleaning	Any-order	Optional	False	0	1
Overall discrepancy of plan Having Breakfast is 3					

Table 7 - ADL discrepancies for 'Having Breakfast'

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.

While the discrepancy is computed whenever there is any missing mandatory action/task, such as "*Make Tea*" for the ADL "*Having Breakfast*", the surprise index of a missing sub-activity is the maximum of the conditional probabilities $P[a_i \mid b]$ of its missing sub-activities or actions/tasks occurring [a_i *s*] given that the ADL [*b*] is being conducted. A mandatory task will have probability of 1. The maximum is taken over all the immediate sub-activities or actions, i.e. children. This is clearly a cautious estimate and the approach ad hoc, but the information required to use a more sophisticated approach, such as Bayesian networks would need significant knowledge collection. This could be worthwhile and estimates similar to equation (2) in chapter 2 could be used.

3.3.3 Validation and Verification

The prospect of carers using Asbru to construct ADL and tasks for individual Alzheimer's patients has many benefits as indicated by much of the work in this thesis. It allows tracking of what activity the person is doing given the set of constructed ADLs and tasks. However, the construction of these ADLs could lead to unwanted consequences. For example a plan may have a poor match as the ADL needs redefinition as the disease progresses. ADLs might be modelled that patients may now find difficult conducting. In addition there is also the possibility that a perfectly safe task (e.g. Make Tea) that was constructed six months ago for a particular patient might have suddenly become a dangerous task, as the patients cognitive health has deteriorated. One of the ways to overcome this is by introducing periodic validation and verification that could be implemented so that at the very outset of ADL construction and at 6 month intervals. Also by incorporating the ability to review the ADLs, the review can be used to evolve the ADLs with the patient that it has been modelled for. The system may be able to trigger reviews also. For example, if the recognition system observes the recognition rates for the ADLs and tasks are changing this could help indicate the ADLs and tasks are no longer modelling the behaviour adequately and it might be the time to review the ADLs.

3.3.4 ADL Recognition Experiment and Results

The objective of this experiment was to recognise an ADL assuming that the tasks had been determined. The 1st version of the ADL recogniser was validated with this experiment. The ADLs used for this experiment have a simpler structure than the ADLs that have been used to validate the 2nd version of the ADL recogniser, as they were used to look at the number of tasks needed for an ADL to be recognised as complete. The results for the 2nd version of the ADL recogniser are presented later with the TASE experiment results, as both approaches were combined in the second recogniser in order to obtain more reliable task and ADL recognition. For this experiment the ADLs and tasks that were modelled are shown in Figure 24.

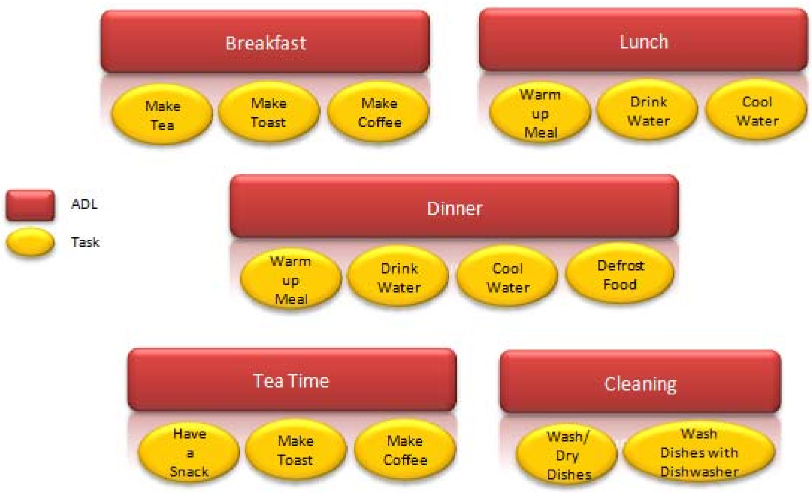


Figure 24 - Modelled ADLs and tasks for ADL recognition experiment

The performance of the ADL recogniser version 1 was analysed by looking at the recognition rate of each ADL given the tasks. This was done by looking at how many tasks were needed for an ADL to be recognised as complete by the ADL recogniser. For example in experiment repetition number 6 in Table 8, the "*Breakfast*" ADL was recognised after two tasks out of a possible four tasks were completed, specifically these tasks were "*Make Toast*" and "*Make Tea*". Note that the each column within the table shows the results for 10 repetitions of the experiment in order to get an average result in terms of performance. The task sequences that have been used for this experiment were the sequences of tasks (ten tasks) that were generated from the MBHMM and Viterbi-based experiments to be described in the next chapter. These task sequences covered each of the ADLs, where one task could belong to more than one ADL. This was done intentionally to see how the ADL recogniser would deal with tasks that belong to more than one ADL. The task sequences for each of the 10 ADL repetitions were in different orders, but on this occasion the ADLs were not interwoven.

		Number of Repetitions of Experiment											
ADLs	Task(s) within ADL	1	2	3	4	5	6	7	8	9	10	Average number of tasks needed to recognise ADL as complete	%
Breakfast	4	3	3	3	3	4	2	4	4	3	3	3.2	80
Lunch	3	3	3	3	2	3	2	3	2	2	3	2.6	86.7
Dinner	4	3	4	4	4	3	3	4	4	4	4	3.7	92.5
Snack	4	2	2	3	4	3	2	4	4	2	4	3	75
Cleaning	2	1	1	2	1	2	1	2	2	1	2	1.5	75

Table 8 - ADL recogniser version 1 experiment results

The percentage column in Table 8 indicates the percentage of tasks that are needed to determine the ADL. The results (Table 8) that were gathered from the ADL recogniser show that the ADL recogniser was successful in recognising the 5 ADLs. As all the concerned ADLs are presumed to remain idle within the ADL library, this gives an indication that even if the task is left in-complete and is executed later, then the ADL Recogniser will still be able to recognise the ADL.
This is a very simple experiment, and clearly the performance depends very much on the degree of overlap between tasks in different ADLs. Because of this different scenarios with different degrees of overlap will be considered.

However, even for such a simple scenario, the ADL recogniser still needs to be improved. The percentage column in Table 8 indicates the percentage of tasks needed to determine an ADL. The percentages are between 75% and 92% which means that the ADL recognisor can need at least 80% of the tasks completed in order to recognise that the "*Breakfast*" ADL has been completed. This of course, depends on the nature of the ADL. The more optional sub-activities and the more sharing of sub-activities the more difficult it is to be absolutely sure. However, even if ADLs are not identified uniquely, the set of possible ADLs may be enough to a) give feedback to the task identification system and b) support context sensitive help - as the ADLs may be related. Introduction of the temporal constraints into the recognition process will add increased discrimination. This will be explained in the following chapters.

3.4 Task Recognition

A range of models have been developed to identify tasks from a stream of object usage data. The algorithms that have been developed and tested use HMMs as a framework, a task segmentation approach and an extension that generates a set of alternative task sequences based on the conjunction of the disjunction of task possibilities for each sensor event used to determine the task. As well as presenting these algorithms, the results from the experiments conducted in order to validate the algorithms are also presented.

3.4.1 Hidden Markov Model Modelling for Tasks

A Hidden Markov Model (HMM) models transitions between states, where the observables are related to the state that is active. Here the hidden states are the possible tasks (note, not the ADLs) the elderly person is carrying out, e.g. *Making Tea*. The observables parameters are the sensor events e.g. switch on kettle. Two different types of HMM have been used and compared in two episode recovery experiments, one for each model. By episode recovery is meant determination of the correct sequence of tasks from a stream of object use data.

The first HMM model used for the episode recovery experiment was based on the approach developed by Wilson [131]. This used a simple HMM and the Viterbi algorithm, which was bootstrapped with knowledge mined from the Internet. The Viterbi algorithm computes the most likely sequence of the sensor readings. After this each sequence of sensor readings was segmented into the ADLs that they belong to. In order to measure the accuracy of the segmentation of sensor readings Wilson et al used the P_k metric. This is formulated as a probability that two sensor readings at a distance of k from each other are incorrectly segmented. This statistical approach for segmenting the sensor readings was inspired from Beeferman et al [132] who used a similar approach to automatically partition text into coherent segments.

Below is a formal definition of the Viterbi algorithm as used by Wilson et al, which is also used for the episode recovery experiments that are described later in this chapter. Denote the transition probability matrix of moving from state *i* to state *j* by $[a_{ij}]$, the probability of observing sensor *j* given in state *i* by the confusion matrix by $[b_{ij}]$, and the prior probabilities of being in each state by $[\pi_1, \pi_2 .. \pi_n]$. $a_{ij} = prob[T_{t+1}(j) \mid T_t(i)]$, where $T_t(i)$ represents being in state *i* at time step *t*. While $b_{ij} = prob[observation = j \mid T_t(i)]$. There are *n* hidden states. Here the sensors are RFID tags. It is assumed that sensors are either triggered or it is not, so the domain of every sensor reading is Boolean. There can however be many different Boolean sensors. In this model time increments with each observation.

The algorithm begins by initialising the probability calculations of each *i*, *i=1,…n* in T_0 by taking the product of the initial probabilities $[\pi_1, \pi_2 .. \pi_n]$ of the hidden states with the observation probabilities $[b_{i_{k_k}}]$, e.g. $T_0(i_1) = \pi_1 \cdot b_{i_{k_k}}$ is the probability of the path ending with observation i at time1.

The Viterbi algorithm then works out the probability of the most probable route to the next state.

$$T_1(i) = \max\lfloor T_{1-1}(j) a_{ji} b_{i_{k_t}} \rfloor \tag{3}$$

This is achieved by firstly working out all the products of transition probabilities with the maximum of the probabilities from the preceding step, e.g. in the case of T_1 the proceeding step would T_0. This is then multiplied with the conditional probabilities of the observations $[b_{i_{k_k}}]$ for the current T_n which is T_1 for this example.

The algorithm applies this method to each step, $T_2(i) = \max\lfloor T_{2-1}(j) a_{ji} b_{i_{k_t}} \rfloor \ldots$ $T_n(i) = \max\lfloor T_{n-1}(j) a_{ji} b_{i_{k_t}} \rfloor$, which in turn determines the most probable route to next state, which in the case of the episode recovery experiments is the most likely sequence of the sensor events.

The HMM model used by Wilson was a fairly simple model with several limitations, which are discussed later. Another HMM model, developed as part of this research, eliminated some of the deficiencies of the simple model used by Wilson, was also used for episode recovery. Comparisons on which model is better suited to carry out the identification tasks at the low level have been made. In this thesis this model is called the Multiple Behavioural Hidden Markov Model (MBHMMs). It is vital that the recognition of the tasks is accurate in the low level of HADL, as this will have an effect on the recognition that will be carried out in the higher levels. Note that for the episode recovery experiment conducted for this thesis no form of bootstrapping has been used for the HMM model, as the idea is to see if the task recognition can be conducted with task being conducted in more than one variation of how it can be done.

3.4.2 Multiple Behavioural Hidden Markov Model

The hidden states on this occasion are the steps (states) which are taken to complete a task. The steps will be referred to as states. For example, a simple model of the task "*Make Tea*" could be to *switch the kettle on*, followed by *putting sugar in cup* then *adding milk to the cup of tea*. Rather than having one Hidden Markov Model which determines the tasks from the observables (the sensors), this approach determines which task is currently active, under the hypothesis that the model is the correct one.

Figure 25 - Multiple variations for *Make Tea*

A task is a reasonable simple sequence of steps and does not require to be modelled by a plan. Rather the steps are modelled by a simple probabilistic state transition sequence diagram. Because even for simple models different sequences need to be modelled (whether you put the milk in before the sugar or vice versa is perhaps irrelevant) multiple models, called variants, are used to represent the small set of different orderings (Figure 25).

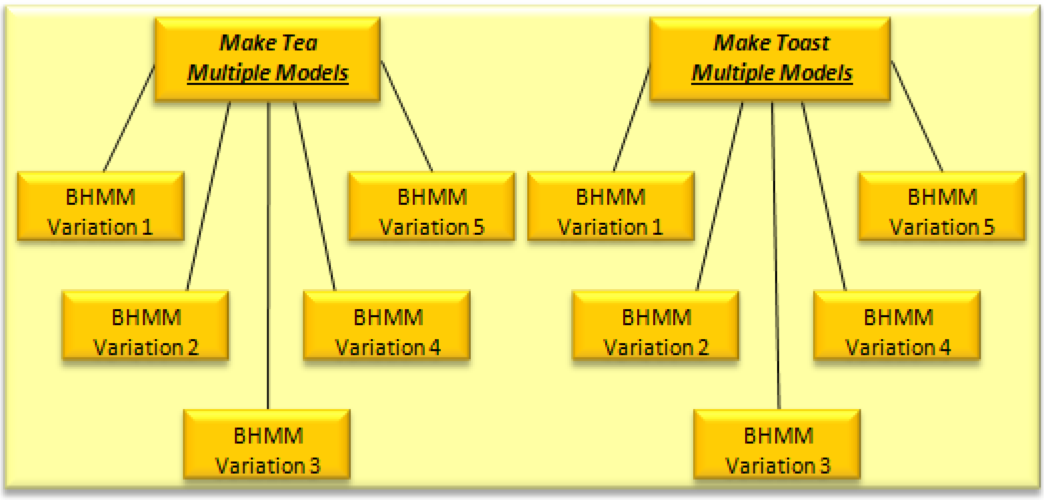


Figure 26 - A set of task models for and their variants

Multiple models are created for each task and the one that fits the sensor readings best is chosen as identifying the task. There is still the problem of episode recovery for example, there will be separate variants for different tasks such as "*Make Tea*" or "*Make Toast*". This is because the elderly may carry out the task (e.g. *Make Tea*) in a different way. Figure 26 shows an example of five different models for the "*Make Tea*" task and five for "*Make Toast*".

In Figure 26 any variation could be the true model, such as "*Make Tea*" Variation 3 or even "*Make Toast*" Variation 2. Whichever model has the highest probability given the observations is chosen.

One of the advantages of this approach is that even if the elderly person has not finished completing the task it is still possible for the MBHMM to determine which task is currently active. This is because the probability of being in the final state of the model is computed as each sensor reading is read, which means that the provisionally identified task can be used in the plan recognition modules. Sensor events are mapped onto a trellis where each column corresponds to a sensor event. Each row gives the probability of being in that state given the observations up to and including the time corresponding to the column. An example of a variation of the model of "*Make Tea*" showing a trellis after two observations is shown in Figure 27.

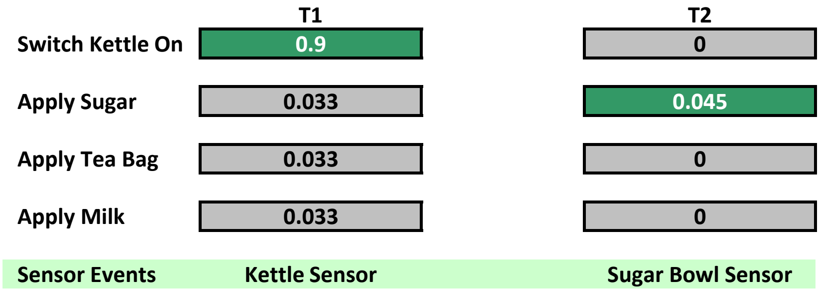


Figure 27 - Make tree trellis

Figure 28 shows how the state transition diagram and table of prior probabilities (Table 9) for the model "*Make Tea*" variation 1 will look like, which also includes unexpected states.

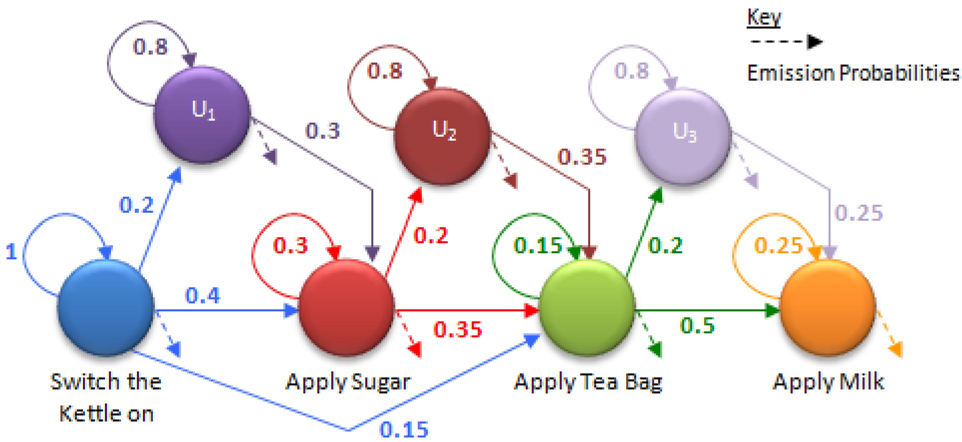


Figure 28 - State transition diagram

States	K	S	T	M	U_1	U_2	U_3
Switch Kettle on (K)	1	0.4	0.15	0	0.2	0	0
Apply Sugar (S)	0	0.3	0.35	0	0	0.2	0
Apply Tea Bag (T)	0	0	0.15	0.5	0	0	0.2
Apply Milk (M)	0	0	0	0.25	0	0	0
Unexpected State 1 (U_1)	0	0.3	0	0	0.8	0	0
Unexpected State 2 (U_2)	0	0	0.35	0	0	0.8	0
Unexpected State 3 (U_3)	0	0	0	0.25	0	0	0.8

Table 9 - Priori state transition probabilities for BHMM

The state transition probabilities for the multiple models of each task will be different. However, the emission/confusion probabilities will remain the same for all the variations for each task. For example, all the different variations of the multiple models for "*Make Tea*" will have the same emission probabilities.

Before discussing the HMM based episode recovery experiments the mathematical formalisation of the algorithm is described. The algorithm is illustrated by using the "*Make Tea*" Variation 1 model.

For this particular model there are five types of object mapped as sensor events used as observations. These are Kettle Sensor triggered = *k*, Sugar Bowl Sensor triggered = *s*, Tea Bag Bowl Sensor triggered = *t*, Fridge Sensor triggered = *f*, and any other sensor event which is not associated with this model will be referred to as x.

The states in this are model are as follows:

- Switch the Kettle on (K)
- Apply Sugar (S)
- Apply Tea Bag (T)
- Apply Milk (M)
- Unexpected State 1 (U_1)
- Unexpected State 2 (U_2)
- Unexpected State 3 (U_3)

The algorithm works out the probability of being in a particular state given the observed sensor event(s), in relation to "*Make Tea*" Variation model 1 it is represented like this:

$\underline{a_{ij} = P[new\ state = j | old\ state = i]}$, this probability will be determined by the values in the transition matrix. The values in the transition matrix have been assigned based on the sequence that should match the model variant.

$\underline{b_{ij} = P[observe\ \ sensor\ \ reading\ \ j | in\ state\ \ i]}$, this probability will be determined by the values in the emission/confusion matrix.

So that the states can be referred to as indices the state will be labeled as numbers:

K = 1, S = 2, T = 3, M = 4, U_1 = 5, U_2 = 6, U_3 = 7

Also the sensor objects will be labelled as numbers:

k = 1, *s* = 2, *t* = 3, *f* = 4, *x* = 5

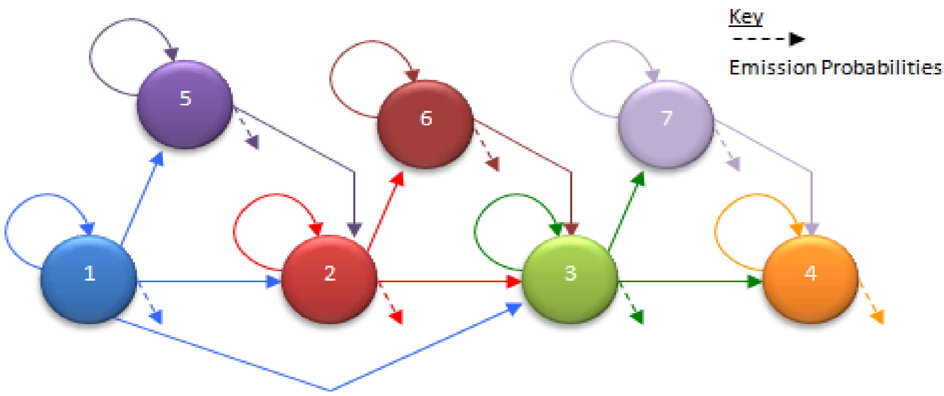


Figure 29 - Numerically labelled states and sensor events

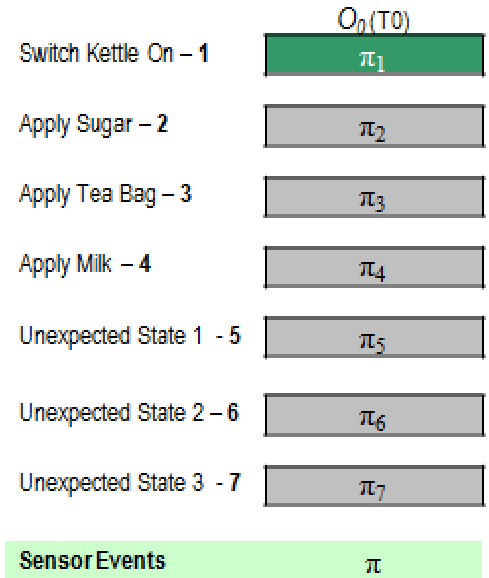


Figure 30 - Initialisation of states

The states are assumed to have initial (π) probabilities in O_0 (T0).

$P[S_0 = 1]$ is the probability that we are in a state when no additional information is available, i.e. it is π_1. $P[S_0 = 2]$ is the probability in state 2 initially and is π_2.

Let O_1 represent the first observation. O_1 can be *k*, *s*, *t*, *m*, or *x*; in the numerical scheme, this will be 1, 2, 3, 4 or 5.

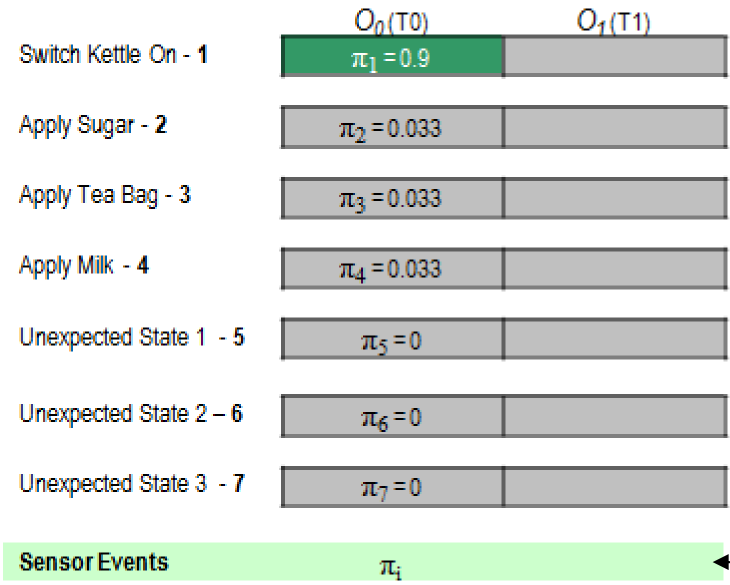


Figure 31 - Possible observations within a variation model for make tea

After the first observation O_1, we want to update the probabilities of being in each state, e.g.

$P[S_1 = i | O_1 = j]$ where e.g. $P[S_1 = 5 | O_1 = 1]$ means the probability of being in state 5 (unexpected state 1) given the first observation is 1 (e.g. k, which is the Kettle Sensor).

