Department of Computing Science and Mathematics
University of Stirling

A Scalable Home Care System Infrastructure Supporting Domiciliary Care

P. D. Gray, T. McBryan,
N. Hine, C. J. Martin, N. Gil,
M. Wolters, N. Mayo,
K. J. Turner, L. S. Docherty, F. Wang, M. Kolberg

Technical Report CSM-173

ISSN 1460-9673

August 2007

Department of Computing Science and Mathematics
University of Stirling

A Scalable Home Care System Infrastructure Supporting Domiciliary Care

P. D. Gray, T. McBryan
Computing Science
University of Glasgow, Glasgow G12 8QQ, UK
Email pdg, mcbryan @dcs.gla.ac.uk

N. Hine, C. J. Martin, N. Gil
School of Computing
University of Dundee, Dundee DD1 4HN, UK
Email nhine, cjmartin, nubiagil @computing.dundee.ac.uk

M. Wolters, N. Mayo
Human Communication Research Centre
University of Edinburgh, Edinburgh EH8 9LW, UK
Email mwolters, nmayo @inf.ed.ac.uk

K. J. Turner, L. S. Docherty, F. Wang, M. Kolberg
Computing Science and Mathematics
University of Stirling, Stirling FK9 4LA, UK
Email kjt, lsd, fw, mko @cs.stir.ac.uk

Technical Report CSM-173

ISSN 1460-9673

August 2007

Abstract

Technology-mediated home care is attractive for older people living at home and also for their carers. It provides the information necessary to give confidence and assurance to everyone interested in the well-being of the older person. From a care delivery perspective, however, widespread deployment of home care technologies presents system developers with a set of challenges. These challenges arise from the issues associated with scaling from individual installations to providing a community-wide service, particularly when each installation is to be fitted to the particular but changing needs of the residents, their in-home carers and the larger healthcare community. This paper presents a home care software architecture and services that seek to address these challenges. The approach aims to generate the information needed in a timely and appropriate form to inform older residents and their carers about changing life style that may indicate a loss of well-being. It unites sensor-based services, home care policy management, resource discovery, multimodal interaction and dynamic configuration services. In this way, the approach offers the integration of a variety of home care services with adaptation to the context of use.
.

Keywords: Home Care, Software Architecture, Policy-Based Management, Context-Sensitive System, Multimodal System, Dynamic Reconfiguration, Systems Integration

1 Introduction

1.1 Home Care Systems

Home care is seen as a fundamental element of home-based support for extended independent living at home by older people. It can allow care services to reduce the number of speculative care visits and instead concentrate on situations of greatest need. It can provide older people with information to help them in their own self care, and the confidence to know that, should a care need arise, this will be recognised by the care services, understood and acted upon.

This context does, however, present a set of exciting system development opportunities. At the level of sensors and device management, home care is caught up in the explosion of ubiquitous computing and wearable medical devices. It offers new ways of doing old things (e.g. home based blood sugar monitoring), and also methods of performing new tasks (e.g. body area physiological monitoring). Developments in human-computer interaction, especially in multimodal interaction and speech technology, mean that home care systems can, in principle, communicate more intuitively and be more satisfying to use. Distributed and component-based system architectures make it possible to link home-based services (e.g. condition monitoring, appliance control, alerts and alarms, medication management). These services are connected with each other, and also with other services and users outside the home (e.g. clinical and social services, friends and family, peer health support groups).

However, with these opportunities comes a set of rather daunting challenges. The scaling of an individual installation to a community-wide service suggests a new way of thinking for older people living in the community and carers, as well as real technical challenges. These changes are at the very heart of the privacy and independence issues that characterise home life, and can lead to conflict among the service stakeholders [11]. Care still tends to be thought of as a set of discrete and individual packages provided by formal or informal caregivers. An integrated, multi-agency approach to care means that technologies and systems have to work together as well as people, so new strategies for cooperation at the human/service agency and the technical level need to be defined. This integration of the technology building blocks remains a major issue in research and development, as single integrated systems seek to cater for the interests of all the stakeholders.

