CogWatch – Cognitive Rehabilitation of Apraxia and Action Disorganisation Syndrome

D3.1 Report on action recognition techniques

Deliverable No.		DN.3.1	
Workpackage No.	**WP3**	Workpackage Title	**Activity Recognition & Prediction**
Task No.	**T3.1**	Activity Title	**Action Recognition**
Authors (per company, if more than one company provide it together)		**Author (s) Name (s) (Company)** **Martin Russell, Chris Baber, Manish Parekh (UOB)** **Charmayne Hughes, Joachim Hermsdörfer (TUM)**	
Status (F: final; D: draft; RD: revised draft):		**F**	
File Name:		**CogWatch_D3.1_Report_ActionRec_Final.docx**	
Project start date and duration		**01 November 2011, 36 Months**	

EXECUTIVE SUMMARY

Automatic Activity Recognition (AAR) refers to technology for monitoring of a participant engaged in an activity for everyday living (ADL). The CogWatch AAR system should know the stage that the participant has reached, it should be able to estimate the likelihood of successful completion, and it should be able to synthesise useful cues and feedback to the participant to redirect action. The CogWatch AAR system will monitor an activity using sensors attached to tools and objects, plus video-based estimates of the participant's hand positions.

This report discusses the issues that are relevant to the development of the CogWatch AAR system. Its main purpose is to explain the rationale for the design of the first prototype system, which is to be operational in month 16 of the project. The report is presented in four main sections, addressing the activity scenario, instrumentation, the task model, and activity recognition.

The report begins with a review of the tea-making task chosen as the application for the first prototype (Section 2). A hierarchical tree description of tea-making (taken from CogWatch D1.1) is a reference for the discussions of action recognition models and task models which follow. The goal of 'tea-making', at the root of the hierarchical tree, is split into sub-goals (for example, 'add water in the cup'), which in turn are described in terms of tasks.

Section 3 discusses the instrumentation available to the prototype system. This includes the CogWatch Instrumented Coaster (CIC) (an instrumented "mat" that can be fixed to the base of a mug or jug), RFID tags, and 3D hand location estimated using the Kinect system. The utility of these sensors for monitoring tea-making is discussed in Section 3.4. The sensors will communicate with the AAR system wirelessly via Bluetooth. The set of sensors that will be used in the prototype will be agreed between WP2 and WP3 in July 2012.

Section 4 is an overview of the CogWatch AAR system, and Sections 5 and 6 describe its two main components, the task model (TM) and the action recogniser.

The strengths and weaknesses of various candidate TMs are considered in Section 5. These include the psychological models proposed by Cooper and Shallice, and Botvinick and Plaut; the Hierarchical Task Analysis (HTA) model from Ergonomics, the automated probabilistic models of everyday activities (AM-EvAs) that are being developed at TUM, and Markov Decision Processes (MDPs) and Partially Observable MDPs (POMDPs).

Section 6 discusses the pattern recognition component of the AAR system. The task of AAR is to convert sequences of measurements from the sensors into estimates of the sub-goals and tasks that the participant is performing. Hidden Markov Models (HMMs) are chosen for this, because they are an appropriate technology for processing time-varying sequences of data and because many of the issues that arise in the context of the CogWatch application have already been addressed for HMMs in the context of automatic speech recognition (ASR).

The final prototype specification is summarised in Section 7. The prototype CogWatch TM will be based on MDPs, whose structure and parameters are determined using HTA. Future TMs will be data-driven, based on either MDPs, POMDPs or AM-EvAs. Activity recognition will be performed using sub-goal level HMMs, using multiple decoders configured as a set of parallel sub-goal detectors (late integration – sub-goal level fusion). This architecture is novel from the perspective of ASR, and was chosen for its ability to cope with asynchronous overlapping sub-goals.

TABLE OF CONTENTS

1. INTRODUCTION 12

1.1 Activity recognition and prediction 12

1.2 Objectives for months 1 to 16 12

1.3 Interactions between the partners 12

1.4 Proposed timetable for the report 13

2. SCENARIO 15

2.1 The tea-making task 15

2.1.1 Terminology 16

2.2 Strengths and limitations of hierarchical descriptions 16

2.2.1 Utility of the hierarchical task description 16

2.2.2 Limitations of the hierarchical description 16

2.2.3 Implications for activity recognition 17

3. INSTRUMENTATION 18

3.1 Sensor technologies 18

3.1.1 Vision-based systems 18

3.1.2 Radio frequency identification (RFID) 18

3.1.3 Accelerometers 18

3.1.4 Force sensitive resistors (FSRs) 19

3.1.5 Force sensitive handles (FSHs) 19

3.1.6 Wireless connectivity 19

3.2 Functionality of sensors for tea-making 20

3.3 The CogWatch instrumented coaster 21

3.3.1 Specification 21

3.3.2 Functionality of the CIC 21

3.3.3 Output of the CIC 21

3.4 Instrumentation in the first CogWatch prototype 21

4. THE FIRST PROTOTYPE ACTION RECOGNITION SYSTEM 23

4.1 Components of the action recognition system 23
4.1.1 Sensor data capture and pre-processing 23
4.1.2 Automatic action recognition (AAR) 23
4.1.3 The Task Model (TM) 24

5. MODELS OF HUMAN TASK EXECUTION (THE "TASK MODEL") 25

5.1 Role of the Task Model (TM) in the CogWatch system 25
5.1.1 Inference of the 'belief state' 25
5.1.2 Sub-task history 25
5.1.3 Sub-task prediction 25
5.1.4 Failure prediction 25
5.1.5 Cue / feedback generation 25
5.1.6 Task execution recording 25
5.1.7 Knowledge-driven versus data-driven 26
5.1.8 Psychological plausibility 26
5.1.9 Computational utility 26

5.2 Candidates for the CogWatch Task Model 26
5.2.1 Contention Scheduling Model – Norman and Shallice (1986) 26
5.2.1.1 Errors: 28
5.2.2 The interactive action model (IAN) - Cooper and Shallice (2000) 28
5.2.2.1 Errors 29
5.2.3 Simple recurrent network (SRN) - Botvinick and Plaut (2002, 2004) 30
5.2.3.1 Errors: 31
5.2.4 Automated probabilistic models of everyday activities (AM-EvAs) – Beetz, Tenorth, Jain, and Bandouch (2010) 31
5.2.5 Hierarchical Task Analysis (HTA) 34
5.2.5.1 Errors 34
5.2.6 Markov Decision Processes (MDPs) 35
5.2.6.1 Errors 36
5.2.7 Partially Observable MDPs (POMDPs) 36

5.3 Choice of Task Model in the first CogWatch prototype system 36
5.3.1 Discussion 36
5.3.2 Summary 38

6. PATTERN-BASED ACTION RECOGNITION 39

6.1 Introduction 39

6.2 Hidden Markov Models (HMMs) 39

6.3 Application of HMMs to action recognition 40

6.3.1 Unit selection 40

6.3.1.1 Task-level HMMs 41

6.3.1.2 Sub-goal level HMMs 42

6.3.2 Action recogniser architectures 42

6.3.2.1 Sensor integration (early integration) 42

6.3.2.2 Late integration (object level) 44

6.3.2.3 Late integration (sub-goal level fusion) 45

6.3.2.4 Background model 45

6.4 Choice of HMM architecture for first CogWatch prototype 46

7. SPECIFICATION OF THE FIRST COGWATCH PROTOTYPE 47

7.1 System inputs 47

7.1.1 The CogWatch instrumented coaster 47

7.1.2 RFID tags 47

7.1.3 Kinect 47

7.2 Automatic action recognition (AAR) 47

7.2.1 Specification of the HMM-based AAR system 47

7.3 The Task Model 47

7.3.1 Specification of the MPD-based TM 47

8. CONCLUSIONS 48

TABLE OF FIGURES

Figure 1: WP3 view of interactions between CogWatch Work Packages 13

Figure 2: Hierarchical tree representation of the tea making task (from D1.1 Report on scenarios) 15

Figure 3: Functionality of sensors for the tea-making task 20

Figure 4: The CogWatch instrumented coaster attached to the base of a mug 21

Figure 5: Instrumentation for the 1st CogWatch prototype 22

Figure 6: Prototype CogWatch action recognition system 23

Figure 7: The Contention Scheduling model (Adapted from Norman and Shallice, 1986) .. 27

Figure 8: Principal components of the Interactive Action Model (Cooper & Shallice, 2000). 28

Figure 9: The architecture of the overall Simple Recurrent Network (Botvinick & Plaut, 2004). Open arrows indicate the connections between units in each layer of the system. 31

Figure 10: From several observations of the same task (left), the system learns the partial order of actions in that task (right) using statistical relational learning models (Tenorth, 2011) 33

Figure 11: Action-level HMM for "*tilt the kettle until cup is full*" 41

Figure 12: Early integration of sensor information 43

Figure 13: Late integration (object-level fusion) 44

Figure 14: Late integration (sub-goal level fusion) 45

REVISION HISTORY

Revision no.	Date of Issue	Author(s)	Brief Description of Change
Version 1	20/05/2012	UOB, TUM	First draft for circulation
Version 2	23/05/2012	UOB, TUM	Comments on first draft incorporated.
Version 3	24/05/2012	UOB, TUM	Additional comments from AW incorporated.
Version 4	25/05/2012	UOB, TUM	Additional comments from JH incorporated
Version 5	28/05/2012	UPM, UOB	Formatting corrections
Final	28/05/2012	UOB, TUM	Final version

LIST OF ABBREVIATIONS AND DEFINITIONS

Abbreviation	Abbreviation
AADS	**Apraxia and Action Disorganisation Syndrome**
AAR	**Automatic Activity Recognition**
ADL	**Activity of daily living**
ASR	**Automatic speech recognition**
CIC	**CogWatch Instrumented Coaster**
CSC	**Contention scheduling system**
DBN	**Dynamic Bayesian network**
FMEA	**Failure Modes Effects Analysis**
FSH	**Force Sensitive Handle**
FSR	**Force Sensitive Resistor**
GMM	**Gaussian mixture model**
HMM	**Hidden Markov model**
IAN	**Interactive action model**
MDP	**Markov decision process**
PDF	**Probability density function**
POMDP	**Partially observable Markov decision process**
PSR	**Pressure Sensitive Resistor**
RFID	**Radio Frequency Identification**
SAS	**Supervisory attentional system**
TM	**Task model**

1. INTRODUCTION

1.1 Activity recognition and prediction

This is the first report from CogWatch Work Package 3 “Activity Recognition and Prediction”. WP3 involves two of the CogWatch partners, the University of Birmingham (UOB) and Technische Universität München (TUM). Its objective is to establish techniques that will be used to map the raw measurements from the monitor devices developed in WP2 to a description of patient behaviour. The task will explore two approaches for action recognition: model based and pattern based recognition.

Model based recognition refers to the use of psychological models (e.g., Cooper et al., 2005) to provide a behavioural platform for the hierarchical labelling relating to tasks performed by the patients. For example, Cooper and colleagues (2005) suggested a model based on a hierarchy of schemas that are interconnected with object representations. They state that explicit hierarchically organized and causally efficacious schema and goal representations are required to provide an adequate account of the flexibility of sequential behaviour in everyday life. Botvinick and Plaut (2004) and Botvinick and colleagues (2009) offered an alternative model based on recurrent connections within a network mapping from environmental inputs to actions in everyday tasks. One of the main differences between the two approaches relates to the assumed relationship between actions that are performed; for Cooper and colleagues, actions occur in a sequence towards a goal (so there is always a teleological explanation for the ordering of actions), and for Botvinick, actions co-occur at specific points in time (some of these correlations *could* be related to a task sequence but this does not have to be the case). Both models will be considered in providing a theoretical framework for action recognition and new models will be developed.

Pattern based recognition refers to the use of more direct methods from pattern recognition and signal processing. In this subtask we will define multilevel, hierarchical labelling conventions for the data. Labels might relate directly to individual sensors (e.g. force transducers, accelerometers, eye-trackers, body-motion trackers, etc), or they might correspond to the integration of multiple sensors (e.g. revealing the cognitive performance of the patient), or they might be task related (e.g. a convention for describing the various stages in preparing and eating breakfast, brushing teeth, or dressing). The hierarchical labelling will initially be manual, performed using labelling tools such as the AMIDA NITE XML tool. The resulting labelled data will subsequently be used to train and evaluate alternative automatic systems.

1.2 Objectives for months 1 to 16

The most important immediate target for WP3 is MS5, the completion of the first action prediction model, in month 12, leading to the evaluation of the first CogWatch prototype system in month 16. The main objective of this report is to define this model and explain the rationale for the choices that have been made.

1.3 Interactions between the partners

Figure 1 shows the interactions between Work Packages in the first 12 months of the CogWatch project, which are most important from the perspective of WP3.

- The CogWatch sensor recorder, developed under WP2, is software to record the outputs of the sensors (force sensitive resistors, accelerometers, body-motion trackers, RF-ID tags) attached to the task objects and the participant's body as well as hand position information from Kinect during the execution of a task in the trials performed in WP1. These outputs will be saved in files and used to train the AAR system.
- Within WP2, UOB and UPM will jointly determine the precise set of sensors that will be available for the sensor recorder and the first prototype CogWatch system.
- WP1 will conduct trials with healthy subjects and patients. The outputs of the sensors will be recorded during these trials and provided to WP3, where they will be used to train and test model and pattern based action recognition systems.
- Within WP3, UOB and TUM will collaborate on the development suitable Task Models.
- WP1 will provide WP3 with a specification of the outputs that are required from the prototype CogWatch system, to enable useful feedback and cues to be provided to the participant.
- WP2 and WP3 will collaborate on system integration, to ensure that components developed under WP3 will integrate properly into the CogWatch system.

