

A Component Framework for Telecare and Home Automation

Claire Maternaghan and Kenneth J. Turner
Computing Science and Mathematics, University of Stirling
Stirling FK9 4LA, UK
cma | kjt @cs.stir.ac.uk

Abstract – Many sensors, appliances and smart devices for within the home are becoming readily available to the general public. These can not only enhance everyday living but also allow people requiring care to stay in the comfort of their own homes for longer. Users need to be provided with the ability to easily customise their home and the devices within it. A Service Component Architecture is described for representing various services within a home environment. The resulting system has a simple plug-and-play mechanism for easy addition and integration of a range of components. A collection of services is presented using both off-the-shelf and novel components.

Keywords – Component Framework, Home Automation, Service-Oriented Architecture, Telecare

I. INTRODUCTION

Background and Context

The world population is gradually ageing, with the percentage of older people (over 65) expected to rise by 2050 to 19.3% worldwide [1]. Although the population is ageing, older people are generally healthier and more active than in previous generations. They usually prefer to stay in their own homes for as long as possible.

Telecare systems are computer-based systems that support delivery of care to the home. They can provide the user with advice, identify trends that may need intervention, monitor for undesirable situations, reassure family members and informal carers, and relieve professional carers of low-level monitoring tasks. Telecare systems should be appropriate (reflecting different stakeholder viewpoints), customisable (tailored to specific user needs), flexible (supporting a range of solutions), and adaptive (as care needs and conditions evolve). Unfortunately, most telecare solutions are relatively fixed. Where alteration is possible, detailed technical knowledge and re-programming are typically required.

Home automation allows users to interact with their home as a whole, and to have flexible control over their domestic devices and appliances. Many services required for telecare are also useful in home automation. In fact, telecare services significantly overlap those of smart homes.

A component-based architecture is an obvious choice for supporting both telecare and home automation. The associated devices and appliances can then be shared components within an overall framework. An appropriate framework should support a customisable and flexible system that can be used for different purposes such as telecare and home automation.

SOA (Service-Oriented Architecture [2]) has emerged as a popular and malleable approach to building systems out of loosely coupled services. The key idea is that component functionality is given a service wrapping. The overall framework then acts as a container for components as services. It is relatively easy to take even legacy components and give them a service wrapping.

It is also highly desirable for home users to have a choice of interaction modalities and interfaces, allowing them to personalise how they interact with the home.

Motivation

This paper reports work to develop a component architecture that supports a range of interfaces, devices and services, allowing them to evolve as user needs change. This is a new application of Service Component Architecture, with potential for greater flexibility and (re)configurability.

A contention of this paper is that there is significant commonality among the requirements of telecare and home automation. For example, there are common needs for appliance control, communication, devices, entertainment, information, monitoring, security and services. The goal is therefore to create an overarching framework and common components that allow more specific ‘profiles’ (subsets of services and devices) to be supported for applications like telecare and home automation. Of course, each such profile will have additional more specialised requirements that in turn require more specific components and services. For example, telecare has particular needs such as remote health monitoring, help with medication compliance, and analysis of ADLs (Activities of Daily Living).

This aim of this work is therefore to develop a flexible and dynamic framework that supports any type of device, appliance or home service. The framework focuses on how to represent components in a way that makes them easy to use as building blocks in higher-level services. The framework can then combine these building blocks in a variety of ways to create applications, services, rule-based systems, etc. It is possible to use the same components for completely different purposes, and to reconfigure them dynamically as requirements change. This supports the design principles recommended by Davidoff *et al.* [3] for developing end-user programming systems within a smart home environment.

Considerable device functionality is available within a typical home. Unfortunately, it is not yet common to combine

the capabilities of individual devices. Audio-visual systems are the exception, but even there the combination is limited to simple and fixed interconnections. Basic tasks like managing home appliances from a remote location are possible, yet integrated solutions are not readily available. A home system should allow the user to manage and interconnect home devices as required. Work carried out by Newman *et al.* [4] demonstrates such a system, but it is limited to media devices within the home.

As part of the work reported here, a wide range of users were canvassed as to their requirements for home automation. There is a strong demand for trivial tasks to be automated (e.g. greater control over heating, getting status reports about the home), accessed within the home or from a remote location. Another recurring requirement is energy efficiency. An in-depth analysis of the user requirements is being used to guide the development of the home (care) system discussed here.