We want $P[S_1 = i | O_1 = j]$ for i = 1, …, 6 and for j, which is the observable sensor event. In particular, we want $P[S_1 = 6 | O_1 = j]$, this is decomposed as:

$$P[S_1 = i | O_1 = j] = \frac{P[S_1 = i \wedge O_1 = j]}{P[O_1 = j]}$$

$$P[S_1 = i | O_1 = j] = \frac{P[S_1 = i \wedge O_1 = j]}{P[O_1 = j]}$$

$$\begin{aligned} P[S_1 = i \wedge O_1 = j] &= P[O_1 = j | S_1 = i] \cdot P[S_1 = i] \\ &= b_{ij} \cdot P[S_1 = i] \\ &= b_{ij} \sum_{k=1}^{6} a_{ki} \cdot \pi_k \end{aligned}$$

The observation number
(e.g. O_1, also known T1)

Call $P[S_1 = i \wedge O_1 = j]$ $\quad \alpha_i^1(j)$

The sensor event
(e.g. Sugar Bowl Sensor)

So $\quad P[S_1 = 1 \wedge O_1 = 1]$ is $\alpha_1^1(1)$

$\quad P[S_1 = 6 \wedge O_1 = 2]$ is $\alpha_6^1(2)$

The unknown state
(e.g. Apply Sugar)

$$P[S_1 = i | O_1 = j] = \frac{P[S_1 = i \wedge O_1 = j]}{P[O_1 = j]}$$

$$
\begin{aligned}
&P[O_1 = j] \\
&= P[(S_1 = 1 \wedge O_1 = j) \vee (S_1 = 2 \wedge O_1 = j) \vee \ldots (S_1 = 6 \wedge O_1 = j)] \\
&= P[S_1 = 1 \wedge O_1 = j] + P[S_1 = 2 \wedge O_1 = j] + \ldots P[S_1 = 6 \wedge O_1 = j] \\
&= \alpha_1^1(j) + \alpha_2^1(j) + \ldots + \alpha_6^1(j) \\
&= \sum_{k=1}^{6} \alpha_k^1(j)
\end{aligned}
$$

$$\therefore P[S_1 = i \mid O_1 = j] = \frac{\alpha_i^1(j)}{\sum_{k=1}^{6} \alpha_k^1(j)}$$

Therefore in order to work out each state 1 to 6 after the first observation O_1, which is j (the sensor event), we have to compute the following:

$\alpha_1^1(j), \ldots \alpha_6^1(j)$

followed by $\sum_{k=1}^{6} \alpha_k^1(j)$ and hence $P[S_1 = 1 | O_1 = j], P[S_1 = 2 | O_1 = j] \ldots P[S_1 = 6 | O_1 = j]$

This was the first part of the algorithm , which was to work out the probability of each state given the first observation O_1. The second part of the algorithm will work out the probability of the each state given the observations which follow the first observation O_1, which are $O_2, O_3, O_4, O_5, O_6 \ldots\ldots O_n$.

In the following notation *k* represents the new observation.

$$P\left[S_2 = i \middle| O_1 = j \wedge O_2 = k\right]$$
$$= \frac{P\left[S_2 = i \wedge O_1 = j \wedge O_2 = k\right]}{P\left[O_1 = j \wedge O_2 = k\right]}$$

$$P\left[S_2 = i \wedge O_1 = j \wedge O_2 = k\right]$$
$$= P\left[S_2 = i \wedge O_2 = k \wedge O_1 = j\right]$$
$$= P\left[S_2 = i \wedge O_2 = k \wedge \{(S_1 = 1 \wedge O_1 = j) \vee (S_1 = 2 \wedge O_1 = j) \vee \ldots \vee (S_1 = 6 \wedge O_1 = j)\}\right]$$
$$= P\left[S_2 = i \wedge O_2 = k \wedge (S_1 = 1 \wedge O_1 = j)\right] + P\left[S_2 = i \wedge O_2 = k \wedge (S_1 = 2 \wedge O_1 = j)\right] + \ldots + P\left[S_2 = i \wedge O_2 = k \wedge (S_1 = 6 \wedge O_1 = j)\right]$$
$$= P\left[S_2 = i \wedge O_2 = k \middle| S_1 = 1 \wedge O_1 = j\right] \cdot P\left[S_1 = 1 \wedge O_1 = j\right] + P\left[S_2 = i \wedge O_2 = k \middle| S_1 = 2 \wedge O_1 = j\right] \cdot P\left[S_1 = 2 \wedge O_1 = j\right] + \ldots + P\left[S_2 = i \wedge O_2 = k \middle| S_1 = 6 \wedge O_1 = j\right] \cdot P\left[S_1 = 6 \wedge O_1 = j\right]$$
$$= P\left[S_2 = i \wedge O_2 = k \middle| S_1 = 1 \wedge O_1 = j\right] \cdot \alpha_1^1(j) + P\left[S_2 = i \wedge O_2 = k \middle| S_1 = 2 \wedge O_1 = j\right] \cdot \alpha_2^1(j) + \ldots + P\left[S_2 = i \wedge O_2 = k \middle| S_1 = 6 \wedge O_1 = j\right] \cdot \alpha_6^1(j)$$
$$= a_{1i} b_{ik} \alpha_1^1(j) + a_{2i} b_{ik} \alpha_2^1(j) + \ldots + a_{6i} b_{ik} \alpha_6^1(j)$$
$$= b_{ik} \left\{ \sum_{p=1}^{6} a_{pi} \alpha_p^1(j) \right\}$$
$$= \alpha_j^k(j,k) \quad (say)$$

$$P\left[S_2 = i \wedge O_2 = k \middle| S_1 = 1 \wedge O_1 = j\right]$$
$$= P\left[O_2 = k \middle| S_2 = i \wedge S_1 = 1 \wedge O_{1=j}\right] \cdot P\left[S_2 = i \middle| S_1 = 1 \wedge O_1 = j\right]$$
$$= b_{ik} \cdot a_{1i}$$

$$P[S_2 = i | O_1 = j \wedge O_2 = k]$$
$$= \frac{P[S_2 = i \wedge O_1 = j \wedge O_2 = k]}{P[O_1 = j \wedge O_2 = k]}$$

$$P[O_1 = j \wedge O_2 = k]$$
$$= P[(S_2 = 1 \wedge O_1 = j \wedge O_2 = k) \vee (S_2 = 2 \wedge O_1 = j \wedge O_2 = k) \vee$$
$$\ldots \vee (S_2 = 6 \wedge O_1 = j \wedge O_2 = k)]$$
$$= P[S_2 = 1 \wedge O_1 = j \wedge O_2 = k] + P[S_2 = 2 \wedge O_1 = j \wedge O_2 = k] +$$
$$\ldots + P[S_2 = 6 \wedge O_1 = j \wedge O_2 = k]$$
$$= \alpha_1^2(j,k) + \alpha_2^2(j,k) + \ldots + \alpha_6^2(j,k)$$
$$= \sum_{p=1}^{6} \alpha_p^2(j,k)$$

So therefore

$$P[S_2 = i | O_1 = j \wedge O_2 = k] = \frac{\alpha_i^2(j,k)}{\sum_{p=1}^{6} \alpha_p^2(j,k)}$$
$$where \quad \alpha_i^2(j,k) = b_{ik} \left\{ \sum_{p=1}^{6} a_{pi} \alpha_p^1(j) \right\}$$

In order to compute the probabilities for any new sensor events like $O_3, O_4, O_5, O_6 \ldots O_n$, then the formula for determining the states given the previous observations is used as follows, e.g.:

$$P[S_3 = i | O_1 = j \wedge O_2 = k \wedge O_3 = l]$$
$$= \frac{\alpha_i^3(j,k,l)}{\sum_{p=1}^{6} \alpha_p^3(j,k,l)}$$
$$where \quad \alpha_i^3(j,k,l) = b_{ik} \left\{ \sum_{p=1}^{6} a_{pi} \alpha_p^2(j,k) \right\}$$

The Multiple Behavioural Hidden Markov Models approach is derived from an approach that was developed by Han [133]. The aim of the approach was automated recognition of robot behaviour, where the author mapped each behaviour of the robot as a state. For example, the positioning of robot in the left direction, corresponding to the robot turning left, is treated as a state. The way in which the model we have developed differs from the one developed by Han is that in our case each sensor reading could correspond to a variety of states. For example, if the fridge sensor is detected then this may correspond to the state of "*milk being used*" or "*something else being taken out of the fridge*". Another way the model we have created differs from the mentioned model is that we use model variations for each task recognition. There will be further discussion about Han's approach in the chapter.

3.4.2.1 Task Recognition Experiments and Results

Task Recognition from a sequence of events, which is also referred to as 'Episode Recovery', determines which tasks were active and so could have generated the observed stream of sensor data. The objective of these episode recovery experiments described below was to assess the performance of the MBHMM approach in comparison to a standard Viterbi-based HMM. Therefore two sets of episode recovery experiments were performed. The first set of experiments was done using simple Hidden Markov Models with the Viterbi algorithm applied to them in a similar manner to that proposed by Wilson [131], without any bootstrapping. The second set of experiments used MBHMMs to achieve the same objective. Figure 32 shows the state transition diagrams of a similar model proposed by Wilson and the model for the MBHMM.

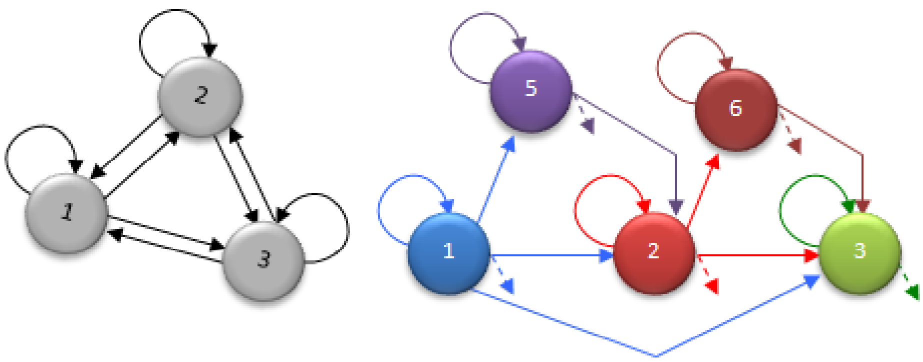


Figure 32 - State Transition Diagrams of Viterbi-based HMM and MBHMM

In the left hand model the states correspond to the possible tasks, and after each transition (and so observation) the system can stay in the same state (task) or move to any other task. There is only one model. The right hand shows one of the many models used. Here the states correspond to stages in a task. The state with no outputs (number *3*) is the final state and indicates that this task has been completed. The model allows for missing observations and extraneous observations. However, this is at the cost of having a model not only for each task, but each variant of a task.

The design was the same for both sets of the experiments, with the kind of HMM being the only difference. The ADLs were kitchen oriented as the experiments took place in a kitchen. For these experiments the tasks ranged from making tea to putting dishes into the dishwasher; these tasks are shown in Table 10. The experiment was split into two parts, in the first part the subjects carried out each task in a given prescribed order, while in the second part the subjects were asked to carry out each task in any order that they wished, and to record this order. Each task was atomic, in the sense that a task such as "*Make Tea*" was not interrupted with task "*Drink Water*".

Tasks
1. Make Tea
2. Make Toast
3. Drink Water
4. Make Coffee
5. Warm up meal
6. Defrost food in microwave
7. Wash/dry dishes manually
8. Have a cool glass of water
9. Wash dishes via dishwasher
10. Have a snack (Biscuit/Crisps)

Table 10 - Tasks for MBHMM vs. Viterbi-based episode recovery experiments

The accuracy for the episode recovery experiment results was determined as a percentage of times the task was correctly identified in comparison to the ground truth collected during the experiment. When a task was identified successfully it was scored as a true positive, while an incorrect claim would be scored as false positive. If a task occurred and the algorithms (BHMMM and Viterbi-based HMM) did not report it (meaning a task going unnoticed) then this is treated as a false negative.

Tasks	True Positive [%]	False Positive [%]	False Negative [%]
Make Tea	85	15	0
Make Toast	80	15	5
Drink Water	60	35	5
Make Coffee	85	15	0
Warm up Meal	60	30	10
Defrost Food	95	0	5
Wash/Dry Dishes	15	65	20
Have a Cold Glass of Water	25	65	10
Wash Dishes with Dishwasher	100	0	0
Have a snack (Biscuit/Crisps)	85	10	5

Table 11 - Predefined state order - Viterbi-based HMM episode recovery results

Tasks	True Positive [%]	False Positive [%]	False Negative [%]
Make Tea	65	5	30
Make Toast	50	10	40
Drink Water	50	15	35
Make Coffee	65	5	30
Warm up Meal	40	25	35
Defrost Food	60	15	25
Wash/Dry Dishes	10	30	60
Have a Cold Glass of Water	20	25	55
Wash Dishes with Dishwasher	70	20	10
Have a snack (Biscuit/Crisps)	50	30	20

Table 12 - Any state order - Viterbi-based HMM episode recovery results

The Viterbi-based episode recovery experiment results in Table 11 show that a majority of tasks like "*Make Tea*", "*Make Toast*" and "*Make Coffee*" were correctly determined when the subjects carried out the tasks in the prescribed order. In contrast, when the subjects carried out the task in their chosen order (Table 12) then this had led to significant downfall in terms of accurately determining the task, as the subjects were allowed to choose an arbitrary order of states to carry out a task.

This shows the inefficiency of Wilson's approach as it could not accommodate different variations in the order a task may be carried out. In the predefined state experiment, the accuracy rate for "*Wash/Dry dishes*" and "*Have Cold Glass Water*" was very low in comparison to the other tasks. This was because these tasks did not have a sensor reading which was exclusive to that task. For example, the dishwasher has a sensor that is only triggered when undertaking a dishwashing task and not for any other task. So triggering the dishwasher sensor leads to a high probability that a dishwashing task is taking place. However, the task "*Have a Cold Glass of Water*" needed a water dispenser/cooler sensor to be triggered. This sensor event is mutually exclusive from sensors associated with all other tasks but still the accuracy rate was only 10%. This is because the other states (stages) of this task were very similar to the states of the task "*Drink Water*" from the tap. Therefore, whenever the task "*Drink Water*" was identified then this lead to the task "*Have a Cold Glass of Water*" going unnoticed.

Tasks	True Positive [%]	False Positive [%]	False Negative [%]
Make Tea	100	0	0
Make Toast	100	0	0
Drink Water	80	15	5
Make Coffee	100	0	0
Warm up Meal	80	15	5
Defrost Food	100	0	0
Wash/Dry Dishes	90	5	5
Have a Cold Glass of Water	35	55	10
Wash Dishes with Dishwasher	100	0	0
Have a snack (Biscuit/Crisps)	100	0	0

Table 13 - Predefined state order - MBHMM episode recovery results

Tasks	True Positive [%]	False Positive [%]	False Negative [%]
Make Tea	95	5	0
Make Toast	95	5	0
Drink Water	80	10	10
Make Coffee	95	5	0
Warm up Meal	85	10	5
Defrost Food	90	0	10
Wash/Dry Dishes	85	5	10
Have a Cold Glass of Water	30	60	10
Wash Dishes with Dishwasher	95	5	0
Have a snack (Biscuit/Crisps)	80	15	5

Table 14 - Any state order - MBHMM episode recovery results

The results from the MBHMM episode recovery experiment (Table 13 and 14) show an improved level of accuracy in task recognition from the sensor readings. The reason why this approach had higher accuracy rates than the Viterbi-based approach is because of the MBHMMs different feasible orderings of sensor readings. The simpler model of Wilson does not impose any order, but it does assume that each task generates, on average the same number of sensor readings, and indeed the probability of returning to the same state is chosen so that the expected number of visits to a state corresponds to this average. This is a rather strong assumption as there is no good reason for supposing that each task should generate the same number of sensor readings. Naturally the model could be augmented by expanding the state space to include not only the task as a state but e.g. the number of times the task had been executed. Then there would not be self loops as the states would be different. Of course this way the state space becomes very large. Unlike Wilson's approach, the MBHMM also takes into consideration all the possible variations of each task. A reason why the MBHMM outperforms the Viterbi approach is because the objective of the Viterbi algorithm is to determine the most likely sequence, whereas the MBHMM determines the probability of being in a finish state of a model given the window of observations being considered. As well as that, this approach was able to solve the problem of missing sensor readings to a certain extent. This is because the models that were constructed for each activity modelled the possibility of an unexpected sensor event occurring between expected sensor events. The idea of the unexpected state being modelled is similar to the concept of Profile Hidden Markov Models, where any unexpected sequence data which occurs in a DNA motif is substituted with an insertion. However this approach only works to a certain extent where the task models are small and manageable (e.g. make tea). If a task model is large and has a few missing sensor readings then the ontology approach could add additional benefit. Yet as seen in the results the unexpected state approach does allow for different variations of one task to be detected better than existing approaches.

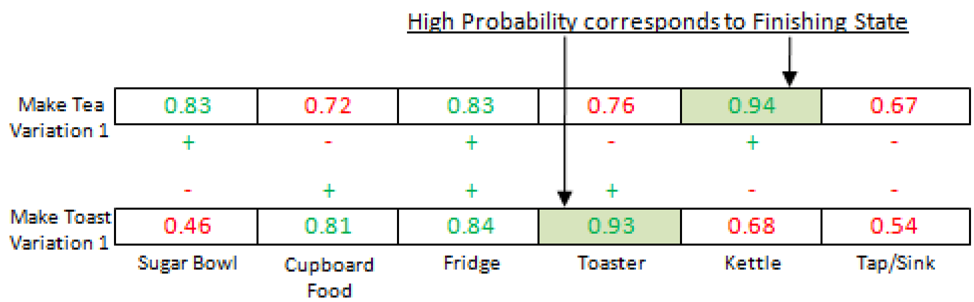


Figure 33 - Defining the finishing state with MBHMM

Figure 33 illustrates why the MBHMM approach is able partially to accommodate concurrent tasks. As well as mapping all the variations, this approach can determine when a finishing state is about to be reached or has been reached, as the probability of each task increases when a related sensor event occurs and decreases when an unknown event for a specific model occurs.

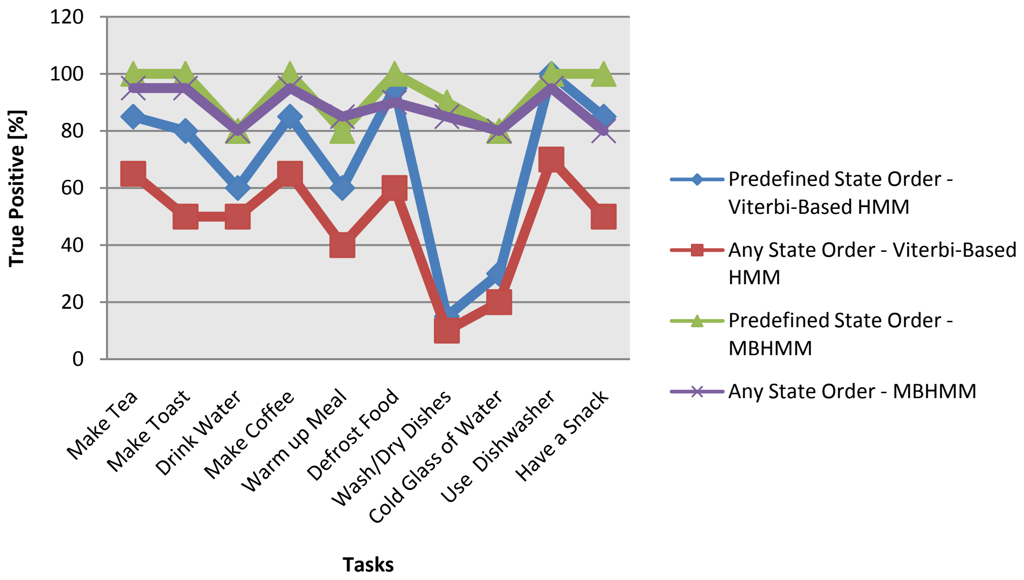


Figure 34 - Comparison of Viterbi and MBHMM approach

From Figure 34 it is clearly evident that the MBHMM highly improved the accuracy of determining the task which is currently active. The Viterbi approach determines the most likely sequence, where as the MBHMM determines:

- The probability of being in a finishing state.
- Most likely sequence and integrates multiple models to accommodate different ordering variations of a task.

3.4.3 Task Associated Sensor Event Segmentation

The previous section showed that the initial identification of tasks can be carried out by simply segmenting sensor events into segments that correspond to a particular task. This section looks at more approaches.

One of the objectives of this thesis is to enhance the accuracy by allowing feedback e.g. the priors, based on the identification of higher level goals. Another approach was developed that allows explicit enumeration of the possibilities. This is used to test the learning and feedback approaches at the ADL level. Since tasks are considered to be short activities, essentially atomic, the stream of sensor events from different objects will be small, and so the enumeration is feasible as long as the combinations are explored in an ordered manner.

For each task (a) and sensor event (b), we can assigned a probability $P[b \mid a]$. This is required when carrying out task segmentation in this type of 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.

In order to accommodate this type of task identification approach a Task Association of Sensor Events (TASE) function is introduced.

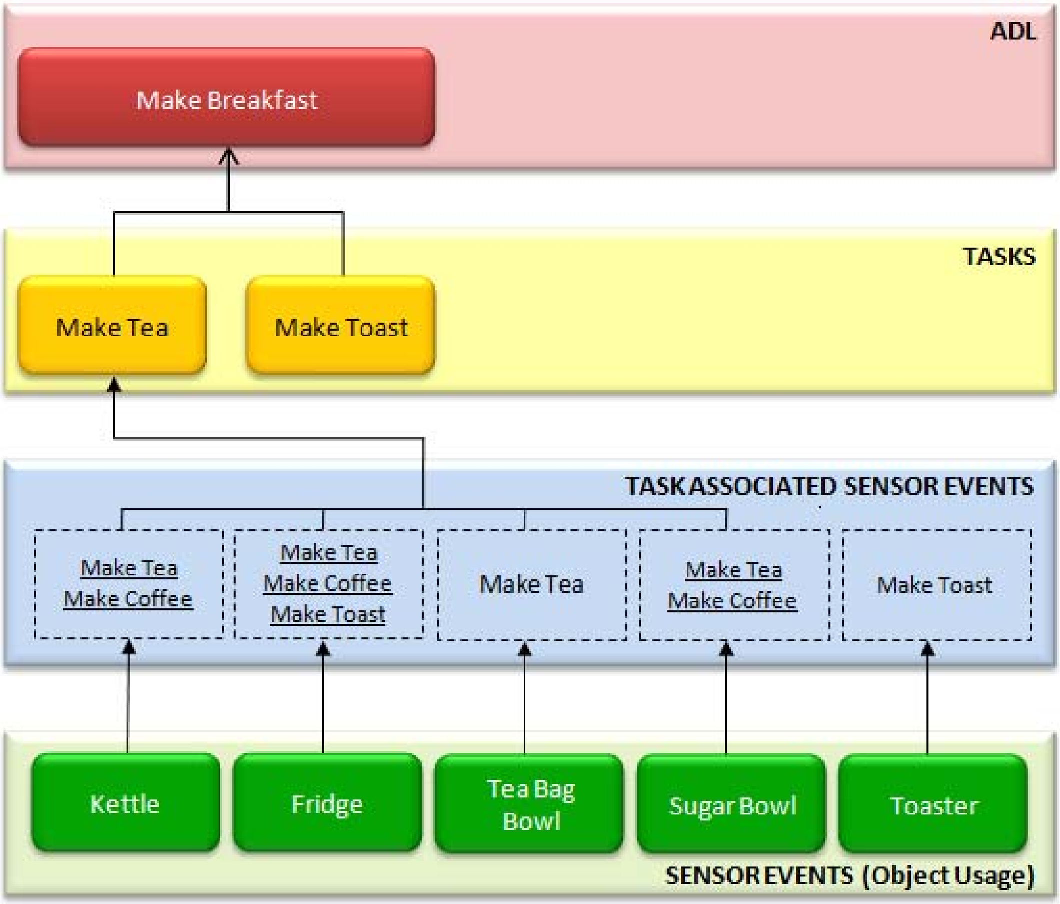


Figure 35 - Modified HADL structure

The lowest tier deals with the incoming sequence of sensor events that have been detected. These sensor events are then associated with the tasks. For example in Figure 35, kettle sensor event can be associated with "*Make Tea*" or "*Make Coffee*". Once the sensor events have been mapped into the associated tasks then an algorithm is applied in order to segment the tasks efficiently. The algorithm described here was based on a statistical model created for text segmentation by Utiyama et al [134].