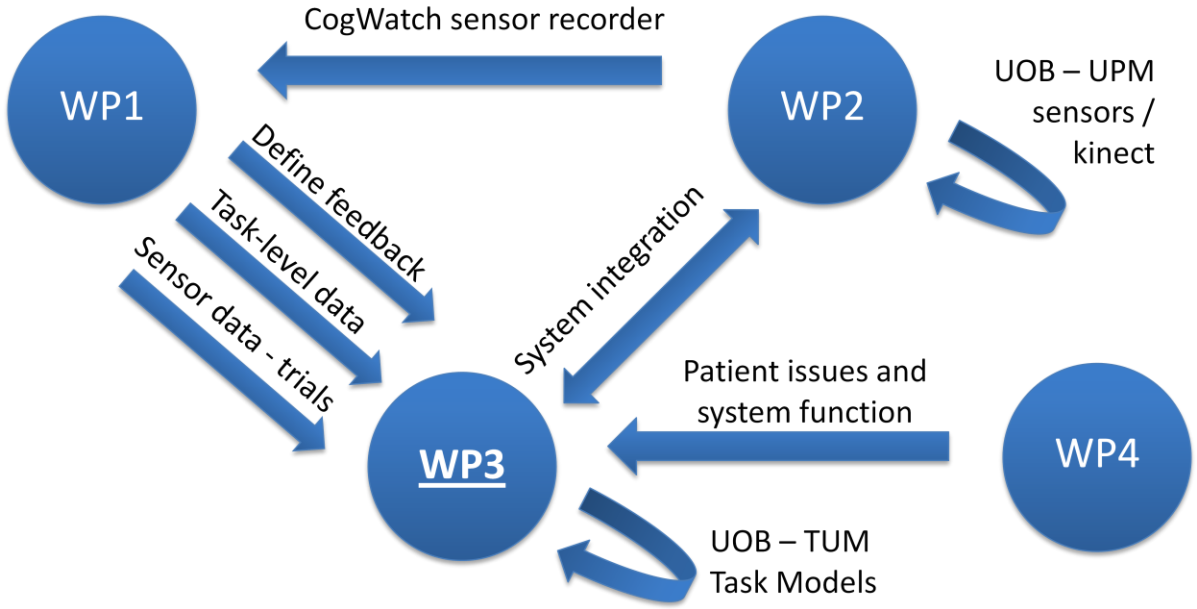

Figure 1: WP3 view of interactions between CogWatch Work Packages

1.4 Proposed timetable for the report

- First draft end April
- Presentation at UPM meeting in Madrid, May 10/11
- Completion mid May
- Reviewed via UPM Quality management mid May
- Copy editing end May

- Submitted to EU end May (well within 45 days of nominal deadline).

2. SCENARIO

During the CogWatch project a number of different ADLs will be considered. However, the application for the first prototype system is a simple tea-making task.

2.1 The tea-making task

The tea-making task, together with the rationale for choosing it as the task for the first CogWatch prototype system, are described fully in CogWatch deliverable D1.1 “Report on scenarios”. Figure 2 shows a hierarchical tree based description of the task, taken from D1.1.

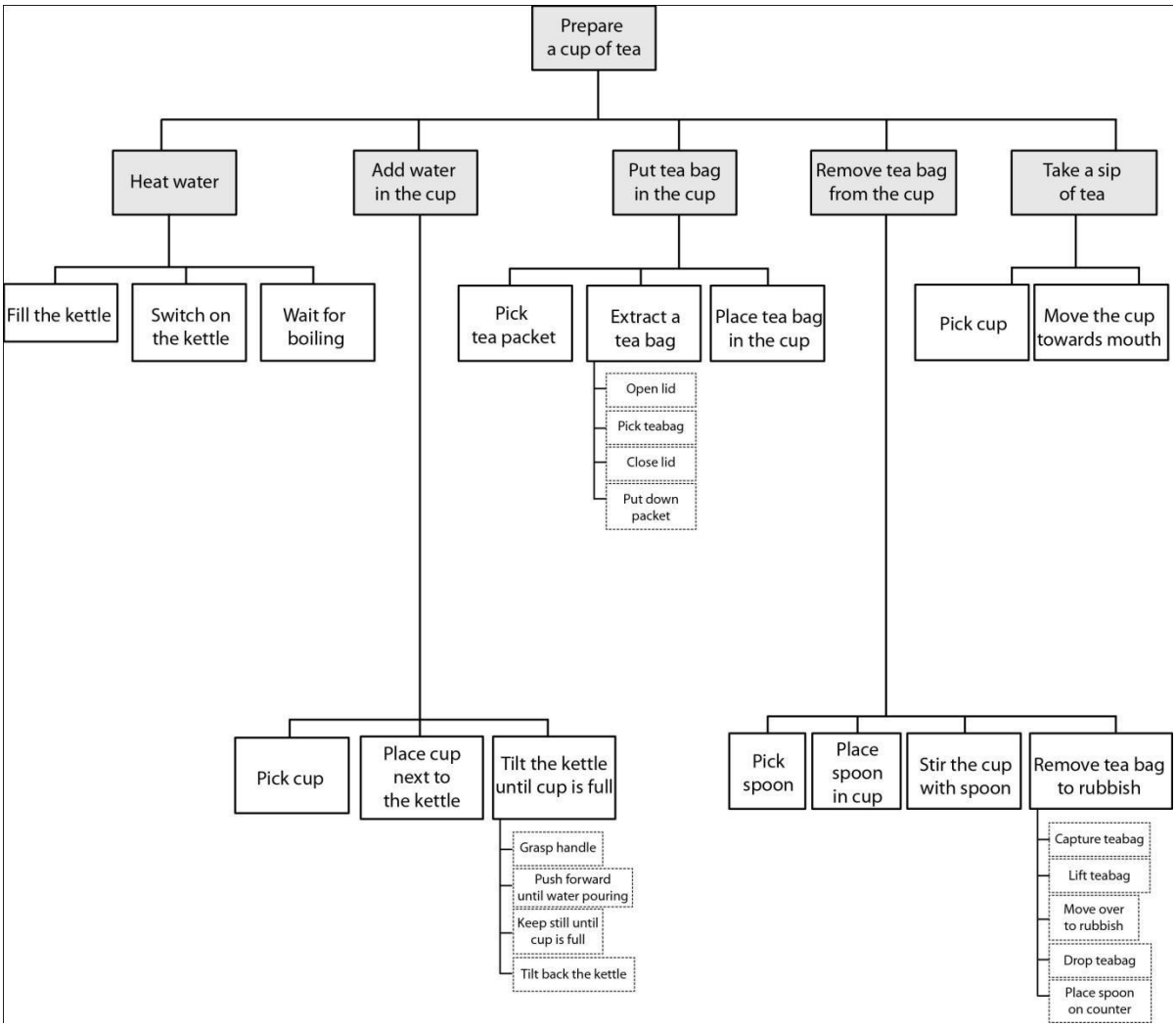

Figure 2: Hierarchical tree representation of the tea making task (from D1.1 Report on scenarios)

2.1.1 Terminology

In this report the following terminology is used to describe items in this hierarchical tree:

- The goal corresponds to the root of the tree and is “prepare a cup of tea”. The whole description is aimed at achieving the goal (which, therefore, can be defined in terms of completion)
- The items at the next level of the tree, namely “heat water”, “add water to cup”, “put tea bag in the cup”, “remove the tea bag from the cup” and “take a sip of tea”, will be referred to as sub-goals.
- The items at the third level of the tree, for example “fill the kettle”, “switch on the kettle” and “wait for boiling”, will be referred to as tasks.
- The leaves of the tree, for example “grasp handle”, “push forward until water pouring”, “keep still until cup is full” and “tilt back the kettle”, will be referred to as sub-tasks.

2.2 Strengths and limitations of hierarchical descriptions

2.2.1 Utility of the hierarchical task description

From the perspective of activity recognition, a diagram like Figure 2 is useful because by breaking the ADL down into its basic component actions it suggests how the action recognition model should be structured. It also exposes relationships with other pattern classification problems. For example, a similar hierarchical description is used in automatic speech recognition (ASR), where an application is described in terms of a grammar, the grammar is described in terms of words, and the words in terms of their phonetic pronunciations. These parallels motivate the proposed application of hidden Markov models (HMMs) to action recognition described in Section 6 of this report.

2.2.2 Limitations of the hierarchical description

A significant limitation of the hierarchical description in Figure 2 is that it gives no indication of the sequential structure of the task. For example, it does not specify whether the sub-goals should be executed in parallel or in sequence, or more generally whether the time intervals over which sub-goals are executed can overlap or must be disjoint. The same is true at the task and sub-task levels. From a mathematical perspective the issue is whether, at the sub-goal, task or sub-task levels, an instantiation of the ADL should be thought of as a well-ordered set or a partially-ordered lattice.

In the specific case of tea making, and assuming that only one participant is involved, a key issue is whether the participant tries to achieve the goal using one hand or both hands. If only one hand is used then the sub-goals, tasks and sub-tasks will take place in sequence. However if both hands are used there is clearly an opportunity for overlap (for example, the participant may pick up the cup with one hand and a tea bag with the other).

As substantial quantities of real data collected from healthy participants and patients executing the ADL become available, these issues will be resolved. However in the short-term (which includes the period up to the creation of the first prototype) it will be necessary to make assumptions.

2.2.3 Implications for activity recognition

In summary, the choice of tea-making as the scenario for the first CogWatch prototype system has a number of implications. First, the description of tea-making as a hierarchical tree facilitates the use of methods from other fields, such as ASR. However, the sequential structure of the tea making activity, and in particular whether its components can be thought of as a well-ordered or partially-ordered set, has implications for the choice of architecture for the action recognition system. These issues are discussed in more detail in Section 6, which discusses the proposed approach to action recognition.

3. INSTRUMENTATION

The action recognition system will track a participant's progress in achieving the goal by processing the outputs of sensors attached to the tools and objects involved, sensors attached to the participant's body, and estimates of body position resulting from applying image processing to video data. The set of potential sensors and the rationale for the choice of sensors is presented in detail in CogWatch deliverable D2.1. The following text is based on D2.1 and is included here for completeness.

3.1 Sensor technologies

The sensors that are potentially available to the first prototype CogWatch include:

3.1.1 Vision-based systems

Vision-based hand position data estimated, for example using the Kinect system,

3.1.2 Radio frequency identification (RFID)

Radio frequency identification (RFID) technology can be used to wirelessly detect and identify objects fitted with RFID tags. The tags are small and require no power supply of their own so that they can be inconspicuously attached to objects. An RFID reader can be attached to the user's wrist so that tagged objects that are handled by the user fall within the detection range of the reader. For the CogWatch project this method will be used to supply the computer with a history of all objects that have been interacted with. This data will be used to support the recognition algorithms that will be used to detect and identify errors in the sub-goals and tasks performed by patients.

This RFID based method has been used in past research to automatically recognise ADLs by looking at the sequence of objects interacted with. Similar methods could be used to detect sequence and omission errors in tasks, which is a required capability of the CogWatch system. However a system that only uses RFID cannot detect what actions are performed with a single object – only the fact that it was handled. Many objects will require other sensors to enable the detection of usage errors. For very small objects, which are too small to support wireless electronics and a battery, RFID is the only suitable method of detecting interaction. Camera based methods can be used but the camera's sight of small objects is easily obscured.

There are many options for the tags that can be used. Due to high availability and low price, Mifare Ultralight tags are suitable for this project. The 45mm RFID sticker labels are well suited for tagging large items such as mugs, and 20mm sticker discs and 13mm laundry tags are available for tagging smaller objects.

3.1.3 Accelerometers

Accelerometers will be attached to the tools and objects involved in the task. Usage errors, such as holding an object in the wrong orientation or shaking a kettle instead of pouring from it, cannot be detected from simple RFID sensors. The use of vision based techniques using the Kinect camera system may make it possible to detect these kinds of error; however this can be very difficult and may fail completely if vision is obscured. For this scenario accelerometers attached to the objects can be extremely helpful in providing the

data required. Accelerometers provide data about movement as well as orientation. They are commonly used in many activity recognition applications due to the quality of data they provide. They are also very small and have low power consumption, making them ideal for embedded electronics. For these reasons it can be argued that accelerometers should be embedded on all mobile ADL objects that can accept the electronics without hindering use.

3.1.4 Force sensitive resistors (FSRs)

FSRs can be used to measure a physical force applied to their surface. Applying these sensors to the bottom of objects such as a mug or a kettle results in data that can be used in two ways:

- The force data could be used to detect that the object has been picked up off the work surface, as the object's own weight would no longer be detected.
- If the quantity of water within the object changes (relevant to mug/kettle in the tea-making task) then this would be detected by the force sensors through a change in the weight in the object.

3.1.5 Force sensitive handles (FSHs)

The instrumented handle is a wireless sensor for handles of objects such as cutlery and tools. The instrumented handle is not just embedded electronics; it is a replacement for the entire handle of the object. The handle incorporates an accelerometer plus strain gauges fitted to its sides to measure changes in grip force.

Objects with handles, such as cutlery, are handled in a rather complex manner, and correct grip plays an important role in successfully completing any task. Grip force sensing can be used to detect errors in patients involving significant insufficient or excessive grip force. Furthermore, the changes in grip over time can be used to assist the automated recognition of the activity that is being carried out with the instrumented object. Research has been done with multiple prototypes to ascertain the potential of strain gauge grip sensing in the CogWatch project.

The sides of the instrumented handle are made out of strips of steel. Strain gauges are used to measure the small amounts of bending that occurs when any force is applied. This gives a generalised measure of grip force applied across the handle. Adding more than one strain gauge to each side of the handle could help to build a more accurate calibrated model of grip force, but the increase in circuit size, power consumption and cost would be unacceptable for the CogWatch project.