Related Work

The work in this paper covers many areas, so only a brief review of related work with a few citations is practicable. The activities discussed below are well documented on the Web.

Telecare The authors are extending work on the MATCH project (Mobilising Advanced Technologies for Care at Home, www.match-project.org.uk). There have been numerous projects on technologies for home care. This research includes e-health (e.g. e-HealthCare, HAVEN, MIRTH, SAPHIRE, UBICARE), independent living (e.g. AMI, ALIP, EQUAL, PERSONA, SOPRANO, SPARC), smart houses (e.g. AMIGO, Bath, Cortexa, Gator, House_n, Millennium Homes), and telecare (e.g. Continua Health Alliance, ETSI, SAPHE). The work reported here differs in its focus on managing home-based devices and services in an easily adaptable, extendable and flexible way. A higher-level framework allows the home system to support different profiles of usage – notably telecare and home automation.

Architectures Many frameworks and architectures could be appropriate for representing components within a smart space. Jini (www.jini.org) is a Java-based, service-oriented architecture for distributed systems. This could allow components to be located within the home, while accessing common services such as weather forecasting on a central server. However, Jini is a rather general approach that does not address the specific challenges the authors wish to meet.

Sensor Networks Atlas [5] supports sensor-actuator networks in a plug-and-play, service-oriented manner. It provides a framework that can automate the sensor-to-service (hardware-to-software) conversion. Specialised hardware modules are needed to connect devices, thereby generality in a domestic setting. Atlas also does not focus on user interfaces, which are left to the developer to create. The middleware simply offers services to a system, relying on specialised hardware for input and custom interfaces for usability. Atlas is therefore not a solution to the goals of this paper.

Home Automation Cortexa (www.cortexatechnology.com) is a top-of-the-range home automation solution that claims to offer ‘the most user-friendly, secure, powerful and simplistic system available’. It supports a wide range of off-the-shelf home automation hardware. The package attempts to combine many services and applications through a simple and consistent user interface in the style of Windows Media Centre™. However, this system has limited support for users customising the home to their needs. It does support a simple rule creator and editor, as well as a complicated administrative rule editor. However, the authors’ view is that sufficient functionality and flexibility have not yet been combined with adequate usability for ordinary users.

Home Networks There are several international and many proprietary standards for home networks. X10 is widely used to control home devices using existing mains cabling, though there is some wireless support. KNX (www.knx.org) is a derivative of EIB (European Installation Bus), and is the only open standard for home and building control. It can work over various communications media, including mains cables. Wireless standards in the home include Bluetooth, WiFi and Zigbee. The work reported in this paper is focused on higher-level interconnection at the component and service level. It is therefore independent of the lower-level networking aspects, which are handled by separately provided components. For example, open-source or commercial solutions exist for all the key home network standards.

II. A FRAMEWORK FOR HOME SERVICES

A. The Service Component Architecture

SCA (Service Component Architecture, www.osoa.org) is a collection of standards describing a component-based framework that complies with the principles of SOA (Service-Oriented Architecture). The idea behind SCA is that component implementations should be separable from how they are interconnected. It should be possible to develop components in a range of languages. The framework should allow these components to be ‘wired up’ in a variety of configurations, without having to worry about the component implementations. Furthermore, reconfiguration should be easy – perhaps because of a new component implementation, or because new requirements need a different configuration.

Tuscany (tuscany.apache.org) is an Apache open-source implementation of SCA. It supports several implementation languages such as BPEL (Business Process Execution Language), C++, Java, JavaScript, Spring, and various scripting languages. An Eclipse IDE plug-in offers graphical combination of components, hiding much of the complexity of the underlying XML. However, Tuscany has little support for service discovery and management. Certain aspects, such as persistent services, can also be obscure to specify.

Adaptation of SCA for Home Networks

The home system architecture is illustrated in Figure 1. Networks are fundamental to both telecare and home automation. They link devices and services in the home, and to services provided in the wide area. Home networks connect a broad range of devices, appliances, sensors and actuators.