This method was used to find the maximum-probability segmentation of text, and does not need any training data, as it estimates the required probabilities from the stream of text. In the context of segmenting tasks and using the Task Associated Sensor Event (TASE) segmentation algorithm the tasks are denoted by letters so that a stream of tasks appear as 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).

Figure 36 below shows the different levels of conversion from sensor event to task associated sensor event to stream of letters. The probability values assigned for the letters in the letter stream are based on the number of associations each task has with the total the number of sensor events.

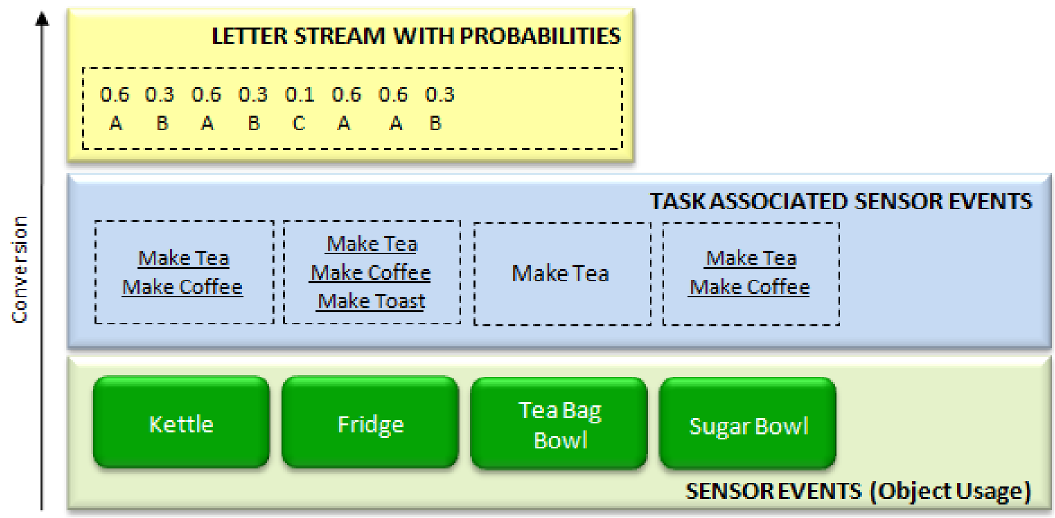


Figure 36 - Levels of conversion with TASE

The stream of letters is then converted into tasks by 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 which lead to stream of letters with different segmentation points:

- A|B|C
- A|BC
- AB|C
- ABC

Each segmentation has a cost associated with it. The cost function (4) is applied to each segment within each stream of letters, which outputs an overall cost for each stream.

$$\sum_{j=1}^{n_i} \log \frac{n_i + k}{p+1} + \log n_i * 0.2 \tag{4}$$

For example, let AB|C be the stream of letters that cost function (4) is going to be applied to. n_i represents the length of the segment within the stream of letters, (AB) n_1 =2, (C) n_2 =1.

While k represents the frequency of each letter in the stream of letters, k(A) =1, k(B) =1, k(C)=1. n represents the total length of the text stream, while p is the prior probability assigned to each letter. The length of the text stream is used to assign the prior probability for each of the letters (tasks).

Below is an example of how the probabilities are generated for the stream of letters AABACA. This works out the prior probability of a letter (task), which is based on the proportion of the letter given the letter stream.

$$p = \frac{k(i)}{n}$$

n =6

$k(A)$ =4, $k(B)$ = 1, $k(C)$ =1

A= 0.68, B=0.16, C=0.16

Table 15 shows the cost of stream AABCA for the prior probabilities:

A=0.6

B=0.2

C=0.2

The stream of letters that has the lowest cost is generally close to a correct segmentation or has in fact been correctly segmented and a sample of the 10 lowest cost segmented streams, gives a good idea of which task is actually being conducted by the person. It is evident that on many occasions the results provided are not perfect in terms of accuracy, but this is where the higher tier is used to refine the interpretation.

The task segmentation in Table 15 shows the cost of each stream for AABCA with different segments, with the lowest cost in 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.

Cost of Stream	1st Segment	2nd Segment	3rd Segment	4th Segment	5th Segment
2.91349	A	AB	CA		
2.91349	AA	B	CA		
2.91353	AA	BC	A		
2.915046	A	A	B	CA	
2.915047	A	A	BC	A	
2.915051	A	AB	C	A	
2.915055	AA	B	C	A	
2.915059	A	A	B	C	A
2.915063	A	A	BCA		
3.054124	A	ABC	A		
3.054129	AAB	C	A		
3.082083	AA	BCA			
3.082090	AAB	CA			
3.332142	A	ABCA			
3.332143	AABC	A			
3.744727	AABCA				

Table 15 - Cost of text segmentation for task recognition

3.4.3.1 Segmentation Collaboration with ADL recogniser

This text based segmentation approach works in conjunction with the ADLs modelled in Asbru and the ADL recogniser version 2 with the aim of achieving reliable ADL recognition. As mentioned in the earlier sections, ADL recogniser version 2 takes into consideration the surprise indexes as well as the discrepancy count for each ADL. The collaboration between these two approaches is formed when the higher tier retrieves the tasks that have been segmented from the stream of data and identifies where the task fits into the plans. For example, if "*Enter Kitchen*" has been segmented correctly then the higher tier goal identification tool could 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 can look at the number of times a task that 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 is worked out by an occurrence model, an example of this is shown in Figure 37. The occurrence model shows the frequency of each task given the sensor observations.

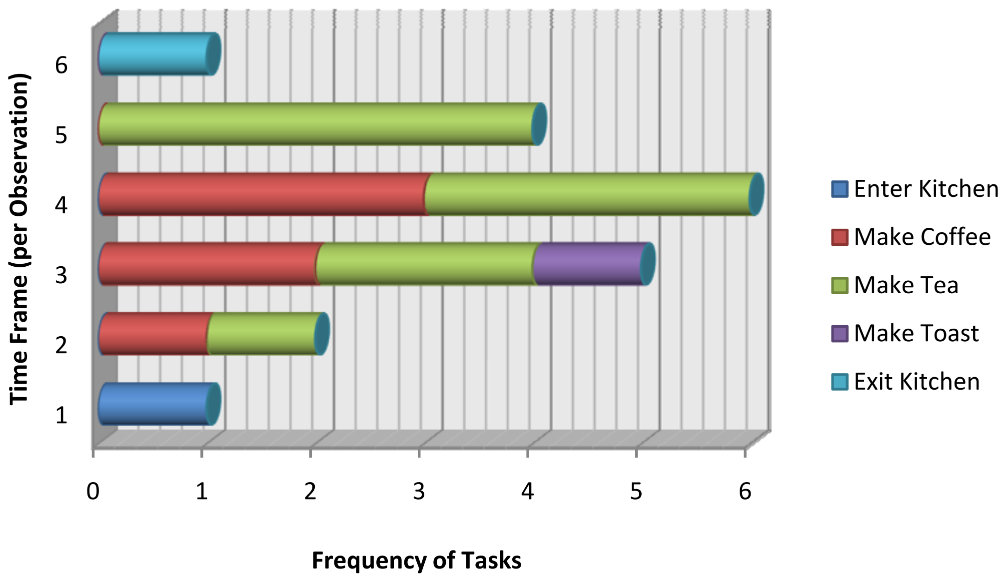


Figure 37 - Occurrence model for tasks given the number sensor observations

In Figure 37 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*" increments with each time frame until time frame five, which suggests 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 and once it is evident that make tea is the correct task as shown in time frame five then task "*Make Coffee*" can be eliminated.

3.4.3.2 ADL Recognition Experiment and Results

The objective of this experiment was to determine which ADLs are active from a collected sensor data stream. The accuracy of these experiments was determined by the proportion of times an ADL was detected correctly, i.e. the detection rate. For each possible plan, the ADL recogniser computes a discrepancy count and most importantly a surprise index, whenever a new task is recognised in the lower tier.

ADL/ Sub Activities	Surprise Threshold	Execution Order
Breakfast	1.0	Sequential
Prepare Food	0.5	Any Order
Clean Dishes	0.5	Unordered
Laundry	1.0	Sequential
Wash Clothes	0.5	Sequential
Dry Clothes	0.5	Sequential
Put Shopping Away	1.0	Any Order
Unpack Shopping	1.0	Any Order
Prepare Meal	1.0	Any Order
Make Chicken Curry	0.33	Sequential
Make Fish & Chips	0.33	Sequential
Warm up Meal	0.33	Sequential
Clean up Kitchen	1.0	Any Order
Clean Dishes	0.5	Unordered
Dish wash Dishes	0.5	Sequential

Table 16 - Surprise threshold and execution order of the ADL/Sub Activities

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 then that means the ADL has not been detected correctly. The semantics of the execution order of the ADL has a significant effect on the appropriate level for the surprise threshold for each ADL. For example, the ADL "*Make Breakfast*" has a sequential execution order and has surprise index threshold of 1. While carrying out the experiment if we get a surprise that is below 1 then it is highly likely that the ADL has been detected successfully. However, if the surprise index is 1 or greater then this means the correct ADL has not been identified. This is the reason why the detection rates for these experiments are based on the surprise index of each ADL. Table 16 shows the surprise threshold for each ADLs and sub activities that have been used in the experiment.

These 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 (objects) where there is usually a determining object used for each ADL. For example the "*fairy bottle*" sensor is exclusive to the task "*washing up dishes*", which makes it a distinctive sensor event that 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 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 17 shows the objective of each experiment conducted.

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 interwoven.
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 interwoven.

Table 17 – ADL recognition experiment objectives

This experiment is modelled around 5 ADLs, which consist of 25 tasks and 45 sensor events, Figure 38 shows the ADLs with their associated tasks that have been used for the experiments.

More ADLs have been modelled as plans in Asbru intentionally, 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, TASE Segmentation and the ADL Recogniser.

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

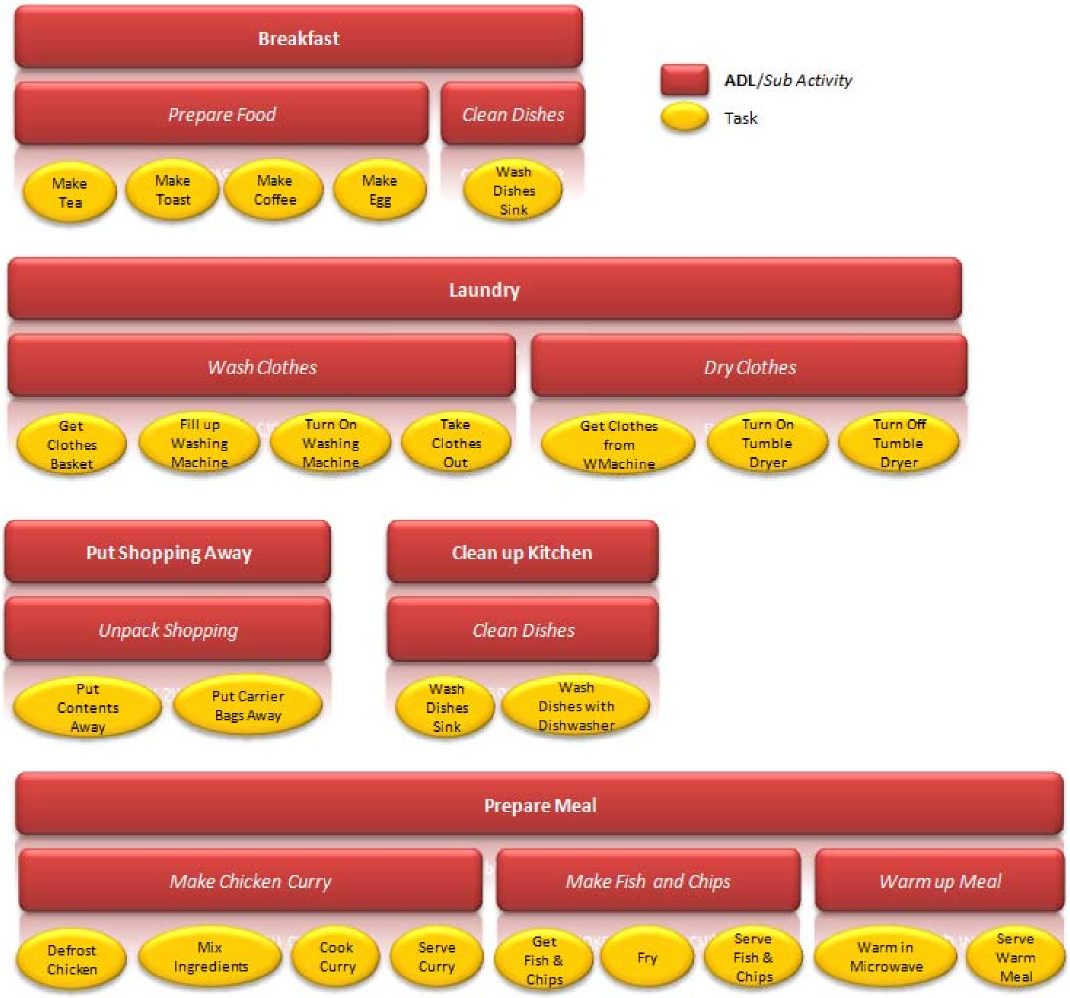


Figure 38 - Modelled ADLS for the TASE-ADL experiments

The results of the experiments carried out with the set of distinctive sensor events (Table 18) 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 up 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 the 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.

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

Table 18 - Distinctive series results for experiments 1, 3, and 5

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

Table 19 - Non-distinctive series results for experiments 2, 4, and 6

The results from non-distinctive experiments (Table 19) 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 39 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.

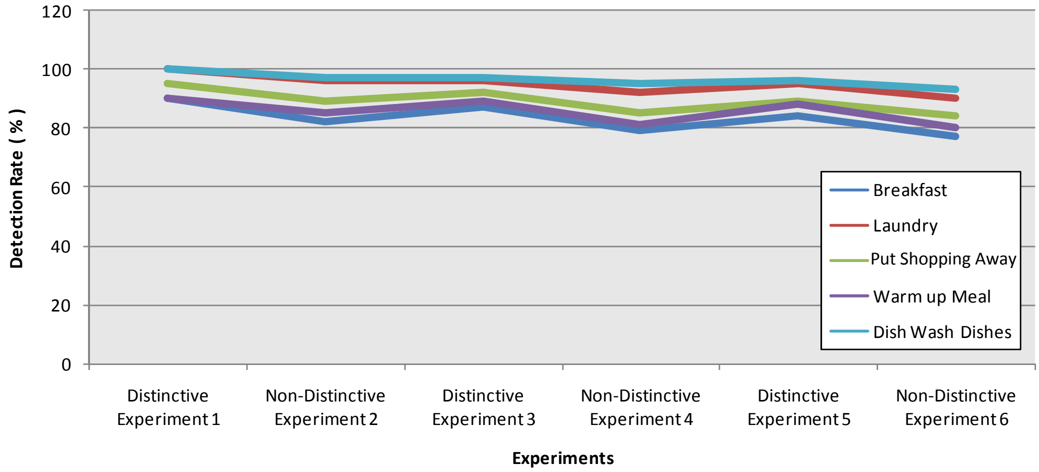


Figure 39 – Results comparison of ADLs that are reliant on distinct objects

The reason why this 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 and to detect 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. 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 dealing with the missing sensor events at the lower tier, the text based segmentation provided all the possible task associations for the sensor event.

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. In order to overcome this limitation, further on in the thesis the higher tier of the recognition approach will incorporate task (and goal) durations. Also since the detection system will be aware of the time of day it will prefer the ADL plans that are usually executed around that time. In addition timing plays an important part in the lower tier of the hierarchal approach, as time can be used to measure how long it takes on receiving different type of sensor events.

3.4.4 Generating Alternative Task Sequences

In the previous section of this chapter it was shown that task associated sensor events could be segmented into appropriate tasks using a range of techniques. The difference between this approach and the statistical approach used in the previous section is that this approach employs a simple algorithm which works out the all possible combinations for each task given the sensor event. While this may seem computationally expensive when performed, however a best first identification in synchronization with the plan recognition it could provide a simple but effective approach, especially as each task is not associated with a large number of distinct sensor events.

The tasks will be denoted by characters for simplicity. For example;

Task(Make Tea)	=	Character (A)	
Task (Make Coffee)	=	Character (B)	
Task (Make Toast)	=	Character (C)	Make Tea or Make Coffee is denoted by A+B
…	…	…	
…	…	…	

For each task (a) and sensor event (b), we assigned a probability $P[\ a \mid b]$. These are assigned as prior probabilities, as no training is assumed. However, using identification from the higher tier it is possible to estimate the probability proportions of $P[\ a \mid b]$. The feedback approach is explained further in the following chapters.

3.4.4.1 Enumeration of Task Sequences with a Scenario

The tasks that can potentially be carried out in this example are *Make Tea*, *Make Coffee*, *Make Toast* and *Make Egg*.

Make Tea maps into the following sensor events:

e_1 =Kettle, e_2 =Sugar Bowl, e_3 =Fridge, e_4 =Tea Bag Bowl, e_5 =Coffee Bowl, e_6 =Food Cupboard, e_7 =Toaster, e_8 =Gas Cooker and e_9 =Frying Pan.

While tasks that can potentially be carried out in this example are *Make Tea*, *Make Coffee*, *Make Toast* and *Make Egg*.

Object	Sensor Event	Task Associated with Sensor Event	Character Representation
Kettle	e_1	Make Tea, Make Coffee	A+B
Sugar Bowl	e_2	Make Tea, Make Coffee	A+B
Fridge	e_3	Make Tea, Make Coffee, Make Toast, Make Egg	A+B+C+D
Tea Bag Bowl	e_4	Make Tea	A
Coffee Bowl	e_5	Make Coffee	B
Food Cupboard	e_6	Make Toast, Make Egg	C+D
Toaster	e_7	Make Toast	C
Gas Cooker	e_8	Make Egg	D
Frying Pan	e_9	Make Egg	D

Table 20 - Task associated sensor events and its character representation

These sensor events are associated with its respective tasks (Table 20) and each task is then assigned prior probabilities given the sensor event (Table 21). The probability values were based on the number of associations each task has with the sensor event.

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9
Make Tea (A)	0.5	0.5	0.25	1	0	0	0	0	0
Make Coffee (B)	0.5	0.5	0.25	0	1	0	0	0	0
Make Toast (C)	0	0	0.25	0	0	0.5	1	0	0
Make Egg (D)	0	0	0.25	0	0	0.5	0	1	1

Table 21 - Prior probability distribution for each task given the sensor event

Given the following sensor events were detected e_1, e_2, e_3, and e_4, the stream of possibilities will be as follows:

(A+B)(A+B)(ABCD) A = (AA+AB +BA+BB) (A+B+C+D) A

Now if A cannot be repeated then AA can be simplified to A. The probability to associate with this reduced A is taken as the maximum of the two conditional probabilities. For some tasks the reduction of AA to one instance of the task A may be inappropriate. For example, if A is the task of adding a spoonful of sugar, then it may well be sensible to have repeated occurrences. It is expected that such cases are relatively rare for the applications in mind, and even when they do occur the plan may often be indeterminate in the number of repetitions, and so it is not an important issue. However, such considerations are a result of the characteristics of the task, which can be presumed to be known, and can be modelled. We will assume here that AA can be reduced to A.

So;

(A+B)(A+B)(ABCD) A

= (A+AB +BA +B) (A+B+C+D) A

= (A+ABA+BA+AB+BAB +B+AC+ABC+BAC+BC +AD+ABD+BAD+BD) A

= A+ABA+ACA+ADA+ABA+ABCA+ABDA+BA+BABA+BACA+BADA+BDA

This gives the possible segmentations in disjunctive normal form, where each possible task sequence is separated by a +. The cost of each conjunct is computed using product of the probabilities.

When there are repetitions of the same task in immediate sequence then the maximum of the probabilities is used rather than the product. This is easily done by computing the product as a tree and remembering the maximum probability of each task in the leaf of the tree.

Task stream =
A+ABA+ACA+ADA+ABA+ABCA+ABDA+BA+BABA+BACA+BADA+BDA

e_1		e_2		e_3				e_4
A +	B	A +	B	A +	B +	C +	D	A
0.5	**0.5**	0.5	0.5	0.25	0.25	**0.25**	**0.25**	**1**

Table 22 - Conditional probability values for each task given the 1st four events

For example:

After two events (Table 22) we have the possibilities (A+B) (A+B) = (AA+AB +BA+BB) here each of the A tasks came from a different event. When AA is reduced to A then the maximum of the two proportions is taken. Here they are both 0.5, so 0.5 is taken. The cost of AB is 0.25.

Table 23 shows the possible tasks sequences and their associated cost for the detected sensor events e_1, e_2, e_3, and e_4:

A+ABA+ACA+ADA+ABCA+ABDA+BA+BABA+BACA+BADA+BDA

Task Stream	Sequence				Cost
1	A				
	1				1
2	A	B	A		
	1	0.5	1		0.5
3	A	C	A		
	1	0.25	1		0.25
4	A	D	A		
	1	0.25			0.25
5	A	B	C	A	
	1	0.5	0.25	1	0.125
6	A	B	D	A	
	1	0.5	0.25	1	0.125
7	B	A			
	0.5	1			0.5
8	B	A	B	A	
	0.5	1	0.5	1	0.25
9	B	A	C	A	
	0.5	1	0.25	1	0.125
10	B	A	D	A	
	0.5	1	0.25	1	0.125
11	B	D	A		
	0.5	0.25	1		0.125

Table 23 - Associated cost for task streams

As each event is input the products can be computed, and the disjunctive form found. To avoid a computational explosion, the products can be computed in a best first manner (keeping a solution front of a given size) and each of these fed to the ADL recognition system and the compliance (as measured by discrepancies and surprise) established.

To help identify the ADLs each conjunct of tasks has a cost associated with it and a sequence of ADLs, each with a surprise index associated with each ADL. In the case where ADLs are not interwoven when another ADL is added to the sequence of ADLs the completion surprise will have been computed for the ADL that has finished. If ADLs can be interwoven there can be a large number of possible ADL sequences. To reduce the unnecessary complexity on this possible approach is to consider certain ADLs such as "*Answer the Door*" as being interruption type ADLs, and only allow interweaving of an ADL with an interruption type ADL.