An initial prototype force sensitive handle with only 2 grip sensing sides was rejected because during use a lot of grip force is applied to the wrong sides of the steel strips, where it cannot be properly detected. With three sensing sides most grip patterns apply force in the correct sensing directions and so this has been chosen as the ideal design.

3.1.6 Wireless connectivity

Bluetooth is used for the wireless connection between the sensor units and the computer. Bluetooth is a well supported, low power standard for wireless communication for small devices, with a communication suited to rooms in a standard house. Due to its wide support, using Bluetooth opens up the possibility of direct connection to many devices such as mobile phones and tablet computers, allowing for future expandability.

Zigbee wireless (and similar proprietary protocols such as MiWi) based technologies may provide better power management (and hence better battery life) than Bluetooth and more flexibility for the management of large networks of sensor units. However, the amount of sensor data that can be transmitted using such technologies is much smaller and the implementation would be much more complex. The viability of such technologies cannot be confirmed until the minimal useful sensor data is confirmed through trials.

3.2 Functionality of sensors for tea-making

Figure 3 indicates the utility of each type of sensor for the tea-making task. The rows of the table correspond to different sensors and the columns to 'events' that may or may not be detectable with that sensor. In the figure, a solid disk indicates that a particular event can be detected using the relevant sensor, and a hollow disk indicates that detection may be possible. At present this table is based largely on knowledge and expectation rather than practical experience with the sensors. The table will be updated as the project progresses.

In Figure 3 'contact' refers to actual contact or very close proximity to an object. It is distinguished from proximity to differentiate between Kinect and RFID technologies. The extent to which proximity is detectable by RFID will depend on the range of the receiver, which varies between 30cm and 1cm according to the literature. FSRs are the only type of sensor capable of detecting weight changes, and the FSH is the only sensor capable of measuring grip.

The event 'use' refers to factors such as tilting a cup to take a sip of tea, or tilting a milk jug to pour milk into a cup. Although vision-based sensors and the FSH will give clues to this type of activity, it is expected that the most reliable information will come from an accelerometer. It should also be noted that some types of event will be best detected using combinations of sensors. For example, in pouring milk from a jug into a cup, the change in orientation of the jug will be indicated by accelerometer data, but also through the FSRs attached to the cup due to the increase in its weight. The issue of 'coupling' between the sensors will be revisited in Section 6 where pattern-based action recognition is discussed.

	Contact	Proximity	Weight	Lift	Grip	Motion	Position	Use
Kinect	○	●		●		●	●	○
RFID	●	○						
Accelerom.	○			○		●		●
FSR			●	●				
FSH	○				●			○

Figure 3: Functionality of sensors for the tea-making task

3.3 The CogWatch instrumented coaster

3.3.1 Specification

The CogWatch Instrumented Coaster (CIC) is an electronic coaster (or drink mat) which can be attached to the base of a mug or jug.

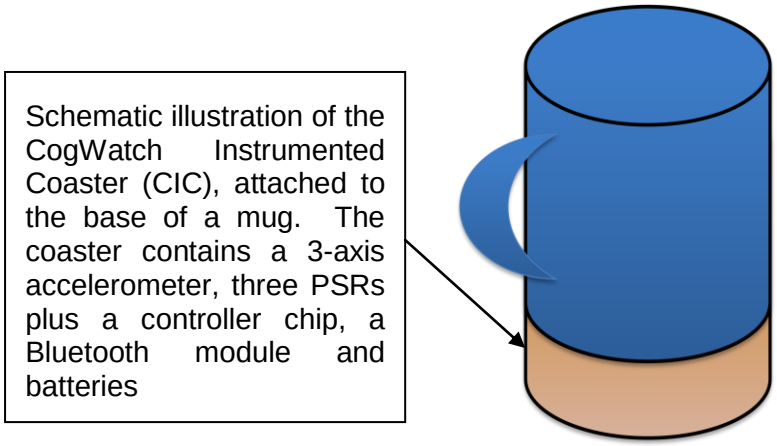

Figure 4: The CogWatch instrumented coaster attached to the base of a mug

The CIC contains a 3 axis accelerometer, three FSRs, a microcontroller and a Bluetooth module. Full details of the CIC are included in CogWatch report D2.1.

3.3.2 Functionality of the CIC

Referring to Figure 3, for an object fitted with a CogWatch instrumented coaster it will be possible to detect the fact that the object has been lifted from the work surface, that the object is in motion, changes in weight of the object (due to changes in its contents) and the use of the object (for example, tilting a cup to take a sip of tea).

3.3.3 Output of the CIC

The CIC will output 6 measurements: *x*, *y* and *z* measurements from the accelerometer plus 1 measurement from each of the FSRs.

3.4 Instrumentation in the first CogWatch prototype

The proposed instrumentation of the first CogWatch prototype is shown in Figure 5. The precise instrumentation of the kettle will depend on whether it is a normal free-standing kettle, with a stand connected with a mains connection (in which case a CIC and an RFID tag are appropriate), or a kettle mounted in a frame for safety reasons, so that it can only be tipped in the correct manner.

Object	Sensors	Number of outputs
Cup/Mug	CIC	6
	RFID tag	1
Jug/Milk Container	CIC	6
	RFID tag	1
Sugar Container	CIC	6
	RFID tag	1
Kettle*	CIC	6
	RFID tag	1
Tea spoon	RFID tag	1
Tea bag	RFID tag	1
Left hand	Kinect	3
Right hand	Kinect	3
Total		**36**

Figure 5: Instrumentation for the 1st CogWatch prototype

4. THE FIRST PROTOTYPE ACTION RECOGNITION SYSTEM

Figure 2 shows the action recognition system that will be at the centre of the M16 CogWatch prototype system. The purpose of the diagram is to indicate the component parts of the system that need to be considered, and it should not be seen as a definitive statement of the system architecture. For example, it would be equally valid to show the complete system as a subset of the task model.

4.1 Components of the action recognition system

4.1.1 Sensor data capture and pre-processing

The left hand side of the figure shows the array of sensors attached to the objects involved in the task and to the participant's body. These are input via a wireless connection to the action recognition system. In the pre-processing (feature extraction) stage, the synchronized measurements from the sensors are transformed into a feature vector. The objective being to emphasize components of the measurements which are important for activity recognition, and to de-emphasize those which are not (the noise).

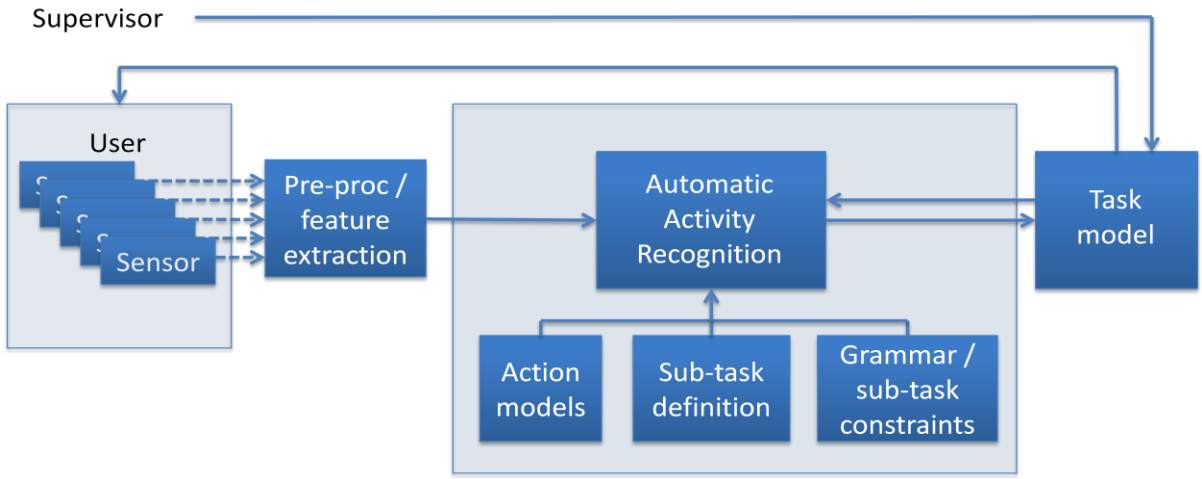

Figure 6: Prototype CogWatch action recognition system.

4.1.2 Automatic action recognition (AAR)

The purpose of the AAR component of the system is to interpret the sequence of feature vectors constructed from the outputs of the sensors in terms of the tasks and sub-goals performed by the participant. The key challenge is to accommodate intra- and inter-participant variability in these measurements, for participants engaged in the same activity. There are many possible approaches to this problem in the pattern recognition literature, and a number of these have been applied to action recognition from instrumented body and object data in the past. Candidates include Hidden Markov Models (HMMs) (e.g. Patterson et. al. (2005), Wang et al. (2007), Ward et al. (2006)), Dynamic Bayesian Networks (DBNs) (e.g. Patterson et. al. (2005), Wang et al. (2007)), Decision Trees (e.g. Hong et al. (2008)) and Finite State Machines (e.g. Stiefmeier et al. (2008)).

The CogWatch prototype places very specific requirements on the pattern recognition component. In addition to achieving sufficiently good activity-recognition accuracy to support the CogWatch application, the system must run in real-time and must be capable of processing a continuous stream of data. Continuous recognition is a much more difficult problem than many of the 'activity classification' tasks that are described in the activity recognition literature. In classification, the data is pre-segmented so that the pattern recognition system is presented with a finite sequence of feature vectors that corresponds to a single activity, and the task is to correctly identify that activity. In continuous recognition the activity boundaries and the number of activities that have been performed by the participant are unknown, leading to potential segmentation, deletion and insertion errors in addition to classification errors.

In the CogWatch prototype we propose to use HMMs for action recognition. HMMs have been applied to automatic speech recognition (ASR) since the mid 1970s (Baker, 1975). They were popularised a decade later by researchers at AT&T Bell Laboratories (Rabiner et al., 1985) and now form the basic 'acoustic-phonetic' component of all commercial speech recognisers and the majority of research systems. Bridle et al. (1982) showed how Viterbi decoding, the recognition algorithm that underpins HMM-based ASR, could be extended from simple word recognition to connected speech recognition, and a technique called 'partial traceback' (Spohrer et al., 1980) allows these systems to run continuously without running out of memory.

In a HMM-based speech recognition system the grammar, or syntax, defines the recogniser's vocabulary and captures constraints (normally in the form of probabilities) on word order. The pronunciation dictionary expresses each vocabulary word as one or more sequences of phones (allowing for alternative pronunciations), and the phones are modelled as context-sensitive HMMs. By analogy, in our proposed pattern based activity recognition system, the task model expresses a goal as a hierarchy of sub-goals, which in turn are described in terms of (one or more) sequences of basic tasks. We propose to model either these basic tasks or the sub-goals using HMMs.

There are, of course, some important differences between speech and activity recognition. If it is assumed that words in ASR correspond to sub-goals in activity recognition, then the vocabulary size is typically very much smaller in activity recognition. In addition, speech is sequential. One word follows another, and there is no possibility that a speaker will produce two words at the same time. The same is not true in activity recognition, where a participant may, for example, move a jug and use it to pour milk into a mug with one hand, while at the same time (or at least in overlapping time) initiate another action with the other.

These issues, and the definition of the HMM-based activity recognition system, are explored in more detail in Section 6.

4.1.3 The Task Model (TM)

The purpose of the Task Model (TM) is to collate the sub-goal labels that are output from the AAR system as sub-goals are achieved and use these to maintain a record of the participant's progress with respect to goal completion. The TM needs to be able to determine whether or not a particular set of sub-goals is likely to be extensible to a successful achievement of the whole goal, and it must contain sufficient information to allow useful cues and feedback to be given. A range of psychological, ergonomic and statistical candidates for the CogWatch TM are discussed in the next Section.

5. MODELS OF HUMAN TASK EXECUTION (THE "TASK MODEL")

5.1 Role of the Task Model (TM) in the CogWatch system

In the context of automatic human activity recognition in the CogWatch project, the term 'Task Model' (TM) refers to a computational model that satisfies the following requirements:

5.1.1 Inference of the 'belief state'

The TM should be able to infer the participant's status with respect to the current goal from the outputs of the decoder. With reference to spoken dialogue processing (e.g Levin et al., 2000) this status is be referred to as the system's 'belief state'. In a probabilistic system a more subtle objective is to maintain a probability distribution over the set of possible belief states as the participant's sequence of actions unfolds.

5.1.2 Sub-task history

Part of the Belief State should be the sub-goal history - a record of the sub-goals that have been completed by the participant at any point in the execution of the activity. This will be needed by the system designer to construct appropriate feedback or cues to the participant.

5.1.3 Sub-task prediction

Given the current Belief State (or distribution over possible Belief States) the TM should be able to predict the next most probable sub-goal. Again, this will be needed by the system designer to construct appropriate feedback or cues to the participant.

5.1.4 Failure prediction

The TM should be able to detect when the participant is unlikely to achieve the goal successfully. This is needed to prompt the provision of feedback or cues to the participant. The implication of this requirement is that the TM belief state should include some measure of the 'cost' of reaching the current state. The cost could be a function of transaction time or the number of incorrect actions performed.

5.1.5 Cue / feedback generation

The contents of the belief state should be sufficient to enable the system designer to synthesise useful cues and feedback to the participant. In general, the feedback and cue will depend on the specific error that has occurred, and the information contained in the belief state must be sufficient to support this. For example, the cue will depend on the best next action given past history, and whether the error committed needs corrective action (for example, "remove X from Y", and if so this action may also be subject to error). The precise design of the feedback or cues should be specified in WP1/4 via WP2. For WP3 the requirement is to liaise with WP1, WP2 and WP4 to determine the information that is needed to construct appropriate cues or feedback.