This complexity requires input from a number of academic disciplines. As will be seen, this work requires the collaboration of professionals from computer science, telecommunications, speech, human factors, healthcare, social science, and housing policy. In response, the work reported here is a fusion of rather different technical research areas: service-oriented computing, middleware, networking, policy-based management, behavioural analysis, speech communication, and multimodal interfaces.

1.2 The MATCH Project

MATCH (Mobilising Advanced Technologies for Care at Home, *www.match-project.org.uk*) is a four-year project supported by the Scottish Funding Council. The project is a collaboration among the Universities of Dundee, Edinburgh, Glasgow and Stirling, joined by 11 external partners in areas such as social work, health care and assistive technology. The research on MATCH is developing home care technology in four key areas:

- home networks: provision of care services, flexible service discovery, and policy-based management of home care
- lifestyle monitoring: automated recording and analysis of daily behaviour, identification of trends and deterioration in the user's well-being
- speech communication: speech recognition and speech synthesis, speech corpora for the target client groups, and automated reminder systems
- multimodal interfaces: use of non-speech audio, haptic interfaces, and identification of stakeholder requirements.

Figure 1 shows the high-level technical approach taken by the MATCH project. The work reported in this paper focuses on the design of the residential gateway and the way that gateway meets the need of the care stakeholders.

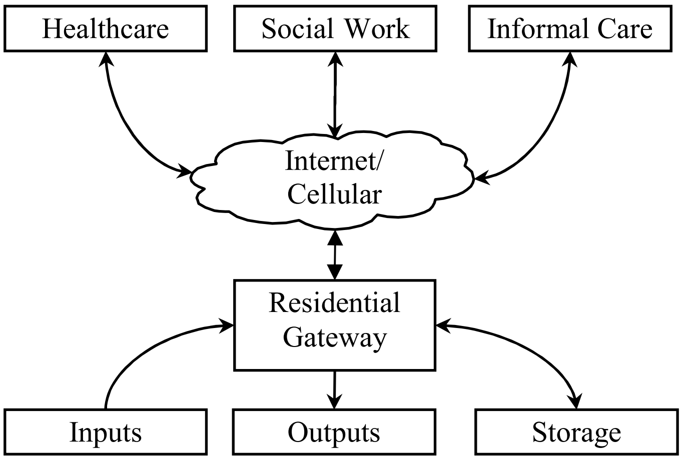

Figure 1. High-Level Approach of MATCH

MARCH (MATCH Architecture) is being developed as part of the MATCH project as the environment within which care services run. MARCH is a software infrastructure designed to address two of the most important home care development challenges: supporting system integration, and coping with change.

This paper addresses how these challenges have been met in the home care infrastructure, including the set of care services, their method of communication, and how they relate. Section 0 presents an overview of the architecture. Sections 3 to 9 discuss the major components in MARCH. Section 10 describes the current status of the system and the plans for future development.

1.3 Positioning of The Research

The MATCH project is distinctive in a number of respects. The emphasis is on delivery of care services to the home. Social care plays a dominant role, though healthcare issues are also accommodated. This requires a wide range of situations in the home to be monitored and managed. For a similar reason, assistive technologies are also important. MATCH is seeking to interface to care services, and as such it is integrating a wide variety of care monitoring devices.

MATCH is focused on home care services. As a result, the MATCH approach needs to be seen in the context of home networks rather than healthcare information systems. The work on smart houses (e.g. [10]) tends to concentrate on home automation (e.g. appliances, entertainment, security). Delivery of care is of lesser interest. Smart houses often emphasise device control, with service provision being secondary. Older people living at home need an integrated home care and smart home approach to support their independence at both the practical and the care levels.