3.4.4.2 Task Recognition Experiments and Results

The objective of this experiment was to work out the accuracy rate of the enumeration approach against the text segmentation approach for task recognition. The same sensor data (mixture of distinctive and non-distinctive objects) and ADLs/tasks from the activity recognition experiment conducted in section 3.4.3.2 have been used to determine the precision rate for both approaches. The precision rate is determined by the following:

$$precision = \frac{\left|\{relevant\_tasks\} \cap \{segmented\_tasks\}\right|}{\left|\{segmented\_tasks\}\right|} \tag{5}$$

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. On the other hand the recall rate will be determined by the following formula:

$$recall = \frac{\left|\{relevant\_tasks\} \cap \{segmented\_tasks\}\right|}{\left|relevant\_tasks\right|} \tag{6}$$

The results that will be returned by the recall formula will always be trivial to achieve a rate of 100% as it will always return all the relevant tasks in response to a relevant ADL being carried out, e.g. there may be 5 relevant tasks from 10 tasks that were actually segmented, so the recall for this will 5/5, i.e.100%. Hence, the precision rate is of more importance as it measures relevant task as opposed to the non relevant tasks from the segmented data.

$$\det ection = \frac{\left|\{\det ected\_tasks\} \cap \{\exp ected\_tasks\}\right|}{\left|\exp ected\_tasks\right|} \tag{7}$$

The overall detection rate of the tasks on this occasion has been determined by the (7), where the detected tasks are tasks that have been correctly detected and are relevant to the ADL that is being conducted. While the expected tasks are the number of tasks that are expected to be conducted within the ADLs, this detection rate is validated with ground truth data collected when the experiments where being initially conducted.

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. For example, an ADL like "*Put Shopping Away*" has two constituent tasks but if the approach manages to segment out at least one task correctly that belongs to this ADL, then it makes it possible to calculate the precision rate. However, as detection is also another important aspect of this work, the calculation of the detection rate of this approach is based on the number of tasks which belong to an ADL. The results of this approach have been compared with the text segmentation method. 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. These set of results were also compared with the text segmentation approach. Note that the detection rates of the text segmentation approach in these experiments are different to the results in the previous section, as the detection rate for this instance has been calculated with (7). Results for both experiments are in Table 24 and Table 25.

ADLs	Enumeration Segmentation			Text Segmentation		
	Precision [%]	Recall [%]	Detection [%]	Precision [%]	Recall [%]	Detection [%]
Breakfast	100	100	89	96	100	90
Laundry	100	100	95	90	100	97
Put Shopping Away	90	90	45	90	90	45
Prepare Meal	30	30	35	22.5	30	40
Clean up Kitchen	100	100	90	100	100	92

Table 24 - Results for task relevance experiment

Table 24 indicates that the precision rate for 'enumeration segmentation' approach is slightly greater than the 'text segmentation' approach. The reason for this is because the 'enumeration segmentation' approach only takes into account the segmented task with the highest product, which is always determined by a distinct object, therefore the precision rate is high. While the 'text segmentation' approach takes into account segments that are determined by distinct and non-distinct object that are associated with the task, which can sometimes have tasks in those segments that are not totally relevant to the ADL being carried out.

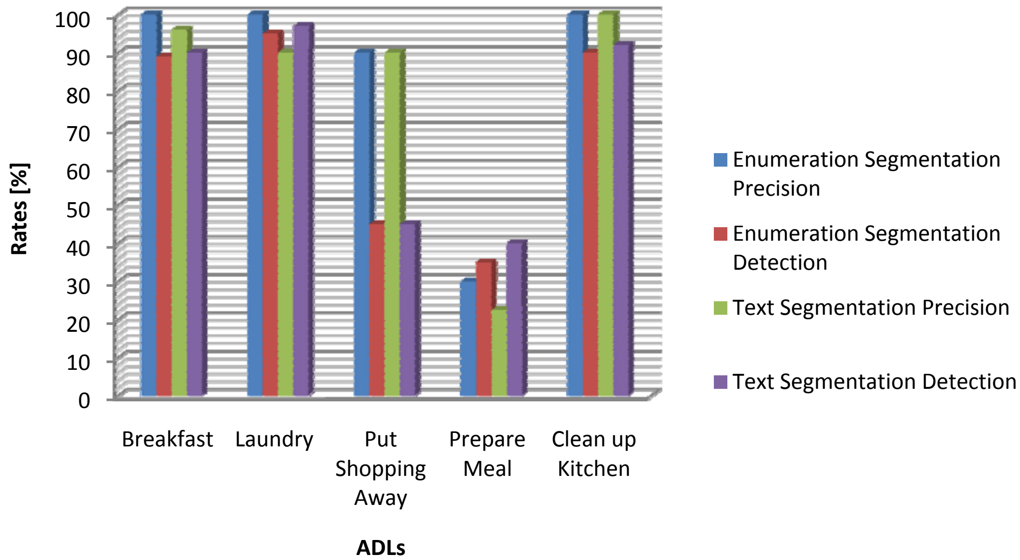


Figure 40 - Task relevance results graph

There was an insignificant difference in the detection rates using both approaches, as both were able to detect tasks which were relevant to the actual ADL with success. This is because both approaches were able to take into consideration a variety of segments that may correspond to the task that belongs to the ADL being conducted.

It is evident from Figure 40 that an ADL like "*Prepare Meal*" generally had a poor precision rate with both approaches, this is due to the lack of distinct objects being used while the ADL was being conducted.

The objective of the second set of experiments was more focused on seeing how both approaches deal with multiple ADLs and tasks. Therefore even if one of many tasks is detected that helps determine between the two ADL this is seen as a positive result. Also for this particular type of experiment, the object data (collected sensor events) that has been used does consist of considerable amount of distinct objects, as the main focus is to see how both approaches fair against mixture ordering of task and activity traces. What can be learnt from Table 25 is that both approaches have over 80% detection rate for multiple ADLs, as multiple ordering of tasks does not influence both of the approaches. Again as expected the precision rate of the 'enumeration segmentation' is higher than 'text segmentation' approach, due to the 'enumeration segmentation' being able to segment the relevant distinct object related task, rather than segmenting all the possible tasks which are associated with object.

	Enumeration Segmentation			Text Segmentation		
	Precision [%]	Recall [%]	Detection [%]	Precision [%]	Recall [%]	Detection [%]
Multiple ADLs	100	100	85	86	100	89

Table 25 - Results for multiple ADL recognition

In relation to these experiment results, the 'enumeration segmentation' is an approach that always has a high precision rate, as this method is able to filter out the correct task that is being conducted given that the task trace has an distinct object used. In relation to distinct objects being used, some tasks can have more than one distinctive object, which can be useful for activity recognition via object usage. Another positive aspect of this approach is that ordering of tasks and objects does not make an influence on the detection rate, as this approach is able to deal with multiple tasks and ADLs being conducted at the same time. The 'enumeration segmentation' approach does have the capability to take into consideration the other segments which have a product of less than 1. However, for this set of experiments in this chapter we have intentionally just concentrated on segmented tasks with a product of 1 in order to compare the precision of both task segmentation approaches. In the following chapter, this task recognition approach will work in conjunction with higher tier ADL recognition, where the other segments (which are less than 1) are taken into consideration in order to generate associated costs for ADL sets. The 'enumeration approach' mitigates the chances of not being able to recognise tasks that have been carried out via different variations, as it takes into consideration all the possible types of task sequences given the task associated sensor events. Unlike other task recognition approaches such Hidden Markov Models (HMM), the ordering of the objects does not influence the recognition process, as the enumeration approach does not rely on the transition value between the previous objects. It does rely on effective pruning (or delayed evaluation) of unlikely candidate task sequences.

Chapter 4

Interaction between Levels

In the previous chapter ADL recognition and task recognition methods were described separately and experimental results that measured task recognition rates and ADL recognition rates were given. This chapter will describe an approach that enhances the ADL recognition process, by exploiting the information from models at both levels. In addition to this, this chapter will describe a decision tree-based approach that uses ADLs in partially recognised plans to support the prediction of future ADLs, even in the case where the set of plans is not complete.

In addition to being able to carry out activity/task recognition the work described in this chapter has the following capabilities:

- The use of timing constraints, expressed as intervals, enhances the pruning process when trying to identify which ADL is being carried out at a particular time. For example, it is highly likely that an activity such as 'Make Lunch' will not be conducted at 8.00am.
- Determine the future intentions of the subject, from the individual ADLs and from the overall ADL schedules that are specifically modelled for Alzheimer's patients. (The schedule is in fact a simple sequential plan at the top level).
- Track interweaving between tasks and activities, as a person can conduct two activities at one time (e.g. putting away shopping while making the

tea). Also there can be occasions when a person is conducting an activity and has to stop the current activity to perform another one (e.g. answering the door or phone). These activities can be referred to as interruption ADLs.

- Recognise a task from different sequences of sensor event data, where each sequence has different orderings of how the objects are used when a task is being performed. Also to recognise tasks from data where there is noise in the data (e.g. missing or unrelated sensor events) based on how well the identified tasks match an ADL.

4.1 Interaction between Tiers

This section will describe how the enumeration task recognition approach is merged with high tier activity recognition. In principle, any of the lower tier task recognition approaches could be used. The merging between the tiers is formed by generating the costs for task sequences based on the lower tier task recognition process. These costs are then used to generate ADL set utilities, which are used to determine the activity that is currently being conducted in the higher tier recognition process.

4.1.1 Associated Costs for Alternative Task Sequences

Task recognisers were described in Chapter 3. One of them being the enumeration approach, which is used to output ordered lists of alternative task sequences given an input set of events. Each of these task sequences 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 non-compliance of the task sequence with the event sequence and the relative frequencies of ADLs. A heuristic approach was used to compute the calculations for each task sequence, where the 'probability' was taken into consideration. The function of the low level task recogniser can be represented as:

$$e1, e2, \ldots en \rightarrow \{<TS1, c1> + <TS2, c2> + <TSm, cm>\} \tag{8}$$

where *TS* represents a task sequence. *m* is a parameter chosen when the task recogniser is asked for its set of task sequences that match the events. *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 a list of possible task sequences, ABC and ABD, might be generated where A, B, C and D are tasks. ABC will have an associate cost and so will ABD. The set of alternatives (and mutually exclusive) task sequences and their costs will be represented as {<ABC, c1> + <ABD, c2>}.

The list of possible task sequences 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 have a prescribed upper-bound, but the task recogniser has the capability of generating more tasks sequences (if there are any) should the higher tier 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 $\{<TS_{m+1}, c_{m+1}> + \ldots\ldots <TS_{m+n}, c_{m+n}>\}$ task sequences.

As has been shown in Chapter 3, this functionality of generating task sequences can be achieved in several ways. Each method has different parameters, such as transition probabilities and confusion matrices. Feedback from the higher layer can be used to modify some of these parameters, so that the task recogniser is a better model of the current event stream and can re-assess the task sequences based on the feedback.

As a new event arrives the task recogniser is invoked computing a new set of task sequences. Making the output of this new invocation relate to the previous is a function of the task recogniser. It recognises when computing the cost function associated with a task sequence that the more recent tasks are more important. Typically this is handled by exponential weighting of costs, where the match of the tasks to the more recent events is given more weight.

4.1.2 Generating ADL sets

An ADL set is a group of probable ADLs that are generated given the task sequences that have been recognised and generated by the lower tier. The utility of each of these ADL sets is based on the cost of each task sequence. By generating the associated utility for ADL sets, this will provide the high tier ADL recognition component with an accurate task sequence as input, as well as forming the initial interaction between the higher tier and lower tier. The utility function itself is based on the 'degree of match' between generated ADLs within the ADL sets and the modelled ADLs. Each ADL set consists of a set of some incomplete ADLs and its complete predecessors. If there are no interwoven tasks then the ADL set will consist of a sequence of abutting ADLs, a *single* 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. ρ_i is the utility of the ADL set.

$$t1, t2, \ldots tn \rightarrow \{<ADLS1, \rho_1> + <ADLS2, \rho_2> + <ADLSm, \rho_m>\} \quad (9)$$

Again the utility function should give a higher weight to discrepancies/surprise levels in the more recent ADLs. + is interpreted as or.

There is an issue as to which events are used. The first option could be to use all events. However, this could be very inefficient as only the most recent events are of interest. The other option is to keep a sliding window of events. However, this raises another issue as to where the sliding window should start from. A sensible approach is to ensure that the 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, there can be a situation where ADLs are interwoven, which leads to a large number of events. Also it is possible that a task sequence may have different interpretations in terms of ADLs, and these may mean that for the same task sequence a different length and order of tasks need to be remembered, and hence a different number of events. Therefore a better option is to define a window of tasks and hence events for each ADL set under consideration. Of course there is a chicken and egg problem here. We cannot choose the event sequence unless we have an ADL set, but we cannot get an ADL set, without using an event sequence. The task sequences are only ever hypothesised. They can only be confirmed after checking with the ADL recognition component.

The output of the ADL recognition component is, for each task sequence considered, a list of active ADLs and any immediate predecessors of each active ADL.

This component is able to utilise temporal information, specifically the time intervals associated with each ADL and temporal information across ADLs, namely the location of the ADL within the schedule of ADLs.

4.1.2.1 ADL Schedules

Morning	Afternoon	Evening
• Wash, brush teeth, get dressed • Prepare and eat breakfast • Discuss the newspaper or reminisce about old photos • Take a break, have some quiet time	• Prepare and eat lunch, read mail, wash dishes • Listen to music or do a crossword puzzle • Take a walk	• Prepare and eat dinner • Play cards, watch a movie or give a massage • Take a bath, get ready for bed

Table 26 - Daily Activity Plans constructed by Alzheimer's Association

In a real life scenario, the instantiation of the ADLs will be different depending on the individual who is being monitored, therefore in order to achieve reliable modelling the ADLs modelled in this thesis are based on planned activity examples constructed by the Alzheimer's Association for people with dementia (Table 26). The objective of the association is to help people with dementia by providing assistance via a caregiver in order to organise the person's day. The organised day consists of activities that are modelled to meet each individual's preference, as well as that the objective of these planned activities is to enhance the individual's self esteem and improve quality of life by providing them with purpose and meaning to their life [135].

All the ADL plans are held in a library (The question about completeness of the library is addressed later). Generating the utility of ADL sets given all possible ADLs in the library at the time of an activity taking place can be difficult process. In order to overcome this, the utilities are based on ADL schedules within a certain time frame (e.g. 9.00am to 10.00am), as this is more manageable and provides accurate recognition. This also allows the high tier ADL recognition component to only consider the ADL schedule within a given time frame, e.g. 9.00am to 10.00am, hence eliminating some of the unlikely possibilities at the very outset of the recognition process. However interruption ADLs (e.g. answer phone) can occur while any ADL is being carried out (by definition). This is made possible as the representation language 'Asbru' can model timing intervals between ADLs.

4.1.2.2 ADL Sets with Associated Utilities

Before generating ADL sets with their associated utilities, the cost of each task sequences needs to be determined within the lower tier. The following example will first work out the cost of each task sequence, from which the ADL sets will be generated, each with an associated utility.

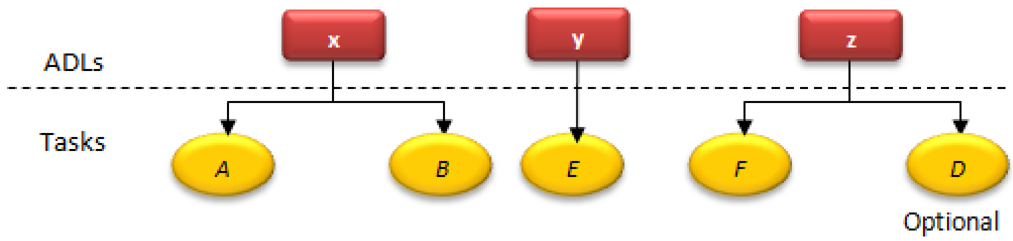


Figure 41 - Example of an ADL Schedule (9.00am to 9.15am)

Given the ADL schedule in Figure 41, suppose the following sensor events are detected:

e_1 , e_2 , e_3 , e_4 , e_5 , e_6 , e_7

Table 27 shows these events being mapped as task associated disjunctions and given probability values, with the highest value for each letter highlighted in bold.

e_1	e_2	e_3	e_4	e_5	e_6	e_7
A+B	A	B	C+E+F	C+E	E	F
0.5 0.5	**1**	**1**	**0.33** 0.33 0.33	0.5 0.5	**1**	**1**

Table 27 - Probability values for each task given the sensor events

Based on the task associations in Table 27, the following task sequences are generated:

= (A+B)(A)(B)(C+E+F)(C+E)(E)(F)
= <u>ABCEF+ABECEF+ABEF+ABFCEF+ABFEF</u>
<u>+BABCEF+BABECEF+BABEF+ BABFCEF+BABFEF</u>

The letters in these task streams are then reassigned their initial highest probability values, which helps determine the cost for each task sequence (Table 28).

Task Stream	Sequence							Cost
1	A	B	C	E	F			
	1	1	0.5	1	1			0.5
2	A	B	E	C	E	F		
	1	1	1	0.5	1	1		0.5
3	A	B	E	F				
	1	1	1	1				1
4	A	B	F	C	E	F		
	1	1	1	0.5	1	1		0.5
5	A	B	F	E	F			
	1	1	1	1	1			1
6	B	A	B	C	E	F		
	1	1	1	0.5	1	1		0.5
7	B	A	B	E	C	E	F	
	1	1	1	1	0.5	1	1	0.5
8	B	A	B	E	F			
	1	1	1	1	1			1
9	B	A	B	F	C	E	F	
	1	1	1	1	0.5	1	1	0.5
10	B	A	B	F	E	F		
	1	1	1	1	1	1		1

Table 28 - Task sequences and associated costs

The utility of an ADL is relative to the number of constituent tasks. We define a base utility and the total utility is the sum of the base utility and the number of tasks matched. The base utility is allocated initially. The base utility of an ADL is determined by how many mandatory tasks each ADL consists of, which means an ADL that consists of two necessary tasks has a base utility of 2. However, optional tasks are not counted when calculating the base utility. For example in Figure 42, ADL 'z' has two tasks, but one of them is optional, therefore the initial utility is 1. Any ADLs outside the schedule window has an initial minimum utility of 0 regardless.

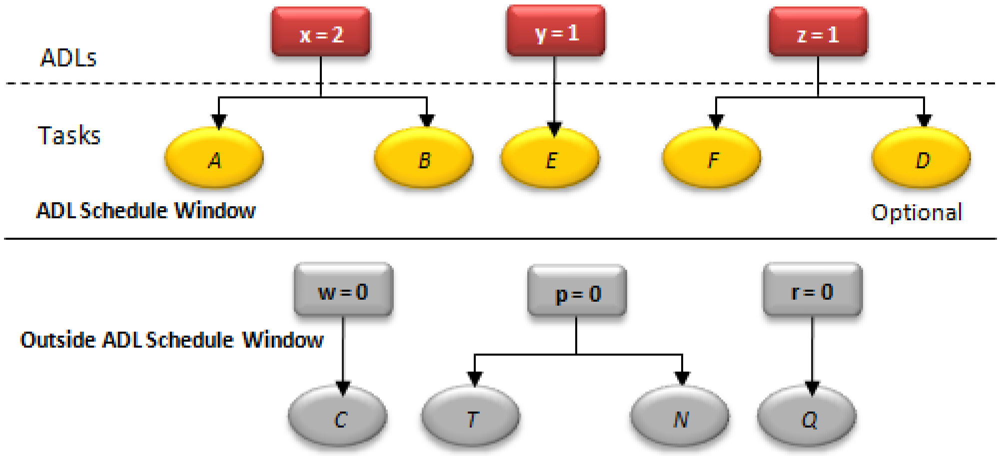


Figure 42 - ADL schedule with minimum utilities

Once the base utility for the ADLs in the schedule window and the generated segments of the task sequences are known then the following is carried out for each task sequence.

Example continued: Task Stream 1 and 3 from Table 28 are used to illustrate how the associated utility of an ADL set is generated.

Task Stream 1: A=1, B=1, C=0.5, E=1, F=1

Within Task Stream 1 the first task is 'A', within the current schedule window 'A' belongs to ADL 'x'. The revised utility of ADL 'x' is computed in the following way:

Let t =1 if the hypothesised task is a mandatory task in the ADL to be matched, 0 otherwise.

Task 'A' is indeed one task within 'x', hence $t = 1$

Base utility of ADL 'x' is 2,

Cumulative utility of ADL 'x' after one task $= t + base$ =1+2

Task 'B' is the next task in the stream of tasks.

Task 'B' is a task within 'x', therefore $t = 1$

As 'B' is a task which belongs to the same ADL as the previous task 'A' the current utility of the ADL set is simply incremented with the value of t.

Cumulative utility of ADL 'x' after two tasks= 3+1 = 4

The next task in the task stream is 'C', this task is within ADL 'w', therefore t= 1. However 'w' does not belong to the current schedule window, which means the base utility of ADL 'w' is 0.

The cumulative utility of ADL 'w' after three events is 1+0 = 1

'w' is a new ADL and so the cost of ADL 'x' and ADL 'w' have to be combined to get the aggregate cost of the ADL set 'xw'.

The utility of this ADL set 'xw' is set equal to $4\alpha + 1$, here α is a discount factor.

The next task is 'E', this task is within 'y' ADL (t= 1) and this is also within the current ADL schedule window. Therefore the utility of ADL 'y' is 1+1.

The utility of this ADL set 'xwy' is now equal to $4\alpha^2 + 1\alpha + 2$.

The final task in this task sequence is 'F', this task is within 'z' ADL and also within the current ADL schedule window, and so the minimum utility of ADL 'z' is 1+1, as one of the tasks is optional.

The total utility for the ADL set 'xwyz' based on task sequence 1 is $4\alpha^4 + 1\alpha^2 + 2\alpha + 2$.

This total utility is now to be compared with the other total utility of the ADL set that is generated by task sequence 3.

Task Stream 3: A=1, B=1, E=1, F=1

First task is 'A' belongs to ADL 'x'.

t= 1. Minimum utility of ADL 'x' is 2.

Hence the utility of task stream 'x' after 1st task=3. This is then followed by task 'B'. Again as 'B' is a task which belongs to the same ADL as the previous task 'A', rather than using the minimum utility of 'x', the current associated cost of the ADL set is incremented with the value of t, which means the cumulative utility of this ADL set is currently 4.

The next task is 'E', this task is within 'y' ADL and this again is also within the current ADL schedule window. Since the base utility of ADL 'y' is 1, Hence, the cumulative discounted cost of the ADL set 'xy' is now equal to $4\alpha+2$.

The final task in this task sequence 3 is 'F', which is within 'z' ADL and within the current ADL schedule window, which means the minimum utility of ADL 'z' is 1 and the cumulative discounted utility for this ADL set based on task sequence 3 is $4\alpha^2+2\alpha+2$.

In comparison to task stream 1($4\alpha^4+1\alpha^2+2\alpha+2$), the associated utility of the ADL set which is based on task stream 3 ($4\alpha^2+2\alpha+2$) is greater if, $\alpha<\frac{\sqrt{3}}{2}$.

These associated utilities suggest that the ADL set based on task stream 3 is the correct set, as all the tasks have been identified correctly and match the current ADL schedule window.

This process is carried out for all the task sequences. The ADL set with highest utility is taken and its corresponding task sequence is used as input for the higher tier ADL recognition component.