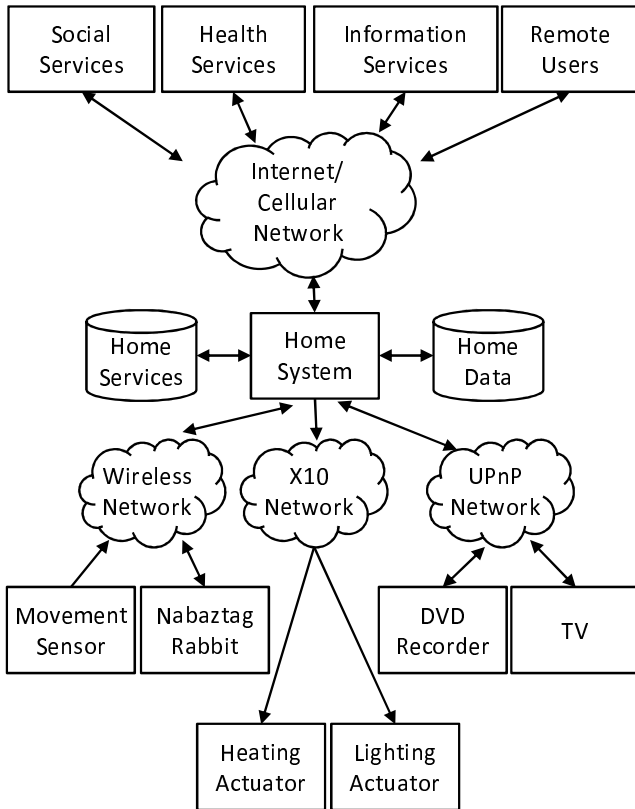


Figure 1: Sample System Environment

There is considerable variety in the protocols used within and to the home. A home platform needs to be as device- and protocol-independent as possible in order to handle this variability. The solution adopted here is to abstract the details by treating devices as providing services. It is then possible to treat everything in the home as a service. This gives a uniform and flexible architecture, and also benefits from the advantages of a SOA. Fortunately, interfaces (drivers) for many protocols exist in commercial or open-source form. It has therefore been possible in this work to focus on the higher-level framework and services.

There are many component frameworks. The attraction of using services as components is that the benefits of SOA can be realised, such as loose coupling and distribution of components. Although physical distribution in the home is limited, there is a need to link to wide-area resources. For example, a weather forecast service can inform how home heating is managed, a health centre can be given regular updates on the user's vital signs, and a social worker can be reassured that the user is performing normal daily tasks.

For the work reported here, SCA has been adopted as the component framework. This couples the benefits of SOA with the additional freedom to implement components as convenient, and to (re)configure them as required. This is especially important for telecare as the system must be adaptable to the user's care needs, and these often evolve over time. The approach proposed in this paper is therefore an application of SCA. The general principles of the framework apply, and a range of components has been developed for the specific applications of telecare and home automation.

III. COMPONENTS FOR HOME SERVICES

A. Philosophy

The authors have observed that smart homes often focus on smart devices. These often need specialised development work that cannot readily be re-used for new purposes. Commercial solutions are also often fairly inflexible, requiring reprogramming (and therefore expert knowledge) for changed functionality. Instead, the authors believe that it is better to combine dumb devices in intelligent and flexible ways. Using off-the-shelf devices also reduces the cost of equipping the home. Of course some requirements may involve specialised devices, but the aim is to reduce the need for these as far as possible. Some devices not normally associated with telecare or home automation might also be adapted for these purposes.

The emphasis is therefore on components and their configurations rather than on devices. Furthermore, there is a strong need for simple configuration that can be performed by the ordinary user. The framework should support more than just device-based services. It should be possible to create higher-level services based on combinations of these. There is also the need for 'glue' logic that combines existing services – something that is normally called service orchestration. SOA and SCA are capable of achieving these goals. Since it is important to make the home system friendly for lay users, the work will also focus on a variety of novel interfaces.

B. Sample Off-The-Shelf Components

As examples for off-the-shelf components, SCA wrappings have been provided for a variety of devices. For new-build housing, wired connections (e.g. using KNX) may be appropriate. However for existing housing, the cost and disruption of new wiring makes it unattractive. The authors have hence mainly focused on wireless home interconnection.

For telecare, a range of home devices produced by Tunstall (www.tunstall.com) are supported. This includes basic devices such as flood detectors, gas detectors, movement detectors and pressure mats, as well as more specialised devices such as medicine dispensers and door entry systems. Similar Visonic devices (www.visonic.com) are also supported, though these are designed more for building management and security than for telecare. These kinds of devices can be used for

monitoring in the home: use of appliances, use of rooms and facilities in the house, visitor and intruder detection, etc.