5.1.6 Task execution recording

The TM must be able to record a specific sequence of sub-goals which are sufficient to achieve the goal, demonstrated by the participant's helper or clinician. The TM must be

configurable automatically to incorporate this sequence (or minor variations of this sequence) as a guide or constraint for the participant.

5.1.7 Knowledge-driven versus data-driven

Experience in other fields which involve modelling human behaviour suggests that in order to model variability in human performance, it is necessary to use complex models whose parameters (and to some extent structure) are learnt automatically from data. Therefore, once sufficient data becomes available through WP1, it should be possible to use this data to train the parameters and structure of the TM. However, in the first 12 months of the project this data will not be available. Hence the TM in the first prototype CogWatch system will need to be at least partially manually configured.

5.1.8 Psychological plausibility

Ideally, the TM should be psychologically plausible. In order to satisfy 3.4.1.1 to 3.4.1.6 the model must be an accurate model of human goal execution. In other words, it must give high scores to sequences of actions that a patient (or a relevant clinician) would consider to be a successful achievement of the goal, and low scores to sequences which are unlikely to result in successful goal completion, or which simply take too long. In addition it is desirable (but not essential) that the model is 'psychologically plausible'. For example, the mechanisms that the model uses to characterise task execution might reflect known cognitive processes, or the model might respond in a similar way to a human participant to factors such as distracter objects.

5.1.9 Computational utility

The TM must be computationally useful. In other words it must be sufficiently well-defined to be implemented in software, and, in the case of the CogWatch prototype, it must be sufficiently simple for real-time execution.

In the next section possible TMs are reviewed against the criteria in 3.4.1.

5.2 Candidates for the CogWatch Task Model

5.2.1 Contention Scheduling Model – Norman and Shallice (1986)

Norman and Shallice (1986) proposed a dual-systems account of action selection. According to their Contention Scheduling Model, routine well-practiced actions are controlled via the Contention Scheduling System (CSC), whereas actions that require attentional supervisory control are governed by the Supervisory Attentional System (SAS). In the SAS, Normal and Shallice propose that actions are represented as schemas and that these schemas reflect the learned relationship between an action and a set of features perceived in the environment. Thus, an individual will learn to pair sets of features with specific actions. However, there might be situations in which a set of features could be related to more than one action, e.g., when standing in front of a door, one might perceive a set of features (handle, hinges etc.) and have a set of actions that could be performed (such as turn handle and push door, or turn handle and pull door). In situations where there is not just one action that is immediately apparent to apply, one needs to select between different schemas. Contention Scheduling is the basic mechanism by which one schema is selected over another competing schema. Contention Scheduling assumes that selection is dependent on the schema being activated above threshold, and that schemas are triggered

via both top-down and bottom-up processes. Further, when a schema is selected it activates its hierarchical 'component' schemas and/or controls the execution of the requisite actions. When the action sequence is executed, the component schemas form a "horizontal thread", which is a processing structure that enables the routine action to be carried out without intervention.

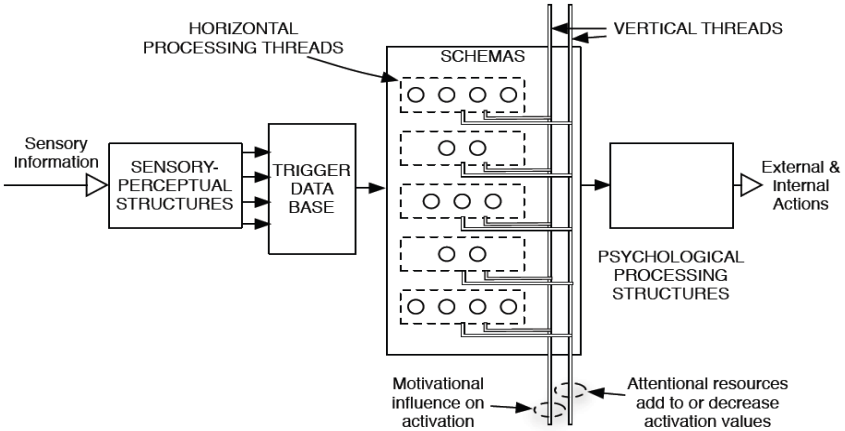

Figure 7: The Contention Scheduling model (Adapted from Norman and Shallice, 1986)

The Contention Scheduling Model also proposes that attention is not essential for the execution of routine well-practiced actions, but that action selection is modulated by attention from the SAS. Within the Contention Scheduling Model, the SAS forms the "vertical thread" which is activated when a novel task is performed, or when attention to the task is required.

The Contention Scheduling Model is a psychological model that requires a total of eight parameters that control the flow of activation within and between various networks of the model. Two parameters control the more general aspects of network dynamics, namely rest activation and persistence. Rest activation refers to the activation level to which activations in all domains tend in the absence of any net input. Persistence refers to the degree to which activation values persist over time with a net input of zero. Together, these parameters coalesce and provide smooth activation profiles through processing. A third parameter governs the standard deviation of normally distributed random noise that is added to the net influence in all domains, and accounts for variation in behaviour that occurs in biological systems. If the noise (or random variability) is set too high, then it may lead to spurious action selection. The CS model also includes three balance parameters which govern the contribution of the various activation influences to the net input within all networks. The four inputs (excluding noise) are: self influence, lateral influence, an internal influence, and an external influence. Self influence and lateral influence are competitive processes, with the parameter Self:Lateral controlling the relative proportion of self influence and lateral influence in the final influence on a node. Internal influence and external influence are non-competitive, and the Internal:External parameter controls the proportion of internal and external influence on a node. The third balance parameter controls the proportion of competitive and non-competitive influences that contribute to the total excitation or inhibition of a node. Within the schema network, a ninth parameter, the selection threshold, governs schema selection and specifies the activation level above

which a schema node must be excited in order to be selected. When competition is functioning appropriately, the model is relatively insensitive to this parameter. However, if the selection threshold is extremely high (> 0.95), schema selection tends to fail because schemas cannot become sufficiently active. If the selection threshold is too low (< 0.50), schemas tend to be selected before competitive processes have achieved their purpose, and as such spurious selection of schemas are more likely to occur.

5.2.1.1 Errors:

Cooper & Shallice (2000) evaluated the Contention Scheduling Model in a coffee preparation task (Schwartz et al., 1991, 1995, 1998), with the goal to assess the models ability to produce well-structured action sequences in complex tasks, to determine the susceptibility of the normally functioning model to action lapses (capture, omission, anticipation, perseveration, object substitution), and to assess whether the model would yield behaviour that is qualitatively similar to that observed in individuals with action disorganization syndrome (Schwartz et al., 1991, 1995, 1998). The results indicated that the CS model is able to explain actions errors that occur during routine tasks, such as logging into a personal email account using work login details. These 'capture errors' occur when environmental cues for a different, but familiar, action 'capture' behaviour, and are thought to occur when insufficient attention is paid to the intended task.

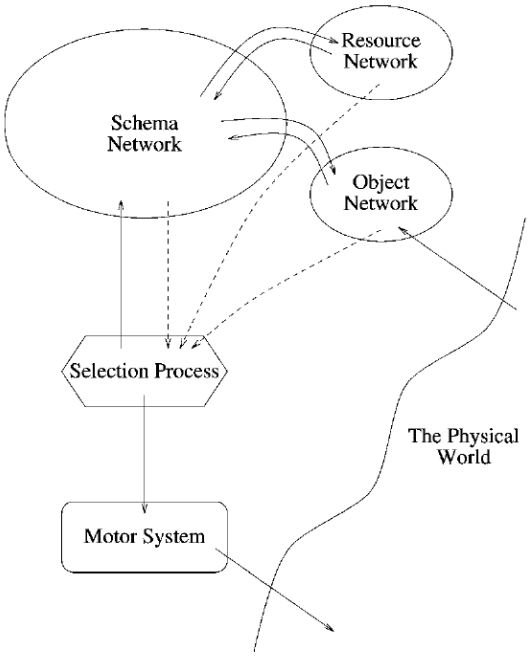

Figure 8: Principal components of the Interactive Action Model (Cooper & Shallice, 2000).

5.2.2 The interactive action model (IAN) - Cooper and Shallice (2000)

The general principles of the Contention Scheduling Model were instantiated in the Interactive Action (IAN) model of Cooper and Shallice (2000). Central to the IAN model is the hierarchically organized schema network, within which individual nodes correspond to action schemas. Each node has an independent activation value, which are triggered when a node exceeds a given threshold. These activations can occur via top-down-,

environmental-, lateral-, and self- influences. The IAN model proposes that selection of high level schema (e.g., 'make coffee') result in high excitation of component schema. In contrast, selection of low level schema (e.g., 'pick up spoon') enables allocation of object representation and resources from two separate networks, which precedes action execution.

As in the CSC model, activation flow within the schema network is controlled via four parameters: *SS* (degree of self influence), *LS* (degree of lateral influence), *IS* (degree of intrinsic, or schema-to-schema, influence), and *ES* (degree of extrinsic, or object-to-schema, influence). Similar parameters control activation flow within and between the other networks, with the subscript *S*, *O* and *R* used to indicate schema, object representation, and resource network parameters respectively. The impact of activation on any node within each network is subject to normally distributed random noise, with the standard deviation of the noise distribution is given by the noise parameter, *N*. Last, the parameter, *P*, controls the degree to which activation of nodes persists from cycle to cycle in the absence of other influences.

5.2.2.1 Errors

The IAN model was evaluated by Cooper et al. (2005) during a lunch packing task to data reported in healthy and neurological patients (cf. Schwartz et al. 1998). To this end, an appropriate schema hierarchy was first developed for the lunch packing task, and featured an object representation network that comprised of nodes for each object that could be used in the task. Each object was represented by a set of features that related to the object (e.g., shape and size), and were used to determine how the objects behaved when acted upon, and the extent to which the objects triggered schemas or were triggered by schemas.

Because the model's behaviour is strongly dependent on the relative activation flow within and between the three activation networks, Cooper et al. (2005) performed a number of simulations to determine the appropriate values for each parameter that would yield well-structured behaviour. When $SS = 0.23$, $LS = 0.46$, $IS = 0.50$, $ES = 0.10$, $P = 0.87$ and $0.00 < N \leq 0.01$, Cooper et al. (2005) found that the model was able to perform the complete task without error. Deviations from this parameter space (e.g., by increasing the noise parameter) result in errors that are similar in frequency and type observed in healthy individuals. Cooper et al. (2005) then tested the ability of the IAN model to account for behavior in neurological populations in three ways. First, they varied the balance between top-down excitation of schemas and bottom-up, environmental, triggering of schemas. Second, they increased noise in the schema network. Third, they increased noise in the object representation network.

Simulations in which the top-down/bottom-up balance within the schema network were altered revealed that while an imbalance in top-down and bottom-up activation reproduces significant correlations between accomplishment and error rates, it does not account for the relative frequency of commission errors that have been reported in behavioral studies (cf. Schwartz et al. 1998). Further, manipulating the balance of top-down/bottom-up activation fails to replicate behavioral observations in the presence of distractors. Cooper et al. (2005) hypothesized that the inability of the IAN model to account for behavior in ADS patients is due to the use of strict pre- and post-conditions introduced in order to simulate behaviour on the more complex lunchbox task, and/or the scoring system used was too conservative and did not count action additions as errors.

Cooper et al. (2005) also ran simulations designed to degrade the effectiveness of selective schema network excitation and inhibition by manipulating the noise parameter (i.e., the standard deviation of noise within the schema network ranged from 0.01 to 0.25 at intervals of 0.01). The results indicated a strong positive correlation between task accomplishment score and the number of omission errors, and a mild negative correlation between accomplishment score and commission errors. The simulation results also demonstrated an increase in both omission and commission errors with increasing levels of noise. There was also a strong influence of task distractors on total errors, the number of omission errors, and the number of commission errors. Based on these results, Cooper et al. (2005) concluded that increasing noise in the schema network captures many of the hallmark characteristics of ADS.

Simulations in which noise within the object representation networks were increased (i.e., the standard deviation of noise within the object representation networks ranged from 0.01 to 0.25 at intervals of 0.01) was also examined. As in the previous simulations, the simulations revealed a strong negative correlation between accomplishment score and the number of omission errors. However, in contrast to the two aforementioned simulations, there was a mildly positive correlation between accomplishment score and the number of commission errors. The simulation results also revealed a number of findings that are similar to that produced by noise in the schema network and to neurological populations: omission, substitution and action addition errors increased when distractor objects were present, omission errors are the most common type of error, and the proportion of commission errors is greater for low error producers than for high error producers. These results were interpreted as evidence that the interactions between schema nodes and object representation nodes are reciprocal, such that noise in one network is propagated to the other network. The effect of noise is modulated by schema triggering functions, with noise in the object representation network causing a degradation of schema excitation activation.

Based on the simulation results, Cooper et al. (2005) suggested that errors occur in response to the interaction between environmental and top-down activation influences. 'Perseveration' errors occur most frequently and are performed when a schema is not deselected, which result from either too much self-activation or a lack of inhibition. 'Capture' errors occur when an environmental source of activation is relatively stronger than the top-down activation. 'Omission' and 'anticipation' errors occur due to insufficient activation of appropriate schema. 'Omission' errors are a result of poor environmental cues or self-activation, whereas 'anticipation' errors occur when an action cannot take place because a pre-condition has not been met (e.g., toast cannot be buttered if the lid is still on the butter).