4.1.3 ADL Prediction

The high level ADL recognition component returns the name, discrepancy count and surprise index value of the ADL that it is examining. In the system constructed, plans are at two levels only. The top level schedule is a plan written using Asbru, and the schedule consists of plans (ADLs) that can be to any level of nesting. The recogniser is able to recognise the high level ADLs that are children of the schedule, as well as sub-activities that are nested within the ADLs. In addition the system also returns the name of the ADL that is currently scheduled to be conducted next in that particular ADL schedule. This information is useful, as it can be used to adjust from prior to posterior probabilities for the next iteration of task recognition. For example, if it has been discovered that "*Make Tea*" has been detected, which is belongs to the ADL "*Make Breakfast*" and the next task in the schedule is "*Read Newspaper*", which belongs to ADL "*Reading*" then probability of using objects associated with the task *"Read Newspaper"* have a probability weighting greater than other objects. So from a technical recognition perspective prediction of possible of tasks can provide useful mechanisms to adjust, e.g. the transition probabilities in a HMM model, allowing more efficient recognition. Also this prediction of the ADLs can also enable other services around the home to be more tailor-made to each elderly person, e.g. turn on music in room after eating lunch.

This type of prediction is supported by looking at the schedule of ADLs. In prescribed daily activity plans this can be seen as a single top level plan, which has a sequence of ADLs. The experiments conducted were based around ADLs modelled on the lines of typical daily activity schedules constructed by the Alzheimer's society.

Each of the ADLs in the sequence is called a coordinate. The depth of the task or sub-activity in the top level sequence is another coordinate. Figure 43 shows an example of an ADL schedule the coordinates of task 'A' and task 'B' are both 'A3'.

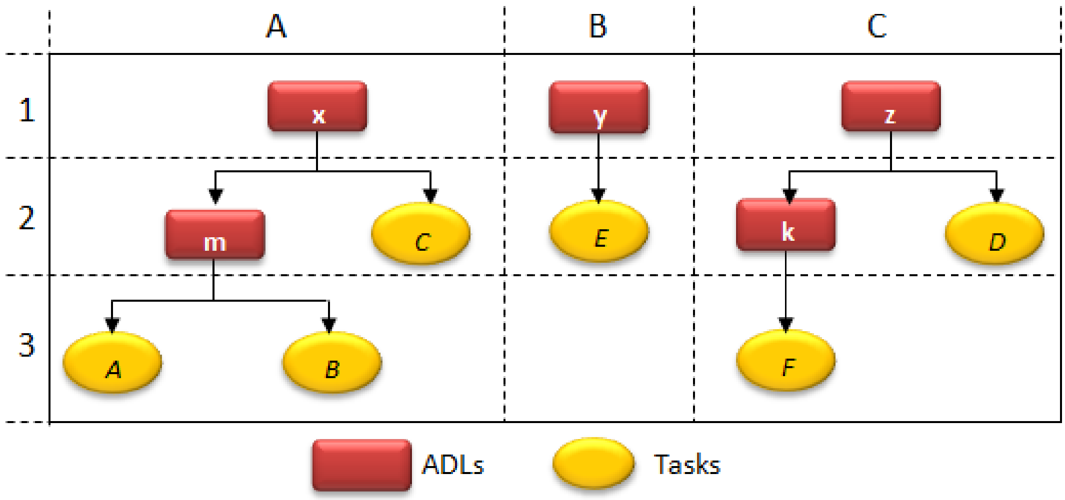


Figure 43 - Co-ordinate based mapping within ADL schedules

Once the location of the currently executed ADL is determined, the co-ordinates allows the high level recognition component to make a judgement on which ADL will probably be conducted next based on the co-ordinate that follows the current co-ordinate position. This does not mean that the search within the ADL schedules is restricted, it simply orders the search, and allow the tracking of which ADL, sub-activity or task is being executed and which one might be executed next.

However before considering another co-ordinate a search is performed to see if any other task exists that belongs to ADL ‘m’ that has a co-ordinate value of A2. When task ‘A’ has finished task ‘B’ should execute, hence the next likely co-ordinate remains as A3. This search is carried out each time an ADL and its executed task has been detected. However, if task ‘C’ has been executed then the next probable ADL is ‘y’ which is executed by task ‘E’, as the co-ordinate for task ‘C’ is A2, which means the next likely co-ordinate is B1.

4.1.4 ADL Recognition Experiments and Results

Experiment Number	Type of Experiments
1	Each subject carries out the tasks in Table 31 separately, but the task has to be conducted using the constituent objects in any order. For example, making a cup of tea can be conducted with a different ordering of objects used, e.g. 1st Kettle, 2nd Sugar Bowl. The objective is to see how well the enumeration approach is in recognising tasks performed with different orderings of objects. Note that there is no concept of ADL here, as the objective is to solely focus accurate lower tier task recognition.
2	Subjects are asked to carry out multiple tasks and activities simultaneously. This is to see how well the approach can recognise tasks and activities that are interweaved.
3	Subjects carry out ADLs as normal, however the streams of object data collected are then modified, by deleting segments of object data from the streams. This is to see how well the approach performs when a stream of object data is incomplete.
4	Subjects perform two ADLs at a time, which are captured in two separate steams of object data. The first stream is then used to determine which ADL is being conducted in the second stream of data. This experiment is to see how well the approach discovers the intentions of the subject.
5	Subjects are asked to perform an ADL, however they are also asked to conduct an interruption ADL (Table 30) simultaneously. The objective of this experiment is to see how well the high level activity recognition component is capable of managing two concurrent ADLs that have nothing in common, e.g. “Preparing Food” and “Answering the Door”.

Table 29 - Experiment objectives for complete HADL approach

One of the key objectives of these experiments is to see how well combining the lower and higher level recognition steps works. As this approach also has other capabilities such as tracking interweaving between activities five sets of experiments (Table 29) have been conducted to illustrate the identification potential.

Each subject carried out the ADL schedules in Table 30. The schedules were carried out in separate segments, which consisted of one or more ADLs within a certain time frame. For example, subject 1 conducted ADL "*Morning Refresh*" between 9.00am and 9.30, hence the object data for this was stored with a timestamp. This would lead to each subject having 9 sets of object data each with different time frames of the day per experiment.

Morning Schedule
• Morning Refresh
o Oral Cleaning
o Clean Face
• Prepare and Eat Breakfast
o Prepare Food
o Consume Food
o Cleaning
• Relaxing Leisure
o Reading
Afternoon Schedule
• Prepare and Eat Lunch
o Make Fish and Chips
o Warm Up Meal
o Drink Water
• Prepare and Eat Snack
o Prepare Snack
o Consume Snack
• Daily Puzzle
o Complete Crossword
Evening Schedule
• Prepare Dinner
o Make Chicken Curry
o Drinking
• Laundry
o Wash Clothes
• Dry Cleaning
o Ironing
Interruption ADLs (are modelled in within all schedules)
• Phone Call
• Someone at the Door
• Going to W/C

Table 30 - ADL schedules used for activity recognition experiments

Note that the ADL schedules in Table 30 consisted of 9 ADLs, however each ADL (●) consisted of multiple sub-activities (○) and these would be made up of tasks, which are shown in Table 31.

Task List	
1 - Brush Teeth	17 - Meal in Microwave
2 - Wash Face	18 - Serve Warm Meal
3 - Make Tea	19 - Having a Drink
4 - Make Toast	20 - Eat Biscuits
5 - Have Cereal	21 - Doing Crossword
6 - Make Egg	22 - Defrost Chicken
7 - Make Coffee	23 - Mix Ingredients
8 - Eat and Drink	24 - Cook Curry
9 - Wash Dishes	25- Serve Curry
10 - View Photo Album	26 - Gather Clothes
11 - Read Newspaper	27 - Use Washing Machine
12 - Read Book	28 - Iron Clothes
13 - Read Mail	29 - Answer Phone
14 - Get Fish and Chips	30 - End Phone Call
15 – Frying	31 - Attend Front door
16 - Serve Fish and Chips	32 - Close Front door

Table 31 - Corresponding tasks for modelled ADL schedules

The overall recognition rate of the tasks and ADLs has been determined by (10), where the recognised tasks/ADLs are tasks and ADLs (including nested sub-activities) that have been correctly recognised given the stream of object usage data. The expected tasks and ADLs are the number of tasks and ADLs that are expected to be conducted, which is based on the collected ground truth data.

$$recognition = \frac{\left|\{recognised\_tasks/ADL\} \cap \{\exp ected\_tasks/ADL\}\right|}{\left|\exp ected\_tasks/ADL\right|} \tag{10}$$

The results for experiment 1 (Table 32) show that the enumeration approach was effective in recognising single tasks that were conducted with streams of object data that had a variation in the order when the objects were used. As mentioned in the previous chapter, unlike approaches like Hidden Markov Models this approach does not rely on a particular ordering of usage of the objects when conducting an activity. However, the results for experiment 2 show a slight decrease in the average recognition rate from 100% to 94%. Tasks such as "*Serve Warm Meal*", "*Read Book*" and "*Gather Clothes*" brought the average recognition rate of the experiment down. This is due to these tasks not be recognised when multiple tasks were being conducted. The reason why these may have not been recognised correctly is because these tasks did not have objects that were exclusive to the task, for example "*Serve Warm Meal*" makes use of objects that are all but associated with other tasks, in contrast to a task like "*Make Tea*" which has a kettle that is an mutually exclusive object to "*Make Tea*".

Tasks	Experiment Recognition Rates 1 [%]	2 [%]	3 [%]
Brush Teeth	100	100	80
Wash Face	100	100	100
Make Tea	100	100	100
Make Toast	100	100	83
Have Cereal	100	100	100
Make Egg	100	100	100
Make Coffee	100	85	69
Eat and Drink	100	100	100
Wash Dishes	100	100	100
View Photo Album	100	83	67
Read Newspaper	100	100	60
Read Book	100	75	60
Read Mail	100	83	60
Get Fish and Chips	100	100	100
Frying	100	100	100
Serve Fish and Chips	100	80	100
Meal in Microwave	100	100	67
Serve Warm Meal	100	67	60
Having a Drink	100	90	80
Eat Biscuits	100	100	100
Doing Crossword	100	100	100
Defrost Chicken	100	90	70
Mix Ingredients	100	100	100
Cook Curry	100	100	90
Serve Curry	100	90	90
Gather Clothes	100	70	60
Use Washing Machine	100	100	90
Iron Clothes	100	100	100
Answer Phone	100	100	100
End Phone Call	100	100	100
Attend Front door	100	100	100
Close Front door	100	100	100

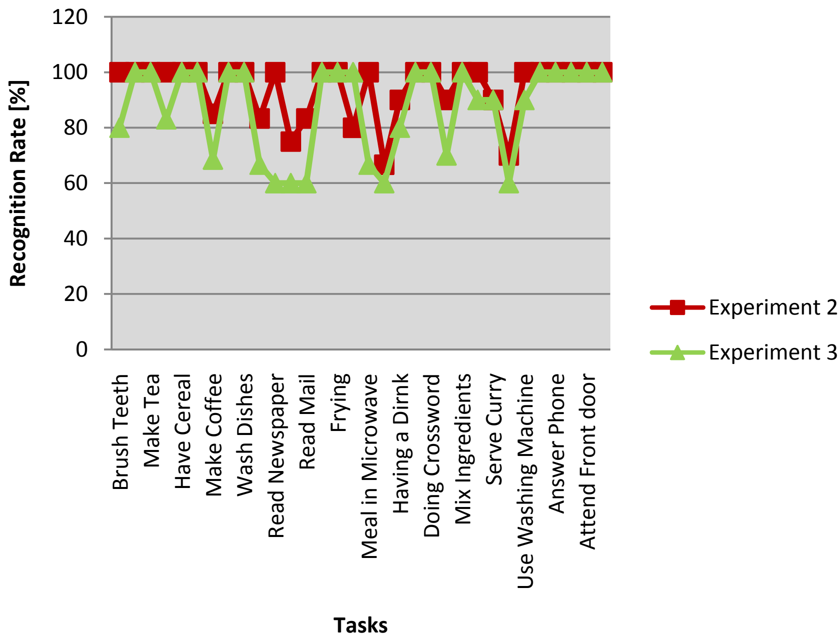
Table 32 - Task recognition experiment results

As expected the average task recognition rate fell significantly further, down to 84% for experiment 3. On this occasion tasks such as "*Read Newspaper*", "*Read Book*", "*Read Mail*", "*Meal in Microwave*", "*Defrost Chicken*", and "*Gather Clothes*" had lower recognition rates. The reason for the decline in recognition rate for these tasks is that they are generally performed with fewer objects than other tasks, and as segments of objects were deleted from the object stream this then handicapped the recognition process.

	Experiment Recognition Rates			
ADL/Sub Activities	2 [%]	3 [%]	4 [%]	5 [%]
Morning Refresh	100	100	100	100
Oral Cleaning	100	80	100	100
Clean Face	100	100	100	100
Prepare & Eat Breakfast	100	100	100	90
Prepare Food	90	100	100	90
Consume Food	100	100	100	88
Cleaning	100	100	100	86
Relaxing Leisure	100	100	90	90
Reading	100	100	90	90
Prepare & Eat Lunch	100	100	100	80
Make Fish & Chips	100	100	100	50
Warm Up Meal	100	75	100	83
Drink Water	90	80	100	89
Prepare/ Eat Snack	100	100	90	90
Prepare Food	100	90	90	90
Consume Food	100	100	90	90
Daily Puzzle	100	100	100	100
Complete Crossword	100	100	100	100
Prepare Dinner	90	100	100	70
Make Chicken Curry	90	100	100	70
Drinking	70	73	100	68
Laundry	100	100	100	80
Wash Clothes	100	100	100	80
Dry Cleaning	100	100	100	100
Ironing	100	100	100	100

Table 33 - Higher tier ADL recognition experiment results

Even though the task recognition rates have fallen for experiment 2 and 3 this did *not* affect the average ADL recognition rate, as the results for experiment 2 for the high level ADL (Table 33) have an average recognition rate of 97%. The average recognition rate for experiment 3 is 95%. Figure 44 shows the comparison of the experiment 2 and 3, which shows the ranges covered of both of the detection rates of the task and ADL recognition experiments. Both experiments have used the same stream of object data, and it can be seen that even if all of the tasks have not been recognised, it is still possible to carry out accurate ADL recognition. This is made possible because of the plans of the higher tier.



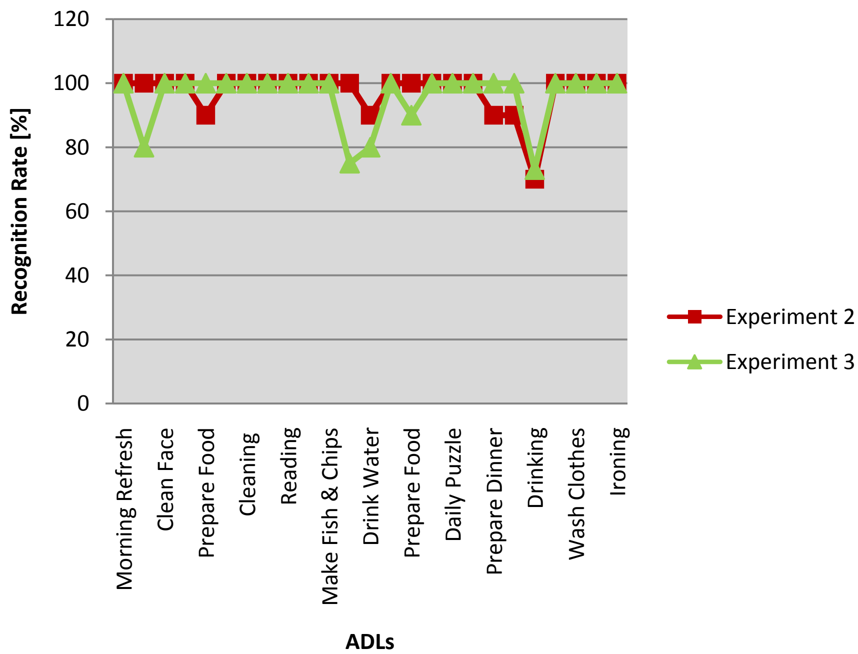


Figure 44 – Task and ADL Recognition Graphs for Experiment 2 and 3

The plan representation 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. This is because of the timing intervals that have been associated with each task and ADL, which allows the higher tier to take the time frame of the object data stream into consideration. For these experiments the timing intervals were applied to the tasks and ADLs, in terms of their duration and what time of the day they were conducted. The timing intervals were used to eliminate the possibilities when a task belongs to more than one ADL. However in order for the higher tier to function properly, a window had to be assigned, which was based on the timing interval of one ADL, and consisted of different intervals (durations) of the nested sub-activities and tasks.

The average recognition rate for experiment 4 is 98%, hence detecting the intentions of the subject given the stream of object data works well for this example. The recognition rate for this experiment should have been 100%, but the decrease in two percents was due to a set of tasks not being recognised in the lower tier.

In contrast to the high recognition rates in experiment 4, the average recognition rate fell to 87% for experiment 5. However 87% is still good, as the recognition approach was capable of keeping a track of an activity being conducted while the activity was interrupted and then resumed. ADLs such as "*Make Fish and Chips*" and "*Make Chicken Curry*" were the ADLs that had low recognition rates when an interruption ADL was recognised. This is due to the task recognition component not being able to distinguish between the tasks that relate to these ADLs when an interruption task was recognised. A reason could be that the tasks associated with both of these ADLs have larger streams of object data in comparison to other tasks. Additionally, these tasks have many objects which are associated with numerous tasks and when the object data that is related to an interruption task occurred then the task recognition component was unable to carry on recognising the tasks which was initially being carried out. This is something that is addressed in the following section of the thesis, where the feedback is used to enhance the recognition process in the lower tier.

This will enable the lower tier to take into consideration the information from the higher tier when assigning posterior probabilities for the enumeration approach, which will improve the overall recognition process of the lower tier. As seen from the results currently the higher tier is capable of making up for any tasks that have not been recognised by the lower tier, however it is still vital that the task recognition component provides at least some sequences of tasks to higher tier.

4.2 Using Feedback to Enhance the Lower Tier Recognition of Tasks

The use of feedback between the higher tier recognition component and the lower task recognition is something that is important as it can make the prior probabilities used relevant to the context. It is legitimate as it is not using the same data. The structure of the plans and other contextual information is knowledge that is not available to the lower tier. As an extreme case at certain times, certain tasks need not be considered from the very outset of a task recognition instance. For example, if an ADL such as "*Make Breakfast*" is currently being conducted then the probability of the incoming task that belongs to the activity "*Make Breakfast*" will be high, hence the probability values of these related tasks given the sensor events in the lower tier are adjusted accordingly. This section demonstrates how feedback can improve the task recognition process given different timing intervals within an ADL schedule. This has also been validated with experimental results.

Figure 45 illustrates a scenario where feedback information from the higher tier has been used to enhance the lower tier recognition probabilities. Note that in this example the task recognition approach that has been used to demonstrate the effects of feedback is the task associated sensor events technique that is used by text segmentation and enumeration segmentation approach. This does not mean that the following scenario cannot be applied to task recognition carried out by the Multiple Behavioural Hidden Markov Models (MBHMM) or any other type of task recognition approach.

The scenario in Figure 45 shows two examples of an ADL schedule that is made up of two ADLs x and y within the time frame of 9.00 to 11.00. Both sets of examples are the same, with the difference being feedback being applied to one and not the other. The examples show the tasks that are conducted in order for the ADLs to be discovered and the task sequence options (i.e. the disjunction of conjunctions of task sequences indicating the options arising from the event stream) that are used to carry out the task identification. The probability values assigned to the task associated sensor events in the example of where no feedback has been applied are based on the probability of the each task given a sensor event. This leads to a sensor event being associated with many tasks that may not even be considered for execution, e.g. sensor event *1* is associated with task *A* and *F*, this could be fine if it is the first task of the day, however by 10.15 we can tell from the higher tier that task *A* has already been conducted but because there is no feedback in the first example therefore it is not able to reflect this change in the task recognition component.

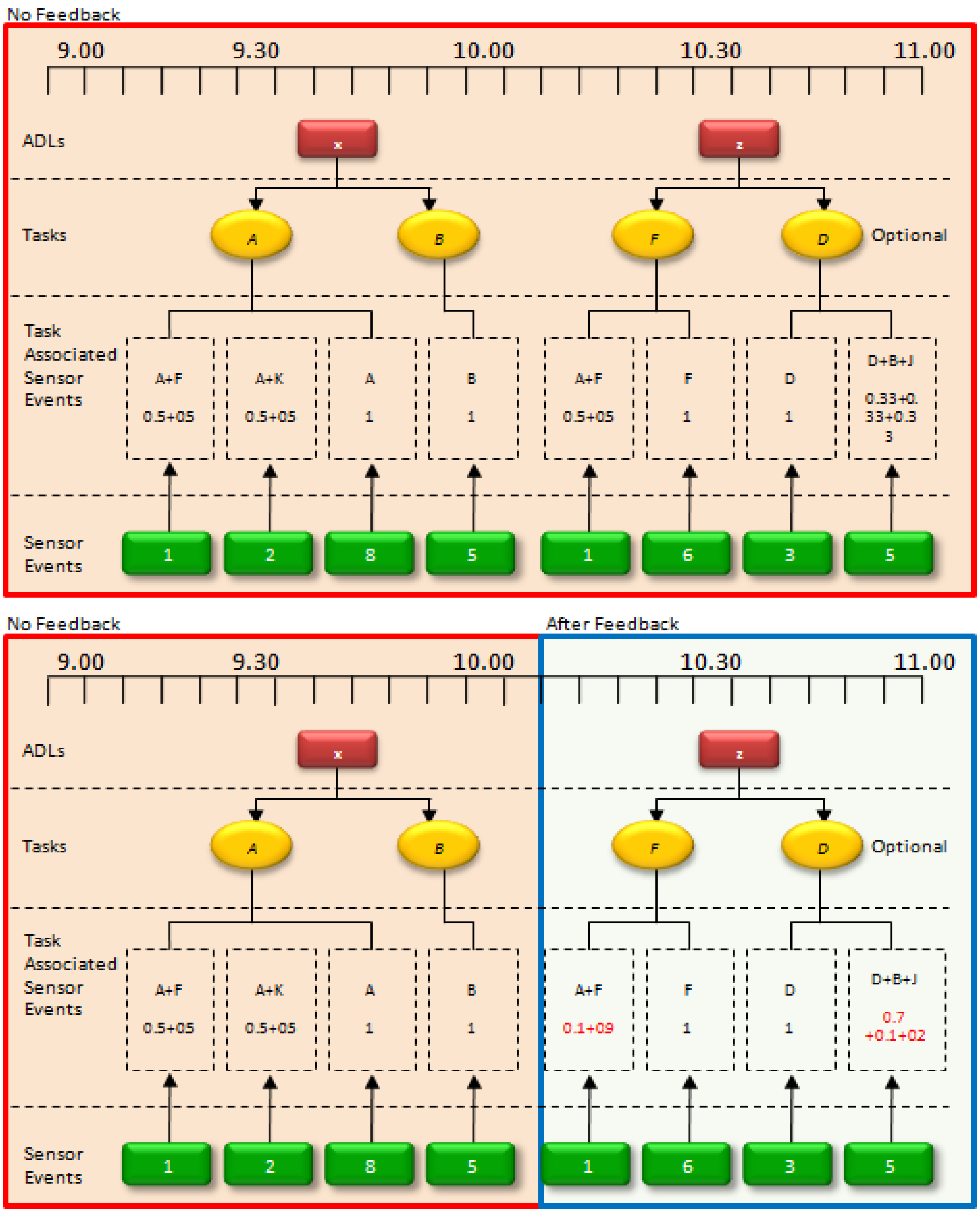


Figure 45 – Probabilities of task sequences after Feedback

In contrast, the second example in Figure 45 does make use of feedback, by using knowledge from the higher tier in order to influence the task recognition. For example, the probabilities have been adjusted accordingly for the task possibilities *A+F*, as from the high level ADL schedule it is possible to distinguish that the person is more than likely to carry out task *F* rather than *A*, as it can be seen from the ADL schedule that task *A* has already been conducted and that the next probable task is *F*. Therefore the probability of task *F* is given a more weight as opposed to *A*. This is also applied to task the possibilities *D+B+J*.