As the base architecture, MATCH has adopted OSGi (Open Services Gateway initiative, *www.osgi.org*). This is ideal as the approach is vendor-neutral, device independent, and focused on service provision. Several projects have used OSGi in healthcare, e.g. e-HealthCare (*ehealth.sourceforge.net*), Home HealthCare (*www.ida.liu.se/%7estuha/anna-web/ projects/HHC-overview.htm*) and SAPHIRE [9]. [1] defines a widely used standard for exchange of healthcare information. This is supported by open-source projects like MIRTH (www.mirthproject.org). A middleware standard for healthcare information systems is defined in [7]. This addresses middleware for storage and retrieval of shared healthcare data. The Continua Health Alliance (*www.continuaalliance.org*) is particularly concerned to ensure interoperability of telecare solutions. A number of specifications have been developed to support healthcare applications of CORBA (Common Object Request Broker Architecture, *www.corba.org*). However, all these approaches are exclusively for healthcare applications (typically electronic patient records), and so are of only peripheral relevance to MATCH. As far as the authors are aware, MATCH has a unique emphasis on social care using OSGi.

Other differentiating factors in MATCH include the use of ontologies to enhance discovery of home care services, the use of policies to manage these services, and the fusion of multiple technical disciplines – activity monitoring, home networks, multimodal interfaces, speech technology, stakeholder analysis.

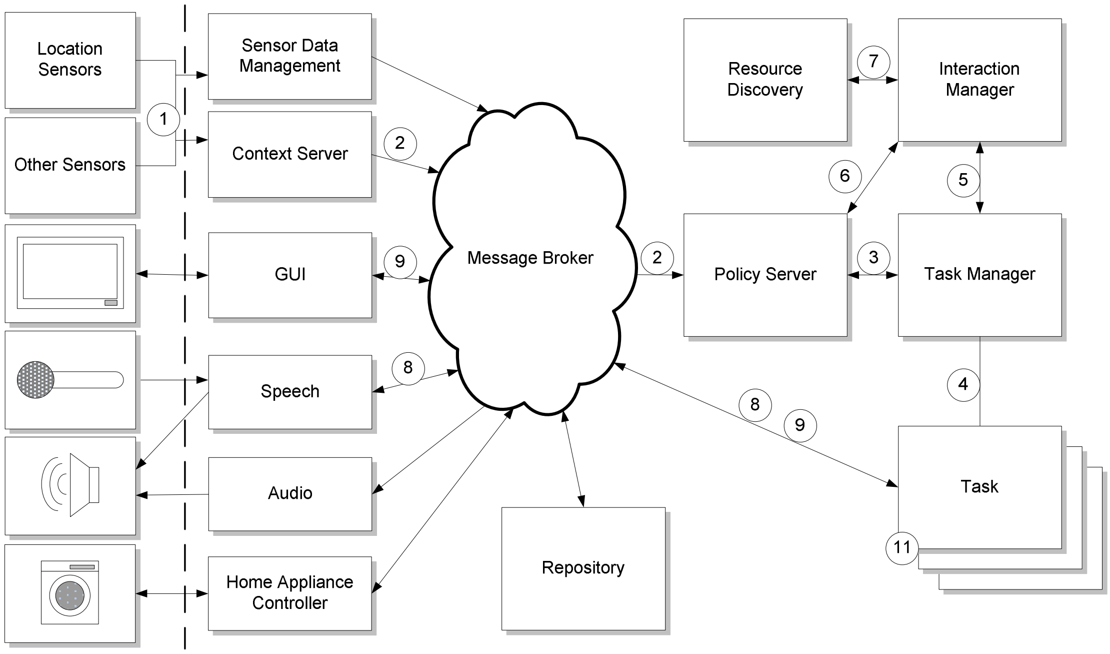

Figure 2. March – The MATCH Architecture

2 The Match System Architecture

2.1 Architecture Overview

Figure 2 gives an overview of MARCH as the MATCH system architecture. To introduce the overall system and its constituent services, consider the following simple example.