5.2.3 Simple recurrent network (SRN) - Botvinick and Plaut (2002, 2004)

In contrast to the two aforementioned action recognition models (i.e., CSC, IAN), the Simple Recurrent Network (SRN) model proposed by Botvinick and colleagues (Botvinick and Plaut 2004; Botvinick et al. 2009) is based on recurrent connections within a network mapping from environmental inputs to actions in everyday tasks. The SRN model makes a distinction between the hierarchical structure of a task and its cognitive representation. Sequential behaviour is modelled using a parallel distributed processing (PDP) account, which can be thought of as a general learning mechanism that learns from samples in the environment. Whereas the CSC and the IAN models feature hierarchical schema structures, the SRN postulates that understanding the task structure is controlled via the emergent properties of

the processing system. The PDP model is comprised of individual nodes, which are activated via excitation and inhibition from the nodes that are linked to it (Botvinick and Plaut 2004). The SRN is divided into three layers: an 'input' layer that provides a representation of the perceived environment, a 'hidden' layer which transforms the input information, and an 'output' layer that represents the action taken. In the SRN model, every hidden node is connected to all nodes in both the input and output layers, and each unit in the "hidden" layer is connected to every other unit in the hidden layer. Furthermore, each step of processing carries information about the state of the system at the previous time point, and as with recurrent connectivity principles, information can be preserved and transformed across time. In this way, the SRN is sensitive to temporal context.

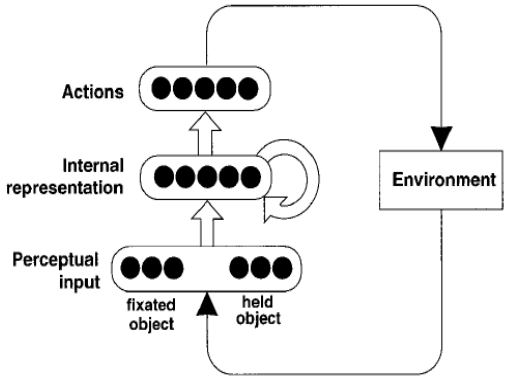

Figure 9: The architecture of the overall Simple Recurrent Network (Botvinick & Plaut, 2004). Open arrows indicate the connections between units in each layer of the system.

5.2.3.1 Errors:

Botvinik and Plaut (2004) tested the efficacy of the SRN model in a coffee preparation task. Performance of the SRN model was first compared with data collected from healthy individuals who did not make performance errors. The model was then subjected to action slip by adding zero-mean, normally distributed, random noise to activation values in the hidden layer at the end of each processing cycle. Lastly, the model was evaluated again the behaviour of patients with ADS. The results indicated that the SRN was able to model error-free behaviour, as well as task performance with low error rates (noise values below 10%). At low noise values, the model produced errors that are typically observed in patients with neurological impairments; namely omissions and errors in action sequence. Similarly, when the model was evaluated with respect to the performance of ADS patients, it reproduced several hallmark characteristics of ADS behaviour: performance deteriorated as the level of noise increased, errors were either of the omission or action sequence type, and there was a marked increase in the proportion of omission errors with overall error rate.

5.2.4 Automated probabilistic models of everyday activities (AM-EvAs) – Beetz, Tenorth, Jain, and Bandouch (2010)

An alternative approach to action recognition modelling has been developed by the Intelligent Autonomous Group (IAS) at the Technical University of Munich (TUM). The automated probabilistic models of everyday activities (AM-EvAs) advanced by Beetz et al.

(2010) are detailed, comprehensive models that describe human actions at various levels of abstractions: from raw poses and trajectories, to motions, actions, and activities. AM-EvA's consist of automated activity observation systems, interpretation and abstraction mechanisms for behaviour and activity data, a knowledge representation and reasoning system for symbolically representing the activity data, and a query system that allows AM-EvA's to answer semantic queries about the observed activities. A knowledge based framework integrates methods for human motion tracking, for learning continuous motion models, for motion segmentation and abstraction, and for probabilistic reasoning. At each level, all information in the system is represented in combination with its semantic meaning, which enables automated reasoning on the observations.

For the purposes of the CogWatch Task Model, the most important aspect of AM-EvA's is the partial-ordered learning models. The advantage of this approach is that the system *learns* a model that is able to describe complex tasks including their partial order from observed data. Using Bayesian Logic Networks (BLNs), the joint probability distribution over the actions in an activity, their properties, and their pairwise ordering constraints, can be extracted. These pairwise ordering constraints result in statistical models that describe the partial order imposed on all actions in a task, as well as the general relations between consecutive actions and their properties. From training data partial-ordered models can learn which actions are relevant and which ordering relations are important, such that actions that occur in all observations of a task are considered more relevant than those that are only rarely observed, and ordering relations that consistently hold are also more likely to be important.

The approach of the partial-ordered learning models is illustrated in Figure 10. The task of making brownies can be reached by significantly different action sequences (Figure 10: left most panels), which may be influenced by individual preferences or task context. The colors indicate the dependencies among the actions, which are also shown in the partial-order graph (Figure 10: rightmost panel). The arrows in the partial-order graph indicate the precedence relation between actions; an arrow from A to B means that A happens before B. The goal of is to learn the partial-order graph from a multitude of diverse action sequences like those in the left part of Figure 10. In many cases the training set does not equally cover all alternatives ways that an action can be performed, but shows some bias, introducing soft precedence constraints in addition to the causal dependencies between the actions. These soft constraints can be represented using a statistical model that can describe the probability of a precedence relation based on how consistently it was observed in the training data (visualized by the gray arrows in Figure 10).

5.2.4.1 Errors

The partial-ordered learning model has been evaluated on both synthetic data and two real-world data sets of human activities ('making brownies' and 'cooking an omelette'), (Tenorth, 2011). Testing the models on synthetic data tests the extent to which the models are influenced by noise in the data, and to check if the actual partial-order graph can be reconstructed. The real-world data sets provide realistic and more complex test data, and provide the opportunity to verify that partial-order models perform well on ADL scenarios.

To test the influence of irrelevant actions in between important actions, the sampling algorithm was modified so that a "noise" action may be chosen instead of a relevant action with a certain probability. In the experiments, the probability of selecting a "noise" action was set to 10%, 20%, and 50%. As shown in Figure 11, the partial-ordered model was about to learn a model that allows for good classification of action sequences, even with

very noisy sequences (50%) in which about half of the actions are not relevant for the task. The classification results were also compared to Hidden Conditional Random Field models (HCRFs, Quattoni et al., 2004). It was found that HCRFs could directly model the sequence of actions, but were unable to take into account longer-range dependencies such as global ordering constraints. Further, HCRFs could model the data adequately at when the level of noise was low (Figure 11, lines without markers), the performance of HCRFs decreases substantially when the proportion of irrelevant actions increases. In sum, the results of the model evaluation shows that AM-EvA's outperformed models often used in activity recognition (e.g., Conditional Random Field models, Hidden Markov models) for common tasks since they are much less confused by the variation inherent in human activities.

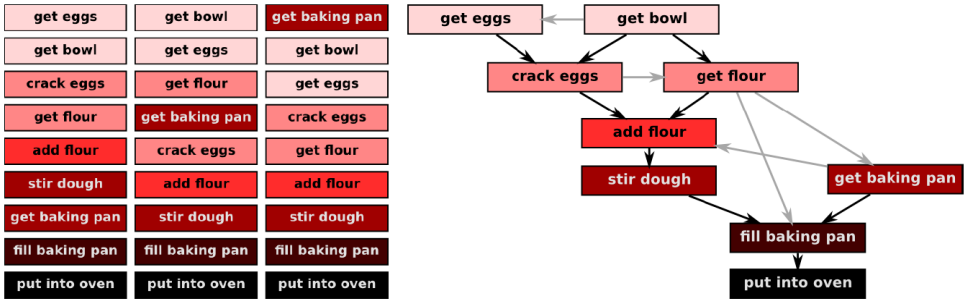

Figure 10: From several observations of the same task (left), the system learns the partial order of actions in that task (right) using statistical relational learning models (Tenorth, 2011)

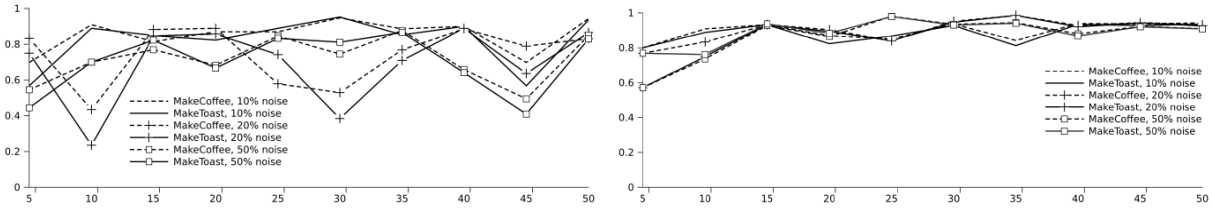

Figure 11: Recognition rates on synthetic data with different noise levels (10%, 20% and 50% probability of choosing a noise action) and sizes of the training and testing set (5 to 50 samples, see x-axis). Left: Hidden Conditional Random Field models (HCRF). Right: partial-ordered models. (Tenorth, 2011).

Tenorth (2011) also evaluated whether the partial-ordered learning model could infer the types of single actions in a task by randomly sampling sequences from the nosiest version of both activities (50% noise actions), removing the type of an arbitrary action in the test sequence, and inferring this given the rest of the sequence. As shown in Figure 12, the model was able to infer the type of an action given the type of the activity and the surrounding actions. Action N8, for example, is always the last non-noise action in every sequence and can thus easily be identified (seq. 12, 33). When there is confusion, it is mostly between actions on a similar level of the precedence graph (e.g. N4 and N1 in seq. 37) or between direct predecessors and successors (as in seq. 25, where N5 and N6 are direct predecessors of N7).

ID	activityT	actionT	most likely types
12	MakeCoffee	N8	N8(0.5760), N7(0.4135), X5(0.0042)
25	MakeCoffee	N7	N7(0.4837), N5(0.2022), N6(0.0846)
33	MakeCoffee	N8	N8(0.7667), N7(0.2211), X5(0.0117)
43	MakeCoffee	N1	N1(0.5303), N3(0.4243), N2(0.0447)
24	MakeToast	N6	N6(0.2867), N3(0.2498), N7(0.1395)
37	MakeToast	N4	N4(0.5940), N1(0.3800), N5(0.0220)
48	MakeToast	N4	N4(0.3950), N2(0.2860), N5(0.1860)

5.2.5 Hierarchical Task Analysis (HTA)

While the decomposition of activity into component tasks is common across a range of disciplines, Human Factors (particularly in the UK) employs a methodology called HTA, Hierarchical Task Analysis (Annett et al., 1971; Shepherd, 2001). What is important in this approach is not simply the hierarchical decomposition but also the definition of 'plans'. The hierarchy is typically described in terms of decomposition of a 'goal' into 'sub-goals', moving from a high-level objective to lower-level tasks, as in Figure 2. However, as discussed in Section 2.2, this hierarchy gives little indication of either the sequence in which tasks need to be performed or the conditions under which task completion is achieved. By separating tasks from conditions, HTA provides a simple but powerful means of creating a description. Furthermore, subsequent analyses can be built on HTA which generate predictions of errors, e.g., using techniques borrowed from Failure Modes Effects Analysis (FMEA) such as SHERPA (Embrey, 1986; Stanton and Baber, 1996).

In the notation for plans, '>' signifies "followed by" to indicate sequence, numbers indicates sub-goals in the hierarchy, and text indicates 'conditions'. In this example, there are two alternative plans.

Subgoal	Plan
0.0 Make tea	a.) 1.0 > while waiting > 2.0 > when water ready > 3.0 > if required 4.0 + 5.0 > 6.0 > 7.0 exit b.) 2.0 > 1.0 > 3.0 > if required 4.0 + 5.0 > 6.0 > exit

5.2.5.1 Errors

For each sub-goal, the analyst infers which of the Error Modes could potentially apply. The Error Modes were originally defined for power station control rooms and other process industries and, while they have been used in the analysis of ticket vending machines and similar products, it is not obvious that they can be directly translated to the CogWatch scenarios without modification. However, the following table gives an outline of their application.

Subgoal	Error mode	Consequence
0.0	A8	Tea is not made
1.1	A4	Kettle has too much / too little water
	A6	Container other than kettle is filled
	A8	Kettle is not filled

5.2.6 Markov Decision Processes (MDPs)

Markov Decision Processes (MDPs) have been used in spoken language processing research as models of human interaction for dialogue processing. There are several similarities between the requirements for a TM and those for a Spoken Dialogue Model (SDM). In both cases the system needs to keep a record of the sub-sequence of sub-goals that have been completed, and the sub-goals that still need to be completed in order to complete the goal. Both types of system need to be able to deal effectively with errors and to be able to provide suitable cues and feedback to the participant. In the case of a SDM, the problem is compounded by potential recognition errors made by the automatic speech recognition component (and this may also be a problem in our ADL system).

One approach to SDM is to model the dialogue as A Markov Decision Process (MDP) (for example, Levin et al., 2000). A MDP consists of:

1. A finite set ***S*** of *N* states. In SDM these are often referred to as belief states,
2. A finite set ***A*** of actions,
3. For each pair of states $\boldsymbol{s_1}$ and $\boldsymbol{s_2}$ in ***S*** and action ***a***,
    a. $P_a(\boldsymbol{s_1},\boldsymbol{s_2})$ is the probability of being in state $\boldsymbol{s_2}$ at time t+1 given state $\boldsymbol{s_1}$ at time t and that action ***a*** was taken
    b. $R_a(\boldsymbol{s_1},\boldsymbol{s_2})$ is the corresponding reward/cost

For example, in a TM a state could represent a particular stage in successful tea-making – a sequence of tea-making sub-goals that can be extended to successful tea making. In this case the state space would consist of all valid sub-sequences of sub-goals, implying a large value on *N*. The actions could be the set of sub-goals. If the system is currently in state $\boldsymbol{s_1}$ and the participant completes sub-goal ***a*** and the sequence of sub-goals obtained by adding ***a*** to state $\boldsymbol{s_1}$ is a valid state $\boldsymbol{s_2}$, then the system would move to state $\boldsymbol{s_2}$ and no cost would be incurred other than a fixed cost for completing an additional sub-goal. Otherwise, the system would remain in state $\boldsymbol{s_1}$, a cost would be incurred for completing an incorrect sub-goal, and a message would be transmitted indicating the state of the system and the fact that an error had occurred.