4.2.1 Feedback Experiments and Results

The objective of these experiments was to see how well the feedback between the levels performed with tasks that were not recognised as well when no feedback was applied. The experiment is solely looking at how information from the higher tier has been used to improve the task recognition rate. Data from previous experiments (section 4.1.4, task recognition experiment 3) were used to carry out and to compare recognition rates before and after the use of feedback knowledge. The tasks that have been used to validate how well feedback works are the tasks that did not perform well when there was incomplete object data i.e. sensor events.

Tasks	Recognition Rate Before Feedback [%]	Recognition Rate After Feedback [%]
View Photo Album	67	95
Read Newspaper	60	98
Read Book	60	98
Read Mail	60	96
Meal in Microwave	67	100
Serve Warm Meal	60	98
Defrost Chicken	70	100
Gather Clothes	60	97

Table 34 - Feedback Experiment Results

Initially the reason for the low recognition rates for the tasks in Table 34, was because these tasks were generally performed with fewer objects than other tasks, as segments of sensor events were missing. The feedback can help deal with this type of scenario as it makes use of the higher tier knowledge represented as ADL plans. From the results it can be seen that there was a significant change in the recognition rates when the feedback was applied and when it was not applied. The results indicate that even when it is difficult to carry out task identification due to missing data, then the higher tier knowledge can be exploited to provide some sort of clue as to which task is most likely to be conducted next.

4.3 Decision Trees to Support Prediction and Recognition of ADLs

The previous section has illustrated the benefits of feedback. This section addresses the issue of completeness and the performance of the system when the library of plans is not complete.

While it is reasonable to imagine that the most common ADLs will be modelled in the library of plans, it is impossible to imagine that the library contains plans for every hierarchical ADL. In order to generalise the activity and intention recognition capability outside the framework of the core ADLs constructed to support recognition, decision trees are constructed using a well known induction algorithm during a training period. Once the tree has been developed the trees are used as a support tool to support recognition of the ADL at each current iteration of the recognition process. For example, each time a new task is hypothesised by the low level event to task recogniser component, an ADL recognition iteration is performed at the higher level, which is also used to predict the next ADL. This capability sits on top of the core recognition process that finds the best match in the kernel of ADLs. Experiments have been conducted to assess the added value of the decision tree approach over the core recognition process. It is intuitively obvious that if the ADL to be recognised is in fact one of the core, then recognition and prediction could be good. However, the plan representing an ADL may have many optional components. Even though relative frequencies of optional tasks and sub goals can be collected during a training period, recognition using decision trees will in fact also perform this function, and additionally suggest goals for sensed actions of the subject that could be outside the plans in the core ADLs.

While the primary motivation is to mitigate the lack of library completeness, such a learned decision tree will enhance the run time efficiency of the task recognition process. By doing this the lower tier process will be able to assign the probabilities to the objects given the tasks based on the learning function’s outcome on which task is most likely to be conducted next given the posterior information collected. This will make the feedback in the ADL recognition process more efficient.

For the recognition process, a decision tree is generated for each ADL schedule, which is used to classify the correct task/ADL that is being conducted within the current ADL schedule given the current instance and taking into account the training data.

The decision tree is learned during a training period. As one of the options in the following example is the leaves of the tree are the labels of the task immediately after the most recent observed task, i.e. the tree predicts the next task. The other alternative for the labels is the parent ADL of the next task to be performed. A training instance is a set of features and the classification label.

The data for the training can be generated in two ways. In either case the training is done using information taken from the core ADLs.

In the first case the training is based on subjects performing ADLs from the core ADLs only. The information used is based on the tasks and sub-activities actually undertaken by the subject.

In the second case, the subject may follow other plans, not necessarily one of the core ADLs during training and the information used in the training instance is based on tasks actually observed and the best match to ADLs in the core ADL library. Even though none of the core plans are necessarily being followed, the system will find a nearest match to use in the training instance.

This learning instance is created when each task is labelled during training. The features of the training instances are simply name value pair structures and an example is:

{Room of observed task* = Kitchen, *Time Frame of observed task*= 9.00-9.15, *Parent ADL of observed task*=Make Breakfast, *Previously Observed Task*=Brush Teeth, *ADL of Previously Observed Task*=Morning Refresh*}

The above uses the frame feature to support the current co-ordinate based approach for recognising the task that will be conducted next, as well as making sure that the results of this instance are similar to higher tiers classification of the

ADL that has been recognised. Notice that ADLs recognised so far are used in the feature space.

The instance above can be referred to as an unlabeled instance, as it is not matched to an outcome. The outcome is e.g. the name of the task that is going to be conducted next or the parent ADL that is conducted next. The training data is made up of labelled instances that have been collected each time a new task is performed, and the plan recognition process is initiated.

The role of the decision tree is then to act as a classifier that predicts the class label for all unlabeled instances. In order to determine an outcome for an instance a decision tree needs to find an appropriate node to split in order to form the branches and leaves of the tree, which will lead to a predicted outcome. In order for the decision tree to generate small and consistent nodes information theory is used to split the sets of training instances associated with each node in the tree. The algorithm used is ID3. The root represents all the training instances. A leaf can represent training instances all with the same label. However, this is not necessarily the case and to support more accurate prediction on new data, the leaves may correspond to instances that are predominately of one label.

4.3.1 Information Gain Split Decision Trees

The following is an example of how a decision tree based on information gain splitting criteria can be used to support the current hierarchal approach for activity recognition described.

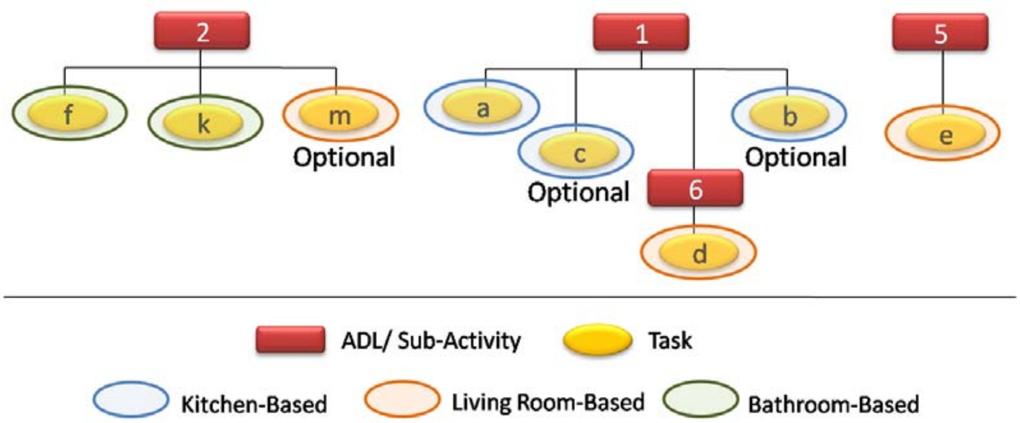


Figure 46 - ADL schedule modelled for decision trees

Figure 46 shows an ADL schedule modelled for the time interval 9.00- 10.00. This is a typical structure of an ADL schedule, however one of the differences with this ADL schedule to the ones used in previous experiments is the incorporation of the location of where each task is conducted. In general other data about the physical context relevant to the application, such as temperature (as a determinant of heating decisions, exact time as a determinant of postal arrivals, television schedules for favourite programs) could be added to enhance the predictive ability. When a task is recognised in the lower tier, the location of where the task was conducted does get recognised, however we make full use of this information when constructing a decision tree based on the ADLs within the ADL schedule that this task belongs to. Table 35 shows labelled instances that are used as training data to build the decision tree.

Room of observed task	Time Frame of observed task	Parent ADL of the observed task	Grandparent ADL of the observed task (Root or ADL)	Previously observed Task	ADL of previously observed task	Outcome (observed Next Task)
Bathroom	9.00-9.15	2	Root	f	2	k
Bathroom	9.00-9.15	2	Root	f	2	k
Bathroom	9.15-9.30	2	Root	f	2	k
Living Room	9.15-9.30	2	Root	k	2	m
Kitchen	9.15-9.30	1	Root	m	2	a
Kitchen	9.15-9.30	1	Root	k	2	a
Kitchen	9.15-9.30	1	Root	a	1	c
Living Room	9.30-9.45	6	1	c	1	d
Living Room	9.15-9.30	6	1	a	1	d
Kitchen	9.15-9.30	1	Root	k	2	a
Kitchen	9.15-9.30	1	Root	k	2	a
Living Room	9.30-9.45	6	1	c	1	d
Kitchen	9.30-9.45	1	Root	a	1	c
Kitchen	9.30-9.45	1	Root	d	6	b
Kitchen	9.45-10.00	1	Root	d	6	b
Living Room	9.45-10.00	5	Root	b	1	e
Living Room	9.30-9.45	5	Root	d	6	e
Living Room	9.45-10.00	5	Root	b	1	e
Living Room	9.30-9.45	5	Root	d	6	e
Living Room	9.45-10.00	5	Root	b	1	e

Table 35 - Training data based on ADL schedule 1

Decision trees use the features from the training instances to build the tree. This is done by taking into account the eligibility of the attributes to see if they have not already been used in the path of the chosen node. Typically decision tree learning algorithm computes the quality of each possible split that can be produced by each attribute and chooses the attribute which has the highest utility based on the quality of the split.

The splitting approach based on information gain, also known as ID3 algorithm is adopted and illustrated in Figure 47. ID3 is a greedy algorithm, which makes a decision by looking at the best next split and choosing it. Looking two steps ahead could, in principle create a better tree.

ID3 measures the quality of attribute based on the average of the entropies of the nodes produced by the split. This is based around the idea of trying to gain the most useful information for classification, which is made possible if the decision tree produces splits where the entropy is small. For example when the entropy of a split is 0, this split is better than a split which has an entropy of 1, as entropy is a measure of disorder. The entropy formula (11) is an idea formulated in information theory that is used to measure the amount of information in an attribute. Given a collection S (entire sample set) of m outcomes:

$$Entropy(S) = \sum_{i=1}^{m} p_i \log p_i \qquad (11)$$

where p_i is the proportion of S belonging to class i, while $\sum$ is over the m labels. Note that a entropy formula normally uses log base 2, however on this occasion we use log base 10 as we are simply looking to get to a classification point where the lowest entropy, rather than an absolute value.

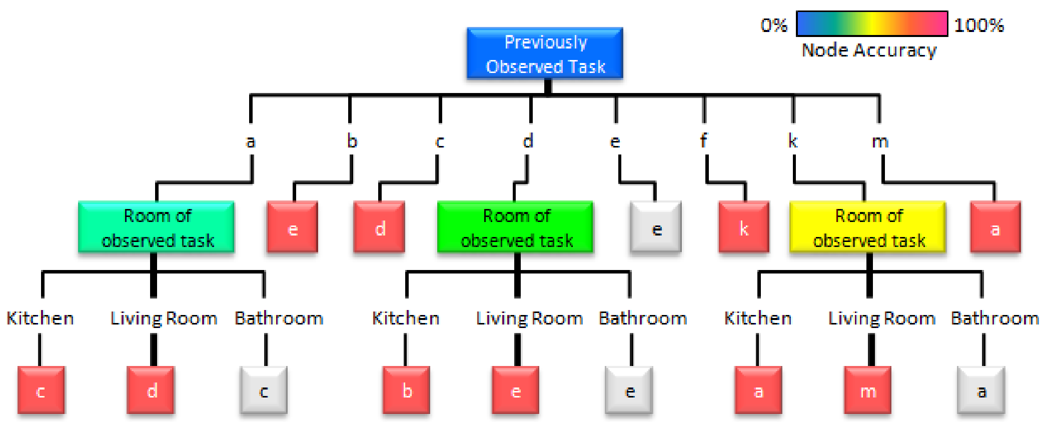


Figure 47 - Decision tree (ID3 Splitting) based on ADL schedule 1

In relation to ADL schedule 1 (Figure 46) and its associated training data (Table 35), the decision tree in Figure 47 is built using information gain as the splitting criteria. This tree is built based on the 20 instances in Table 35, where the outcomes of these instances and there entropies are initially calculated as follows:

- 3 outcomes out of the 20 instances are task ‘k’
- 1 outcome out of the 20 instances are task ‘m’
- 4 outcomes out of the 20 instances are task ‘a’
- 3 outcomes out of the 20 instances are task ‘d’
- 2 outcomes out of the 20 instances are task ‘c’
- 2 outcomes out of the 20 instances are task ‘b’
- 5 outcomes out of the 20 instances are task ‘e’

$$Entropy(S) = -\left(\frac{3}{20}\log\left(\frac{3}{20}\right)\right) - \left(\frac{1}{20}\log\left(\frac{1}{20}\right)\right) - \left(\frac{4}{20}\log\left(\frac{4}{20}\right)\right) - \left(\frac{3}{20}\log\left(\frac{3}{20}\right)\right) - \left(\frac{2}{20}\log\left(\frac{2}{20}\right)\right) - \left(\frac{2}{20}\log\left(\frac{2}{20}\right)\right) - \left(\frac{5}{20}\log\left(\frac{5}{20}\right)\right) = 1$$

This is then followed by computing the expected entropy for each attribute to see which attribute has the highest gain so that it can be used as a split to build the tree further. The gain for each attribute is determined is a follows (12):

$$Gain(A) = S(current\_set) - \sum S(child\_sets) \tag{12}$$

The gains for each of the attributes are shown in Table 36, which shows that attribute ‘*Previous Task*’ has the highest gain value, hence in Figure 47 it is chosen as the node which is split.

Attributes	Gain
Room	1.457
Time Frame	1.128
ADL	1.903
Previous Task	2.165
Previous ADL	1.276

Table 36 - Gain for each attribute to determine where to split node

This splitting process continues until a situation is reached were the remaining entropy is equal to 0.

Given the following instance after a task has been identified, we can identify by looking at the decision tree (Figure 47) that task that has been conducted is task ‘c’.

{Room of Observed Task* = Kitchen*, Time Frame of Observed Task*= 9.15-9.30, *Parent ADL of the Observed Task* =1, *Grandparent ADL of the Observed Task* = Root, *Previously Observed Task*=a, *ADL of Previously Observed Task*=1*}

We can see that information gain is good as a quality measure for the decision trees that we have constructed for correctly classifying a task within the ADL schedule. However there are several limitations for using information gain as a splitting criterion:

- Only one attribute is tested at time for making a decision, therefore it cannot take into consideration other future child nodes, as its priority is to split the attribute it is currently at.
- Can be computationally expensive when classifying continuous data, as the full tree generally needs to built in order to see where to break the continuum.

4.3.2 Gain Ratio Splitting Trees

Another method that can be used as splitting criteria is Gain Ratio, which is a way of compensating for a large number of attributes by normalising. This is done by computing the information gain for an attribute, which is then followed by dividing the gain for the attribute by the information associated with that attribute that is based only on the set of values for that attribute. Figure 48 shows a tree constructed from the training data in Table 35. It can be seen that both of the trees generated via two different splitting methods are different, however both of the generated trees are correct in terms of current training data that we have and we already know. It is important to evaluate both sets of trees to see which would be best suited for carrying out classification if an unlabeled instance occurred.

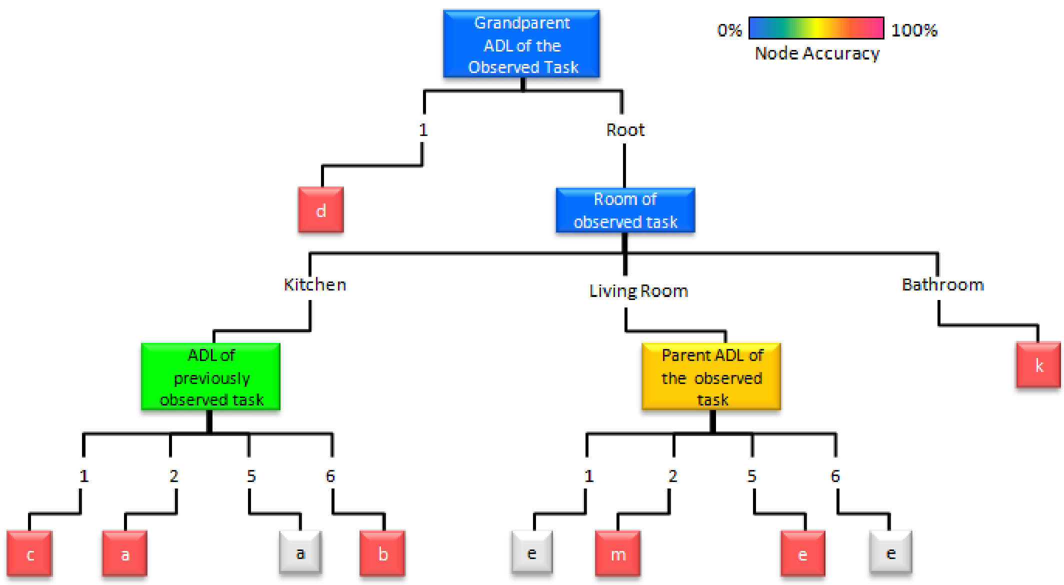


Figure 48 - Decision tree (Gain Ratio Splitting) based on ADL schedule 1

In order to evaluate these two learning trees with unlabeled instances is by using instances that the classifying trees cannot see but we can see, for example the training data in Table 35 is made of 20 labelled instances from which the tree constructed is based on, so the idea is to use only 19 labelled instances to construct the decision tree, and use this tree to see if it classifies the 20th instance correctly (which is unseen by the decision tree which is referred to as holdout sample).

The validation process will be varied with the size of labelled instances and holdout sample treated as unlabelled instances that are used, for example different variations: 15 labelled instances and 5 unlabelled instances or 10 labelled instances and 10 unlabelled instances. This will provide a good measure of which splitting criteria to be used for the constructing the decision tree. Increasing the holdout sample and decreasing the labelled is likely to reduce the predictive ability.

4.3.3 Splitting Criteria Experiment and Results

The objective of these experiments is to see which splitting criteria is best suited to construct the decision trees and to assess the potential of the decision tree approach in predicting the next task or ADL in a context where the performed activities do not correspond exactly to any of the plans associated with the ADLs in the core. Each splitting criteria will be tested with different combination ranges of labelled and sample holdout instances. The training instances used to see if any of these splitting criteria are efficient enough to support the hierarchal ADL approach are based on two ADL schedules constructed from the object data collected from the experiments conducted in section 4.1.4.

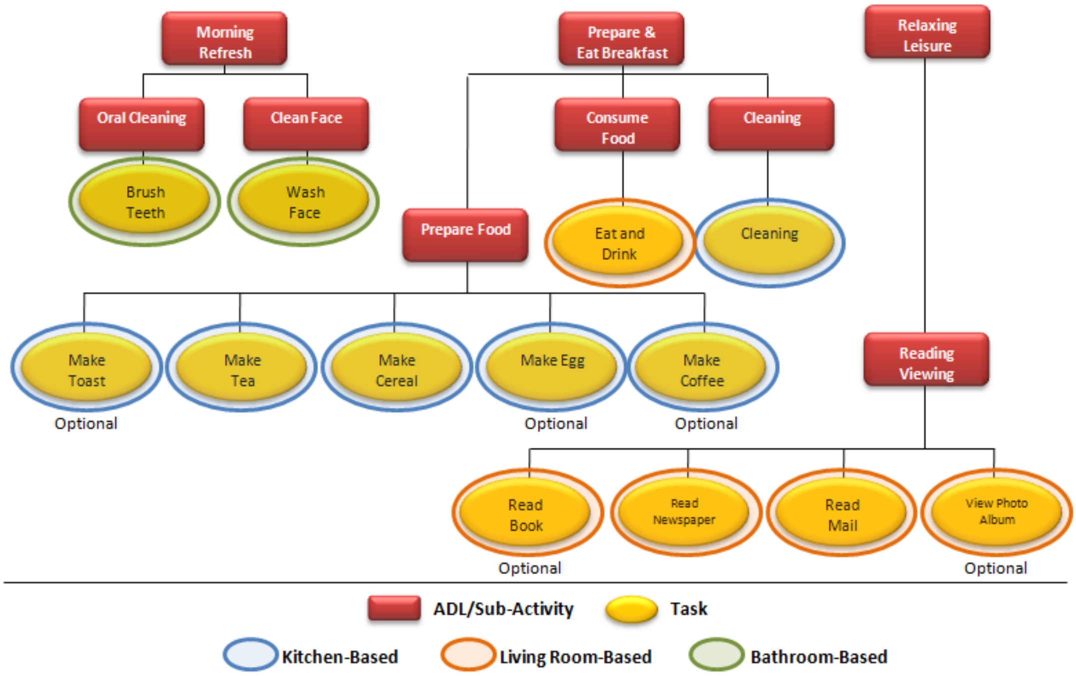


Figure 49 - ADL morning schedule constructed for decision tree experiments

These two schedules constructed are based around 'Morning' (Figure 49) and 'Afternoon' (Figure 50) activities. Both schedules have been modelled similarly to the ADL schedule in Figure 46, as they takes into consideration the location of where the tasks have been conducted. For both of the schedules, two sets of decision trees have been constructed from two sets of training data, one is used to classify the outcome of the next task, while the other tree is classifying the parent ADL of the next task being conducted.

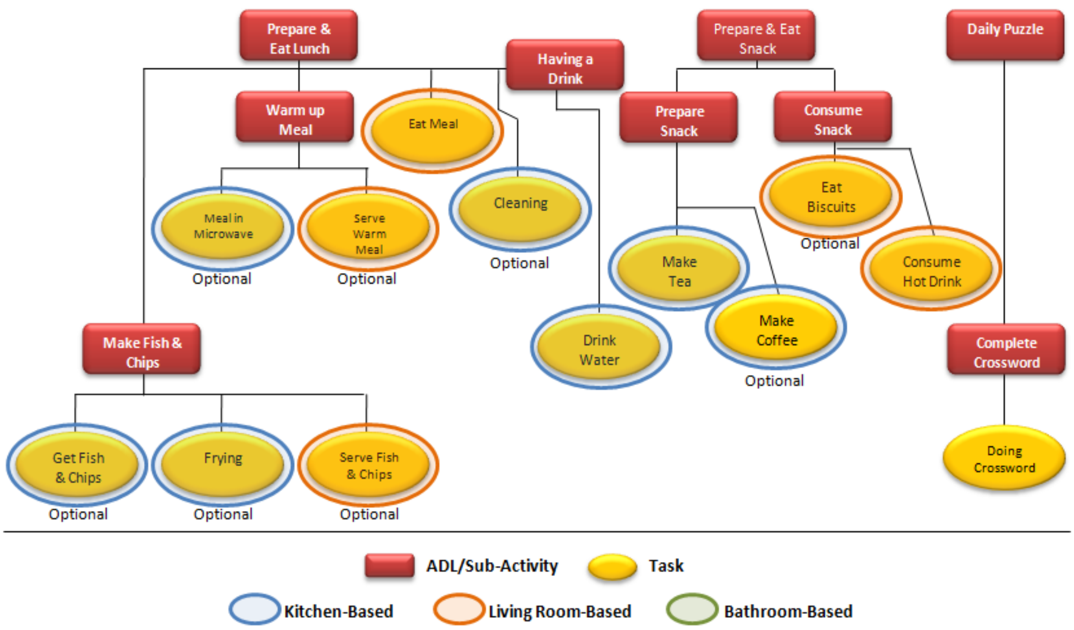


Figure 50 - ADL afternoon schedule constructed for decision tree experiments

Like the ADL schedules that were constructed for the experiments in section 4.1.4, these ADL schedules for morning and afternoon will also incorporate Interruption ADLs (Figure 51), such as a phone call, someone at the door or going to the toilet.