In consultation with a community nurse, the level of activity in the living room during the normal night time sleeping period of a resident has been identified as rising in a way that is potentially problematic. Understanding this unusual level of 'busyness' has been chosen as a key goal. At this stage, it is not known if the change in activity is an indicator of a problem (resident unable to sleep), or an indicator of a beneficial situation (the resident is a fan of NASCAR racing, shown on TV only late at night). A policy is therefore formulated to alert the appropriate stakeholders, including the resident, if this level of activity rises above a particular threshold. This would result in information that would inform the dialogue between the resident and the carers.

In terms of the technology involved, as illustrated in figure 2, the flow of events to realise this alert is as follows. Sensor data, such as from movement or door sensors, is forwarded to the Context Server (1). The Context Server processes the sensor data into activity data (e.g. movement in the living room) and sends this information as events to the Policy Server (2). This is facilitated by the Message Broker, which operates as the main message distributor in the system. The Policy Server retrieves policies that are triggered by these events.

Long-term information (e.g. policies) and transient information (e.g. from the Context Server) are stored in the Repository. As the action prescribed by the policy for this example is to inform the stakeholders, the Policy Server sends a 'Start Task' command to the Task Manager (3). This creates a task (4) and passes it to the Interaction Manager (5). The Interaction Manager queries the Policy Server (6) and Resource Discovery (7). This determines an appropriate interaction technique for the current context. For example, this might depend on where the stakeholder is, the appropriate ways of interacting with this particular user, and what devices are available.

With this information, the task begins to execute. It sends an alert using the assigned modality such as speech (8), and awaits a reply. On receiving the message, the user responds via a convenient device, e.g. by clicking a button (9). This returns a response to the task which then terminates (10). The task might have a timeout waiting on a user reply. For example, if the user is inactive and does not respond to the alert, the timeout could trigger a corrective action – a friend or other informal carer could be contacted

2.2 Base Architecture

OSGi and the home care services developed for MATCH are located in a residential gateway installed in the home. OSGi is able to access and use a wide variety of protocols that are becoming increasingly prevalent in various white goods and entertainment systems in the home such as X10 and UPnP, and also wireless and wired networks. OSGi uses a model that allows components known as bundles to be dynamically added and removed from a running JVM (Java Virtual Machine). The choice of OSGi makes it easier to incorporate a multitude of disparate devices (white goods, home appliances, environmental control systems, as well as special activity or presence detection devices), in addition to allowing dynamic system reconfiguration. OSGi has a strong focus on services, which makes it very appropriate for home care delivery, as it can readily be used to detect busyness in the home as residents go about their normal daily lives. As demonstrated by the enthusiastic uptake of SOA (Service Oriented Architecture), a service-oriented approach lends itself well to integration of loosely coupled components.

2.3 Message Broker

Because the stakeholders are in a variety of different physical locations, and the details of their information needs are different, this can be considered to be a distributed computing problem. OSGi, however, lacks native support for distribution of bundles across multiple JVMs or physical machines. MARCH achieves this by using a Message Broker for communication between components likely to be on different physical devices, or that are designed to broadcast or consume streams of data. Components register a subscription with the Message Broker to receive messages matching their request. The Message Broker itself is an OSGi bundle within the system, running in each JVM. We initially envisioned using an off-the-shelf Message Broker such as Elvin, but its withdrawal from sale convinced us to abstract this component into a generic interface. This allows us to easily change the Message Broker implementation. The current implementation uses the REDS message broker (Reconfigurable Dispatching System, *zeus.elet.polimi.it/reds*). In this way, we can retain the benefits of the OSGi functionality in the dwelling, whilst extending into the distributed computing situation of telecare.

2.4 Repository

A home care system generates data. This data may contain sensitive information of a personal nature, so data management is a critical aspect of any home care system. MARCH uses a Repository for storing and accessing home care data, including policies, user profiles, system configuration, and system state. Configuration information is necessary so that users and policies can refer symbolically to things like 'the front door', 'the community nurse' or 'the security service'. System state is needed so that policies can be interpreted dynamically in context. Status variables keep track of device and service state, such as whether a bed is occupied or whether a service is online.