For environmental monitoring, the wireless devices produced by Oregon Scientific (www.oregonscientific.com) are supported. These are mostly used for data such as room temperatures and humidity levels. This information is used for controlling the household environment. Other devices such as weather forecasters are also supported.

Many home appliances can be controlled via infra-red. The infra-red controllers made by IRTrans (www.irtrans.com) have been given a service wrapping for the component framework. This allows audio-visual devices to be managed, and also other domestic appliances such as air conditioners, curtain closers and garage openers.

Simple domestic appliances are controlled via X10 services. More complex ones (typically audio-visual) are controlled via UPnP (Universal Plug-and-Play). The approach is also open to a variety of other protocols such as EIB, Jini and KNX (using externally provided drivers that are given a service wrapping).

Communication via email and SMS (Short Message Service) is supported. This is two-way: messages can be sent (e.g. to alert a carer or to provide house status) or received (e.g. to activate functions within the home).

Although not yet supported, the authors also plan to incorporate other kinds of off-the-shelf devices such as electrical power monitors (for energy management) and water flow detectors (e.g. for toilet usage).

C. Novel Interface Components

Besides more conventional devices, the authors have adapted more unusual devices for the home framework. As an example, the Nabaztag ‘Internet rabbit’ (www.nabaztag.com) has been adapted as a user-friendly interface device. As a non-threatening interface to technology, this is ideal for technophobic or technically inexperienced users. The rabbit provides an interface which supports speech recognition, text-to-speech conversion, and audio or gestural alerts.

RFID (Radio Frequency Identification) tags are used to ‘label’ various items around the home. For example, keys can be tagged to recognise the user approaching the front door. If the house is unoccupied, the user’s partner can be told by text message that the user is back home.

The WiiMote (a hand-held controller for the Nintendo Wii™ games console, www.nintendo.com/wii) has been given a service wrapping, allowing it to be used for gestural input. For example, the WiiMote can mimic nodding or shaking the head in response to questions, while button inputs can be used for control functions. This is a good example of how a mass-market device, originally for a completely different purpose, can be adapted for use in telecare or home automation.

The i-mate Momento digital picture frame (www.momentolive.com) has been adapted as a way of providing visual reminders anywhere in the house. This is done by translating user messages into images which are then transmitted wirelessly to the picture frame.

A touch-screen SDK (Software Development Kit) has been created. This supports a variety of common components that allow the user to interact with the home in an easy manner. The services developed so far include touch-based home control such as allowing the user to remotely turn on lights, heating and appliances. A touch-screen with a built-in microphone and loudspeaker allows use of voice commands and voice prompts, using the Cerevoice software (www.cereproc.com) provided by the authors’ University of Edinburgh collaborators.

D. Policies for User Control

Many home management tasks take the form of rules. The second author and his colleagues have developed a comprehensive policy-based management system that is applicable in a number of areas [6]. The ACCENT system (Advanced Component Control Enhancing Network Technologies, www.cs.stir.ac.uk/accent) supports a wide variety of capabilities. A policy server is responsible for managing and executing user-defined policies. This is complemented by a goal server that allows the user to define high-level objectives (here, for telecare or home automation). Various policy wizards allow non-technical users to define policies easily. Conflicts among policies are automatically detected and resolved (e.g. the user wishes the house to be warm, but also wishes to save energy). The policy system is interfaced to the target underlying system to be managed (here, the home system).

Policies are in the widely used ECA form (Event-Condition-Action). Triggers (i.e. events), conditions and actions can all be combined in a variety of ways. A telecare example could be: ‘when the user is late in rising (trigger), if it is not the weekend (condition), then alert a relative to this by text message (action)’. A home automation example could be: ‘when freezing weather is forecast, if the user is not on holiday, then switch the central heating on earlier than normal’.

The ACCENT policy system is comprehensive but complex. For now, the authors have created a simple policy system for use in home control. In time, the ACCENT system will be used in the home – though more work is needed on making it user-friendly (e.g. supporting fuzzy policies and explaining policy actions to the user).

E. System Implementation

The home system has a library of components represented as SCA (Tuscany) services for actuators, appliances, devices, information resources, sensors and other services. These services advertise the triggers, conditions and actions that they support. Multimodal interfaces are supported as specialised components that interact with other system services. The policy/goal system is another set of components that manage system services.