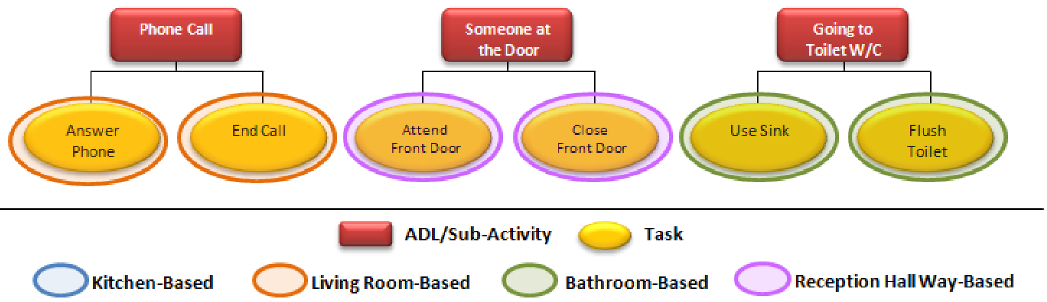


Figure 51 - Interruption ADLs for morning and afternoon ADL schedules

Each of the ADL Schedules used for these experiments has different training data sets used to build its decision tree. As well as having instances which correspond to the different timings of the day (e.g. morning and afternoon), each of these decision trees built from the training data also have different characteristics that imposed to validate different types of schedules. For example, training data for morning ADL schedule has incorporated instances which have an outcome of an interruption ADL differently to the way the instances are incorporated in the training data for afternoon ADL schedule. The characteristics for each of the training data used for each ADL schedule are as follows:

- Morning ADL schedule: The training data constructed for this schedule is made up of 212 labelled instances based on the ADL schedule being performed 20 times, where approximately 10 instances correspond to an ADL schedule being performed once. For this training data, an interruption ADL does not occur in the same position within the 10 instances of the schedule being carried more than three twice. For example, the task '*phone call*' will not follow the task '*make tea*' on more than two occasions within the complete training data set. This is done intentionally to see how well the classifying rate for the decision tree given tasks such as the ones within the interruption ADLs that do not have many labelled instances.
- Afternoon ADL schedule: For this ADL schedule the training data is made up of 202 labelled instances, which are also based on the ADL schedule being performed 20 times, with again approximately 10 instances corresponding to an ADL schedule being performed. In contrast to the

Morning ADL schedule, the training data will incorporate instances corresponding to tasks within interruption ADLs in similar positions within the 10 instances corresponding to each ADL schedule. For example, tasks associated with Interruption ADL '*Going to WC*', may occur more regularly after the subject has had lunch and just finished consuming their afternoon tea time snack. Again, this has been done intentionally, as the intention is to see a comparison of how each splitting criteria deals with two different types of training data made up of varying labelled instances that correspond to the two ADL schedules that have been constructed for these experiments.

	Holdout Sample [%]	Training Data	Holdout Sample
Morning ADL schedule	20	176	46
Morning ADL schedule	50	111	111
Morning ADL schedule	90	22	200
Afternoon ADL schedule	20	162	40
Afternoon ADL schedule	50	101	101
Afternoon ADL schedule	90	20	182

Table 37 - Holdout samples for splitting criteria experiments

Using different size variations of the labelled data as holdout samples was to see how well the splitting approaches work with different sizes of holdout samples. Table 37 shows the variations of holdout samples that were used for these experiments. Three variations of holdout sample have been used, these are 20%, 50% and 90% of the complete training data size, which is 222 instances for morning ADL schedule and 202 instances for afternoon ADL schedule.

Holdout Sample [%]	Morning ADL Schedule		Afternoon ADL Schedule	
	ID3 [%]	Gain Ratio [%]	ID3 [%]	Gain Ratio [%]
20	91	93	98	99
50	75	82	96	98
90	62	71	78	86

Table 38 – Results of holdout samples correctly classified

The results in Table 38 indicate that for both ADL schedules, gain ratio was more efficient way of splitting the attributes for constructing a decision trees as it had higher percentage of classification results for the holdout samples. One of the reasons why gain ratio performed better as a splitting approach than the ID3 is because in contrast to the gain ratio splitting approach, the ID3 tends to learn the training set too well when attributes have a large number of distinct values, which can also be its downfall when trying to classify instances that have not occurred before. For example, suppose that we are trying to construct a decision tree for describing which task is most likely to be carried out within the ADL schedule, and then ID3 on most occasions decides which attributes is the most relevant attribute by choosing the attribute with highest gain. In relation to the task being carried out, the attribute with the highest gain might be previous task within the current ADL schedule, as this will also be able to uniquely identify a task given the previous task. However this is not always suitable, as a tree which focuses its classification based on a previous tasks is unlikely to recognise a task that has not been witnessed before.

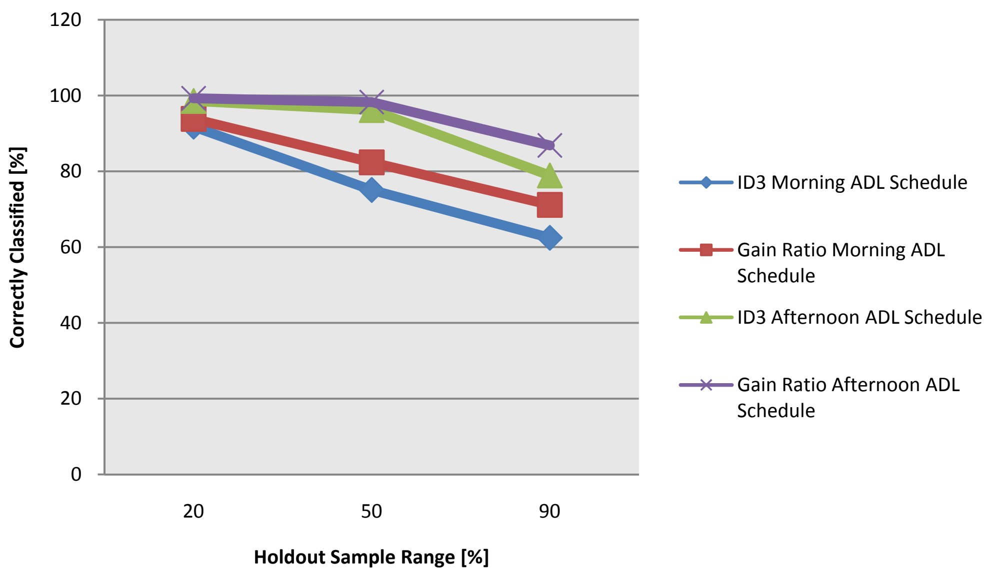


Figure 52 - Comparison of splitting techniques used for classification

The results in Table 38 reiterate the fact the gain ratio splitting is better at considering unknown tasks or unlabelled instances, as gain ratio splitting performed better with all holdout samples for the morning ADL schedule, which consisted of tasks from interrupted ADLs occurring at random junctures within the constructed training data.

Figure 52 show that the both splitting methods classified the holdout samples better for the afternoon ADL schedule than the morning ADL schedule. This was expected as the morning ADL schedule was intentionally constructed with infrequent and inconsistent appearance of tasks with no particular order. However, this does not imply that training data constructed for the afternoon schedule was simply easy for classification, as it was constructed keeping in mind the general slower pattern of how activities and tasks would normally be conducted by Alzheimer's patients.

The integration of decision trees gives the potential of being able to carry out activity recognition, with the intention of being able to learn and predict the likelihood of what task within an activity may be conducted next. However, the interaction of these approaches is only successful when consistent and cohesive training data is available.

Chapter 5

Conclusion and Future Work

This chapter is a summary of the contributions of this thesis and outlines issues that could be investigated as an extension of the research conducted in this thesis.

5.1 Conclusion

The aim of this research was trying to develop an approach to activity recognition which is reliable, i.e. with low false positives and false negatives, even though sensor data could be sparse and the activities could be complex. Another factor was the wish to reduce the demands on the sensing systems to that ADL recognition could be done in a less-intrusive manner than existing techniques. This is potentially possible through the development of new techniques for finding ADLs from object usage data, and using that data to interpret the intentions of the elderly people who suffer from Alzheimer's disease.

Chapter two introduced relevant literature on the elderly with Alzheimer's disease in smart homes. In addition, Chapter two looked at current techniques that have been applied to activity recognition in the home environment, as well as closely looking at the three important components to reliable activity recognition, namely feature detection, feature selection and models for recognition. The literature review research focused on techniques for determining the ADL an elderly person is carrying out.

Chapter three, Method, described in detail the implementation of the algorithms that have been generated to distinguish between different ADLs and for analysing the intentions of the elderly. The chapter was split into two main sections, with each of the sections focusing on the two tiers of the approach. The chapter demonstrated with experimental results how the different components that have been developed for the higher and lower tiers of were able to fulfil the experimental objectives, and how the combination of the models and algorithms for the low level and high level models could determine which ADL is currently active. The deliverables from the work in this chapter was two developed versions of the ADL recogniser. In addition, programmes based on the task recognition algorithm were developed, e.g. MBHMM and TASE segmentation.

Chapter four addressed how the two levels interact with each other. The lower level approach task enumeration approach was used to illustrate how the interaction is formed with higher tier recognition component based on the plan representation language Asbru. In addition, this chapter also looked at how the overall recognition was improved once the information at both levels are used, e.g. taking timing intervals into consideration, interweaving ADLs and enhancing the intention analysis, and with the incorporation of feedback. The application of decision trees to the current to mitigate the lack of completeness in the models and to provide efficient feedback was one of the highlights of this chapter. It took the recognition process beyond the framework of the core ADLs constructed to support recognition, and decision trees were constructed using two well known splitting algorithms during a training period. Splitting methods able to recognise tasks that occurred and were not frequent within the training data were investigated. In addition to the work stated above, another deliverable of this chapter was the enhanced ADL recogniser program. This was integrated with the GATS task recognition approach in order to validate the use of both approaches together. This programme includes enhanced features that improved different aspects of the recognition process, e.g. taking timing intervals into consideration.

In conclusion, the proposed hierarchal framework is an approach which is built on techniques that can potentially address the complexity of peoples' plans and goals within the home and recognise the intentions using object usage data. The thesis described several approaches that have been developed for each layer, each with their benefits and shortcomings. In addition the interaction of the levels between the levels is through an approach that is simple. The next step would be to build an integrated system using more sophisticated data capture equipment. However, when doing so it is important to take into consideration how this hierarchal approach will need to be scaled up for real life applications and what benefits this proposed hierarchal approach could bring. Many applications can be built around this, such as:

- An alert system to notify children that their parents are safe in their home while doing a particular activity.
- Integrate with current smart home applications, as these applications are reliant on contextual information, which is something the proposed approach can provide.
- Tracking the functional decline based on the recognition rates, which can be used to assess whether the elderly person is still able to lead an independent life.

5.2 Further Work

The work conducted within this thesis primarily focused on activity recognition using object usage data in the home environment, however there certain aspects which can be investigated further in relation to work conducted in this thesis.

One such is privacy, as it is an area of prime importance. Sensors should not be needlessly intrusive or old people will simply refuse to use them, despite their potential benefits. For this reason visual sensors have not been used. However, even RFID and sound sensors can be intrusive and one approach which could be investigated is the integration of privacy policies into the system. A person may want to switch some or all of the sensors off from time to time, or may opt for a programmed approach where more sensors can be used at certain times of the day, or if the system believes that the person is in need of help. The question of accuracy is a difficult one as increased detection usually means false positives and a trade off between the two is necessary. Policies for when more information can be used would mitigate this problem. One of the benefits of intruding higher level models into the system is that if the system has more knowledge about behaviour associated with goals then the dependency on many sensors can be reduced and when privacy requirements may need adjusting can be based on beliefs about the higher level intentions of the individual and the individual's context.

Another area of concern is standardisation, as there are many assistive technologies for use in smart homes that sometimes do not follow standard protocols for the management of the elderly. Having standardisation in place will aid the whole process of carers being able to prescribe and construct a set of ADLs for each individual patient, as part of the protocols will require periodical checkups in place in order to accommodate this. In terms of technical standardisation it is very important, as the incorporation of different monitoring devices and a wide range of applications will make it easier to integrate different approaches to carry out activity recognition, e.g. incorporating a standardised GPS system that can be embedded with the proposed hierarchal approach and smart homes.

Appendix

This appendix contains a description of the Java software programmes that were developed to validate the algorithms developed in this thesis.

ADL Recogniser v1

This was the first version of the ADL recogniser. It was used to determine an ADL given a stream of tasks. At the initial setup the software reads in ADLs (constructed in XML) and stores them into memory as a DOM tree. Once the ADLs have been loaded into memory, the ADL recogniser then acts as a server which listens for incoming task notifications. The program then reads in a task to calculate the discrepancies of each of the ADLs based on the inputted task. The output of this programme was a list of probable ADLs that may be currently active. The list is in an ascending order, with the most probable being at the top and the least probable at the bottom. The ADL that has the lowest discrepancy is considered the most probable ADL. This version of the program does not deal with interweaving tasks.

ADL Recogniser v2

The second version has the same function as the previous version; however, it has been enhanced by the incorporation of surprise indexes for each ADL. In addition this version is able to deal with interweaving of tasks. This programme was used in collaboration with TASE segmentation programme, where the outputted tasks form the TASE programme were used as input for this ADL recogniser.

MBHMM Task Recogniser

This programme was developed to validate the MBHMM work (section 3.4.2) conducted in this thesis. The inputs for this program were object usage data represented as sensor events in the form of words. The program allowed transition and confusion matrices to be modified for the different variants of a task. The output of this programme was the name of task variant that was currently being conducted given the incoming sensor events, which were mapped as observations.

TASE Segmentation

The input for this program was a stream of sensor events, which were used to output a list of possibilities of how the sensor events could be segmented into possible tasks. The output was in ascending order, with the most probable (lowest cost) segmentation of the tasks being at the top and the least probable at the bottom. The costing mechanism was based on the algorithm developed in this thesis (Section 3.4.3).

GATS Task Recognition and Activity Recognition

This programme enhanced the ADL recogniser v2 by incorporating new features:

- As opposed to reading in a library of single ADLs, this version reads in a library of constructed ADL schedules (morning, afternoon and evening).
- The use of timing constraints, expressed as intervals, in order to enhances the pruning process of identifying that an ADL is being carried out at a particular time.
- Incorporating a reading mechanism for co-ordinates (also integrated within constructed XML based ADLs), in order to determine the future intentions of the subjects.

This program also incorporated the GATS task recognition approach (section 3.4.4) with ADL recogniser, which was done by generating ADL sets from generated task sequences (section 4.1). In addition to the new features above, the following conventional features were incorporated:

- Tracking interweaving between tasks and activities. In addition, the program was able to deal with interruption ADLs.
- Recognise a task from different sequences of sensor event data, where each sequence has different orderings of how the objects are used when a task is being performed.

The program also allowed manual feedback from the ADL recogniser to the GATS approach so that the probabilities assigned to task sequences could be modified given the result of the ADL that has been recognised.

Publications

1. U. Naeem, J. Bigham, "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, UK, 2007, pp. 383-388.

2. U. Naeem, J. Bigham, J. Wang, "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.

3. U. Naeem, J. Bigham, "Activity Recognition using a Hierarchical Framework" in *Proceedings of the 2nd International Conference on Pervasive Computing Technologies for Healthcare, Ambient Technologies for Diagnosing and Monitoring Chronic Patients Workshop,* Tampere, Finland, 2008, pp. 24-27.

4. U. Naeem, J. Bigham, "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.

5. U. Naeem, J. Bigham, "Recognising Activities of Daily Life through the Usage of Everyday Objects around the Home" in *Proceedings of the 3rd International Conference on Pervasive Computing Technologies for Healthcare, Technologies to Counter Cognitive Decline Workshop*, London, March 2009

6. U. Naeem, J. Bigham, "Activity Recognition in the Home using a Hierarchal Framework with Object Usage Data" in *Journal of Ambient Intelligence and Smart Environments*, IOS Press, 2009 [Submitted]

References

[1] L. Jeffery and M. D. Cummings, "Alzheimer's disease," *The New England Journal of Medicine, Drug Therapy*, vol. 351, no. 1, pp. 56-67, Jul. 2004.

[2] S. Katz, A. B. Ford, R. W. Moskowitz, B. A. Jackson, and M. W. Jaffe, "Studies of illness in the aged: The index of ADL: A standardized measure of biological and psychosocial functio n," *Journal of the American Medical Association*, vol. 185, no. 12, pp. 914-919, Sep. 1963.

[3] M. Philipose, "Large-Scale Human Activity Recognition Using Ultra-Dense Sensing," *The Bridge, National Academy of Engineering*, vol. 35, no. 4, 2005.

[4] M. Ogawa, S. Ochiai, K. Shoji, M. Nishihara, and T. Togawa, "An attempt of monitoring daily activities at home," in *Proceedings of the 22nd Annual EMBS International Conference of the IEEE*, Chicago IL, 2000, pp. 786-788.

[5] D. H. Wilson, S. Consolvo, K. Fishkin, and M. Phillipose, "In-Home Assessment of the Activities of Daily Living of the Elderly," in *Extended Abstracts of CHI 2005: Workshops – HCI Challenges in Health Assessment*, Portland, 2005, pp. 2130-2132.

[6] A. Seyfang, R. Kosara, and S. Miksch, "Asbru Reference Manual, Asbru Version 7.3," Vienna University of Technology, Institute of Software Technology and Interactive Systems, Vienna, Technical Report Asgaard-TR-2002-1, 2002.

[7] Office of National Statistics. (2005, Feb.) Deaths from injury and poisoning: males, external cause, year of occurrence and sex, numbers and percentages, 1997-2000. [Online]. http://www.statistics.gov.uk/STATBASE/xsdataset.asp?vlnk=5680

[8] P. Kannus, J. Parkkari, S. Niemi, and M. Palvanen, "Fall-Induced Deaths Among Elderly People," *American Journal of Public Health*, vol. 95, no. 3, pp.

422-424, Mar. 2005.

[9] Department of Trade and Industry, "Research on the pattern and trends in home accidents," 99/858, 1999.

[10] B. Livesley, "Reducing home accidents in elderly people," *British Medical Journal*, vol. 305, no. 6844, pp. 2-3, Jul. 1992.

[11] M. Evans, *Psychiatry in General Practice*, B. Green, Ed. UK: Kluwer Academic Publishers, 1994.

[12] J. R. M. Copeland, M. E. Dewey, and P. Saunders, "The epidemiology of dementia: GMS-AGECAT studies of prevalence and incidence," *Journal of European Archives of Psychiatry and Clinical Neuroscience*, vol. 240, no. 4-5, pp. 212-217, Apr. 1991.

[13] L. E. Hebert, P. A. Scherr, J. L. Bienias, D. A. Bennett, and D. A. Evans, "Alzheimer disease in the US population: prevalence estimates using the 2000 Census," *Archives of Neurology*, vol. 60, no. 8, pp. 1119-1122, Aug. 2003.

[14] M. Knapp, "Dementia UK," London School of Economics and Institute of Psychiatry at King's College London, 2007.

[15] D. J. Selkoe and D. Schenk, "Alzheimer's disease: Molecular Understanding Predicts Amyloid-Based Therapeutics," *Annual Review of Pharmacology and Toxiocology*, vol. 43, pp. 545-584, Apr. 2003.

[16] M. Goedert and M. G. Spillantini, "A Century of Alzheimer's Disease," *Science*, vol. 314, no. 5800, pp. 777-781, Nov. 2006.

[17] M. P. Mattson, "Pathways towards and away from Alzheimer's disease," *International Weekly Journal of Science*, vol. 430, pp. 631-639, Aug. 2004.

[18] M. N. Haan and R. Wallace, "Can Dementia be prevented? Brain aging in a population-based context," *Annual Review of Public Health*, vol. 25, pp. 1-24, Apr. 2004.

[19] G. C. Roman, "Vascular dementia may be the most common form of dementia in the elderly," *Journal of the Neurological Sciences*, vol. 203-204, pp. 7-10, 2002.

[20] P. Tiraboschi, et al., "What best differentiates Lewy body from Alzheimer's disease in early-stage dementia?," *Brain Advance Access*, vol. 129, no. 3, pp. 729-735, Jan. 2006.

[21] C. W. Olanow and W. G. Tatton, "Etiology and Pathogenesis of Parkinson's Disease," *Annual Review of Neuroscience*, vol. 22, pp. 123-144, Mar. 1999.

[22] J. S. Snowden, D. Neary, and D. M. A. Mann, "Frontotemporal dementia," *The British Journal of Pyschiatry*, vol. 180, pp. 140-143, Feb. 2002.

[23] I. Lavenu, F. Pasquier, F. Lebert, H. Petit, and M. Van der Linden, "Perception of emotion in frontotemporal dementia and Alzheimer disease," *Alzheimer Disease Associated Disorders Journal*, vol. 13, no. 2, pp. 96-101, Apr. 1999.

[24] L. Teri, et al., "Anxiety of Alzheimer's disease: prevalence, and comorbidity," *Journals of Gerontology Series A: Biological Sciences and Medical Sciences*, vol. 54, no. 7, pp. 348-352, 1999.

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

[26] S. M. McCurry, L. E. Gibbons, R. G. Logsdon, and L. Teri, "Anxiety and nightmare behavioural disturbances. Awakenings in patients with Alzheimer's disease," *Journasl of Gerontology Nursing*, vol. 30, pp. 12-20, 2004.

[27] G. D. Barba, Z. Nedjam, and B. Dubois, "Confabulation, Executive Functions, and Source Memory in Alzheimer's disease," *Journal of Cognitive Neurppsychology*, vol. 16, no. 3-5, pp. 385-398, May 1999.

[28] R. Kern, W. Van Gorp, J. Cummings, W. Brown, and S. Osato, "Confabulation in Alzheimer's disease," *Journal of Brain Cognition*, vol. 19, pp. 172-82, Jul. 1992.

[29] I. Tallberg and O. Almkvist, "Confabulation and Memory in Patients with Alzheimer's disease," *Journal of Clinical and Experimental Neuropsychology*, vol. 23, no. 2, pp. 172-184, Apr. 2001.

[30] W. R. Gibb, P. J. Luthert, I. Janota, and P. L. Lantos, "Cortical Lewy body dementia: clinical features and classification," *Journal of Neurology, Neurosurgery, and Psychiatry*, vol. 52, no. 2, pp. 185-192, Feb. 1989.

[31] S. J. Tetewsky and C. J. Duffy, "Visual loss and getting lost in Alzheimer's disease," *Journal of Neurology*, vol. 52, pp. 958-965, Mar. 1999.

[32] V. W. Henderson, W. Mack, and B. W. Williams, "Spatial disorientation in Alzheimer's disease," *Archives of Neurology*, vol. 46, no. 4, pp. 391-394, Apr. 1989.

[33] R. S. Wilson, D. W. Gilley, D. A. Bennett, L. A. Beckett, and D. A. Evans, "Hallucinations, delusions, and cognitive decline in Alzheimer's disease," *Journal of Neurology, Neurosurgery, and Psychiatry*, vol. 69, pp. 172-177, Aug. 2000.