The repository also includes a service that logs system events, for subsequent use in analysis of system behaviour, identification of reusable configurations, verification of system access, and use as a check of security violations.

3 Sensor Data Management

Data collection depends on sensors that are both low cost and aesthetically acceptable for deployment in the homes of people. A variety of wired and wireless sensors are being used in MATCH and have been installed in a domestic dwelling as a living test-bed. These sensors include the following: movement and location are identified by means such as pressure mats, PIR sensors (Passive Infra-Red), and beam-break sensors. Entry to/exit from the home is being sensed using RFID (Radio Frequency Identification). The kitchen is equipped with door opening sensors (on the fridge, freezer and some cupboard doors), force-plates (to detect the use of tinned foods and fresh vegetables), power usage sensors (for the kettle and microwave oven), and infra-red heat detectors (for the gas cooker). The hot water tank is instrumented with bimetallic switches that estimate the quantity of hot water available. The user is informed of this by various means, for example an indicator overlaid on the TV, a set of lights in the garden, and an indicator close to the heating controls in the kitchen.

Approaches to the modelling of domestic activity have been explored previously [2][8], and further investigation is planned in this area. Movement between rooms is analysed using a Hidden Markov Model, with sensor data acting as the observed variables and client location being the hidden variable. The sensor information also provides a rich picture of location and transition between rooms.

OLAP (On-Line Analytical Processing) and data mining are being used to visualise and then quantify 'busyness' as a precursor to predicting changes in the nature of user activity that may reflect a change in health or wellbeing. The practical indication of a loss of well-being is a change from an acceptable and normal lifestyle. Data mining techniques are being used to define normal daily living patterns and to look for deviations.

4 Context Server

Different stakeholders need information at different levels of detail, and using different modalities. For this reason, The Context Server provides information to assist modality choice for the different stakeholders, delivered to them where they can access and use that information. It also provides other components in the system with context information, such as estimated location and activity of the user. This information allows the system to deliver information to the user in the most appropriate manner. If the context information indicates that the user has a visitor, for example, it may be appropriate to alter the modality for a reminder from spoken audio to tactile vibration, or to store the reminder for later attention if it is not urgent.

The choice of output device can also be informed by the context information. If the house is unoccupied, for example, a text message via a mobile device might be the most appropriate way of issuing information to the resident. If the resident is at home watching television in the lounge, a subtle reminder on the television might be the most appropriate.

5 Policy Server

Every resident is different. The nature and degree of risks affecting an individual's well-being could be quite different for others in the same community. The most appropriate form or modality for communicating information to a specific resident or stakeholder could be quite different, and even change depending on the moment and the current situation of the user. To support better flexibility and control, home care services are managed by a policy system. This allows users to formulate policies for how they wish the care system to behave. A 'user' in this context includes any stakeholder such as residents, professional care providers and informal carers. Policy support has been integrated into the OSGi framework, interacting with it to manage devices and services and to deal with user interaction.

Policies are expressed in APPEL (ACCENT Project Policy Environment/Language, *www.cs.stir.ac.uk/accent*). This provides a set of core constructs that are specialised for each application domain by defining the triggers, conditions and actions in that domain – here, home care. The core language specifies the structure of the policies and is defined by a domain-independent XML schema. For home care, policy support has been extended to include system state, timing, and a variety of sensors and actuators. Policies are held in the Repository. More details of the policy system and the policy language and can be found in [13].

Despite the technical role that it has in the architecture, the policy system must support non-technical users. It is therefore presented through a wizard that helps users to define and edit policies. The wizard allows policies to be defined using a variety of interfaces including near natural language.

Components communicate with the policy system by sending or receiving messages through the Message Broker. Event messages contain a trigger name and parameters (e.g. location for a UPnP movement sensor). Action messages contain an action name and parameters (e.g. house/unit code and dimming level for an X10 lighting module).