A simple interface has been created to allow the user to write new policies. This interface is split into three sections: triggers, conditions and actions. A collection of pull-down

menus is automatically populated from the available services, offering the user an easy and foolproof method for representing what should happen. The menu options also vary with other choices, allowing policies to be formulated in near-natural language (e.g. trigger ‘when the belt clip alarm is activated’ or condition ‘if the hour is between 18:00 and 20:00’).

A new component is added by giving it an SCA wrapper, as well as defining its triggers, conditions and actions. This allows easy combination of all components through the configuration interface. The policy server manages the storage and execution of policies referring to such components.

IV. CONCLUSION

A. Evaluation

The home system has been evaluated in a lab setting. It has been established that non-technical users can define policies for how the house should behave. These rules can make use of any component integrated into the framework. The system requires a component to be wrapped in a certain way in order to advertise their services to the system. Thanks to the SOA approach, the addition and removal of components is quick and easy. This aspect is handled by the developer and requires technical expertise. However, configuration can be carried out by the ordinary home user.

Representing the components as Tuscany services has had both advantages and disadvantages. The SCA specification is still young, with the 1.0 standard released in only 2007. As an implementation of SCA, Tuscany is still evolving and so has limited documentation, tutorials and working examples. It provides very little by way of service discovery or service management, which left a gap that had to be manually filled in the component framework. There are tools for building Tuscany services using a GUI plug-in tool for Eclipse. This automates the somewhat complex XML for the SCA composite files, but the tool is rather restrictive. With few examples of how components are configured and combined, it can be non-obvious how to achieve certain results.

B. Future Work

The experience of Tuscany so far is that it needs to mature further to be fully usable for building a home system. Other implementations of SCA are being investigated as alternatives. Some alternative approaches being considered are ADLs (Architecture Description Languages [7]), Bonjour (developer.apple.com/bonjour), Jini, JXTA (Juxtapose,

jxta.dev.java.net) and OSGi (Open Services Gateway initiative, www.osgi.org).

With the chosen architecture in place, the focus will move to multimodal interfaces that are attractive to end users. Various modalities will be offered (e.g. web, mobile, speech, gesture) to give the user as much freedom and customisability as possible. For example, regular users of a mobile phone may prefer to control their home with such a device, while others may prefer a touch-screen within the home.

Further components will be added. Some ideas currently being implemented are:

- A weather forecast service that gives meteorological information (e.g. for heating and ventilation control).
- Diary support to allow the user to integrate home support with time-dependent requirements (e.g. to secure the house while on holiday).
- Use of dialogue management and Interactive Voice Response for extended speech interaction.

As the system matures, its evaluation will move from the lab to trials in the homes of end users.

REFERENCES

- [1] L. A. Gavrilov and P. Heuveline. Aging of population. In P. Demeny and G. McNicoll, editors, *The Encyclopaedia of Population*, pp. 27–50. MacMillan, London, UK, Jan. 2003.
- [2] B. Margolis and J. L. Sharpe. *SOA for The Business Developer*, MC Press, 2007.
- [3] Davidoff, S. et al. Principles of Smart Home Control. *Proc. UbiComp 2006*, 19–34.
- [4] Newman, M.W., Elliott, A. and Smith, T.F. Providing an Integrated User Experience of Networked Media, Devices, and Services Through End-User Composition In *Proc. of Pervasive 2008*, 213–227.
- [5] J. King, R. Bose, H. Yang, S. Pickles and A. Helal. Atlas – A service oriented sensor platform. In *Proc. 1st IEEE Int. Workshop on Practical Issues in Building Sensor Network Applications*, pp. 630–638. IEEE Computer Society Press, 2006.
- [6] Kenneth J. Turner and Gavin A. Campbell. Goals for Telecare Networks, in Abdel Obaid (ed.), *Proc. 9th Int. Conf. on New Technologies for Distributed Systems*, pp. 270–275, Montreal, July 2009.
- [7] Nenad Medvidovic. “ADLs and Dynamic Architecture Changes.” In Alexander L. Wolf, ed., *Proceedings of the Second International Software Architecture Workshop (ISAW-2)*, pages 24–27, San Francisco, CA, October 14–15, 1996.

ACKNOWLEDGEMENTS

Claire Maternaghan is supported by the Scottish Informatics and Computer Science Alliance, the University of Stirling, and the MATCH project (Scottish Funding Council, grant HR04016). The authors are grateful to their colleagues on the MATCH project for their advice and support.