[34] R. E. Wragg and D. V. Jeste, "Overview of depression and psychosis in Alzheimer's disease," *The American Journal of Psychiatry*, vol. 146, pp. 577-587, May 1989.

[35] D. Aarsland, J. L. Cummings, G. Yenner, and B. Miller, "Relationships of aggressive behaviour to other neuropsychiatric symptoms in patients with Alzheimer's disease," *The American Journal of Psychiatry*, vol. 153, pp. 243-247, Feb. 1996.

[36] S. Gauthier, "Advances in the pharmacotherapy of Alzheimer's disease," *Canadian Medical Association Journal*, vol. 166, no. 5, pp. 616-623, Mar. 2002.

[37] M. P. Lawton and E. M. Brody, "Assessment of older people: self-maintaining and instrumental activities of daily living," *Gerontologist*, vol. 9, no. 3, p. 179–186, 1969.

[38] S. Katz, "Assessing Self-Maintenance: Activities of Daily Living, Mobility, and Instrumental Activities of Daily Living," *Journal of the American Geriatrics Society*, vol. 31, no. 12, pp. 721-726, 1983.

[39] W. A. Rogers, B. Meyer, N. Walker, and A. D. Fisk, "Functional limitations to daily living tasks in the aged : A focus group analysis," *Human Factors*, vol. 40, no. 1, p. 111–125, 1998.

[40] BBC News. (1999, Sep.) Health Elderly abuse: Case study. [Online]. http://news.bbc.co.uk/2/hi/health/441542.stm

[41] D. o. H. UK, "Protection of Vulnerable Adults scheme in England and Wales for adult placement schemes, domiciliary care agencies and care homes," Department of Health, Practical guide for health and social care professionals 6555, 2006.

[42] BBC News. (2005, Jul.) 700 barred from adult care work. [Online]. http://news.bbc.co.uk/1/hi/uk/4718363.stm

[43] G. A. Hancock, B. Woods, D. Challis, and M. Orrell, "The Needs of older people with dementia in residential care," *International Journal of Geriatric Psychiatry* , vol. 21, no. 1, pp. 43-49, Jan. 2006.

[44] J. Hoe, G. Hancock, G. Livingston, and M. Orrell, "Quality of life of people with dementia in residential care homes," *British Journal of Psychiatry*, vol. 188,

pp. 460-464, May 2006.

[45] P. Collinson. (2005, May) Are care homes cheating the aged?.

[46] J. M. Lilley, T. Arie, and C. D. E. Chilvers, "Accidents involving older people: a review of the literature," *Oxford Journals: Age Ageing*, vol. 24, no. 4, pp. 346-365, Jul. 1995.

[47] V. Ricquebourg, et al., "The Smart Home Concept: our immediate future," in *Proceedings of the 1st IEEE International Conference on E-Learning in Industrial Electronics*, Tunisia, 2006, pp. 23-28.

[48] D. Gann, J. Barlow, and T. Venables, "Digital futures: Making homes smarter," Chartered Institute of Housing/ Joesph Rowntree Foundation 1-900396-14-9, 1999.

[49] M. Ghorbel, F. Arab, and M. Mokhtari, "Assistive housing: Case study in a residence for elderly people," in *Proceedings of the Second International Conference on Pervasive Computing Technologies for Healthcare*, Tampere, Finland, 2008, pp. 140-143.

[50] J. Abascal, "Ambient Intelligence for People with Disabilities and Elderly People," in *SIGCHI Workshop Ambient Intelligence for Scientific Discovery*, Vienna, 2004.

[51] N. C. Council. (2006, Jan.) NDIS News: Norfolk's third Smart House to be opened.

[52] K. Harris, "Smart Homes," Department of Computer Science, University of Missouri Computational Intelligence Seminar Series, 2005.

[53] N. Noury, et al., "New Trends in Health Smart Homes," in *Proceedings of the 5th International Workshop on Enterprise Networking and Computing in Healthcare Industry (Healthcom)*, Santa Monica, California, 2003.

[54] D. Elgesem, "Ethical Issues of Using Information Technology," in *ETHICOMP International Conference on the Social and Ethical Impacts of Information and Communication Technologies*, Leicester, 1995.

[55] M. V. Giuliani, M. Scopelliti, and F. Fornara, "Elderly People at Home: Technological Help in Everyday Activities," in *Proceedings of the 14th IEEE International Workshop on Robot and Human Interactive Communication*, Nashville, Tennessee, USA, 2005, pp. 355-370.

[56] B. Abdulrazak, M. Mokhtari, M. A. Feki, and M. Ghorbel, "Integration of home networking in a smart environment dedicated to people with

disabilities," in *Proceedings on Information and Communication Technologies: From Theory to Applications*, Damascus, Syria, 2004, pp. 125-126.

[57] Y.-J. Lin, H. A. Latchman, M. Lee, and S. Katar, "A power line communication network infrastructure for the smart home," *IEEE Wireless Communications Journal*, vol. 9, no. 6, pp. 104-111, Dec. 2002.

[58] G. Dewsbury, K. Clarke, M. Rouncefield, and I. Sommerville, "Home Technology Systems," *Housing Care and Support Journal*, vol. 5, no. 4, pp. 23-26, 2002.

[59] D. Poulson, C. Nicolle, and M. Galley, "Review of the current status of research on smart homes and other domestic assistive technologies in support of the TAHI trials," Loughborough University, UK, Prepared for the Department of Trade and Industry in support of The Application Home Initiative (TAHI), 2002.

[60] K. Nam Ha, K. C. Lee, and S. Lee, "Development of PIR sensor based indoor location detection system for smart home," in *Proceedings of International Joint Conference on SICE-ICCAS*, Busan, Korea, 2006, pp. 2162-2167.

[61] S. Lee, K. Nam Ha, and K. C. Lee, "A pyroelectric infrared sensor-based indoor location-aware system for the smart home," *IEEE Journal on Consumer Electronics*, vol. 52, no. 4, pp. 1311-1317, Nov. 2006.

[62] L. Jiang, D. Liu, and B. Yang, "Smart Home Research," in *Proceedings of the Third International Conference on Machine Learning and Cybernetics*, Shangai, China, 2004, pp. 659-663.

[63] R. Kango, P. R. Moore, and J. Pu, "Networked smart home appliances – enabling real ubiquitous culture," in *Proceedings of the IEEE 5th International Workshop on Networked Appliances*, Liverpool, UK, 2002, pp. 76-80.

[64] O. Alliance, "About the OSGi Service Platform," Technical White Paper Revision 4.1, 2007.

[65] H. Zhang, F. Wang, and Y. Ai, "An OSGi and agent based control system architecture for smart home," in *Proceedings on Network, Sensing and Control*, Tucson, Arizona, USA, 2005, pp. 13-18.

[66] B. A. Miller, T. Nixon, C. Tai, and M. D. Wood, "Home networking with Universal Plug and Play," *IEEE Journal in Communications Magazine*, vol. 39, no. 12, pp. 104-109, Dec. 2001.

[67] T. Gu, H. K. Pung, and D. Q. Zhang, "Toward an OSGi-based infrastructure for context-aware applications," *IEEE Journal in Pervasive Computing*, vol. 3,

no. 4, pp. 66-74, Oct. 2004.

[68] C.-L. Wu, C.-F. Liao, and L.-C. Fu, "Service-Oriented Smart-Home Architecture Based on OSGi and Mobile-Agent Technology," *IEEE Journal on Systems, Man and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 2, pp. 193-205, Mar. 2007.

[69] S. Helal, et al., "Enabling Location-Aware Pervasive Computing Applications for the Elderly," in *Proceedings of the 1st IEEE International Conference on Pervasive Computing and Communications*, Dallas-Fort Worth Metroplex, Texas, 2003, pp. 531-536.

[70] B.-C. Cheng, H. Chen, and R.-Y. Tseng, "Context-Aware Gateway for Ubiquitous SIP-Based Services in Smart Homes," in *Proceedings of the International Conference on Hybrid Information Technology*, Cheju Island, Korea, 2006, pp. 374-381.

[71] Joseph Rowntree Foundation (ND). (2006, Jun.) Things to consider when specifying smart homes equipment. [Online]. http://www.jrf.org.uk/housingandcare/smarthomes/

[72] A. Yamaguchi, M. Ogawa, T. Tamura, and T. Togawa, "Monitoring behaviour in the home using positioning sensors," in *Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Hong Kong Sar, China, 1998, pp. 1977-1979.

[73] T. Choudhury, M. Philipose, D. Wyatt, and J. Lester, "Towards Activity Databases: Using Sensors and Statistical Models to Summarize People's Lives," *IEEE Data Engineering Bulletin*, vol. 29, pp. 49-58, Mar. 2006.

[74] E. M. Tapia, S. S. Intille, and K. Larson, "Activity Recognition in the Home Using Simple Ubiquitous Sensors," in *Proceedings of the 2nd International Conference on Pervasive Computing*, Vienna, Austria, 2004, pp. 158-175.

[75] K. Fishkin, M. Philipose, and A. Rea, "Hands-On RFID: Wireless Wearables for Detecting use of Objects," in *Proceedings of the 9th Annual IEEE International Symposium on Wearable Computers*, Osaka, Japan, 2005, pp. 38-41.

[76] M. Stikic, T. Huýnh, K. Van Laerhoven, and B. Schiele, "ADL Recognition Based on the Combination of RFID and Accelerometer Sensing," in *Proceedings of the 2nd International Conference on Pervasive Computing Technologies for Healthcare*, Tampere, Finland, 2008, pp. 258-263.

[77] D. H. Wilson and C. Atkeson, "Simultaneous Tracking & Activity Recognition (STAR) Using Many Anonymous, Binary Sensors," in *Proceedings of the 3rd*

International Conference on Pervasive Computing, Munich, Germany, 2005, pp. 62-79.

[78] M. Lösch, S. Schmidt-Rohr, S. Knoop, S. Vacek, and R. Dillmann, "Feature Set Selection and Optimal Classifier for Human Activity Recognition," in *Proceedings of the 16th IEEE International Conference on Robot & Human Interactive Communication*, Jeju, Korea, 2007, pp. 1022-1027.

[79] X. Zhu, "Semi-Supervised Learning Literature Survey," Computer Sciences, University of Wisconsin-Madison, USA, Technical Report 1530, 2007.

[80] F. Cozman, I. Cohen, and M. Cirelo, "Semi-Supervised Learning of Mixture Models and Bayesian Networks," in *Proceedings of the 20th International Conference on Machine Learning*, Washington, DC, USA, 2003, pp. 99-106.

[81] C. Pal, X. Wang, M. Kelm, and A. McCallum, "Multi-Conditional Learning for Joint Probability Models with Latent Variables," in *19th Annual Conference on Neural Information Processing Systems Workshop on Advances in Structured Learning for Text and Speech Processing*, Whistler British Columbia, Canada, 2005.

[82] M. Inoue and N. Ueda, "Exploitation of unlabeled sequences in hidden Markov models," *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 25, no. 12, pp. 1570-1581, Dec. 2003.

[83] R. Raina, Y. Shen, A. Y. Ng, and A. McCallum, "Classification with Hybrid Generative/Discriminative Models," in *Proceedings of the 2003 Conference on Advances in Neural Information Processing Systems 16*, 2003, pp. 545-553.

[84] S. Srihari. (2006) Machine Learning: Generative and Discriminative Models. Presentation, Department of Computer Science and Engineering, University at Buffalo.

[85] T. Jebara and A. Pentland, "Maximum conditional likelihood via bound maximization and the cem algorithm," in *Proceedings of the 1998 Conference on Advances in Neural Information Processing Systems 11*, 1998, pp. 494-500.

[86] K. Nigam, J. Lafferty, and A. McCallum, "Using maximum entropy for text classification," in *IJCAI-99 Workshop on Machine Learning for Information Filtering*, 1999, pp. 61-67.

[87] A. Y. Ng and M. Jordan, "On Discriminative vs. generative classifiers: a comparison of logistic regression and naïve bayes," in *Proceedings of the 2001 Neural Information Processing Systems (NIPS) Conference*, 2001.

[88] T. Barger, et al., "Objective Remote Assessment of Activities of Daily Living: Analysis of Meal Preparation Patterns," Medical Automation Research Centre, University of Virginia, Health System Poster Presentation, 2002.

[89] M. Ogawa and T. Togawa, "Monitoring Daily Activities and Behaviours at Home by Using Brief Sensors," in *Proceeding of the 1st Annaul International Conference on Microtechnologies in Medicine and Biology of the IEEE*, Lyon, France, 2000, pp. 611-614.

[90] H. Noury, et al., "Monitoring Behaviour in Home Using Smart Fall Sensor and Positioning Sensors," in *Proceeding of the 1st Annaul International Conference on Microtechnologies in Medicine and Biology of the IEEE*, Lyon, France, 2000, pp. 607-610.

[91] M. Philipose, et al., "Inferring activities from interactions with objects," *Pervasive Computing, IEEE Journal*, vol. 3, no. 4, pp. 50-57, Oct. 2004.

[92] M. Philipose, et al., "Battery-Free Wireless Identification and Sensing," *Pervasive Computing IEEE Journal*, vol. 4, no. 1, pp. 37-45, Jan. 2005.

[93] L. Cheng-Ju, et al., "Mobile healthcare service system using RFID," in *Proceedings of the 2004 IEEE International Conference on Networking, Sensing and Control*, Taipei, Taiwan, 2004, pp. 1014-1019.

[94] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. Rehg, "A Scalable Approach to Activity Recognition based on Object Use," in *Proceedings of the International Conference on Computer Vision*, Rio de Janeiro, Brazil, 2007, pp. 1-8.

[95] D. J. Patterson, D. Fox, H. Kautz, and M. Philipose, "Fine-Grained Activity Recognition by Aggregating Abstract Object Usage," in *Proceedings of the 9th IEEE International Symposium on Wearable Computers*, Osaka, Japan, 2005, pp. 44-51.

[96] M. Kärkkäinen, "Increasing efficiency in the supply chain for short shelf life goods using RFID tagging," *International Journal of Retail & Distribution Management*, vol. 31, no. 10, pp. 529-536, 2003.

[97] B. Logan, M. Healey, M. Philipose, E. Munguia-Tapia, and S. Intille, "A Long-Term Evaluation of Sensing Modalities for Activity Recognition," in *Proceedings of 9th International Conference on Ubiquitous Computing*, Innsbruck, Austria, 2007, pp. 483-500.

[98] J. Lester, T. Choudhury, N. Kern, G. Borriello, and B. Hannaford, "A Hybrid Discriminative/Generative Approach for Modelling Human Activities," in

Proceedings of the 19th International Joint Conference on Artificial Intelligence, 2005, pp. 766-772.

[99] N. Kern, B. Schiele, and A. Schmidt, "Multi-sensor Activity Context Detection for Wearable Computing," in *Proceedings of EUSAI*, Eindhoven, Netherlands, 2003, pp. 220-232.

[100] T. Huýnh and B. Schiele, "Towards Less Supervision in Activity Recognition from Wearable Sensors," in *Proceedings of the 10th IEEE International Symposium on Wearable Computers*, Montreux, Switzerland, 2006, pp. 3-10.

[101] M. Stikic and K. Van Laerhoven, "Recording Housekeeping Activities with Situated Tags and Wrist-Worn Sensors: Experiment Setup and Issues Encountered," in *Workshop on Wireless Sensor Networks for Health Care at International Conference on Networked Sensing Systems*, Braunschweig, Germany, 2007.

[102] K. Jeong, J. Won, and C. Bae, "User Activity Recognition and Logging in Distributed Intelligent Gadgets," in *Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, South Korea, 2008, pp. 683-686.

[103] P. Lukowicz, et al., "Recognizing Workshop Activity Using Body Worn Microphones and Accelerometers," in *Proceedings of the 2nd International Conference on Pervasive Computing*, Vienna, Austria, 2004, pp. 18-32.

[104] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, "Activity Recognition from Accelerometer Data," in *Proceedings of the 20th National Conference on Artificial Intelligence*, Pittsburgh, Pennsylvania, 2005, pp. 1541-1546.

[105] M. Ermes, J. Pärkkä, J. Mäntyjärvi, and I. Korhonen, "Detection of Daily Activities and Sports With Wearable Sensors in Controlled and Uncontrolled Conditions," *IEEE Transactions on Information Technology in Biomedicine*, vol. 12, no. 1, pp. 20-26, Jan. 2008.

[106] L. Bao and S. S. Intille, "Activity Recognition from User-Annotated Acceleration Data," in *Proceedings of the 2nd International Conference on Pervasive Computing*, Vienna, Austria, 2004, pp. 1-17.

[107] S. Wang, W. Petney, A. Popescu, T. Choudhury, and M. Philipose, "Common Sense Based Joint Training of Human Activity Recognizers," in *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, Hyderabad, India, 2007, pp. 2237-2243.

[108] W. Petney, A. Popescu, S. Wang, H. Kautz, and M. Philipose, "Sensor-Based

Understanding of Daily Life via Large-Scale Use of Common Sense," in *Proceedings of the 21st AAAI Conference on Artificial Intelligence*, Boston, USA, 2006.

[109] W. Petney, et al., "Human Estimation through Learning over Common Sense Data," in *Proceedings of NIPS 2006 Workshop on Grounding Perception, Knowledge and Cognition in Sensori-Motor Experience*, Whistler, Canada, 2006.

[110] R. Gupta and M. J. Kochendeerfer, "Building Large Knowledge-Based Systems," in *Proceedings of the 19th National Conference on Artificial Intelligence*, San Jose, California, USA, 2004, pp. 605-610.

[111] O. Etzioni, et al., "Methods for domain-independent information extraction from the web: An experimental comparison," in *Proceedings of the 19th National Conference on Artificial Intelligence*, San Jose, California, 2004, pp. 391-398.

[112] W. Petney, M. Philipose, and J. Bilmes, "Structure Learning on Large Scale Common Sense Statistical Models of Human State," in *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, Chicago, Illinois, 2008, pp. 1389-1395.

[113] W. Petney, M. Philipose, J. Bilmes, and H. Kautz, "Learning Large Scale Common Sense Models of Everyday Life," in *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, Vancouver, British Columbia, 2007.

[114] J. R. Smith, et al., "RFID-Based Techniques for Human-Activity Detection," *Communications of the ACM*, vol. 48, no. 9, pp. 39-44, Sep. 2005.

[115] J. Lester, T. Choudhury, and G. Borriello, "A Practical Approach to Recognizing Physical Activities," in *Proceedings of the 4th International Conference on Pervasive 2006*, Dublin, Ireland, 2006, pp. 1-16.

[116] N. C. Krishnan and S. Panchanathan, "Analysis of low resolution accelerometer data for continuous human activity recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008, pp. 3337-3340.

[117] H. Kanai, T. Nakada, Y. Hanbat, and S. Kunifuji, "A support system for context awareness in a group home using sound cues," in *Proceedings of the Second International Conference on Pervasive Computing Technologies for Healthcare*, Tampere, Finland, 2008, pp. 264-267.

[118] D. H. Wilson, A. C. Long, and C. Atkeson, "A Context-Aware Recognition Survey for Data Collection Using Ubiquitous Sensors in the Home," in *Proceeding of International Conference for Human-Computer Interaction*, Portland,

Oregon, USA, 2005, pp. 1865-1868.

[119] D. H. Wilson and M. Philipose, "Maximum A Posteriori Path Estimation with Input Trace Perturbation: Algorithms and Application to Credible Rating of Human Routines," in *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, Edinburgh, Scotland, UK, 2005, pp. 895-901.

[120] N. Landwehr, B. Gutmann, I. Thon, M. Philipose, and L. D. Raedt, "Relational Transformation-based Tagging for Human Activity Recognition," in *Proceedings of the 6th Workshop on Multi-Relational Data Mining*, Warsaw, Poland, 2007.

[121] M. Perkowitz, M. Philipose, K. P. Fishkin, and D. J. Patterson, "Mining Models of Human Activities from the Web," in *Proceedings of the 13th International Conference on World Wide Web*, New York, USA, 2004, pp. 573-582.

[122] D. Wyatt, M. Philipose, and T. Choudhury, "Unsupervised Activity Recognition Using Automatically Mined Common Sense," in *Proceedings of the 20th National Conference on Artificial Intelligence*, Pittsburgh, Pennsylvania, 2005, pp. 21-27.

[123] E. M. Tapia, T. Choudhury, and M. Philipose, "Building Reliable Activity Models Using Hierarchical Shrinkage and Mined Ontology," in *Proceedings of the 4th International Conference on Pervasive 2006*, Dublin, Ireland, 2006, pp. 17-32.

[124] H. Wang, Z. Xingshe, and Z. Wang, "Supporting the Living of the Elderly with Semantic Collaborative HealthCare," in *Proceedings of the Second International Conference on Pervasive Computing Technologies for Healthcare*, Tampere, Finland, 2008, pp. 192-195.

[125] S. Miksch and C. Fuchsberger, "Asbru's Execution Engine: Utilizing Guidelines for Artificial Ventilation of Newborn Infants," in *Proceedings of the joint Workshop Intelligent Data Analysis in Medicine And Pharmacology and Knowledge-Based Information Management In Anaesthesia and Intensive Care, in conjunction with the 9th Conference on Artificial Intelligence in Medicine in Europe*, Protaras, Cyprus, 2003, pp. 99-125.

[126] C. Fuchsberger, J. Hunter, and P. McCue, "Testing Asbru Guidelines and Protocols for Neonatal Intensive Care," in *Proceedings of the 10th Conference on Artificial Intelligence in Medicine*, Aberdeen, UK, 2005, pp. 101-110.

[127] R. Zurawski and M. Zhou, "Petri Nets and Industrial Applications: A Tutorial," *Industrial Electronics, IEEE Journal*, vol. 41, no. 6, pp. 567-583, Dec.

1994.

[128] E. D. Browne, M. Schrefl, and J. R. Warren, "Goal-focused Self-Modifying Workflow in the Healthcare Domain," in *Proceedings of the 37th IEEE International Conference on Systems Sciences*, Hawaii, 2004.

[129] W. M. P. Van der Aalst, "How to Handle Dynamic Change and Capture Management Information? An Approach Based on Generic Workflow Models," University of Georgia, Department of Computer Science, Athens, Technical Report UGA-CS-TR-99-01, 1999.

[130] R. Kosara, S. Miksch, Y. Shahar, and P. Johnson, "AsbruView: Capturing Complex, Time-oriented Plans - Beyond Flow-Charts," in *Second Workshop on Thinking with Diagrams*, Aberystwyth, United Kingdom, 1998, pp. 119-126.

[131] D. H. Wilson, D. Wyaat, and M. Phillipose, "Using Context History for Data Collection in the Home," in *Proceedings of the 3rd International Conference on Pervasive Computing: ECHISE Workshop*, Munich, Germany, 2005.

[132] D. Beeferman, A. Berger, and J. Lafferty, "Statistical Models for Text Segmentation," in *Machine Learning, Special Issue on Natural Language Learning*, C. Cardie and R. J. Mooney, Eds. 1999, pp. 177-210.

[133] K. Han and M. Veloso, "Automated Robot Behavior Recognition Applied to Robotic Soccer," in *Robotics Research: the Ninth International Symposium*, J. Hollerbach and D. Koditschek, Eds. London: Springer-Verlag, 2000, pp. 199-204.

[134] M. Utiyama and H. Isahara, "A Statistical Model for Domain-Independent Text Segmentation," in *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, Toulouse, France, 2001, pp. 499-506.

[135] The Alzheimer's Association. (2005) Activities at Home, Planning the day for a person with dementia. Brochure retrived from: http://www.alz.org/national/documents/brochure_activities.pdf