A policy system input (typically from a sensor) retrieves policies associated with the trigger. Policies are selected only if they meet certain conditions (e.g. what the policy applies to, the period of validity of a policy, and what profile the policy belongs to). Policies can be grouped into profiles such as 'at night' or 'on holiday', allowing sets of policies to be enabled easily. Variables in the selected policies are retrieved from the Repository (e.g. configuration variables and state variables). Policy conditions are evaluated to determine which policies apply. All enabled policies then dictate the actions to be performed. In the absence of conflict, these actions are issued via the Message Broker. In turn, this causes various outputs via the Task Manager (e.g. to an actuator or to an audio component).

The policy server also supports the detection and resolution of conflicts among policies (typically due to differing stakeholder viewpoints). This is achieved by specialised resolution policies that define what conflicts are and how to deal with them, accompanied by a dialogue between the stakeholders themselves.

6 Task Manager

Given that the dwelling is equipped with a technology platform, this platform can not only gather information, but can also perform actions based on that information and the policies put in place by the stakeholders. These actions take the form of tasks. Tasks represent the actions that the system can take in order to meet the home care objectives of the system. The Policy Server can specify that a specific task is started, providing any parameters that the task may require. Tasks can range from simple atomic actions (e.g. turning on a light) to complex composite ones that may require long-term sensor monitoring and interpretation, and/or a complex interaction with residents, carers or external care services.

Tasks are modelled as directed graphs of subtasks, similar to Concur Task Trees [12]. This provides a bridge between care services and the concrete interaction techniques used to mediate interaction with a user. In this sense, tasks represent abstract dialogue structures, enabling dynamic adaptation of the interaction while minimising change to the care-related functionality.

The Task Manager allows dynamic loading of new concrete tasks to occur. This creates additional tasks from the currently loaded OSGi components rather than relying on a fixed set of predetermined tasks. Dynamically loaded tasks can be atomic tasks designed to perform specific functions, general-purpose tasks for use with multiple components, or control tasks designed to control execution of the task tree. This makes installation of the system in the community responsive to the individual realities of each resident and the changes that naturally occur in their lives, as well as the changes that are pertinent to their well being.

7 Resource Discovery

Every dwelling has many different systems provided by different vendors, such as white goods, burglar alarms, heating systems, and entertainment systems. The goal of ongoing work in home networks is to provide a unified platform that ideally supports multiple devices and protocols, where services from multiple suppliers interoperate seamlessly. A home network is unlikely to exist in isolation, since it must typically cooperate with a variety of external services (e.g. for health and social care). In such an environment, there are likely to be multiple techniques for service discovery, and different ways of describing services. Introduction of new sensors, devices or services into the home environment should be an automated process that does not require detailed technical expertise.

In order to locate suitable components available from the service framework, discovery must abstract away from protocol-specific service descriptions and implementations. A systematic vocabulary is also required for component description. Standard service discovery mechanisms in OSGi fail to meet these requirements.

A Resource Discovery mechanism using ontologies has therefore been developed. Ontologies support a layered approach to developing a common vocabulary that is scalable and extensible to suit the domain.

A vocabulary for home care has been developed in the form of the Home Ontology Stack [13]. Using this vocabulary, descriptions of home care components have been written in OWL (Web Ontology Language). This ensures protocol and supplier independence. A discovery mechanism has been developed to exploit the capabilities of description logics like OWL. The approach supports inference and reasoning, allowing discovery to be based upon absolute or relative requirements.

Suppose the task to be completed is to provide information in a room near where the user is. It can be discovered if a suitable information display exists, without knowing the name of the desired room. The discovery mechanism is able to infer which rooms are near the user.

Having a semantically rich vocabulary allows Resource Discovery to adapt to an evolving domain. This allows immediate update of information held within the Repository. The discovery mechanism is used to locate components/services based on properties such as their type and the request context. Queries can also be issued based on the kinds of subscriptions supported by a service or the inputs that it can handle.