Due to the large value of *N* a substantial collection of examples of task execution would be needed to train a MDP from data.

5.2.6.1 Errors

Errors occur when the current completed sub-goal is not a valid extension of the current state. In this case the system remains in the same state and a cost is incurred. Thus the model is able to accommodate insertion and substitution errors to some degree. Deletions could be accommodated by allowing transitions to additional states, though this is more complex.

5.2.7 Partially Observable MDPs (POMDPs)

In an MDP the state sequence is "visible". There is a direct relationship between the sequence of sub-goals output by the activity recognizer and the current state of the MDP. However, in general this may not be the case. If the output from the activity recognizer is ambiguous (for example, if two sub-tasks are almost equally probable) or if the activity recognizer makes errors, then it may not be possible to identify the actual state of the MPD. In this case the best that can be done is to try to infer the MDP state. The result is that at any particular time during the execution of the task, the belief state of the system is a distribution over the possible states of the MDP. If this is the case then the appropriate model is a Partially Observable MDP (POMDP).

A POMDP is an MDP with the following additional structure. Using the notation from Section 5.2.6, for each output ***o*** from the activity recognizer and each state ***s*** and action ***a*** there is a probability:

c. $P(\boldsymbol{o}|\boldsymbol{s},\boldsymbol{a})$, the probability of the observation ***o*** given state ***s*** and action ***a***.

POMDPs have been applied in Spoken Dialogue Systems research (Young et al (2010)) and, more closely to CogWatch, in automatic hand-washing assistance for dementia patients (Hoey et al. (2007, 2010)).

5.3 Choice of Task Model in the first CogWatch prototype system

5.3.1 Discussion

From the perspective of the CogWatch system, the models proposed by Norman and Shallice (Secion 5.2.1) and Cooper and Shallice (Section 5.2.2) are synthesis models. They are able to simulate error-free goal-directed action sequences, provided that the model parameters are set appropriately. To account for behaviour in AADS populations, the approach of Cooper and Shallice is to add random noise to the system (in Cooper and Schallice (2000) the noise parameter was set such that the noise was randomly distributed with a standard deviation of 10^{-3}). The models are evaluated from a psychological perspective. A particular configuration of parameters and constraints is chosen, and judgements are made about whether or not the resulting sequences reflect patterns of human task execution. If they do, inferences are made about human task execution.

In terms of the criteria in Section 5.1, in order to function as a TM for the CogWatch system, a key attribute is that the model must be able to judge whether or not the sub-sequence of actions performed by the participant up to a particular point in time corresponds to a 'valid' sequence of sub-goals which is likely to result in successful completion of the goal. It is not clear how this can be achieved directly with this model. The Cooper and Shallice model could be run many times, and the proportion of times that the sub-sequence in question occurs in a successful sequence of actions could be used as a quality measure. In effect, this amounts to using sequences of actions synthesised by the Cooper and Shallice model

to train the parameters of some form of statistical task model. The key issue then becomes the choice of this statistical model.

The main difference between the Cooper and Shallice model and that of Botvinick and Plaut (Section 5.2.3) is that the latter is a trainable connectionist model. However, the observations that it is effectively a synthesis model, that it has traditionally been evaluated from a psychological perspective and that from the viewpoint of AAR its main utility is to synthesise data to train a statistical TM, apply equally well.

In contrast to psychological models of action recognition, AM-EvA's allow to query for semantically specified observation data, and to compare the style how the activities are performed with models learned from observations. These observations may either be prior observations of the same subject, which can be used to detect changes in their performance, or observations of a whole group of subjects, which allows one to assess how well a person performs in comparison to a reference group. The main advantage of AM-EvA's is the ability to learn the partial ordering of actions in a task using statistical relational learning methods. Additionally, AM-EvA's are able to learn which actions are relevant and which ordering relations are important, and thus, can be used to classify and verify activities, identify relevant actions in an activity, and infer missing data.

The model that underlies Hierarchical Task Analysis (Section 5.2.5) is very similar to the Cooper and Shallice model, in that it is a hierarchical tree of schema. However, in HTA sequences of actions are synthesised from the tree manually in the form of plans, and potential errors are identified manually using established principles. From the perspective of AAR, the issues are very similar to those raised in the context of the Cooper and Shallice model, namely that HTA is a method for synthesising action sequences that reflect human execution of a particular task. Hence the main utility of HTA may be to determine the structure of a computationally useful statistical TM. An advantage of HTA is that its treatment of errors is systematic rather than relying on corruption of the model parameters with noise.

MDPs appear to satisfy the criteria for aTM for the CoWatch AAR system. The main disadvantage of MDPs is the potential size of the state space, since the states correspond to all possible partial sequences of sub-goals that can be extended to successfully complete the task. However, in the CogWatch tea-making task the number of sub-goals is small. Moreover, if the number of sub-goals that have been executed is too great, then this would trigger feedback and cuing, and ultimately the task would be abandoned. Consequently it is possible to limit the size of the state space. Even so, it is unlikely that enough training material will be available within the timescale for the development of the first prototype to enable the model parameters to be estimated from data. However, an alternative solution is to use the Cooper and Shallice model or HTA to define the state space.

POMDPs are an extension of MDPs. Therefore the need to use POMDPs as a TM is likely to arise from the discovery of limitations of MDPs, for example an inability to deal effectively with ambiguity in the output of the AAR system. Therefore a judgement of the necessity of POMDPs will be deferred until more is known about the utility of MPDs in the context of CogWatch.

5.3.2 Summary

In summary, at this stage of the project the most promising candidate for the CogWatch system TM is a MDP, whose structure and parameters are determined by applying HTA to a hierarchical tree description of the tea-making task.

However, it is anticipated that future versions of the system will use more sophisticated models, such as AM-EvAs, whose structure and parameters are derived more from example data.

6. PATTERN-BASED ACTION RECOGNITION

6.1 Introduction

The objective of pattern based action recognition in the CogWatch system is to classify a sequence of feature vectors measured (using instrumented tools and objects, on-body sensors, or a video-based system such as Kinect) while a participant is engaged in a particular ADL task, in terms of the sequence of tasks and sub-goals that he or she is performing. This information is used to determine the stage in the task that has been reached and whether or not the task is being executed in a way that is likely to result in success. If not, the system should provide sufficient information to enable useful cues and feedback to be provided for the participant.

6.2 Hidden Markov Models (HMMs)

HMMs are a generic, statistical method for modelling time-varying sequential data. They were originally applied to pattern recognition in the context of automaticspeech recognition (ASR).

In general, HMMs represent a compromise between accurate modelling of the physical properties of a particular signal and mathematical and computational tractability. At any given time an HMM assumes that the signal that it is modelling is in one of a finite number of stationary states, that the transitions between these states are instantaneous, and that the length of time spent in a state is governed by a geometric distribution. In addition it assumes that the elements of the sequence are statistically independent of one another. Deviations from stationarity in a state are treated as noise and modelled with a probability density function associated with the state (the 'state output PDF'). Set against these constraints is the existence of a mathematically rigorous and computationally tractable training algorithm, the Baum-Welch or Forward-Backward algorithm (Baum et al. (1970)) for estimating HMM model parameters directly from data. There is also an established decoding algorithm, the Viterbi algorithm, for recognising a given signal in terms of the outputs of a sequence of HMMs.

The extent to which the HMM assumptions are appropriate for a particular signal depends on the properties of that signal. For example, despite their success in ASR, it is clear that the HMM assumptions are not particularly appropriate for speech signals, and this has resulted in the investment of considerable research effort into the development of alternative models. However, so far none of these alternatives have been able to demonstrate equivalent ASR performance to HMMs. It seems that for ASR the limitations of the HMM modelling assumptions are more than offset by these other computational considerations. The extent to which signals in the CogWatch project will match the HMM assumptions is unknown. However, it is clear that the HMM assumptions are well-matched with the outputs of an eye-tracker (Cooke and Russell (2008)) and it is likely that similar trade-offs between modelling accuracy and computational utility will apply to CogWatch signals.

Pioneering work on the application of HMMs to ASR is reported in Baker (1975), which describes the development of the original "Dragon" system at Carnegie Mellon University, and in Bahl and Jelinek (1975) who describe early research at the IBM speech research group. Initially, HMMs were restricted to sequences of discrete symbols, and their application to a continuous signal such as speech required some form of quantization. This

changed in 1982 when Liporace extended the Baum-Welch HMM parameter estimation algorithm to HMMs with Gaussian Mixture Model (GMM) states (Liporace (1982)). However, the application of HMMs to speech recognition was only popularised in the mid-1980 by Rabiner and his colleagues at Bell Laboratories (Rabiner et al. (1985)). In 1989 Cambridge University released the first version of the HMM Toolkit, HTK, (Young (1994)). This is a library of C functions that implements the set of tools needed to develop a HMM based speech recognition system (including the Baum-Welch and Viterbi algorithms). Its most recent versions are used in speech recognition and more general pattern recognition research laboratories across the world.

Over the past 20 years there have been many advances in HMM technology, including the development of discriminative training algorithms (e.g. Jiang (2010)), adaptation algorithms for model parameter estimation from limited data (e.g. Gauvain and Lee (1994)), noise compensation techniques (e.g. Vaseghi and Milner (1997)) and the emergence of HMM-based approaches to modelling parallel asynchronous processes (Ghahramani and Jordan (1997)). It does not appear that these developments have been applied to the CogWatch action recognition application.

From the perspective of the CogWatch application it is also significant that there is an established HMM algorithm for decoding continuous signals, where the boundaries between different events (sub-goals and tasks) are not known (Bridle et al. (1982)) and that this algorithm can operate continuously in real-time using a technique called Partial Traceback to free memory as soon as the decoding decision becomes unambiguous (Spohrer et al. (1980)).

The action recognition component of the first CogWatch prototype system will be based on HMMs.

6.3 Application of HMMs to action recognition

6.3.1 Unit selection

Unit selection refers to the choice of the basic units that will be modelled by HMMs in the action recognition system.

CogWatch deliverable D1.1 “Report on Scenarios” includes a hierarchical tree description of the tea making task. This is reproduced here for convenience as **Error! Reference source not found.**. The tree divides the *goal* (“Prepare a cup of tea”) into five *sub-goals*:

- “Heat water”,
- “Add water in the cup”,
- “Put tea bag in the cup”,
- “Remove tea bag from cup”,
- “Take a sip of tea”).

Each sub-goal is further divided into a set of *tasks*. For example, the sub-goal “Add water in the cup” involves the *tasks*:

- “Add water in the cup”
    - “Pick cup”,

- "Place cup next to kettle"
- "Tilt the kettle until cup is full".

Finally, each task is expressed as a set of sub-tasks. For example:

- "Add water in the cup" (*sub-goal*)
  - "Tilt the kettle until cup is full" (*task*)
    - "Grasp handle" (*sub-task*)
    - "Push forward until water pouring" (*sub-task*)
    - "Keep still until cup is full" (*sub-task*)
    - "Tilt back the kettle" (*sub-task*)

In CogWatch prototype 1 we will have the option of applying HMMs at the task or sub-goal level. This is analogous to phone-level or word-level modelling in ASR.

6.3.1.1 Task-level HMMs

In a task-level HMM system, sub-goals are expressed as one or more sequences of tasks in a text configuration file analogous to the 'pronunciation dictionary' in a standard HMM-based ASR system. Sub-goal level HMMs are then constructed by combining task-level HMMs.

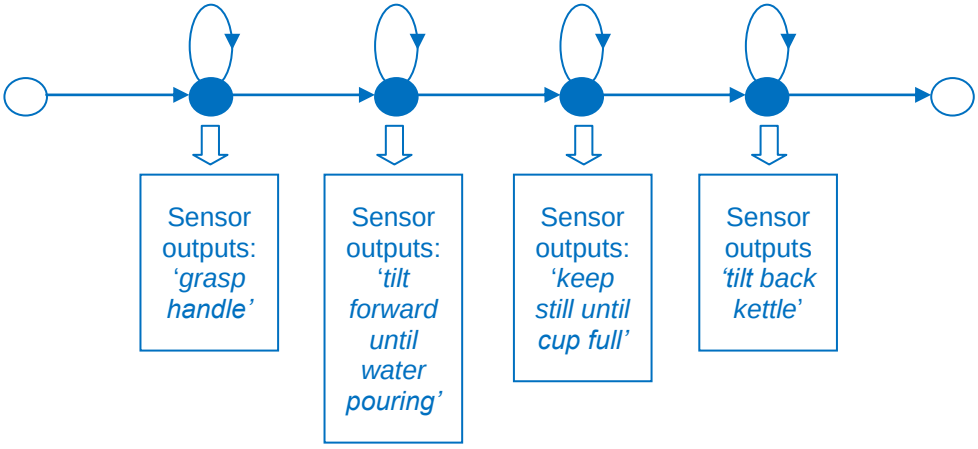

Figure 11: Action-level HMM for "*tilt the kettle until cup is full*"

For example, a HMM of the task "Tilt the kettle until cup is full" might consist of four states $s_1,...,s_4$, where each state s_i is associated with a PDF b_i defined on the set of all sensor outputs, and, for example, b_1 describes the distribution of feature vector values that are output when the handle of the kettle is grasped. Figure 11 shows an intuitive representation of an HMM for the task "tilt the kettle until the cup is full". The initial and final states are "null states" whose function is to facilitate connectivity between models. In reality the parameters of the probability density function (PDF) associated with each state would be learnt from data automatically, so that such a literal interpretation of the model states is generally not accurate. The PDFs associated with the states will be Gaussian mixture models (GMMs).