8 Interaction Manager

Once the Task Manager has built a task structure for the actions dictated by policies, it has still to determine the appropriate modality and interaction technique for each task. This decision takes into account the available devices, the user preferences and other context information. For example, the user might need visual alerts, might prefer to receive messages in German, or might be upstairs. To achieve this, the Task Manager makes use of the Interaction Manager.

The Interaction Manager is responsible for querying Resource Discovery to determine the availability and characteristics of devices. It then uses the Policy Server to determine the best device and interaction technique to use in the current context. The result is mapped onto the task (or subtask) dictated by the active policies and available devices.

It is possible for a task to be performed using one or many devices, and in one or many modalities. Basic modalities include text, speech and audio. More advanced techniques include gestural input and tactile output. Digital pen-and-paper is being used a user-friendly way of gathering input. The approach to interaction management supports static or dynamic binding of tasks to interaction devices and techniques. This supports strategies that allow optimisation of resources. It also permits context-based dynamic adaptation of interaction [3], based on policies defined by the user. Although sophisticated combinations are possible, the work so far has concentrated on speech and non-speech audio.

9 Speech

Speech interaction within MARCH poses four key research problems. Speech synthesisers must be adapted for auditory ageing (which may involve hearing loss). Speech recognisers must be suited to older voices, which may be pathological. Dialogue management and information presentation need to be appropriate for cognitive ageing, which often involves memory loss. Finally, spoken dialogue must be integrated into an architecture without a dedicated dialogue management engine.

The first two problems are incidental to the architecture, but crucial for ensuring system usability. Our approach is to devise strategies that adapt existing solutions for voice synthesis and speech recognition. For synthesis, a general set of guidelines can be implemented using common markup standards such as SSML (Speech Synthesis Markup Language [5]). For recognition, we are working on algorithms to adapt existing state-of-the-art speech recognition techniques to older voices using a small number of carefully selected utterances.

In order to systematically investigate different dialogue management strategies, we are collecting a large corpus of interactions with a simulated dialogue system for appointment scheduling – one of the MATCH services. In order to ensure that this approach is acceptable to residents, performance data is augmented by detailed questionnaire data on user perceptions and preferences.

We have already made good progress on the first three research problems. However the last problem, integrating spoken dialogue into MARCH, is still an open question. In most systems, speech interaction is managed by a dedicated dialogue management engine (e.g. compare [3][4]). Typically, a dialogue manager determines the type of the next utterance (such as a request or confirmation), and provides some of its content. On the other hand, a dedicated language generation module optimises presentation for a particular user in a particular context.

An important function of the dialogue manager is tracking the dialogue history, including a list of the entities that have already been referred to and the questions under discussion. This is important for resolving pronouns in user utterances such as '*It* is not working' (in reference to a device), or 'Can you repeat *that* please?' (in reference to the last spoken message).

In MARCH, there is no separate dialogue manager. Speech is treated as one of several input and output modalities. Speech input can be processed and integrated with other forms of input such as touch-screen and sensor information. The type of message to be generated, as well as the timing and delivery of that message, are determined by the associated task. Hence, the Task Manager takes over much of the functionality commonly associated with a dialogue manager.

The Interaction Manager further constrains where and how information is presented. We have investigated how information that is needed to interpret user utterances should be modelled and stored, and how interactions should be managed. For example a user might be asked for clarification of a request or for confirmation of a selection.

10 Conclusion

Home care technology has the potential to provide tangible benefits to both older home residents and to carers. Because of the range of realities for an individual and their dwelling, a powerful but configurable and intelligent system needs to be deployed. This system needs to be sufficiently generic that the essential building blocks can handle the many contexts of residents within the wider community. For these reasons, MARCH has been developed as a novel architecture for creating distributed home care systems. Some of the component services have been developed in other contexts and represent mature technologies. However, it has been necessary to create a novel integration of all services for use in home care, especially to deal with dynamic adaptation. For this reason, service integration has been an important issue.