The number of states, the connectivity between states, the number of GMM components and the parameter values will be determined empirically.

An advantage of constructing sub-goal models from task models, rather than modelling sub-goals directly as sub-goal-level HMMs, is that it may enable parameter sharing and therefore reduce the amount of training data required. For example, in **Error! Reference source not found.** the "add water in the cup" and "take a sip of tea" sub-goals both begin with the task "pick up cup" and therefore share the same model parameters. More generally, different "pick up object" tasks, where, for example, the object could be a cup or a jug, would also share parameters.

In large vocabulary ASR systems, with vocabularies of tens or thousands of words, the sub-word based approach is essential because it is not practical to estimate the parameters of such a large number of word-level models. In fact, parameter estimation for systems with many thousands of parameters is a major issue for ASR. However, for the present application the 'vocabulary size' is much smaller, with of the order of just 10 sub-goals. Hence sub-goal level modelling is a viable option.

6.3.1.2 Sub-goal level HMMs

In a sub-goal HMM based activity recognition system, each sub-goal is modelled using its own dedicated HMM. These HMMs will typically be more complex, with more states than a task-level HMM. In fact an initial estimate of the number of states required for a sub-goal level HMM might be three times its number of component tasks.

The main advantage of sub-goal level HMMs over task-level HMMs is that the HMM can explicitly model the interaction between the tasks within the sub-goal. Specifically, the precise set of movements and object-interactions that a participant uses to execute an task will, in general, depend on the preceding and following tasks. In ASR the solution to this problem is to use context-sensitive phone-level HMMs. The most common approach is to use so-called "triphone HMMs" (a model of the acoustic realisation of a phone in the context of the immediately preceding and following phones). The analogy for activity recognition would be to build a different task-level HMM depending on the immediately preceding and following tasks. If the number of tasks is N this results in up to N^3 context-sensitive task-level HMMs, but if N is small (as is the case for activity recognition) this may not be an issue.

On balance, for initial activity recognition systems sub-goal HMMs would seem to have advantages, since no assumption needs to be made about the extent of contextual influence on the realisation of tasks.

6.3.2 Action recogniser architectures

A number of different recognizer architectures are possible, each with its own advantages and disadvantages with respect to sub-goal modelling. The issues that determine choice of architecture are the same as those discussed in Section **Error! Reference source not found.** concerning the limitations off the hierarchical task description, namely the ordered or partially ordered nature of a set of sub-goal that instantiate a goal.

6.3.2.1 Sensor integration (early integration)

The simplest approach, referred to as "sensor integration", "sensor fusion" or "early integration" is represented in Figure 12. At each time instant, the full set of sensor outputs is combined in to a single feature vector and processed using a single HMM-based decoder.

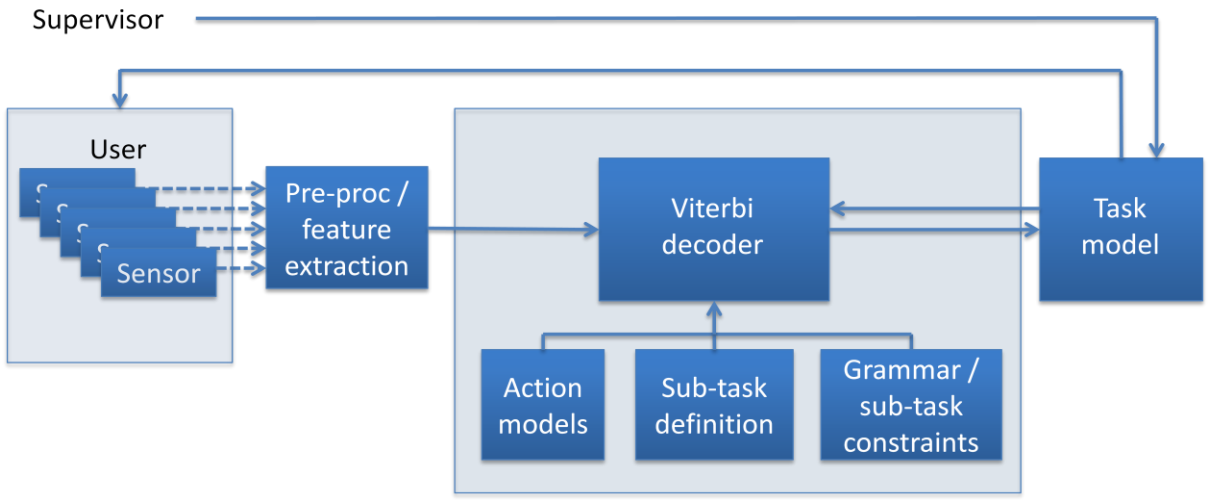

Figure 12: Early integration of sensor information

The advantage of this approach is that it is simple, and that it explicitly captures correlations between the outputs of all of the sensors attached to all of the objects. For example, when milk is poured into a mug, the feature vectors presented to the recogniser will include information from the RFID tag attached to the jug (indicating close proximity of the hand to the jug) and the accelerometer in the CIC attached to the jug (indicating that the jug has been tilted) and information from the FSRs in the CIC attached to the mug (indicating an increase in weight due to the mug filling with milk) and possibly, depending on the range of the RFID sensor, from the RFID tag attached to the mug, indicating proximity of the hand to the mug.

The disadvantage of sensor integration is that the recogniser expects the input to correspond to an ordered sequence of individual sub-goals. This may be the case in tea making, and it will certainly be the case if the participant uses only one hand.. However, if the participant uses both hands, or more than one participant is involved, it is possible for two sub-goals to be executed either at the same time, or at least such that the time intervals over which they are executed overlap. In this case the scheme depicted in Figure 12 will not suffice.

For example, during the sub-goal “pour milk into cup”, the recogniser will expect a sequence of readings from the sensors attached to the jug, corresponding to the jug being lifted, translated and then tilted, and readings from the mug corresponding to a gradual increase in weight.. It will expect null (noise) outputs from all of the other sensors. If during this sub-task the sugar container is moved, either by the participant with his or her other hand or by another person, the unexpected readings from the sugar container sensors will cause confusion.

A potential solution would be to model pairs of sub-goals, but they would need to take place in synchrony, and this is not generally the case. Also, modelling pairs would effectively mean squaring the number of sub-goals, and this would have computational implications.

In summary, sensor integration is the simplest approach and is acceptable if the task is executed as an ordered sequence of sub-goals.

6.3.2.2 Late integration (object level)

An alternative approach, referred to as “late integration” or “decision fusion” is represented in Figure 13. In this scheme the outputs from the set of sensors attached an individual object are processed together but separately from those attached to all other objects. Each object has its own dedicated pattern recognition system. The outputs of the separate classifiers are combined in the task model after classification.

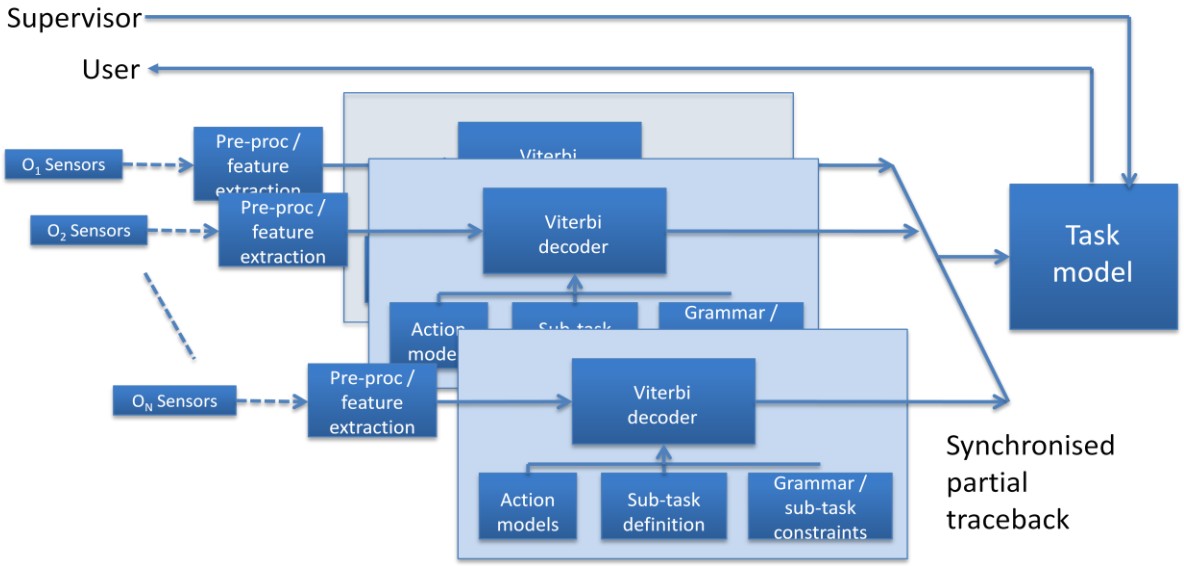

Figure 13: Late integration (object-level fusion)

Late integration is much more flexible than sensor-level integration, because all of the objects are treated independently. However, this is also its main weakness. Returning to the example of pouring milk into the mug from Section 6.3.2.1, in the case of late integration the pattern recognition system responsible for the mug will receive data from the mug’s sensors indicating proximity of the hand to the mug (depending on the range of the RFID sensor) and from the FSRs in the CIC indicating increasing weight. This data is much less ambiguous if the recogniser is also aware of what is happening to the jug, but in object level late integration the sensors on the jug and those on the mug must be interpreted independently, leading to a potential increase in ambiguity.

These types of issues have arisen elsewhere and potential solutions have been proposed.

For example, in Parallel Model Combination (PMC) (Gales and Young (1996)) a noisy speech signal is treated as the combined output of two HMMs, one modelling speech and one modelling noise. A combination function describes how the speech and noise are combined and the probability of a particular feature vector is computed by integrating over all possible combinations of speech and noise that could have resulted in that vector. This method has been applied to the integration of audio and visual speech signals by Tomlinson et al. (1996).

More generally, Factorial HMMs (Ghahramani and Jordan (1997)) provide a framework for including probabilistic constraints between sets of parallel HMMs to try to characterise the dependencies between them. Bayesian Networks (BNs) (Jensen (1997)) and Graphical

Models (Lauritzen (1996)) provide a more general framework for characterising these types of dependencies.

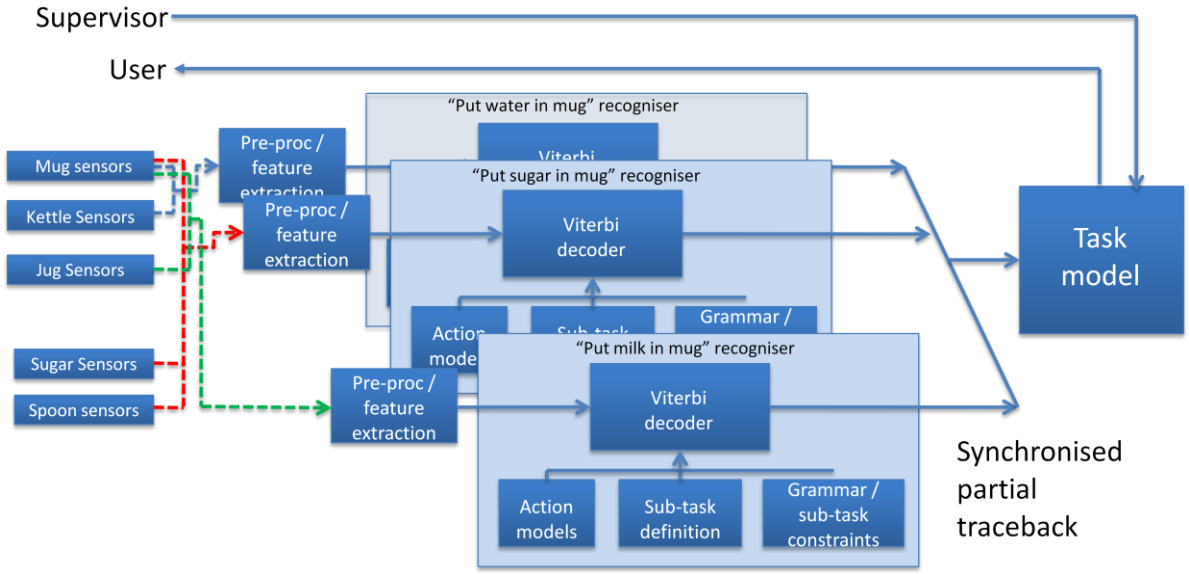

Figure 14: Late integration (sub-goal level fusion)

6.3.2.3 Late integration (sub-goal level fusion)

An alternative to the object-level late integration scheme described in Section 6.3.2.2 is to group together the outputs of sensors associated with objects that are involved with in a particular sub-goal and to employ a set of parallel sub-goal level recognisers. The sub-goal models could be constructed from task-level models or be built explicitly at the sub-goal level, as discussed in Section 6.3.1.

For example, the model for "pour milk into the jug" would take as its inputs the set of all sensors associated with the milk jug and the mug. The advantage of sub-goal level modelling would be a better ability to model the interaction between the two objects, for example the outputs of the sensors when the jug comes into proximity with the mug and when the jug is tilted and milk is transferred from the jug to the mug. Sub-goal level late integration is represented in Figure 14.