A first version of MARCH has been created and evaluated in a lab setting, including a real dwelling equipped as a test house. It exploits intra-service communication to support various home care scenarios, as well as distributed sensor and information distribution. This includes the example of section 2.1, as well as several others that illustrate interaction adaptation and the use of dynamic policy resolution. The next version will reflect more detailed and challenging requirements obtained from end users. This version will be deployed and evaluated in actual homes, using representative sensors and appliances.

Key future work includes the development of a configuration description language. New tools will be created to investigate a home system's past behaviour, its current configuration, and its potential reconfigurations (expressed as policy rules). Field trials will determine the power of our approach to provide a useful, accessible and adaptable home care system.

Acknowledgements

The MATCH project has been made possible through financial support from the Scottish Funding Council under grant HR04016. The authors thank their colleagues and external partners on the MATCH project for their fruitful collaboration.

References

[1] ANSI. Application Protocol for Electronic Data Exchange in Healthcare Environments, ANSI/HL7 V2.5, American National Standards Institute, Washington DC, USA, 2003.

[2] Barger, T.S., Brown, D.E. and Alwan, M. Health-status monitoring through analysis of behavioral patterns, IEEE Trans. Systems, Man & Cybernetics – Part A: Systems & Humans, 35:22–27, Jan. 2005.

[3] Bickmore, T. W. and Giorgino, T. Health dialog systems for patients and consumers. J. Biomedical Informatics, 39(5):556–571, 2006.

[4] Bui, T. H., Zwiers, J., Nijholt, A., and Peol, M. Generic dialogue modeling for multi-application dialogue systems, in Proc. 2nd. Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms, University of Edinburgh, Jul. 2005.

[5] Burnett, D. C, Walker, and Hunt, A. (eds.). Speech Synthesis Markup Language, W3C Recommendation version 1.0, Sep. 2004.

[6] Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Souchon, N., Bouillon, L., Florins, M. and Vanderdonckt, J. Plasticity of User Interfaces: A Revised Reference Framework, in Proc. Tamodia (Task Models and Diagrams for User Interface Design), Bucharest, Jul. 2002.

[7] CEN. Medical Informatics – Healthcare Information Systems Architecture – Part 1: Healthcare Middleware Layer, ENV 12967-1, European Committee for Standardization, Delft, Netherlands, 1998.

[8] Das, S. K., Cook, D., Bhattacharya, A., Heierman III, E.O. and Lin, T.Y. The Role of Prediction Algorithms in the MavHome Smart Home Architecture, IEEE Wireless Communications, 9(6):77–84, Dec. 2002.

[9] Hein, A., Nee, O., Willemsen, D., Scheffold, T., Dogac, A. and Laleci, G. B. Intelligent Healthcare Monitoring based on Semantic Interoperability Platform – The Homecare Scenario, Proc. 1st European Conference on eHealth, Fribourg, Switzerland, Oct. 2006.

[10] Helal, A., Mann, W., Elzabadani, H., King, J., Kaddourah, Y. and Jansen, E. Gator Tech Smart House: A Programmable Pervasive Space, IEEE Computer, 38(3):50–60, Mar. 2005.

[11] McGee-Lennon, M. and Gray, P. D. Including Stakeholders in the Design of Home Care Systems: Identification and Categorisation of Complex User Requirements, in Proc. Include, Apr. 2007.

[12] Mori, G., Paterno, F. and Santoro, C. Design and Development of Multidevice User Interfaces through Multiple Logical Descriptions. IEEE Trans. on Software Engineering, 30(8): 507–520, Aug. 2004.

[13] Wang, F., Docherty, L. S., Turner, K. J., Kolberg, M. and Magill, E. H. Services and Policies for Care At Home, in Proc. 1st. International Conference on Pervasive Computing Technologies for Healthcare, pp. 7.1–7.10, IEEE Computer Society, Nov. 2006.