Rather than thinking of the parallel components in Figure 14 as 'recognisers' it is more accurate to think of them as 'sub-goal detectors'. Each detector continuously monitors the outputs from the sensors related to its sub-goal, looking for evidence that the sub-goal has been completed. In principle this could be triggered by the probability of the sub-goal model exceeding a threshold. However, threshold-based approaches tend to be sensitive to noise and other variability. A more robust approach is to run the sub-goal model in competition with a "background" model of the expected sensor outputs when the sub-goal is not being executed.

6.3.2.4 Background model

Each of the recognition schemes in Sections 6.3.2.1, 6.3.2.2 and 6.3.2.3 will require one or more 'background models' (BMs). The function of the BM is to accommodate the sensor information that is input to the recogniser when the object that the sensor is attached to is

not involved in executing a sub-goal. This will include the object being ‘at rest’, but might also include the object being ‘toyed with’ by the participant.

6.4 Choice of HMM architecture for first CogWatch prototype

The discussions above, and in particular the need to be able to cope with partially ordered sets, rather than sequences, of sub-tasks, point towards object or sub-goal level late integration as the modelling paradigm for the first CogWatch prototype. In addition, given that the number of sub-goals is relatively small, and the disadvantages of ignoring the way in which tasks interact within a sub-goal, sub-goal level late integration is the preferred option (Section 6.3.2.3).

Regarding unit selection, since the number of sub-goals is small and the interactions between tasks within a sub-goal may be complex, we propose to apply HMMs initially at the sub-goal level (Section 6.3.1.2).

It is important to note that the exact decision about the level of integration and modelling unit does not affect the development of the CogWatch HMM decoder at this stage, since the same basic architecture can be configured to accommodate any of these options.

7. SPECIFICATION OF THE FIRST COGWATCH PROTOTYPE

This section summarises the choices and decisions recommended in the previous sections and gives a description of the envisaged first CogWatch prototype system.

7.1 System inputs

The inputs to the system will comprise measurements from sensors attached to the tools and objects involved in the task, plus hand-location derived from Kinect. The precise set of sensors will be agreed with WP2 at the end of July 2012.

7.1.1 The CogWatch instrumented coaster

Each substantial object in the tea making task will be fitted with a CIC (Section 3.3). This comprises a 3-axis accelerometer plus three FSRs. Communication between the CIC and the AAR system will be wireless via Bluetooth.

7.1.2 RFID tags

RFID tags will be attached to all items, including those which are too small to support a CIC. RFID antennas, attached to RFID readers, will be worn on one or both of the participant's wrists.

7.1.3 Kinect

The Kinect system will supply the 3D locations of the participant's hands.

7.2 Automatic action recognition (AAR)

7.2.1 Specification of the HMM-based AAR system

The AAR system will be based on sub-goal level fusion (Section 6.3.2.3) of sub-goal level HMMs (Section 6.3.1.2).

7.3 The Task Model

7.3.1 Specification of the MPD-based TM

The prototype TM will be a MDP (Section 5.2.6), whose structure and parameters are determined from HTA applied to a hierarchical description of the tea-making task (Section 5.2.5).

8. CONCLUSIONS

This report discusses the issues that are relevant to the development of an automatic activity recognition system in the context of the CogWatch project. Its main purpose is to explain the rationale for the decisions that have been made regarding the design of the first prototype system, which is scheduled to be operational in month 16 of the project.

The report begins with a review of the tea-making task, which is the application chosen for the first prototype. A hierarchical tree description of the task (taken from CogWatch deliverable D1.1) is taken as the task definition. This description of the task is a reference that is referred to frequently in the discussions of action recognition models and task models which follow.

Section 3 discusses the types of instrumentation that are potentially available to the prototype CogWatch system. These include the CIC (comprising an accelerometer and three FSRs), RFID tags, and 3D hand location estimated using the Kinect system. The utility of each of these sensors for the tea-making task is discussed in Section 3.4. The sensors will be connected to the CogWatch system wirelessly via Bluetooth. The final set of sensors that will be used in the prototype will be agreed between WP2 and WP3 in July 2012.

The choice of the TM is discussed in Section 5. The strengths and weaknesses of various candidate TMs are considered. These include the psychological models proposed by Cooper and Shallce, and Botvinick and Plaut; the HTA model from ergonomics, the automated probabilistic models of everyday activities (AM-EvAs) that are being developed at TUM, and MDPs and POMDPs. For the first prototype system, the CogWatch TM will be based on MDPs, whose structure and parameters are determined using HTA. However, it is expected that future versions of the CogWatch TM will be data-driven models based on AM-EvAs.

Section 6 discusses the design of the first prototype CogWatch AAR system. The task of AAR is to interpret the sequences of measurements from the sensors (attached to the objects and tools involved in the task and to the participant's body) in terms of the tasks and sub-goals that the participant is performing. HMMs are chosen for this task, because they are an appropriate technology for real-time processing of sequential data and because many of the issues that arise in the context of the CogWatch application have already been addressed for HMMs in the context of speech recognition. However, the chosen architecture of the HMM system, namely parallel modelling of separate sub-goals using sub-goal level HMMs, is novel from the perspective of speech recognition.

The final specification for the prototype system is summarised in Section 7.

REFERENCES

Amft, O. And Tröster, G., (2008). “Recognition of dietary activity events using on-body sensors”, *Artificial Intelligence in Medicine*, 42, 121-136.

Annett, J., Duncan, K. D., Stammers, R. B. and Gray, M. J., (1971), “Task Analysis”, *London: HMSO*.

Baber, C. and Stanton. N.A., (1996), “Human error identification techniques applied to public technology: predictions compared with observed use”, *Applied Ergonomics*, 27,p119-131.

Bahl, L. and Jelinek, F., (1975), “Decoding for channels with insertions, deletions and substitutions with applications to speech recognition”, *IEEE Trans. on Information Theory,* IT-21, p404-411.

Baker, J. K., (1975), "The dragon system - An overview", *IEEE Transactions on Acoustics Speech and Signal Processing*, ASSP-23, pp 24-29.

Baum, L. E., Petrie, T., Soules, G. and Weiss, N., (1970), "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains", *Ann. Math. Statist*., vol. 41, no. 1, pp. 164–171.

Beetz, M., Tenorth, M., Jain, D., & Bandouch, J. (2010). Towards automated models of activities of daily life. *Technology and Disability*, *22*, p 27-40.

Botvinick, M. and Plaut, D. C., (2002), “Representing task context: Proposals based on a connectionist model of action”, *Psychological Research*, 66, p298-311.

Botvinick, M. and Plaut, D. C., (2004), “Doing without schema hierarchies: A recurrent connectionist approach to normal and impaired routine sequential action”, *Psychological Review*, Vol.111, No. 2, p395-429.

Botvinick, M., & Plaut, D. C. (2006). “Such stuff as habits are made on: A reply to Cooper and Shallice (2006)”, *Psychological Review*, 113, 917–928.

Bridle, J.S., Brown, M, and Chamberlain, R., (1982), “An algorithm for connected word recognition”, *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP-82, p899-902.

Cooke, N. J. and Russell, M. J. (2008), “Gaze-contingent automatic speech recognition”, *IET Signal Proc., **2**, (4), pp 369-380.*

Cooper, R. P., and Shallice, T., (2000), “Contention scheduling and the control of routine activities”, *Cognitive Neuropsychology*, 17, p297-338.

Cooper, R. P., and Shallice, T., (2006), “Hiearchical schemas and goals in the control of sequential behaviour”, *Psychological Review*, Vol. 113, No. 4, p887-916.

Cooper, R.P., Schwartz, M. and Shallice, T. (2005). The simulation of action disorganisation in complex activities of daily living. Cognitive Neuropsychology, 22 (8), 959-1004.

Embrey, D. E., (1986), “SHERPA: A systematic human error reduction and prediction Approach”, *Paper presented at the International Meeting on Advances in Nuclear Power Systems*, Knoxville, Tennessee.

Gales, M. and Young, S. (1996). "Robust Continuous Speech Recognition using Parallel Model Combination", *IEEE Trans Speech and Audio Processing* 4(5): 352-359

Gauvain, J-L. and Lee, C-H., (1994), “Maximum A Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains”, *IEEE Transactions on Speech and Audio Processing*, Vol. 2, p 291-298

Ghahramani, Z. And Jordan, M. (1997), “Factorial hidden Markov models”, *Machine Learning*, Vol. 29 Issue 2-3, p245 - 273.

Hoey, J., von Bertoldi, A., Poupart, P., Mihailidis, A. (2007), “Assisting persons with dementia during handwashing using a partially observable Markov decision process”, *Proceedings of the International Conference on Vision Systems*.

Hoey, J., Poupart, P., von Bertoldi, A., Craig, T., Boutilier, C. And Mihailidis, A., (2010), “Automated handwashing assistance for persons with dementia using video and a partially observable Markov decision process”, *Computer Vision and Image Understanding,* 114 p503–519.

Hong, Y.-J., Kim, I.-J., Ahn, S. C. And Kim, H.-G., (2008). “Activity Recognition Using Wearable Sensors for Elder Care”. *2008 Second International Conference on Future Generation Communication and Networking*. IEEE, pp. 302-305.

Jensen, F. V. (1997), “Introduction to Bayesian Networks”, Springer.

Jiang, H. (2010), “Discriminative training of HMMs for automatic speech recognition: A survey”, *Computer Speech and Language, 4,* p589-608.

Lauritzen, S. L., (1996), “Graphical Models”. Oxford Science Publications.

Levin, E., Pieraccini, R., Eckert, W., (2000), “A stochastic model of human-machine interaction for learning dialogue strategies”, *IEEE Transactions on Speech and Audio Processing*, Vol. 8, No. 1, January 2000, pp 11-23.

Liporace, L. (1982), “Maximum likelihood estimation for multivariate observation of Markov sources”. *IEEE Transactions on Information Theory*, IT-28:729-734.

Patterson, D.J., Fox, D., Kautz, H. and Philipose, M., (2005). “Fine-Grained Activity Recognition by Aggregating Abstract Object Usage”**,** *Ninth IEEE International Symposium on Wearable Computers (ISWC’05)*. IEEE, pp. 44-51.

Quattoni, A., Collins, M. & Darrell, T. (2004), “Conditional random fields for object recognition”, in ‘Advances in Neural Information Processing Systems’, MIT Press, p. 1097–1104.

Rabiner, L. R., Juang, B. H., Levinson, S. E., and Sondhi, N. M., (1985), "Recognition of Isolated Digits Using Hidden Markov Models With Continuous Mixture Densities", *AT&T Technical Journal*, 64, pp 1211-1234.

Schwartz, M.F., Montgomery, M.W., Buxbaum, L.J., Less, S.S., Carew, T.G., Coslett, H.B., Ferraro, M., Fitzpatrick-De Salme, E.J., Hart, T., & Mayer, N.H. (1998). Naturalistic action impairment in Closed Head Injury. Neuropsychology, 12(1), 13–28.

Schwartz, M.F., Montgomery, M.W., Fitzpatrick-DeSalme, E.J., Ochipa, C., Coslett, H.B., & Mayer, N.H. (1995). Analysis of a disorder of everyday action. Cognitive Neuropsychology, 12(8), 863–892.

Schwartz, M.F., Reed, E.S., Montgomery, M.W., Palmer, C., & Mayer, N.H. (1991). The quantitative description of action disorganisation after brain damage: A case study. Cognitive Neuropsychology,8(5), 381–414.

Shepherd, A., (2001), “Hierarchical Task Analysis”, *London: Taylor and Francis.*

Spohrer, J.C., Brown, P. F., Hochschild, P. H., and Baker, J.K. (1980), “Partial Traceback in Continuous Speech Recognition”, *Proceedings of the IEEE International Conference on Cybernetics and Society*.

Stiefmeier, T., Roggen, D., Tröster, G., Ogris, G. and Lukowicz, P., (2008). “Wearable activity tracking in car manufacturing”. *IEEE Pervasive Computing*, 7(2), p.42-50.

Tenorth, M. (2011). Knowledge processing for automous robots. Unpublished doctoral dissertation, Technische Universität München, München, Deutschland.

Tomlinson, M. J., Russell, M. J. and Brooke, N. M. (1996), “Integrating audio and visual information to provide highly robust speech recognition”, *Proc IEEE Int. Conf. Acoustics Speech and Signal Processing, ICASSP’96, Atlanta.*

Vaseghi, V. and Milner, B. P. (1997), “Noise compensation methods for hidden Markov model speech recognition in adverse environments”, *IEEE Transactions on Speech and Audio processing*, Vol. 5, Issue 1, pp11-21.

Wang, S. Pentney, W., Popescu, A-M., Choudhury, T. And Philipose, M., (2007). “Common Sense Based Joint Training of Human Activity Recognizers”**,** *Proceedings of the 20th international joint conference on Artifical intelligence*. p. 2237--2242

Ward, J. A., Lukowicz, P., Tröster, G. And Starner, T. E., (2006). “Activity recognition of assembly tasks using body-worn microphones and accelerometers”, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 28, No. 109, 1553-1567.

Young, S. J., (1994), “The HTK Hidden Markov Model Toolkit: design and Philosophy”, CUED/F-INFENG/TR.152, Technical Report, Cambridge University Engineering Department.

Young, S. J. Gasic, M., Keizer, S., Mairesse, F., Schatzmann, J., Thomson, B. And Yu, K., (2010), “The hidden information state model: A practical framework for POMDP-based spoken dialogue management”, *Computer Speech and Language*, 24, p150-174.